การแทนข้อมูลแบบแฟร็กทัลสำหรับข้อมูลอนุกรมเวลาขนาดใหญ่

นายพจน์ สัจจิพานนท์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2551

FRACTAL REPRESENTATION FOR LARGE TIME SERIES DATA

Mr. Poat Sajjipanon

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 2008

หัวข้อวิทยานิพนธ์	การแทนข้อมูลแบบแฟร็กทัลสำหรับข้อมูลอนุกรมเวลาขนาดใหญ่
โดย	นายพจน์ สัจจิพานนท์
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร.โชติรัตน์ รัตนามหัทธนะ

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(รองศาสตราจารย์ ดร.บุญสม เลิศหิรัญวงศ์)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(ศาสตราจารย์ ดร.บุญเสริม กิจศิริกุล)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์
(ผู้ช่วยศาสตราจารย์ ดร.โชติรัตน์ รัตนามหัทธนะ)

..... กรรมการภายนอกมหาวิทยาลัย
(รองศาสตราจารย์ ดร.กฤษณะ ไวยมัย)

..... กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.สุกรี สินธุภิญโญ)

พจน์ สัจจิพานนท์ : การแทนข้อมูลแบบแฟร็กทัลสำหรับข้อมูลอนุกรมเวลาขนาดใหญ่. (FRACTAL REPRESENTATION FOR LARGE TIME SERIES DATA) อาจารย์ที่ปรึกษา : ผู้ช่วยศาสตราจารย์ ดร.โชติรัตน์ รัตนามหัทธนะ, 112 หน้า.

งานวิจัยด้านการทำเหมืองข้อมูลอนุกรมเวลาส่วนมากได้มุ่งเน้นการพัฒนาประสิทธิภาพทั้งในด้านความแม่นยำและความเร็ว อย่างไรก็ตาม สำหรับงานวิจัยที่ผ่านมามักเกิดภาวะถ่วงดุลกันระหว่างประสิทธิภาพทั้งสองด้าน วิธีการค้นหาข้อมูลตามความคล้ายที่ใช้กันทั่วไปและให้ผลความแม่นยำที่ดีมักต้องใช้เวลาในการคำนวณสูง ซึ่งส่งผลกระทบเป็นอย่างมากสำหรับการนำไปใช้ในทางปฏิบัติ การลดขนาดข้อมูลอนุกรมเวลาจึงเป็นวิธีหนึ่งที่สามารถลดเวลาในการประมวลผลได้ แต่ต้องแลกกับผลของความแม่นยำที่ลดลงเมื่อเทียบกับวิธีที่ไม่ทำการลดขนาด ดังนั้น วิธีการลดขนาดของข้อมูลที่มีคุณภาพที่ดีจึงควรให้ผลของความแม่นยำที่ลดลงไม่มากนัก ดังนั้นงานวิจัยนี้จึงได้นำเสนอการแทนข้อมูลแบบแฟร็กทัล โดยเป็นการลดขนาดข้อมูลอนุกรมเวลาที่อยู่บนแนวคิดของมิติแฟร็กทัลมาประยุกต์ใช้กับข้อมูลอนุกรมเวลา ซึ่งสามารถลดขนาดข้อมูลอนุกรมเวลาหนึ่ง ๆ ให้เหลือเพียงเลขจำนวนจริง 2 ค่า สำหรับข้อมูลอนุกรมเวลา 1 อนุกรม ในส่วนของการทดลอง ทำการวัดประสิทธิภาพด้วยการจำแนกข้อมูลและเปรียบเทียบกับงานวิจัยอื่น ๆ ได้แก่ การวัดระยะทางแบบยุคลิด ไดนามิกไทม์วอร์ปปิง ซีดีเอ็ม การแทนข้อมูลแบบแซด และการแทนข้อมูลแบบคลิป ซึ่งจากผลการทดลองสรุปได้ว่า เมื่อชุดข้อมูลมีปริมาณเพิ่มมากขึ้น การแทนข้อมูลแบบแฟร็กทัลจะให้ผลในด้านเวลาที่ดีกว่าวิธีการลดขนาดข้อมูลด้วยวิธีอื่นได้อย่างเด่นชัดยิ่งขึ้น ซึ่งในบางชุดข้อมูล วิธีลดขนาดข้อมูลที่ได้นำเสนอใช้เวลาในการค้นหาข้อมูลน้อยกว่าวิธีไดนามิกไทม์วอร์ปปิงถึงกว่าหลายพันเท่า นอกจากนี้ยังให้ผลความแม่นยำที่เหนือกว่าวิธีการลดขนาดข้อมูลอื่นรวมถึงการวัดระยะทางแบบยุคลิด รวมทั้งได้ผลความแม่นยำใกล้เคียงกับไดนามิกไทม์วอร์ปปิง และซีดีเอ็ม และมีบางชุดข้อมูลได้รับผลความแม่นยำมากกว่าทุกวิธีที่นำมาเปรียบเทียบ

ภาควิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนิสิต .....
สาขาวิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่ออาจารย์ที่ปรึกษา .....
ปีการศึกษา 2551

5070359221 : MAJOR COMPUTER ENGINEERING

KEY WORD : TIME SERIES / DATA MINING / DIMENSIONALITY REDUCTION / FRACTAL / FRACTAL DIMENSION

POAT SAJJIPANON : FRACTAL REPRESENTATION FOR LARGE TIME SERIES DATA. THESIS ADVISOR : ASST. PROF. CHOTIRAT RATANAMAHATANA, PH.D., 112 pp.

Most of the time series mining tasks have focused on increasing both accuracy and speed. However, a tradeoff between accuracy and time consumption needs to be considered. Increasing accuracy of the mining task leads to higher computational cost. The dimensionality reduction techniques can reduce the time complexity of mining tasks, but it hurts the accuracy. In this research, Fractal Representation, a new dimensionality reduction technique, uses merely two real values to represent a time series sequence. To demonstrate effectiveness of fractal representation on classification problems, this research compares the proposed method with existing classification methods, i.e., Euclidean distance, Dynamic Time Warping (DTW) distance, Compression-Based Dissimilarity Measure (CDM), Symbolic Aggregate Approximation (SAX), and Clipped Data Representation, both in terms of accuracy and speed. In the experiments, when amount of time series increases, Fractal Representation greatly outperforms DTW up to 3 orders of magnitude in terms of speed. Moreover, the accuracy of Fractal Representation is comparable to DTW's and CDM's and outperforms the existing methods including SAX, Clipped Data Representation, and Euclidean distance.

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

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

Academic Year ..... 2008 .....

กิตติกรรมประกาศ

ในช่วงระยะเวลาตั้งแต่เริ่มทำการค้นคว้าเพื่อหาหัวข้อวิจัย ได้พบกับอุปสรรคต่าง ๆ มากมาย จนกระทั่งได้รับการจัดทำวิทยานิพนธ์ฉบับนี้ อันเป็นบทเรียนที่ทรงคุณค่ายิ่งแก่ผู้จัดทำ ตั้งแต่ได้รับการฝึกฝนจากอาจารย์หลาย ๆ ท่าน ได้ค้นคว้าเพื่อที่จะเรียนรู้ด้วยตนเอง และแก้ไขปัญหาที่พบได้อย่างถูกต้องและเหมาะสม ซึ่งเป็นประโยชน์อย่างยิ่งสำหรับการทำวิจัยและยังช่วยพัฒนาศักยภาพให้แก่ผู้จัดทำเป็นอย่างมาก รวมถึงวิทยานิพนธ์ฉบับนี้ยังได้รับการสนับสนุนจากบุคคลหลายฝ่าย และขอกล่าวถึงบุคคลที่ช่วยให้งานวิจัยนี้เสร็จสมบูรณ์ไปได้ด้วยดี ข้าพเจ้าจึงขอแสดงความซาบซึ้งเป็นอย่างยิ่งและขอขอบพระคุณสำหรับความช่วยเหลือที่มากมาย

ขอขอบพระคุณเป็นอย่างสูงสำหรับอาจารย์ที่ปรึกษาวิทยานิพนธ์ฉบับนี้ ผู้ช่วยศาตราจารย์ ดร. โชติรัตน์ รัตนามหัทธนะ ผู้ซึ่งอบรม สั่งสอน ดูแล และให้คำชี้นำให้แก่ศิษย์ อันเป็นปัจจัยหลักให้ทำให้งานวิจัยและวิทยานิพนธ์ฉบับนี้สำเร็จไปได้ด้วยดี

ขอขอบพระคุณคณะกรรมการสอบวิทยานิพนธ์ทุกท่าน ที่ให้แนวคิด ความคิดเห็น รวมถึงข้อเสนอแนะที่เป็นประโยชน์ ซึ่งทำให้วิทยานิพนธ์ฉบับนี้ได้รับการพัฒนาให้มีคุณภาพมากยิ่งขึ้นไปอีก ซึ่งคณะกรรมการสอบวิทยานิพนธ์ ประกอบไปได้ ศาสตราจารย์ ดร. บุญเสริม กิจศิริกุล ผู้ช่วยศาสตราจารย์ ดร.สุกรี สินธุภิญโญ และรองศาสตราจารย์ ดร.กฤษณะ ไวยมัย

ขอบคุณเพื่อน ๆ ในห้องปฏิบัติการทุกคนที่ช่วยเหลืองานวิจัยนี้ด้วยดีเสมอมา และให้แนวคิดในการแก้ไขที่เหมาะสมในวิทยานิพนธ์ฉบับนี้ รวมทั้งคอยดูแลเอาใจใส่ซึ่งกันและกัน ทำให้การทำงานเป็นไปอย่างราบรื่น

และบุคคลสุดท้ายที่ขาดเสียมิได้ ขอขอบพระคุณครอบครัวทุกคนที่ช่วยสนับสนุนในหลาย ๆ ด้าน และเป็นกำลังใจที่ดีเสมอมา ซึ่งทำให้วิทยานิพนธ์ฉบับนี้ลุล่วงไปได้ด้วยดี

สารบัญ

หน้า

สารบัญภาพ

หน้า

หน้า

สารบัญตาราง

หน้า

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของปัญหา

ในปัจจุบันมีงานวิจัยจำนวนมากสำหรับการทำเหมืองข้อมูลอนุกรมเวลา (Time Series Mining) มุ่งเป้าไปที่การค้นหาความคล้ายคลึง (Similarity Search) เพื่อจำแนกประเภทข้อมูลเดียวกัน โดยมีวิธีต่าง ๆ ของการทำเหมืองข้อมูลที่นิยมนำมาใช้สำหรับวัดประสิทธิภาพของการค้นหาความคล้ายคลึงสำหรับข้อมูลอนุกรมเวลา เช่น การจำแนกประเภทข้อมูล (Classification) การจับกลุ่ม (Clustering) และอื่น ๆ งานวิจัยสำหรับการค้นหาความคล้ายคลึงมีอยู่หลากหลายวิธี แต่มีบางวิธีที่งานวิจัยในปัจจุบันนิยมนำมาเปรียบเทียบเพื่อวัดประสิทธิภาพ เช่น การวัดระยะทางแบบยุคลิด (Euclidean Distance) เป็นวิธีที่มีประสิทธิภาพในด้านความเร็ว โดยใช้เวลาในการประมวลผล $O(n)$ แต่ผลของความแม่นยำที่ได้ยังไม่มากพอเมื่อเปรียบเทียบกับวิธีอื่น ๆ โดยเป็นการคำนวณหาระยะทางระหว่างจุดต่อจุด ผลรวมของระยะทางในทุก ๆ จุดของข้อมูลอนุกรมเวลาจะถูกเก็บไว้เพื่อเปรียบเทียบความคล้ายคลึงกันระหว่างข้อมูล การวัดระยะทางแบบไดนามิกไทม์วอร์ปปิง (Dynamic Time Warping Distance-DTW) [1, 2] เป็นวิธีที่ให้ผลของความแม่นยำในการวัดความคล้ายคลึงที่มีประสิทธิภาพ แต่เวลาที่ใช้ในการคำนวณค่อนข้างสูง โดยส่วนมากนิยมนำไปใช้กับข้อมูลอนุกรมเวลาขนาดสั้น โดยใช้เวลาในการประมวลผลประมาณ $O(n^2)$ ทำให้การคำนวณกับข้อมูลอนุกรมเวลาขนาดใหญ่ขึ้นใช้เวลาในการคำนวณเพิ่มมากขึ้นหลายเท่า จากปัญหาดังกล่าว จึงมีงานวิจัยเพิ่มเติมเพื่อช่วยลดเวลาการคำนวณของไดนามิกไทม์วอร์ปปิง เช่น ฟังก์ชันขอบเขตล่าง (Lower Bounding Function) [2] และเงื่อนไขบังคับโดยรวม (Global Constraint) [3] เป็นต้น การค้นหาความคล้ายคลึงที่กล่าวมาข้างต้นเป็นการคำนวณความคล้ายคลึงระหว่างข้อมูลอนุกรมเวลาโดยการวัดระยะทาง (Distance Measure) จากปัญหาของทั้งการวัดระยะทางแบบยุคลิดและไดนามิกไทม์วอร์ปปิง จึงมีการนำเสนองานวิจัยเพื่อลดเวลาในการคำนวณให้น้อยลงหรือไม่กระทบกับผลความแม่นยำมากนัก โดยใช้พื้นฐานของหลักการบีบอัดข้อมูลคือ ซีดีเอ็ม (Compression-Based Dissimilarity Measure-CDM) [4] ซึ่งเน้นไปที่ข้อมูลอนุกรมเวลาขนาดใหญ่ อย่างไรก็ตาม การบีบอัดข้อมูลต้องมีการเขียนและอ่านไฟล์ โดยมีการติดต่อกับอินพุต/เอาต์พุต (Input/Output) ซึ่งจะส่งผลให้ความเร็วในการคำนวณช้าลง

จากที่กล่าวมาข้างต้น เห็นได้ว่างานวิจัยที่ให้ผลความแม่นยำสูง โดยส่วนมากใช้เวลาในการคำนวณค่อนข้างสูง ดังนั้น จึงมีงานวิจัยอีกสาขาหนึ่งซึ่งมุ่งเน้นในด้านการลดเวลาสำหรับการประมวลผลเป็นหลัก ซึ่งจะลดทอนขนาดของข้อมูลให้มีขนาดเล็กลง และได้ผลความแม่นยำที่ใกล้เคียงกับการค้นหาความคล้ายคลึง คือ การลดขนาดข้อมูล (Dimensionality

Reduction) เป็นการหาตัวแทนใหม่จากข้อมูลอนุกรมเวลาเดิม ซึ่งคำนวณจากโครงสร้างภายในของข้อมูล เพื่อให้ได้ข้อมูลใหม่ที่สามารถแสดงคุณลักษณะของข้อมูลเดิมได้ด้วยขนาดที่ลดลง สำหรับตัวอย่างงานวิจัยที่นำเสนอเกี่ยวกับการลดขนาดข้อมูลอนุกรมเวลา ได้แก่ การลดขนาดข้อมูลแบบพีเอเอ (Piecewise Aggregate Approximation-PAA) [5] การลดขนาดข้อมูลแบบแซค (Symbolic Aggregate Approximation-SAX) [6, 7] และการแทนข้อมูลแบบคลิป (Clipped Data Representation) [8] สำหรับการลดขนาดข้อมูลอนุกรมเวลาโดยส่วนมาก จำเป็นต้องกำหนดขนาดความยาวของข้อมูลอนุกรมเวลาใหม่ตามความเหมาะสมของแต่ละวิธี ดังนั้น การลดขนาดที่เหมาะสมกับข้อมูลอนุกรมเวลาจึงเป็นสิ่งที่ต้องประมาณโดยผู้ใช้เอง ซึ่งอาจทำให้ความแม่นยำลดลง หรือในอีกกรณีหนึ่งคือ หากลดขนาดข้อมูลน้อยเกินไป เวลาที่ใช้ในการคำนวณจะลดลงไม่มากนัก เมื่อทดสอบด้วยวิธีวัดระยะทางแบบต่างๆ

ผู้วิจัยจึงเล็งเห็นว่า หากมีวิธีที่สามารถลดขนาดของข้อมูลอนุกรมเวลาใด ๆ ให้เหลือเพียงเลขจำนวนจริงหนึ่งค่า และเลขจำนวนจริงมีค่าใกล้เคียงกันมาก ๆ สำหรับข้อมูลอนุกรมเวลาชนิดเดียวกัน ซึ่งสามารถทดสอบเพื่อจำแนกข้อมูลประเภทเดียวกัน เพียงแค่การเรียงลำดับตัวเลขจากค่าน้อยไปหาค่ามากเท่านั้น แต่ในความเป็นจริงแล้วยังไม่มีวิธีที่สามารถลดขนาดข้อมูลให้เหลือเพียงตัวเลขเดียวและให้ผลลัพธ์ที่มีประสิทธิภาพได้ แนวคิดสำหรับงานวิจัยนี้ จึงพยายามหาวิธีที่สามารถลดขนาดของข้อมูลอนุกรมเวลาให้มีขนาดเล็กที่สุดเท่าที่เป็นไปได้ โดยการลดขนาดข้อมูลอนุกรมเวลาทุกตัวจะได้ขนาดเท่ากันเสมอ ทำให้เวลาในการประมวลผลลดลงมาก

สำหรับวัตถุประสงค์ของงานวิจัย เพื่อนำเสนอวิธีการลดขนาดข้อมูลอนุกรมเวลาที่มีขนาดใหญ่ให้เหลือขนาดเล็กลงด้วยเลขจำนวนจริงเพียง 2 ค่า โดยขนาดของข้อมูลอนุกรมเวลาใหม่ที่ได้ ยังคงคุณสมบัติในการจำแนกกลุ่มของข้อมูลประเภทเดียวกันได้อย่างเหมาะสม เรียกวิธีนี้ว่า วิธีแทนข้อมูลแบบแฟร็กทัล (Fractal Representation) แนวคิดของวิธีนี้เน้นไปที่การลดขนาดข้อมูลอนุกรมเวลาขนาดใหญ่เป็นหลัก สำหรับข้อมูลอนุกรมเวลาขนาดใหญ่แสดงถึงข้อมูลที่มีความยาวมาก ๆ ตั้งแต่ความยาวเท่ากับ 1,000 จุด ขึ้นไป โดยวิธีนี้ใช้การค้นหาความคล้ายคลึงของข้อมูลอนุกรมเวลาด้วยการวัดระยะทางแบบยุคลิด และได้รับผลของความแม่นยำและเวลาที่มีคุณภาพเมื่อเปรียบเทียบกับวิธีอื่น ๆ รวมทั้งเปรียบเทียบประสิทธิภาพทั้งความแม่นยำและเวลากับวิธีการลดขนาดข้อมูลแบบอื่น ๆ ด้วยเช่นกัน สำหรับวิธีการทดลองจะประเมินผลจากวิธีการจำแนกข้อมูลแบบเพื่อนบ้านใกล้ที่สุดอันดับที่หนึ่ง (1 Nearest Neighbor) ด้วยวิธีทดสอบแบบการนำออกหนึ่ง (Leaving-one-out) ซึ่งเป็นตัววัดประสิทธิภาพในการค้นหาความคล้ายคลึงของข้อมูลอนุกรมเวลาได้อย่างชัดเจนและนิยมนำมาใช้ซึ่งจะเห็นได้จากงานวิจัยจำนวนมากที่วัดประสิทธิภาพด้วยวิธีนี้

1.2 วัตถุประสงค์ของการวิจัย

1. งานวิจัยนี้เน้นไปที่การลดขนาดข้อมูลอนุกรมเวลาให้มีขนาดเล็กที่สุดเท่าที่เป็นไปได้ สำหรับการลดขนาดข้อมูลอนุกรมเวลาขนาดใหญ่ โดยให้ผลที่มีประสิทธิภาพทั้งความแม่นยำและความเร็วในการทำเหมืองข้อมูล
2. นำเสนอแนวทางใหม่สำหรับการลดขนาดข้อมูลอนุกรมเวลาโดยใช้วิธีแทนข้อมูลแบบแฟร็กทัลบนแนวคิดของมิติแฟร็กทัล
3. วิธีการแทนข้อมูลแบบแฟร็กทัลถูกพัฒนาภายใต้มิติเส้นขอบ โดยปรับวิธีให้เหมาะสมสำหรับข้อมูลอนุกรมเวลา และคำนวณได้อัตโนมัติตามฟังก์ชันที่เหมาะสม

1.3 ขอบเขตของการวิจัย

1. เปรียบเทียบประสิทธิภาพกับการวัดความคล้ายคลึงอื่น ๆ เช่น การวัดระยะทางแบบยุคลิด การวัดระยะทางแบบไดนามิกไทม์วอร์ปปิง และซีดีเอ็ม
2. เปรียบเทียบประสิทธิภาพกับการลดขนาดข้อมูลอื่น ๆ เช่น การลดขนาดข้อมูลแบบแซค และการแทนข้อมูลแบบคลิป
3. ทดสอบผลของความแม่นยำและเวลา โดยประเมินผลจากวิธีการจำแนกข้อมูลแบบเพื่อนบ้านใกล้ที่สุดอันดับที่หนึ่ง ด้วยวิธีทดสอบแบบการนำออกหนึ่ง
4. กลุ่มข้อมูลอนุกรมเวลาขนาดใหญ่ที่นำมาใช้ในการทดลองมีความยาวตั้งแต่ 1,000 จุด ขึ้นไป

1.4 ประโยชน์ที่ได้รับ

งานวิจัยนี้สามารถลดขนาดข้อมูลอนุกรมเวลาให้เหลือเพียงขนาดที่เล็กมากเมื่อเปรียบเทียบกับวิธีการลดขนาดข้อมูลที่ผ่านมา และยังคงได้รับประสิทธิภาพทั้งผลความแม่นยำและเวลาเมื่อเปรียบเทียบกับการวัดความคล้ายคลึงแบบอื่น ๆ

1.5 วิธีดำเนินการวิจัย

1. ศึกษาการทำเหมืองข้อมูลกับข้อมูลอนุกรมเวลา
2. ศึกษาทฤษฎีเคออส แฟร็กทัล และมิติแฟร็กทัล
3. ศึกษาแนวทางเพื่อนำหลักการของมิติแฟร็กทัลมาพัฒนากับการลดขนาดข้อมูลอนุกรมเวลา และนำมาประยุกต์ใช้กับวิธีการทำเหมืองข้อมูลอนุกรมเวลา
4. ออกแบบและพัฒนาวิธีการลดขนาดข้อมูลอนุกรมเวลาโดยใช้หลักการหาค่าของมิติแฟร็กทัลที่เหมาะสมกับข้อมูลอนุกรมเวลา

5. ทดสอบประสิทธิภาพของข้อมูลอนุกรมเวลาด้วยการแทนข้อมูลแบบแฟร็กทัล และเปรียบเทียบผลการทดลองกับวิธีอื่น ๆ ด้วยการประเมินผลวัดจากวิธีจำแนกข้อมูลแบบเพื่อนบ้านใกล้ที่สุดอันดับที่หนึ่ง โดยวิธีทดสอบแบบการนำออกหนึ่ง
6. วิเคราะห์และสรุปผลการทดลอง
7. สรุป เรียบเรียง และจัดทำวิทยานิพนธ์

1.6 ผลงานตีพิมพ์จากงานวิจัย

ส่วนหนึ่งของงานวิทยานิพนธ์นี้ ได้รับการตีพิมพ์เป็นบทความทางวิชาการสองเรื่อง ดังนี้

- “Efficient Time Series Mining using Fractal Representation” โดย พจน์ สัจจิพานนท์ และโชติรัตน์ รัตนามหัทธนะ ในงานประชุมวิชาการนานาชาติ ครั้งที่ 3 “The 2008 International Conference on Convergence and Hybrid Information Technology (ICCIT)” ซึ่งจัดขึ้น ณ เมืองปูซาน ประเทศเกาหลีใต้ ระหว่างวันที่ 11 ถึง 13 พฤศจิกายน 2551 ดังรายละเอียดในภาคผนวก ข
- “A Novel Fractal Representation for Dimensionality Reduction of Large Time Series Data (PAKDD)” โดย พจน์ สัจจิพานนท์ และโชติรัตน์ รัตนามหัทธนะ ในงานประชุมวิชาการนานาชาติ ครั้งที่ 13 “The Pacific-Asia Conference on Knowledge Discovery and Data Mining” ซึ่งจัดขึ้น ณ กรุงเทพมหานคร ประเทศไทย ระหว่างวันที่ 27 ถึง 30 เมษายน 2552 ดังรายละเอียดในภาคผนวก ค

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

เนื้อหาในบทนี้จะเป็นการนำเสนอทฤษฎีต่าง ๆ โดยเริ่มจากทฤษฎีเคออส ซึ่งเป็นแนวคิดพื้นฐานของแฟร็กทัล นำเสนอคุณสมบัติและตัวอย่างข้อมูลที่แสดงคุณลักษณะของแฟร็กทัล ซึ่งจากหลักการของแฟร็กทัลจึงนำไปสู่วิธีการหามิติของข้อมูลใดๆ คือมิติแฟร็กทัล ซึ่งเป็นวิธีพื้นฐานในการวิจัยและพัฒนาวิธีลดขนาดของข้อมูลอนุกรมเวลา โดยแนวคิดในการคำนวณมิติแฟร็กทัลนำเสนอในหลากหลายแบบ และในส่วนสุดท้ายกล่าวถึงงานวิจัยที่เกี่ยวข้องกับกับมิติแฟร็กทัล และวิธีต่าง ๆ ที่ผ่านมา ซึ่งจะนำมาวัดประสิทธิภาพกับงานวิจัยนี้

2.1 ทฤษฎีที่เกี่ยวข้อง

สำหรับทฤษฎีที่เกี่ยวข้องนี้ จะเริ่มต้นนำเสนอจาก ทฤษฎีเคออส โดยกล่าวถึงแนวคิดที่แสดงคุณลักษณะของเคออสและระบบเคออส ตามด้วยแฟร็กทัล ซึ่งจะแสดงคุณสมบัติความคล้ายคลึงตัวเองในรูปแบบที่แตกต่างกัน เช่น ความคล้ายคลึงตัวเองแบบทุกส่วน ความคล้ายคลึงตัวเองบางส่วน และความคล้ายคลึงตัวเองเชิงสถิติ และตัวอย่างที่มีลักษณะของแฟร็กทัลที่เกิดขึ้นจากธรรมชาติหรือสร้างโดยสมการทางคณิตศาสตร์ รวมทั้งวิธีสร้างรูปทรงเรขาคณิตที่แสดงคุณสมบัติของแฟร็กทัล เช่น ฝุ่นแคนทอร์ เส้นโค้งคอช และพรมเซอร์พินสกี ซึ่งแนวคิดของแฟร็กทัล ทำให้เกิดแนวคิดใหม่ที่ใช้ในการหาค่ามิติของข้อมูลใด ๆ เรียกว่า มิติแฟร็กทัล และในส่วนสุดท้ายจะกล่าวถึงคุณสมบัติของมิติแฟร็กทัล และวิธีการคำนวณเพื่อหามิติแฟร็กทัล โดยอธิบายแนวคิดของมิติความคล้ายคลึงตัวเอง ซึ่งเป็นวิธีที่นิยมนำมาอธิบายมิติแฟร็กทัล และกล่าวถึงมิติเส้นขอบเพื่อนำมาประยุกต์กับการแทนข้อมูลแบบแฟร็กทัล

2.2 ทฤษฎีเคออส (Chaos Theory)

ทฤษฎีเคออส [9-11] คือ ปรากฏการณ์ที่เกิดขึ้นอย่างไร้ระเบียบ (Random) แต่ในความเป็นจริงมีลักษณะแฝงเชิงกำหนด (Deterministic) ตัวอย่างของปรากฏการณ์ที่แสดงความเป็นเคออส เช่น

- การทอดลูกเต๋า

  การทอดลูกเต๋า เป็นวิธีที่เกิดการสุ่มค่าจากการเสี่ยงทาย ซึ่งผลของตัวเลขที่ได้รับแต่ละครั้งของการทอดจะเกิดขึ้นอย่างไร้ระเบียบ แต่ไม่มีลักษณะแฝงเชิงกำหนด เนื่องจาก ไม่สามารถหารูปแบบหรือสมการใด ๆ มานิยามผลของการทอดลูกเต๋าในแต่ละรอบ เรียกค่าที่ได้จากการทอดลูกเต๋าว่า เลขสุ่ม (Random Number)

- เลขสุ่มของคอมพิวเตอร์ที่คำนวณจากสมการ

ตัวอย่างของสมการที่ใช้สำหรับคำนวณตัวเลขแบบสุ่มของคอมพิวเตอร์แสดงได้ดังสมการที่ 2.1 โดย n คือ จำนวนครั้งที่ทำการสุ่ม c และ m คือ เลขจำนวนเต็มใด ๆ และ $x(n)$ คือผลลัพธ์ของการคำนวณในรอบที่ n จะเห็นได้ว่า สมการนี้ได้ผลลัพธ์ของการคำนวณในรอบถัดไป มาจากผลลัพธ์จากการคำนวณครั้งที่แล้ว ซึ่งจะเห็นว่า ผลลัพธ์ที่ได้ในแต่ละรอบ ค่าตัวเลขจะเกิดขึ้นอย่างไร้ระเบียบ และมีลักษณะแฝงเชิงกำหนด โดยสามารถบรรยายได้ด้วยสมการ ซึ่งแตกต่างจากการทอดลูกเต๋า ที่ไม่สามารถหากฎเกณฑ์ที่ทำให้เกิดตัวเลขในแต่ละครั้งของการทอดลูกเต๋า

$$x(n+1) = cx(n) \bmod m \tag{2.1}$$

2.2.1 ระบบเคออส

จากทฤษฎีของเคออสข้างต้น ทำให้เกิดแนวคิดของระบบที่แสดงถึงความเป็นเคออส โดยระบบที่แสดงลักษณะของเคออสมีดังนี้

- คุณสมบัติไม่เป็นเชิงเส้น (Non Linearity)

ผลลัพธ์ของระบบทั้งหมดไม่เท่ากับผลรวมของผลลัพธ์ย่อย ๆ รวมกัน แต่ไม่จำเป็นที่ระบบไม่เชิงเส้นทุกระบบต้องเป็นเคออส ดังแสดงในสมการที่ 2.2

$$f(x+y) \neq f(x) + f(y) \tag{2.2}$$

- ระบบเชิงกำหนด (Deterministic System)

เป็นระบบที่เกิดขึ้นภายใต้กฎเกณฑ์ที่แน่นอน ในทางคณิตศาสตร์ คือ มีสมการที่อธิบายโครงสร้างของระบบเคออส โดยไม่มีการสุ่มค่าในสมการ

- ระบบพลวัต (Dynamic System)

กระบวนการของระบบเคออสเกิดจากการกระทำของระบบพลวัต คือ การกระทำครั้งใหม่เกิดจากผลของการกระทำก่อนหน้า

- คุณสมบัติไวต่อสภาวะเริ่มต้น (Sensitivity to Initial Conditions)

ไวต่อสภาวะเริ่มต้น คือ เมื่อกำหนดข้อมูลเริ่มต้นให้กับระบบเคออส ถ้าข้อมูลที่กำหนดเข้าไปมีความแตกต่างกันหรือเกิดข้อผิดพลาดเพียงเล็กน้อย จะส่งผลให้ผลลัพธ์ที่ได้ไม่สามารถคาดการณ์กับผลที่จะเกิดขึ้นในอนาคต

- การทำนายในระยะยาวไม่สามารถทำได้

จากคุณลักษณะการไวต่อสภาวะเริ่มต้นทำให้เกิดความแตกต่างระหว่างข้อมูลเริ่มต้นกับผลลัพธ์เป็นอย่างมาก ซึ่งเป็นผลให้ไม่สามารถทำนายผลลัพธ์ที่ได้ในระยาว แต่เคออสทำให้เกิดผลดีในอีกด้านหนึ่ง คือ เกิดทฤษฎีที่สามารถนำมาทำนายผลในระยะสั้นได้ เช่น เลขชี้กำลังไลยาปูนอฟ (Lyapunov Exponent) [11] เป็นต้น

2.3 แฟร็กทัล (Fractal)

แฟร็กทัล [9-11] เป็นส่วนหนึ่งของทฤษฎีเคออสและมีคุณสมบัติของความเป็นระบบเคออส เพราะฉะนั้น ระบบของแฟร็กทัลมีลักษณะที่เกิดขึ้นแบบเชิงกำหนดและไม่เป็นเชิงเส้น โดยมีการกระทำที่เกิดขึ้นเป็นแบบพลวัต เมื่อคำนวณอย่างต่อเนื่องจะเกิดคุณสมบัติที่ไวต่อสภาวะเริ่มต้น ดังนั้น จึงไม่สามารถทำนายผลในระยะยาวได้ และแฟร็กทัลมีคุณลักษณะเด่นที่สำคัญ คือ คุณสมบัติคล้ายคลึงตัวเอง (Self Similarity) เมื่อพยายามขยายภาพหรือปรับระดับความละเอียดที่ขนาดเท่าใดก็ตาม ก็ยังคงแสดงความคล้ายคลึงตัวเองเสมอ ข้อมูลที่นำเสนอในแนวทางของแฟร็กทัลส่วนมากเป็นข้อมูลรูปภาพ ดังแสดงในรูปที่ 2.1 แสดงลักษณะคล้ายคลึงตัวเองของแฟร็กทัลในลักษณะต่าง ๆ ซึ่งสามารถสร้างโดยสมการทางคณิตศาสตร์ (ซ้าย) หรือความคล้ายคลึงตัวเองที่สร้างโดยธรรมชาติ (ขวา) จะเห็นว่า รูปภาพทั้งหมด เมื่อเราทำการขยายขนาดในระดับที่ละเอียดมากขึ้น ภาพเหล่านั้นยังคล้ายคลึงกับภาพก่อนหน้าเสมอ

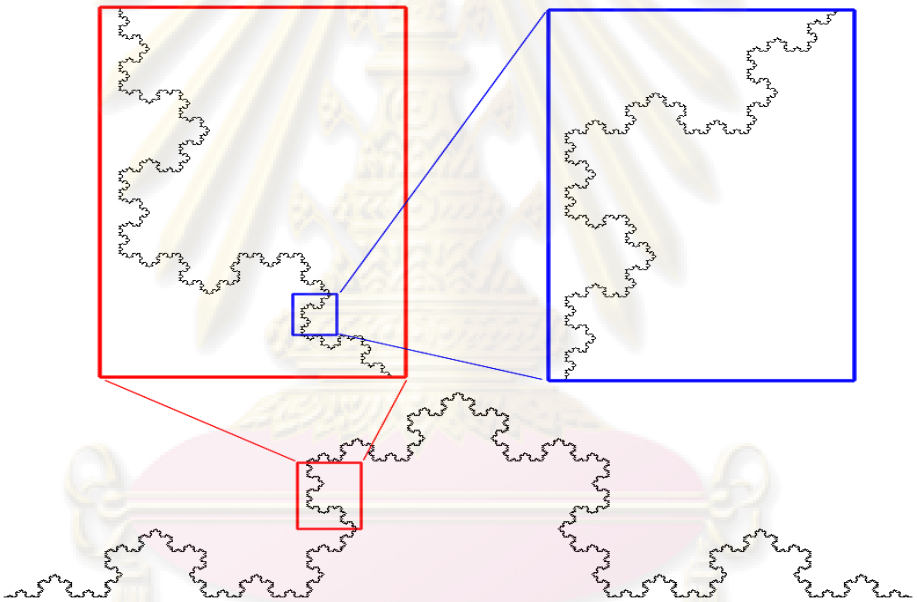
ภาพที่สร้างจากสมการทางคณิตศาสตร์ ภาพที่สร้างโดยธรรมชาติ

รูปที่ 2.1 ลักษณะคล้ายคลึงตัวเองของแฟร็กทัลที่สร้างจากสมการทางคณิตศาสตร์ (ซ้าย) และสร้างโดยธรรมชาติ (ขวา) (ที่มา : Bourke และ Miqel [9, 12])

ข้อมูลที่แสดงความเป็นแฟร็กทัล โดยทั่วไป สามารถจำแนกลักษณะที่แตกต่างกันได้หลายวิธี โดยในบางภาพมีลักษณะคล้ายคลึงตัวเองทุกส่วนไม่ว่าจะสังเกตไปที่ส่วนใดของภาพ หรืออาจมีลักษณะคล้ายคลึงตัวเองบางส่วน และอีกวิธีหนึ่งซึ่งสามารถสังเกตความคล้ายคลึงกันได้ค่อนข้างยาก คือ ข้อมูลที่มีลักษณะคล้ายคลึงตัวเองในเชิงสถิติ คุณลักษณะของแฟร็กทัลที่พบโดยทั่วไปมีดังนี้

2.3.1 ความคล้ายคลึงตัวเองทุกส่วน (Exact Self Similarity)

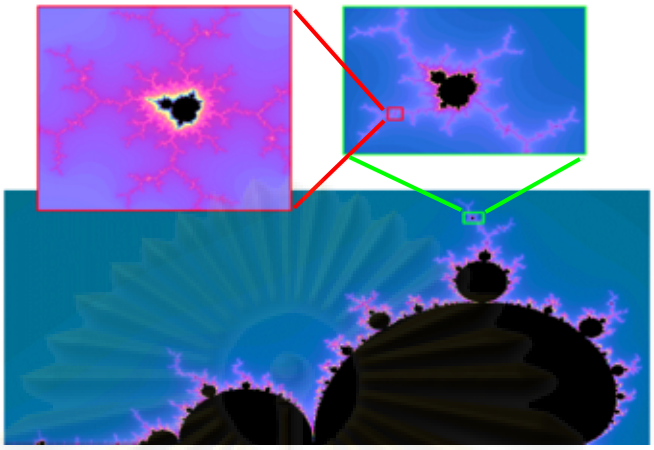
แฟร็กทัลที่มีคุณลักษณะคล้ายคลึงตัวเองทุกส่วน ถึงแม้ว่าจะขยายภาพไปที่ระดับใดก็ตาม จะเกิดคุณสมบัติคล้ายคลึงตัวเองเสมอ ดังแสดงในรูปที่ 2.2 ซึ่งแสดงความคล้ายคลึงตัวเองทุกส่วนของเส้นโค้งคอช (Koch Curve) จะเห็นได้ว่า ภาพในระดับที่ละเอียดขึ้นยังคงไว้ซึ่งลักษณะของภาพที่หยาบกว่า



รูปที่ 2.2 คุณสมบัติความคล้ายคลึงตัวเองทุกส่วนของเส้นโค้งคอช (ที่มา : Bourke [13])

2.3.2 ความคล้ายคลึงตัวเองบางส่วน (Quasi-Self Similarity)

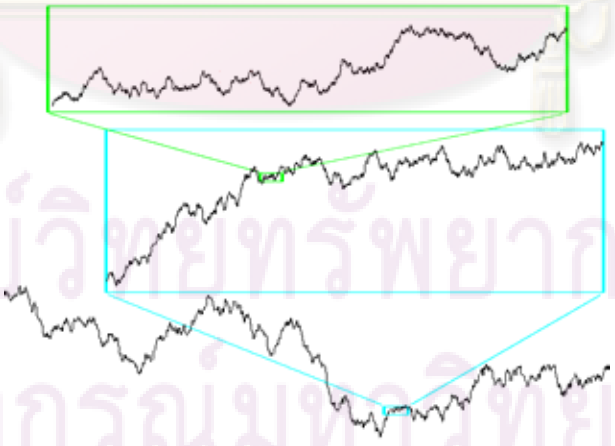
แฟร็กทัลชนิดนี้มีส่วนที่คล้ายคลึงตัวเองในบางส่วน เมื่อเราขยายภาพเข้าไปในส่วนนั้น จะมีลักษณะบางส่วนที่เหมือนกับภาพที่ยังไม่ได้ขยาย ดังแสดงในรูปที่ 2.3 ซึ่งแสดงความคล้ายคลึงตัวเองบางส่วน เมื่อขยายภาพให้ละเอียดมากขึ้นจากภาพตั้งต้น ภาพที่ขายได้ยังคงไว้ซึ่งลักษณะของภาพเดิมในบางส่วน ซึ่งแฟร็กทัลชนิดนี้มีคุณสมบัติคล้ายคลึงตัวเองน้อยกว่าแฟร็กทัลในชนิดแรก



รูปที่ 2.3 คุณสมบัติความคล้ายคลึงตัวเองบางส่วน (ที่มา : Bourke [13])

2.3.3 ความคล้ายคลึงตัวเองเชิงสถิติ (Statistical Self Similarity)

แฟร็กทัลในเชิงสถิติเป็นชนิดที่มีความคล้ายคลึงตัวเองน้อยที่สุด ซึ่งอยู่ในลักษณะของข้อมูลอนุกรมเวลา โดยจะมีคาบที่มีลักษณะคล้ายคลึงกันเป็นช่วง ๆ และเมื่อพยายามตรวจสอบค่าของข้อมูลในบางช่วง โครงสร้างของข้อมูลมีลักษณะคล้ายคลึงตัวเองในเชิงสถิติ ดังแสดงในรูปที่ 2.4 เมื่อทำการขยายเพื่อดูช่วงของข้อมูลอนุกรมเวลาในบางช่วง จะแสดงคุณลักษณะความคล้ายคลึงตัวเอง โดยมีโครงสร้างคล้ายกับค่าของข้อมูลก่อนหน้า ดังนั้นถ้ามีข้อมูลอนุกรมเวลาที่มีโครงสร้างเป็นคาบซ้ำ ๆ กัน ก็น่าจะแสดงคุณลักษณะของแฟร็กทัลประเภทนี้ได้ จึงเป็นจุดเริ่มต้นในการนำแฟร็กทัลมาพัฒนากับข้อมูลอนุกรมเวลา



รูปที่ 2.4 คุณสมบัติความคล้ายคลึงตัวเองเชิงสถิติ (ที่มา : Bourke [13])

2.3.4 ตัวอย่างข้อมูลที่แสดงลักษณะแฟร็กทัล

ข้อมูลที่แสดงลักษณะของความเป็นแฟร็กทัลที่นิยมนำมากล่าวถึง เพื่ออธิบายความคล้ายคลึงตัวเองมีดังนี้

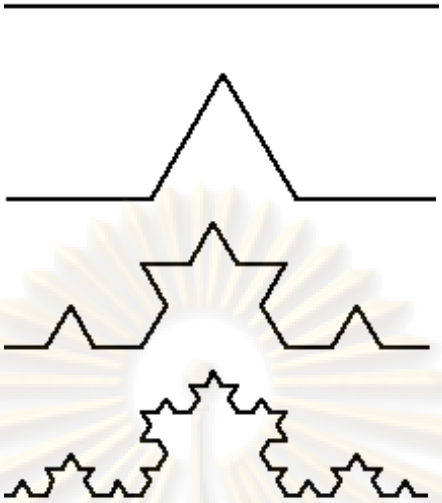
- ฝุ่นแคนทอร์ (Cantor Dust)

ขั้นตอนสำหรับการสร้างฝุ่นแคนทอร์ [14] สามารถแสดงได้ดังรูปที่ 2.5 โดยทำการแบ่งเส้นตรงหนึ่งเส้นเป็นสามส่วนที่มีขนาดเท่า ๆ กัน แล้วนำเส้นตรงในส่วนตรงกลางออก และกระทำซ้ำแบบเดิมโดยเส้นตรงที่เหลือถูกแบ่งเป็นสามส่วน แล้วนำส่วนตรงกลางออก ซึ่งจะกระทำซ้ำจนถึงอนันต์ จะเห็นว่า ภาพที่ถูกแบ่งแล้วเมื่อเปรียบเทียบกับข้อมูลก่อนหน้า ไม่ว่าทำการขยายภาพไปที่ความละเอียดเท่าใด ก็จะมีลักษณะคล้ายคลึงตัวเอง โดยฝุ่นแคนทอร์จะมีมิติอยู่ระหว่างเส้นตรงและจุด

รูปที่ 2.5 การสร้างฝุ่นแคนทอร์ (ที่มา : Cantor [14])

- เส้นโค้งคอช (Koch Curve)

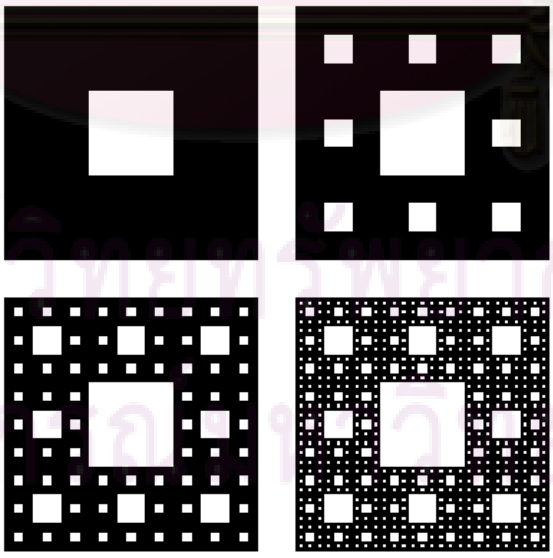
ขั้นตอนการสร้างเส้นโค้งคอช [15] สามารถแสดงได้ดังรูปที่ 2.6 โดยทำการแบ่งเส้นตรงเป็นสามส่วนเท่า ๆ กัน แล้วนำเส้นตรงที่มีขนาดเท่ากันกับเส้นที่ถูกแบ่งสองเส้นมาสร้างเป็นสามเหลี่ยมโดยประกอบกับเส้นตรงกลางแล้วนำเส้นตรงกลางออก และกระทำซ้ำจนถึงอนันต์ เมื่อพยายามปรับความละเอียดเท่าใดก็จะแสดงคุณลักษณะของความเป็นแฟร็กทัลเสมอ โดยเส้นโค้งคอชมีมิติอยู่ระหว่างเส้นตรงและระนาบ



รูปที่ 2.6 การสร้างเส้นโค้งคอช (ที่มา : Clayton [16])

- พรมเซอร์พินสกี (Sierpinski Carpet)

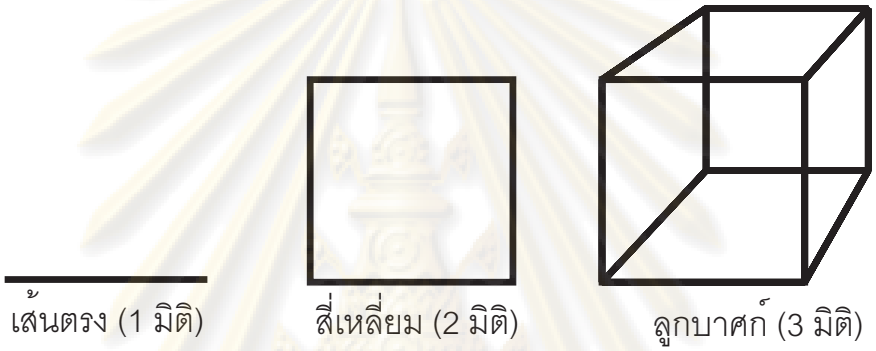
ขั้นตอนการสร้างพรมเซอร์พินสกี [17] สามารถแสดงได้ดังรูปที่ 2.7 โดยทำการแบ่งพื้นที่ของกล่องเป็นด้านละ 3x3 เท่า ๆ กัน และนำกล่องตรงกลางออก แล้วกระทำซ้ำจนถึงอนันต์ ซึ่งยังคงแสดงคุณสมบัติแฟร็กทัลเมื่อพยายามปรับความละเอียดในระดับใดก็ตาม จะเห็นว่า พรมเซอร์พินสกีมีโครงสร้างอยู่ระหว่างเส้นตรงที่เชื่อมต่อกันกับระนาบ



รูปที่ 2.7 การสร้างพรมเซอร์พินสกี (ที่มา : Dickau [17])

2.4 มิติแฟร็กทัล (Fractal Dimension)

ทฤษฎีของแฟร็กทัลที่มีคุณสมบัติความคล้ายคลึงตัวเองนั้น ได้ก่อให้เกิดงานวิจัยในอีกด้านหนึ่งเพื่อใช้ในการคำนวณหามิติของข้อมูล ซึ่งเรียกว่า มิติแฟร็กทัล [10, 11] ก่อนที่จะบรรยายถึงความหมายและวิธีการคำนวณมิติแฟร็กทัล จะกล่าวถึงมิติที่คนทั่วไปเข้าใจและสามารถคำนวณหามิติข้อมูลโดยใช้สามัญสำนึกเมื่อเห็นข้อมูลนั้น ซึ่งเรียกว่า มิติทอพอโลยี (Topological Dimension) [11] ยกตัวอย่างเช่น มิติของข้อมูลจุด มีค่าเท่ากับ 0 มิติ มิติของข้อมูลเส้นตรงหรือเส้นโค้ง มีค่าเท่ากับ 1 มิติ มิติของข้อมูลระนาบ มีค่าเท่ากับ 2 มิติ และมิติของข้อมูลลูกบาศก์ มีค่าเท่ากับ 3 มิติ



รูปที่ 2.8 รูปและมิติของเส้นตรง สี่เหลี่ยม และลูกบาศก์

จากรูปที่ 2.8 แสดงรูปเส้นตรงมีความยาวของข้อมูลเท่ากับ 1^1 หน่วย รูปสี่เหลี่ยมมีพื้นที่ของข้อมูลเท่ากับ 1^2 ตารางหน่วย และรูปลูกบาศก์มีปริมาตรของข้อมูลเท่ากับ 1^3 ลูกบาศก์หน่วย ซึ่งมีมิติเท่ากับ 1 2 และ 3 ตามลำดับ ถ้าขยายขนาดแต่ละข้อมูลเป็น 2 เท่า ค่าของแต่ละชนิดข้อมูลที่ถูกขยายมีขนาดเท่ากับ 2 หน่วย ในเส้นตรง เท่ากับ 4 ตารางหน่วย ในสี่เหลี่ยม และเท่ากับ 8 ลูกบาศก์หน่วย ในลูกบาศก์ ซึ่งสามารถเขียนในรูปยกกำลังเป็น 2^1 หน่วย 2^2 ตารางหน่วย และ 2^3 ลูกบาศก์หน่วย ตามลำดับ ถ้าเพิ่มขนาดของข้อมูลเป็นสามเท่าในแต่ละข้อมูล ค่าที่ได้ในรูปเลขยกกำลังเท่ากับ 3^1 หน่วย 3^2 ตารางหน่วย และ 3^3 ลูกบาศก์หน่วย ตามลำดับ จากตัวอย่างจะเห็นได้ว่า มิติของข้อมูลก็คือเลขยกกำลังนั่นเอง จากที่กล่าวมากำหนดให้ มิติของข้อมูล ให้สัญลักษณ์เป็น D เลขฐานที่ถูกเพิ่มหรือลดตามขนาดข้อมูล ให้สัญลักษณ์เป็น s และขนาดของข้อมูลรวม ให้สัญลักษณ์เป็น N สูตรแสดงความสัมพันธ์ในการหามิติทอพอโลยี คือ

$$N = s^D \tag{2.3}$$

จะเห็นได้ว่า ค่ามิติของข้อมูลที่คนทั่วไปเข้าใจ หรือมิติทอพอโลยีถูกแสดงด้วยเลขจำนวนเต็มบวก แต่ในความเป็นจริงแล้วมีวิธีเพื่อคำนวณหามิติอย่างละเอียด โดยผลของการ

คำนวณมิติที่ได้ อาจไม่เป็นจำนวนเต็มบวก แต่สามารถแสดงได้ด้วยเลขจำนวนจริงบวก วิธีที่ใช้ในการคำนวณหามิตินี้ เรียกว่า มิติแฟร็กทัล โดยส่วนมากนิยมนำไปคำนวณหามิติของข้อมูลชนิดรูปภาพเป็นหลัก

มิติแฟร็กทัลเป็นวิธีคำนวณหามิติของข้อมูลจากโครงสร้างภายในที่มีความสัมพันธ์กัน ผลของการคำนวณได้มิติที่เป็นเลขจำนวนจริงบวก ซึ่งมีหลากหลายวิธีในการคำนวณหามิติด้วยวิธีแฟร็กทัล เช่น มิติความคล้ายคลึงตัวเอง (Self-similarity Dimension) มิติฮอส์ดอร์ฟ (Hausdorff Dimension) มิตินับช่อง (Box-counting Dimension) มิติเรนยี (Renyi Dimension) มิติเส้นขอบ (Compass Dimension) มิติไลยาปูนอฟ (Lyapunov Dimension) มิติสารสนเทศ (Information Dimension) และมิติความสัมพันธ์ (Correlation Dimension)

สำหรับหัวข้อต่อไปจะกล่าวถึงขั้นตอนการคำนวณมิติแฟร็กทัล ซึ่งมีวิธีการคำนวณที่แตกต่างกัน โดยเริ่มจาก มิติความคล้ายคลึงตัวเองซึ่งเป็นวิธีที่แสดงความเป็นแฟร็กทัลได้ดี และกล่าวถึงมิติแฟร็กทัลอื่น ๆ ที่งานวิจัยนี้นำมาใช้กับข้อมูลอนุกรมเวลาเพื่อนำมาลดขนาดข้อมูล

2.4.1 มิติความคล้ายคลึงตัวเอง (Self-Similarity Dimension)

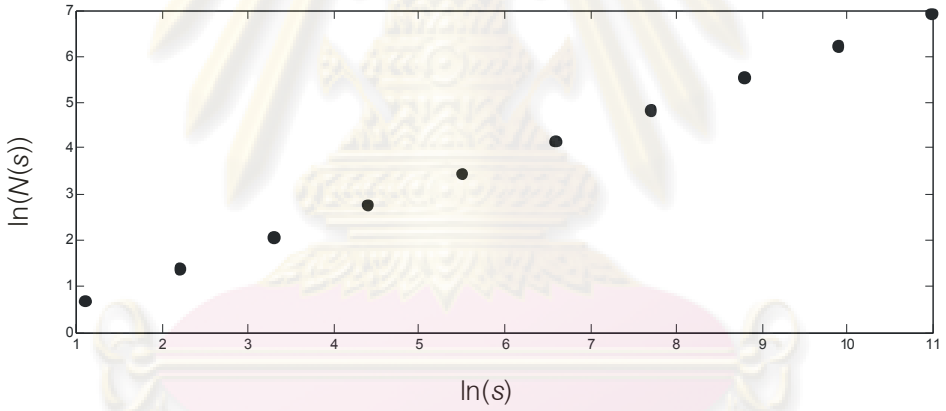
มิติความคล้ายคลึงตัวเอง [10] เป็นวิธีการคำนวณหาค่ามิติวิธีหนึ่งในมิติแฟร็กทัล โดยดูจากโครงสร้างความคล้ายคลึงตัวเองเมื่อเปรียบเทียบกับโครงสร้างข้อมูลก่อนหน้า ยกตัวอย่างเช่น ฝุ่นแคนทอร์ ดังแสดงในรูปที่ 2.5 สถานะเริ่มต้นเป็นเส้นตรงหนึ่งเส้น เมื่อทำการแบ่งเป็นเส้นตรงสามส่วนเท่า ๆ กันแล้วนำเส้นกลางออก ซึ่งในขั้นตอนแรกใช้ตัวแบ่ง s เท่ากับ 3^1 หน่วย ดังนั้น ขั้นตอนที่สอง จะได้เส้นตรงสองเส้นที่มีขนาดเล็กลง ซึ่งจะเห็นว่าเส้นตรงสองเส้นที่ได้ในขั้นตอนที่ 2 มีลักษณะคล้ายคลึงกับเส้นในขั้นตอนแรก เพราะฉะนั้น ในขั้นตอนที่ 2 มีจำนวนเส้นตรงซึ่งคล้ายคลึงกับขั้นตอนแรกเท่ากับ 2^1 เส้น ขั้นตอนที่ 3 แบ่งเส้นตรงสองเส้นในขั้นตอนที่ 2 โดยแบ่งเส้นตรงแต่ละเส้นเป็นสามส่วนเท่า ๆ กัน แล้วนำเส้นกลางออก ซึ่งตัวแบ่ง s ในรอบนี้จะเท่ากับ 3^2 หน่วยและมีจำนวนเส้นตรงที่คล้ายคลึงตัวเองเมื่อเปรียบเทียบกับเส้นตรงเริ่มต้นเท่ากับ 2^2 เส้น ซึ่งเมื่อกระทำไปจนถึงอนันต์ ตัวแบ่งของฝุ่นแคนทอร์ s มีค่าเท่ากับ 3^k หน่วย และจำนวนความคล้ายคลึงตัวเอง $N(s)$ ในแต่ละรอบของตัวแบ่ง s เท่ากับ 2^k เส้น โดยที่ k ลู่เข้าสู่อนันต์ สูตรในการคำนวณหามิติความคล้ายคลึงแสดงได้ดังนี้

$$D_S = \lim_{s \to \infty} \frac{\ln N(s)}{\ln s} \tag{2.4}$$

โดยที่ D_s คือมิติความคล้ายคลึงตัวเอง s คือค่าของส่วนที่ถูกแบ่งในแต่ละขั้นตอน $N(s)$ คือจำนวนที่คล้ายคลึงกันของโครงสร้างภายในข้อมูลเมื่อเปรียบเทียบกับข้อมูลเริ่มต้น ในแต่ละรอบของการแบ่ง s ดังนั้น ค่าของ D_s เมื่อผ่านการกระทำครั้งแรก เท่ากับ

ln2/ln3 = 0.6309 ค่ามิติที่ผ่านการกระทำในครั้งที่สอง เท่ากับ ln4/ln9 =$\ln 2^2/\ln 3^2$ = 0.6309 ดังนั้น เมื่อกระทำไปจนถึงอนันต์ มิติของฝุ่นแคนทอร์ D_s เท่ากับ 0.6039 เสมอ ซึ่งเป็นมิติที่อยู่ระหว่างจุด (มิติเท่ากับ 0) และเส้นตรง (มิติเท่ากับ 1) หรือเมื่อนำค่าของตัวแบ่ง *s* และจำนวนรอบในแต่ละครั้งของตัวแบ่ง *N*(*s*) ของทุกคู่ไปกำหนดจุดบนกราฟ แล้วคำนวณหาความชันของจุดทั้งหมดโดยที่ค่าลิมิต *s* ลู่เข้าสู่อนันต์ กำหนดให้แกนนอนเป็น ln(*s*) และแกนตั้งเป็น ln(*N*(*s*)) จะได้ค่าความชันเท่ากับ 0.6039 ซึ่งก็คือค่ามิติแฟร็กทัลของฝุ่นแคนทอร์ รูปที่ 2.9 แสดงการคำนวณจุดบนกราฟเพื่อหาค่าความชันของจุดทั้งหมดของข้อมูลฝุ่นแคนทอร์ โดยกระทำซ้ำทั้งหมด 10 ครั้ง ค่าความชันที่ได้คือค่ามิติความคล้ายคลึงตัวเองเท่ากับ 0.6039

จากวิธีการมิติความคล้ายคลึง สามารถหาค่ามิติของข้อมูลอื่น ๆ ได้ดังนี้ รูปที่ 2.6 มิติความคล้ายคลึงของเส้นโค้งคอชเท่ากับ 1.2618 ซึ่งคล้ายเส้นตรงค่อนข้างมาก (1 มิติ) และมีบางส่วนที่แสดงลักษณะของระนาบ (2 มิติ) จากรูปที่ 2.7 ค่ามิติความคล้ายคลึงของพรมเซอร์ปินสกี้ เท่ากับ 1.8927 ซึ่งคล้ายระนาบค่อนข้างมาก (2 มิติ) และมีบางส่วนที่แสดงลักษณะของเส้นตรง (1 มิติ)



รูปที่ 2.9 ความชันจากจุดที่วาดบนกราฟเพื่อหามิติความคล้ายคลึงของฝุ่นแคนทอร์

จากสมการที่ (2.3) ถ้านำลอการิทึมธรรมชาติมาใส่เข้าทั้งสองข้างจะทำให้ได้สมการที่เหมือนกับสมการที่ (2.4) ดังนี้

$$N = s^D$$

$$\ln(N) = \ln(s^D)$$

$$\ln(N) = D\ln(s) \qquad (2.5)$$

$$D = \frac{\ln(N)}{\ln(s)}$$

ในสมการที่ (2.3) ค่า *D* กำหนดให้เป็นจำนวนเต็มบวกซึ่งเป็นมิติทอพอโลยี แต่ในสมการที่ (2.4) ค่า *D* กำหนดให้เป็นจำนวนจริงบวกซึ่งเป็นมิติแฟร็กทัล ดังนั้น สูตรการคำนวณหาค่ามิติทอพอโลยีและมิติแฟร็กทัลมีสูตรเดียวกัน แต่มีวิธีการคำนวณที่ต่างกันทำให้มิติแฟร็กทัลสามารถคำนวณมิติได้ละเอียดกว่า

2.4.2 มิติเส้นขอบ (Compass Dimension)

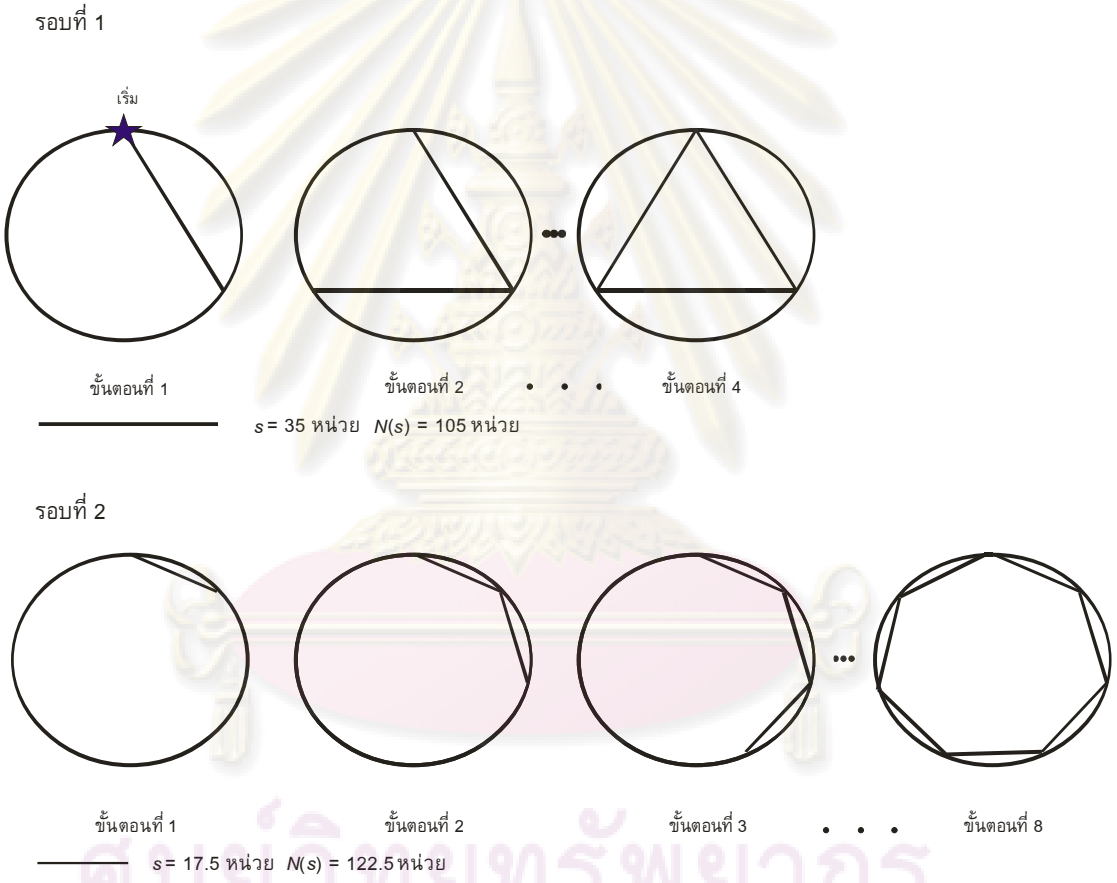
มิติเส้นขอบ (Compass Dimension-D_C) [10] มีหลักการคำนวณที่แตกต่างกับมิติความคล้ายคลึง โดยมิติเส้นขอบเป็นการหาความคล้ายคลึงโดยการอ้างอิงจากความยาวโดยรอบของข้อมูล *N*(*s*) ในแต่ละระดับของความยาวเส้นขอบที่นำมาลาก *s* (Compass Segment) โดยลากจากจุดเริ่มต้นด้วยความยาวเส้นขอบที่กำหนดอย่างต่อเนื่องจนวกกลับมาที่จุดเริ่มต้นอีกครั้ง ซึ่งความยาวทั้งหมดที่ได้คือความยาวโดยรอบของข้อมูล *N*(*s*) นั่นเอง และต่อมาจะทำการวัดความยาวของข้อมูลอีกครั้งด้วยความยาวของเส้นขอบที่ลดลงเป็นครึ่งหนึ่งของความยาวเดิม และกระทำการลากเส้นใหม่อีกครั้ง ซึ่งจะกระทำซ้ำอย่างต่อเนื่องจนถึงอนันต์ในที่สุด แต่ละคู่ของความยาวโดยรอบของข้อมูลในแต่ละรอบ *N*(*s*) และค่าของความยาวเส้นขอบที่ใช้ในการลาก *s* ถูกนำมากำหนดจุดบนมาตราส่วนลอการิทึม (Logarithm Scale) แล้วคำนวณหาค่าความชัน ดังแสดงในสมการที่ 2.6 ค่าของความชันที่ได้คือ มิติเส้นขอบ สูตรการคำนวณของมิติเส้นขอบคือ

$$D_C = \lim_{s \to 0} \frac{\ln N(s)}{\ln(\frac{1}{s})} \tag{2.6}$$

โดยที่ D_c คือมิติเส้นขอบ *s* คือค่าความยาวของเส้นขอบที่ใช้ในการลากรอบรูปซึ่งจะถูกแบ่งครึ่งในแต่ละครั้ง *N*(*s*) คือความยาวรวมของเส้นขอบทั้งหมดเมื่อเปรียบเทียบกับ *s* ในแต่ละครั้งของการคำนวณ จะเห็นว่า สูตรการคำนวณมิติเส้นขอบคล้ายคลึงกับทั้งสมการที่ (2.4) และ (2.5) ดังนั้น สูตรในการคำนวณค่ามิติอยู่บนพื้นฐานเดียวกัน แต่วิธีคำนวณมีความแตกต่างกัน

ในรูปที่ 2.10 แสดงตัวอย่างการลากเส้นขอบ *s* รอบรูปวงกลมเพื่อหาความยาวในแต่ละรอบ สำหรับขั้นตอนในการคำนวณค่ามิติเส้นขอบ เริ่มจากทำการกำหนดความยาวเส้นขอบ *s* แล้วนำเส้นนั้นมาวางรอบขอบรูปวัตถุอย่างต่อเนื่อง และวัดความยาวของเส้นขอบทั้งหมดจากการลากเส้นรอบข้อมูล โดยรอบที่ 1 กำหนดความยาวเส้นขอบ *s* ที่มีความยาวเท่ากับ 35 หน่วย โดยเริ่มจากการสุ่มจุด แล้ววัดความยาวจากจุดตั้งต้นไปตามขอบของวัตถุจนกระทั่งความยาวเท่ากับ 35 หน่วย แล้วลากเส้น ต่อมาเริ่มวัดจากจุดปลายของเส้นก่อนหน้าแล้วทำการลากเส้นให้ได้ความยาว 35 หน่วย อีกครั้ง กระทำซ้ำจนจุดเริ่มต้นของการวัดครั้งแรกพบจุดปลายของการวัดในสุดท้าย แล้วทำการคำนวณความยาวของเส้นขอบที่ลากในรอบนี้

ทั้งหมด จะได้ความยาวรวมของเส้นขอบ $N(s)$ ที่ความยาวเส้นขอบเท่ากับ 35 หน่วย คือ 3 เส้น x 35 หน่วย เท่ากับ 105 หน่วย และในขั้นตอนต่อไป ทำการกำหนดความยาวของเส้นขอบให้สั้นลงเป็นครึ่งหนึ่งของความยาวเดิม จากความยาว 35 หน่วย เป็น 17.5 หน่วย และนำความยาวเส้นขอบนี้ไปใช้วัดในรอบต่อไป ในรอบที่ 2 จะได้ความยาวรวมของเส้นขอบ $N(s)$ ที่ความยาวเส้นขอบเท่ากับ 17.5 หน่วย คือ 7 เส้น x 17.5 หน่วย เท่ากับ 122.5 หน่วย และลดความยาวเส้นขอบ s เป็นครึ่งหนึ่งอีกครั้ง และกระทำการวัดแบบเดิมซ้ำ จนกระทั่งความยาวเส้นขอบลู่เข้าสู่ 0 ในที่สุด นำค่าแต่ละคู่ของความยาวเส้นขอบ s และความยาวรวมทั้งหมดของการลากในแต่ละความยาวเส้นขอบ $N(s)$ มากำหนดจุดบนมาตราส่วนลอการิทึม ความชันที่คำนวณได้ของจุดทั้งหมดคือค่ามิติเส้นขอบ (D_C)



รูปที่ 2.10 วิธีการลากเส้นขอบรอบรูปวัตถุเพื่อหาความยาวรอบรูป ในแต่ละเส้นขอบ s

2.4.3 มิติความสัมพันธ์ (Correlation Dimension)

ในปี 1938 Grassberger และ Procaccia [18] ได้นำเสนอวิธีแนวคิดใหม่สำหรับคำนวณหามิติแฟร็กทัล โดยใช้แนวคิดของความสัมพันธ์ที่เกิดขึ้นระหว่างจุดภายในข้อมูล เรียกวิธีนี้ว่า มิติความสัมพันธ์ (Correlation Dimenson-D_{CR}) [19-21] ซึ่งวิธีนี้ถูกนำไปใช้แพร่หลายในปัจจุบันกับข้อมูลที่มีค่าเป็นตัวเลข มิติความสัมพันธ์คำนวณความคล้ายคลึงของ

ข้อมูลได้จากวิธีอินทิกรัลความสัมพันธ์ (Correlation Integral-CI) คือ เริ่มต้น ทำการกำหนดขีดแบ่งระยะทาง *r* (Threshold Distance) เพื่อใช้ในการคำนวณหาความสัมพันธ์ระหว่างจุดภายในต่อมาทำการวัดระยะทางจากจุดหนึ่งที่กำหนดไว้เปรียบเทียบกับจุดอื่น ๆ ทุกจุดภายในข้อมูล ถ้าระยะห่างระหว่างจุดที่กำหนดไว้กับจุดใด ๆ มีค่าน้อยกว่าหรือเท่ากับขอบเขตระยะทางที่กำหนดไว้ แสดงว่าจุดที่กำหนดไว้และจุดที่นำมาเปรียบเทียบนั้นมีความสัมพันธ์กัน จะถูกนับให้มีค่าเท่ากับหนึ่ง แต่ถ้าไม่ใช่ แสดงว่าจุดที่กำหนดไว้และจุดที่นำมาเปรียบเทียบนั้นไม่มีความสัมพันธ์กัน จะมีค่าเท่ากับศูนย์ และกระทำซ้ำอย่างต่อเนื่องจนกระทั่งครบทุกจุดของข้อมูล และสุดท้ายนำค่าความสัมพันธ์ของทุก ๆ จุดที่คำนวณได้มารวมกันทั้งหมด ซึ่งจะได้ค่าอินทิกรัลความสัมพันธ์มาหนึ่งค่า ที่ขอบเขตระยะทางเท่ากับค่าที่กำหนดไว้ตามความเหมาะสม สูตรในการคำนวณหาอินทิกรัลความสัมพันธ์ แสดงได้ดังแสดงในสมการที่ 2.7

$$C_{CI}(r) = \lim_{N\to\infty} \frac{1}{N}\sum_{i=1}^{N}\frac{1}{N}\sum_{j=1}^{N}\Theta(r - |\vec{x}_i - \vec{x}_j|) \tag{2.7}$$

โดยที่ *N* เป็นจำนวนจุดทั้งหมดของข้อมูล $\vec{x}$ เป็นจุดใด ๆ ภายในข้อมูล ส่วน *r* เป็นขีดแบ่งระยะทาง ที่ถูกกำหนดขึ้นเพื่อหาความสัมพันธ์ระหว่างจุดซึ่งในแต่ละขั้นตอนจะถูกลดลงเป็นครึ่งหนึ่งของค่าก่อนหน้าและ $\Theta(x)$ เป็นเฮวีไซด์สเต็บฟังก์ชัน (Heaviside Step Function) แสดงสมการได้ดังนี้

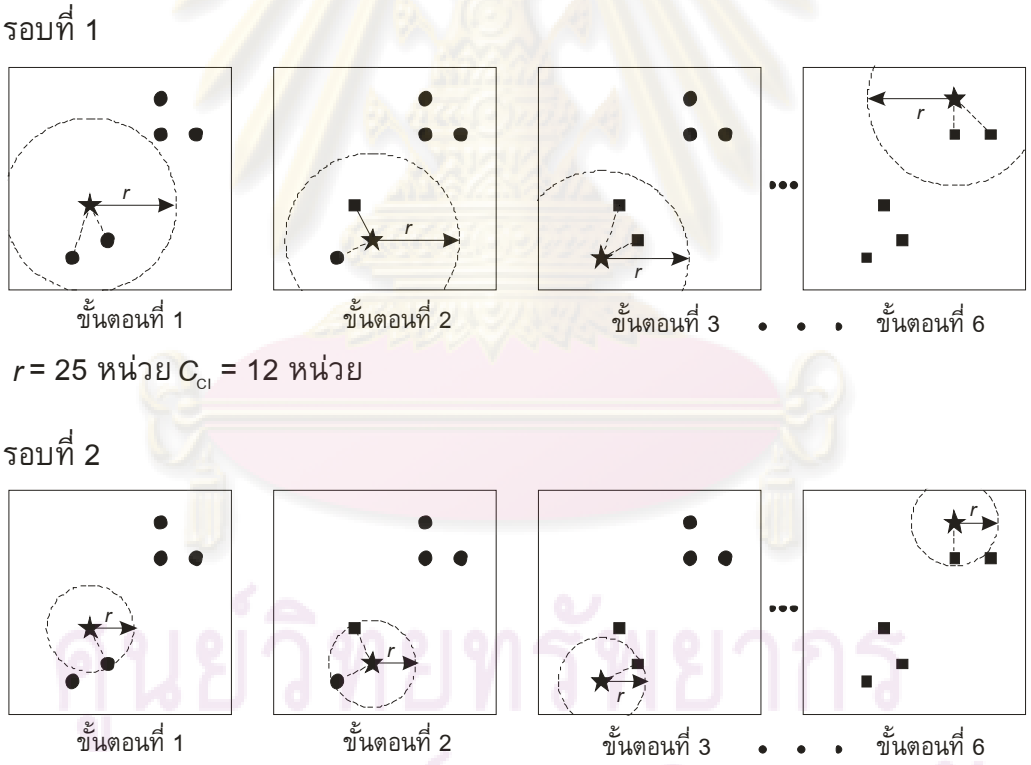
$$\Theta(x) = \begin{cases} 0 & \text{when } x \le 0 \\ 1 & \text{when } x > 0 \end{cases} \tag{2.8}$$

หลังจากการคำนวณหาอินทิกรัลความสัมพันธ์ในครั้งแรก ต่อมาทำการลดขีดแบ่งระยะทาง *r* ให้เหลือครึ่งหนึ่งของค่าก่อนหน้า ต่อมาคำนวณค่าอินทิกรัลความสัมพันธ์ในครั้งต่อไป และกระทำซ้ำจนกระทั่งขอบเขตระยะทาง *r* ลู่เข้าสู่ศูนย์ ดังนั้น แต่ละคู่ของขีดแบ่งระยะทาง *r* และค่าอินทิกรัลความสัมพันธ์ในแต่ละขอบเขตระยะทาง $C_{CI}(r)$ จะถูกนำมากำหนดจุดบนมาตราส่วนลอการิทึม ดังแสดงในสมการที่ 2.9 และความชันที่คำนวณได้จากจุดทั้งหมดคือ มิติความสัมพันธ์ สมการแสดงการคำนวณหาค่ามิติความสัมพันธ์แสดงได้ดังนี้

$$D_{CR} = \lim_{r\to 0} \frac{\ln C_{CI}(r)}{\ln r} \tag{2.9}$$

จากรูปที่ 2.11 แสดงตัวอย่างการคำนวณหามิติความสัมพันธ์ในแต่ละรอบ กับข้อมูลจุดที่มีการกระจายตัวกัน โดยเริ่มต้นจากกำหนดขีดแบ่งระยะทาง *r* เท่ากับ 25 หน่วย เพื่อใช้สำหรับคำนวณหาความสัมพันธ์ระหว่างจุด ในรอบที่ 1 กำหนดจุดเริ่มต้นโดยการสุ่มจุด (รูปดาว) และทำการคำนวณระยะทางกับทุกจุดในปริภูมิ ซึ่งถ้าหากจุดใดมีค่าน้อยกว่าขอบเขตระยะทาง *r* หรือจากรูปแสดงด้วยวงกลมเส้นประ จะถูกนับค่าความสัมพันธ์ของจุดนี้ โดยแต่ละ

จุดจะมีค่าเท่ากับหนึ่ง ดั้งนั้น ในขั้นตอนที่ 1 ได้ความสัมพันธ์ระหว่างจุดเท่ากับ 2 หน่วย ต่อมาในขั้นตอนที่ 2 จะทำการสุ่มจุดที่จะหาความสัมพันธ์ โดยจุดที่ทำการคำนวณแล้วแสดงด้วยรูปสี่เหลี่ยม และหาค่าความสัมพันธ์จากขอบเขตระยะทาง *r* กับทุกจุดในปริภูมิ แล้วจะได้ความสัมพันธ์ของจุดที่สองเท่ากับ 2 หน่วย ซึ่งจะกระทำซ้ำเพื่อหาความสัมพันธ์กับจุดที่เหลือ เมื่อคำนวณครบทุกจุดจะนำความสัมพันธ์ของแต่ละจุดที่ได้มารวมกัน ซึ่งผลรวมที่ได้คือ อินทริกัลความสัมพันธ์ของขีดแบ่งระยะทาง *r* ในครั้งแรก โดยมีค่าเท่ากับ 12 หน่วย ในรอบต่อไป เริ่มจากการลดขนาดของขีดแบ่งระยะทาง *r* ให้เล็กลง โดยส่วนมากจะลดลงเป็นครึ่งหนึ่งของค่าก่อนหน้า จาก 25 หน่วย เป็น 12.5 หน่วย แล้วนำขีดแบ่งระยะทาง *r* ที่ได้ไปหาความสัมพันธ์ระหว่างจุดใด ๆ กับทุกจุดในปริภูมิ ด้วยเพื่อหาค่าอินทิกรัลความสัมพันธ์อีกครั้ง ในรอบที่สอง ซึ่งเมื่อกระทำซ้ำแบบเดิมจะได้ค่าอินทริกรัลความสัมพันธ์กับขีดแบ่งระยะทาง r เท่ากับ 8 หน่วย และกระทำซ้ำจนขีดแบ่งระยะทางลู่เข้าสู่ศูนย์ ในที่สุด แต่ละคู่ของขอบเขตระยะทาง *r* และค่าของอินทิกรัลความสัมพันธ์ในแต่ละขอบเขตระยะทาง $C_{CI}(r)$ จะถูกนำมากำหนดจุดบนมาตราส่วนลอการิทึม ซึ่งความชันที่ได้เป็นค่ามิติความสัมพันธ์ (D_{CR})

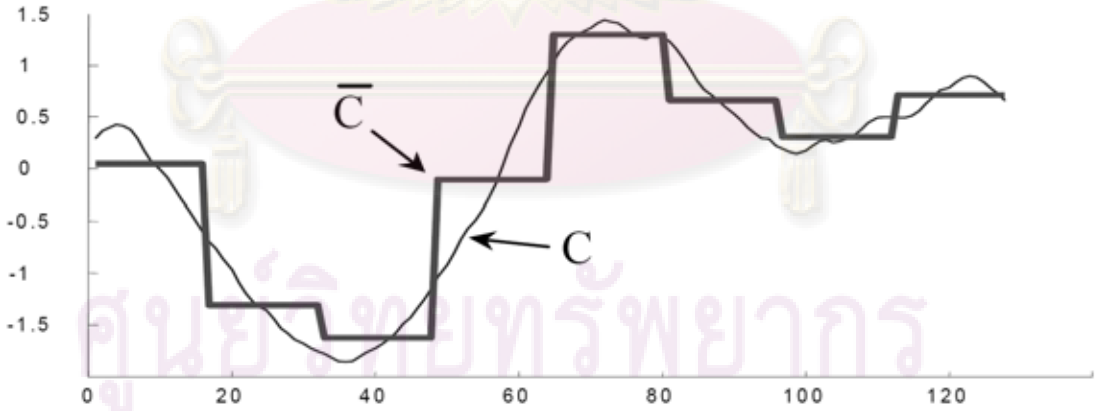


รูปที่ 2.11 วิธีคำนวณอินทริกรัลความสัมพันธ์ ในแต่ละขีดแบ่งระยะทาง *r*

2.5 งานวิจัยที่เกี่ยวข้อง

งานวิจัยของข้อมูลอนุกรมเวลาที่ผ่านมา มุ่งเป้าหมายไปที่สองแนวทางหลัก ๆ คือ เพื่อเพิ่มประสิทธิภาพด้านความเร็ว และความแม่นยำสำหรับการทำเหมืองข้อมูลอนุกรมเวลา มีงานวิจัยจำนวนมากเน้นไปที่การเพิ่มประสิทธิภาพในด้านความเร็ว โดยนำเสนอในวิธีต่าง ๆ เช่น การแทนที่ข้อมูล (Data Representation) หรือการลดขนาดข้อมูล (Dimensionality Reduction) เป็นต้น สำหรับงานวิจัยต่าง ๆ ที่นำเสนอสำหรับการลดขนาดของข้อมูลอนุกรมเวลา มีดังนี้

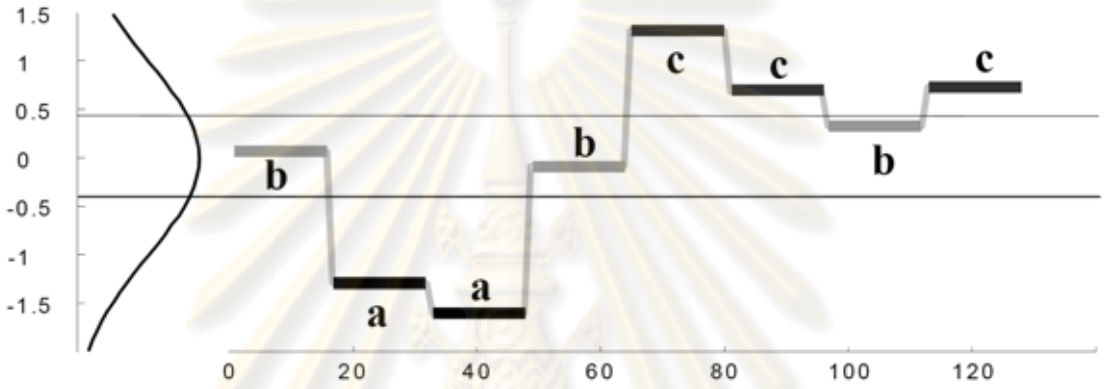
Yi และ Faloutsos [5] เสนอวิธีการลดขนาดข้อมูลแบบพีเอเอ (Piecewise Aggregate Approximation-PAA) ซึ่งเริ่มต้นการคำนวณจากการแบ่งช่วงของข้อมูลอนุกรมเวลาให้มีช่วงที่เท่ากัน แล้วคำนวณหาค่าเฉลี่ยในช่วงนั้นซึ่งได้ค่าหนึ่งค่า แทนช่วงหนึ่งช่วง ดังแสดงในรูปที่ 2.12 โดยเริ่มต้นต้องกำหนดความยาวที่ต้องการลดขนาดที่เหมาะสมกับข้อมูลอนุกรมเวลา ยกตัวอย่างเช่น ข้อมูลอนุกรมเวลาความยาว 128 จุด ต้องการลดลงเหลือความยาว 8 จุด ดังนั้น ทุก ๆ 16 จุด จะถูกลดเหลือ 1 จุด โดยใช้วิธีการหาค่าเฉลี่ยในช่วงนั้นเพื่อลดขนาดข้อมูลให้เหลือ 1 จุด โดยเรียกวิธีนี้ว่าการลดขนาดข้อมูลแบบพีเอเอ จะเห็นว่า วิธีนี้ต้องมีการกำหนดขนาดความยาวที่เหมาะสมสำหรับการลดขนาดข้อมูลอนุกรมเวลา ทำให้การเลือกความยาวเป็นปัจจัยสำคัญ ซึ่งถ้าลดขนาดของข้อมูลมากเกินไป อาจส่งผลต่อความแม่นยำที่ลดลง หรือถ้าลดขนาดของข้อมูลอนุกรมเวลาน้อยเกินไป เวลาที่ใช้ก็ไม่สามารถลดได้มากเท่าที่ควร



รูปที่ 2.12 การลดขนาดข้อมูลอนุกรมเวลาแบบพีเอเอ (ที่มา : Lin และ Keogh [6])

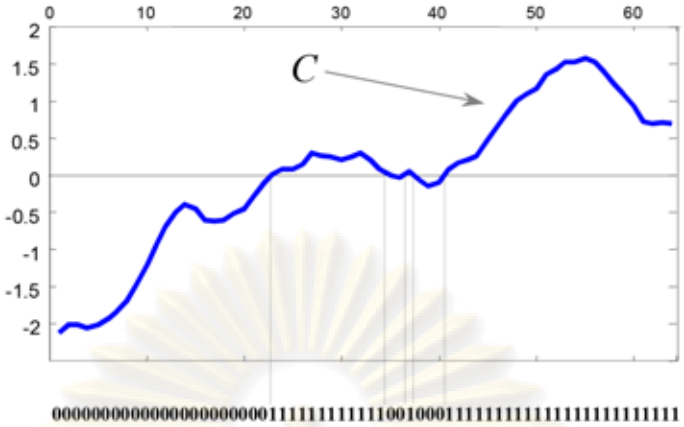
Lin และ Keogh [6, 7] นำเสนอวิธีการลดขนาดข้อมูลแบบแซค (Symbolic Aggregate Approximation-SAX) โดยเริ่มต้นใช้หลักการของวิธีพีเอเอ แล้วทำการเปลี่ยนแปลงข้อมูลพีเอเอเหล่านั้นให้เป็นสัญลักษณ์แทนภายใต้การแจกแจงแบบเกาส์เซียน (Gaussian Distribution) ในการคำนวณทุกครั้ง ข้อมูลอนุกรมเวลาต้องทำการปรับระดับด้วยคะแนน Z (Z-

score Normalization) ยกตัวอย่างเช่น ต้องการลดขนาดข้อมูลอนุกรมเวลาจาก 128 จุด เหลือ 8 จุด โดยแสดงผลเฉพาะสัญลักษณ์ A B C หรือ D ในข้อมูลอนุกรมเวลา ดังนั้น ทุก ๆ 16 จุด จะถูกลดเหลือ 1 จุด และการแสดงผลด้วยสัญลักษณ์จะกำหนดช่วงโดยใช้วิธีแจกแจงแบบเกาส์เซียน ดังแสดงในรูปที่ 2.13 วิธีนี้จะเกิดปัญหาคล้ายกับวิธีพีเอเอ ซึ่งการเลือกขนาดความยาวที่ต้องการลดของข้อมูลอนุกรมเวลาเป็นปัจจัยที่สำคัญเช่นกัน และต้องทำการกำหนดพารามิเตอร์ของจำนวนสัญลักษณ์ที่ต้องการแสดงผลด้วย ทำให้การเลือกจำนวนสัญลักษณ์ที่ต่างกันอาจทำให้ผลความแม่นยำมีความแตกต่างกัน



รูปที่ 2.13 การลดขนาดข้อมูลอนุกรมเวลาแบบแซค (ที่มา : Lin และ Keogh [6])

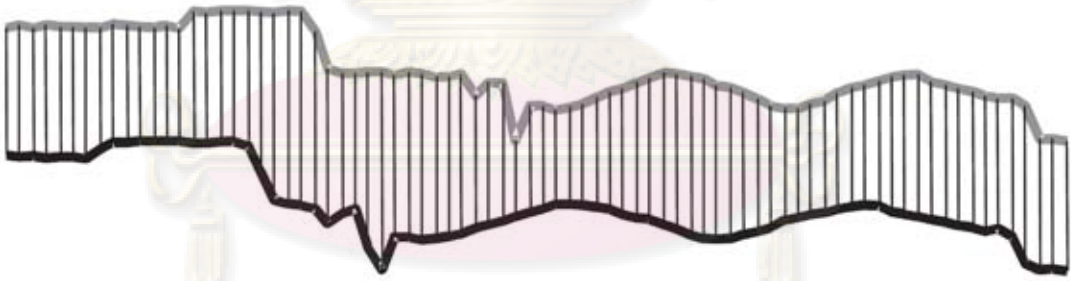
Bagnall และ Janacek [8] นำเสนอวิธีการแทนข้อมูลแบบคลิป (Clipped Data Representation) ซึ่งเป็นวิธีการแปลงข้อมูลอนุกรมเวลาจากเดิมที่เป็นเลขจำนวนจริงให้เหลือเพียงเลขไบนารี (Binary) ดังแสดงในรูปที่ 2.14 โดยเริ่มจากการหาค่าเฉลี่ยของข้อมูลทั้งหมด ถ้าจุดใดมีค่ามากกว่าหรือเท่ากับค่าเฉลี่ยจะแทนค่านั้นเป็นหนึ่ง ถ้าไม่ใช่จะแทนค่านั้นเป็นศูนย์ แล้วทำจนครบทุกจุดของข้อมูลอนุกรมเวลา โดยผลของการแทนข้อมูลแบบคลิปในแนวแกนนอนที่มีค่าเท่ากันจะถูกนับรวมกันเป็นจำนวนครั้งแทน ยกตัวอย่างเช่น ข้อมูลอนุกรมเวลา ซึ่งแสดงในรูปที่ 2.14 มีความยาวเท่ากับ 64 จุด จะถูกแทนข้อมูลด้วยไบนารีเป็นเลข 1 และ 0 ดังแสดงในส่วนล่างของรูป และผลของการเก็บข้อมูลด้วยวิธีคลิปจะแสดงผลเป็น ‘22 zeros|12|2|1|3|24’ คือ มีในช่วงแรกมี 0 ซ้ำกัน 22 ครั้ง ในช่วงที่สองมี 1 ซ้ำกัน 12 ครั้ง จนถึงช่วงที่ 6 มี 1 ซ้ำกัน 24 ครั้ง โดย zeros แสดงถึงค่าเริ่มต้น และตัวต่อไปสลับกันระหว่าง 0 กับ 1 ซึ่งจะทำให้การเก็บข้อมูลมีขนาดเล็กลง จากการลดขนาดข้อมูลด้วยวิธีนี้ จะเห็นว่า ถ้าข้อมูลต่างชนิดกันมีแนวโน้มการขึ้นลงที่คล้ายกันของโครงสร้างโดยรวม แต่ข้อมูลหนึ่งมีโครงสร้างภายในที่ค่อนข้างเรียบ กับอีกข้อมูลมีการขึ้นลงตลอดเวลา จะทำให้มีโอกาสการจับคู่ที่ผิดพลาดได้ง่าย



รูปที่ 2.14 การแทนข้อมูลแบบคลิป (ที่มา : Bagnall และ Janacek [8])

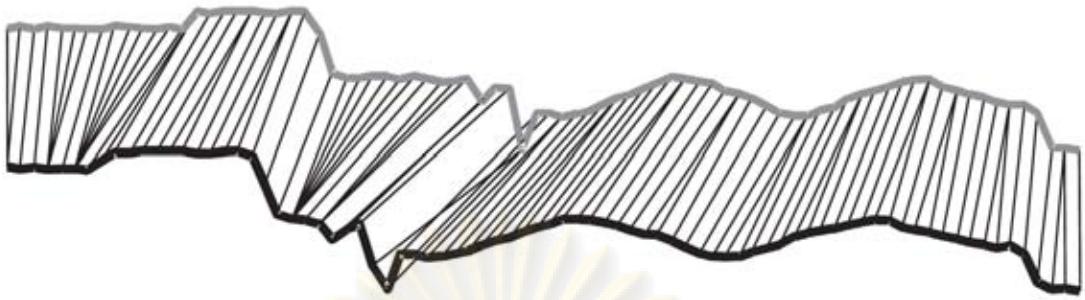
งานวิจัยอีกด้านหนึ่งของการทำเหมืองข้อมูลอนุกรมเวลาคือการพัฒาประสิทธิภาพของความแม่นยำ โดยการค้นหาความคล้ายคลึงด้วยวิธีการวัดระยะทาง ยกตัวอย่างเช่น

การวัดระยะทางแบบยุคลิด เป็นวิธีการวัดระยะทางแบบจุดต่อจุด ดังแสดงในรูปที่ 2.15 ทำให้ใช้เวลาในการคำนวณน้อย แต่ผลความแม่นยำที่ได้ค่อนข้างต่ำ เมื่อพบข้อมูลประเภทเดียวกันที่มีการเหลื่อมกันมาก จะทำให้ค่าระยะทางที่ได้มีความแตกต่างกัน



รูปที่ 2.15 การวัดระยะทางแบบยุคลิด (ที่มา : Keogh [2])

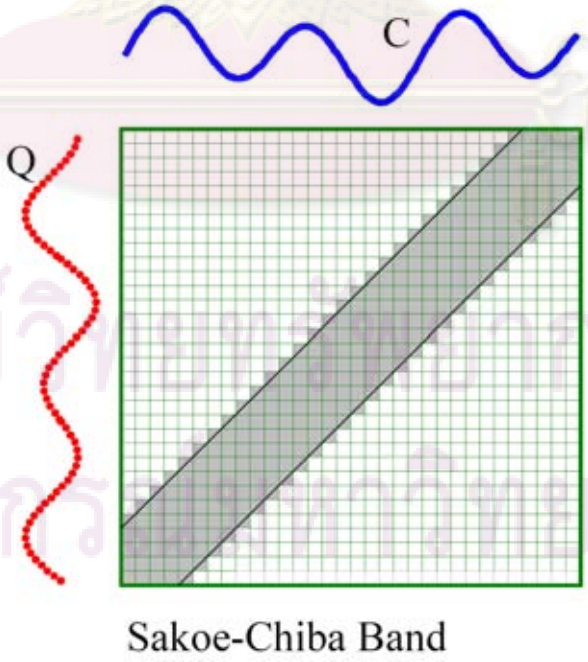
การวัดระยะทางแบบไดนามิกไทม์วอร์ปปิง [1, 2] เป็นวิธีคำนวณหาระยะทางจากจุดหนึ่งเทียบกับทุกจุด โดยใช้กำหนดการพลวัต (Dynamic Programming) ดังแสดงในรูปที่ 2.16 ซึ่งในแต่ละจุดจะจับคู่กับจุดที่วัดระยะทางได้สั้นที่สุดเมื่อเปรียบเทียบกับจุดอื่น ๆ ซึ่งเป็นวิธีที่ดีที่สุดสำหรับข้อมูลอนุกรมเวลาขนาดสั้น แต่เวลาที่ใช้ในการคำนวณที่มากทำให้การนำไปใช้กับข้อมูลอนุกรมเวลาขนาดใหญ่ยังไม่เป็นที่นิยมมากนัก



รูปที่ 2.16 การวัดระยะทางแบบไดนามิกไทม์วอร์ปปิง (ที่มา : Keogh [2])

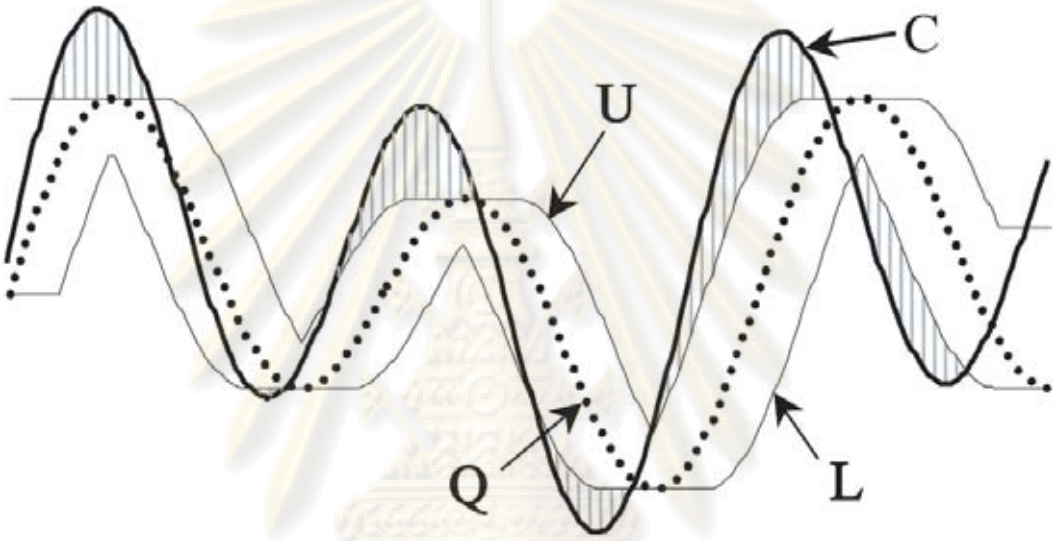
สำหรับการวัดระยะทางแบบไดนามิกไทม์วอร์ปปิงได้มีงานวิจัยเพิ่มเติมเพื่อพัฒนาประสิทธิภาพเพิ่มขึ้นซึ่งจะนำมาทดสอบร่วมกันในงานวิจัยนี้คือ เงื่อนไขบังคับโดยรวมซาโก-ชิบะ และฟังก์ชันขอบเขตล่าง ดังนี้

เงื่อนไขบังคับโดยรวมซาโก-ชิบะ [1] ดังแสดงในรูปที่ 2.17 เป็นวิธีที่พัฒนาขึ้นเพื่อเพิ่มประสิทธิภาพในด้านความเร็วสำหรับไดนามิกไทม์วอร์ปปิงโดยจะกำหนดขอบเขตในการคำนวณของการโปรแกรมแบบไดนามิก (Dynamic Programing) ให้มีการคำนวณในช่วงที่เหมาะสม เนื่องจากการวอร์ปไปยังที่ตำแหน่งไกล ๆ ไม่น่าจะเกิดขึ้นได้ และยังเพิ่มประสิทธิภาพของความแม่นยำด้วยในกรณีที่ข้อมูลอาจจะวอร์ปผิดตำแหน่งไปยังตำแหน่งที่ไกลเกินไปทำให้การกำหนดขอบเขตการวอร์ปเป็นช่วยเพิ่มประสิทธิพภาพทั้งสองได้เป็นอย่างดี



รูปที่ 2.17 เงื่อนไขบังคับโดยรวมซาโก-ชิบะ (ที่มา : Ratanamahatana และ Keogh [1])

ฟังก์ชันขอบเขตล่าง [2] ดังแสดงในรูปที่ 2.18 พัฒนาขึ้นมาเพื่อเพิ่มประสิทธิภาพในด้านความเร็วเป็นหลัก โดยเป็นเทคนิคในการตัดข้อมูลที่ไม่น่าจะเป็นประเภทเดียวกันทิ้งด้วยการสร้างซอง (Envelope) มาครอบข้อมูลอนุกรมเวลา ซึ่งจะได้ขอบเขตบน *U* และขอบเขตล่าง *L* ต่อมาเมื่อนำข้อมูลอนุกรมเวลาตัวอื่นมาวัดระยะทาง *C* จะใช้ซองที่สร้างขึ้นนี้เป็นตัวเปรียบเทียบด้วยการวัดระยะทางแบบยุคลิดก่อน โดยถ้าระยะทางที่วัดได้จากซองน้อยกว่าค่าที่น้อยที่สุดของการเปรียบเทียบกับข้อมูลอนุกรมเวลาตัวก่อน ถึงจะเข้าไปทำการคำนวณไดนามิกไทม์วอร์ปปิงจริง *Q* แต่ถ้าไม่ใช่จะไม่ต้องเข้าไปคำนวณข้อมูลดังกล่าว



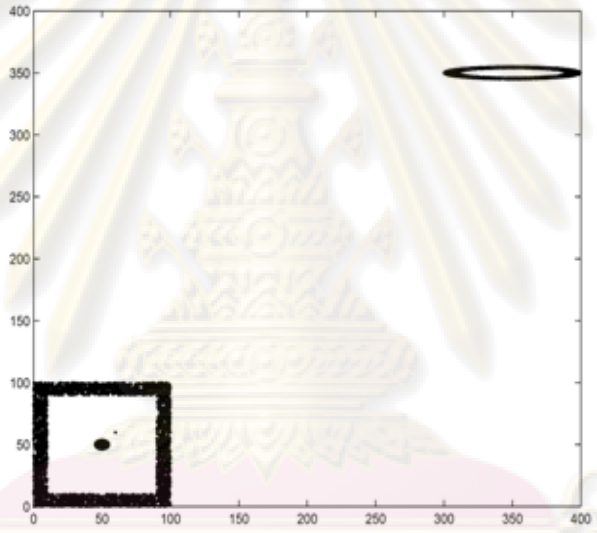
รูปที่ 2.18 การคำนวณฟังก์ชันขอบเขตล่าง (ที่มา : Keogh [2])

วิธีซีดีเอ็ม [4] เป็นวิธีที่พัฒนามาจากวิธีความซับซ้อนโคลโมโกรอฟ (Kolmogorov Complexity) โดยวิธีซีดีเอ็มเป็นการลดขนาดข้อมูลภายใต้การบีบอัดข้อมูล และทำการพัฒนาให้ง่ายขึ้นโดยใช้วิธีการบีบอัดข้อมูลทั่วไป เช่น วินซิป (WinZip) จีซิป (GZip) และวินราร์ (WinRar) เป็นต้น โดยการนำข้อมูลอนุกรมเวลาสองตัวมาคำนวณหาความคล้ายคลึงจากการบีบอัดข้อมูลแต่ละตัวแล้วนำขนาดที่บีบอัดได้มาบวกกันเปรียบเทียบกับการนำข้อมูลอนุกรมเวลาสองตัวมาเชื่อมต่อกันเป็นหนึ่งข้อมูลก่อนทำการบีบอัดข้อมูล แต่เนื่องจาก เวลาที่เสียไปในส่วนของอินพุต/เอาต์พุต (Input/Output) มีผลกระทบค่อนข้างมาก และการนำมาใช้กับข้อมูลอนุกรมเวลาที่ถูกปรับระดับด้วยคะแนน Z (Z-score Normalization) จะส่งผลให้ช่วงค่าของข้อมูลอยู่ในระดับเดียวกัน ทำให้ผลของการค้นหาความคล้ายคลึงด้วยวิธีซีดีเอ็มแย่ลง

ส่วนงานวิจัยที่ใช้หลักการของมิติแฟร็กทัลยังไม่เป็นที่แพร่หลายในการทำเหมืองข้อมูล ทั้งข้อมูลอนุกรมเวลาและข้อมูลแบบต่าง ๆ โดยมากวิธีเหล่านี้นิยมนำมาพัฒนากับข้อมูลรูปภาพเป็นหลัก และมีบางงานวิจัยสำหรับข้อมูลจุดที่กระจายตัวกัน ซึ่งในที่นี้จะ

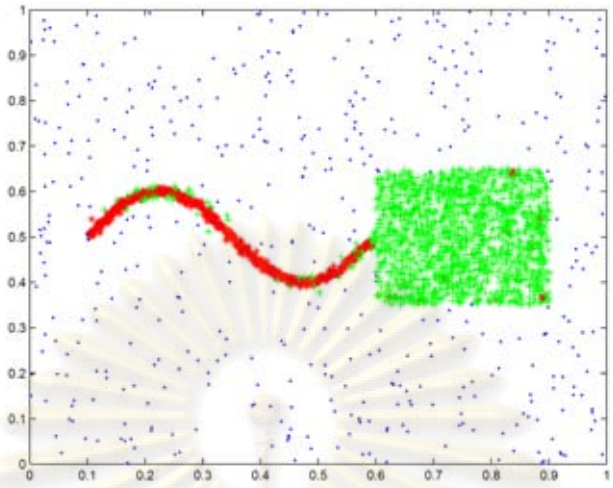
ยกตัวอย่างเฉพาะข้อมูลที่เป็นจุด ซึ่งมีลักษณะคล้ายคลึงกับข้อมูลอนุกรมเวลามากกว่าข้อมูลรูปภาพ ยกตัวอย่างเช่น

Barbarà และ Chen [22] ใช้มิติแฟร็กทัลเพื่อจำแนกประเภทของข้อมูลจุดที่กระจายตัวกันเป็นกลุ่ม ดังแสดงในรูปที่ 2.19 โดยเริ่มจากการสุ่มตัวอย่าง เพื่อเรียนรู้จำนวนประเภทที่มีทั้งหมดในกลุ่มข้อมูลนั้น เมื่อคำนวณหาจำนวนประเภทที่มีแล้ว ซึ่งได้ค่าเท่ากับ 2 กลุ่มคือ จุดที่กระจายตัวเป็นรูปสี่เหลี่ยม และจุดที่กระจายตัวเป็นรูปวงรี ต่อมาข้อมูลที่เหลือทั้งหมดจะถูกนำมารวมกับข้อมูลแต่ละประเภทแล้วคำนวณค่ามิติแฟร็กทัลโดยใช้มิตินับช่อง ถ้าผลที่ได้มากกว่าขีดแบ่ง (Threshold) ที่กำหนดไว้ แสดงว่าข้อมูลจุดไม่อยู่ในประเภทนั้น และถ้าคำนวณกับทุกประเภทแล้วไม่สามารถหาประเภทของข้อมูลนั้นได้ จะสร้างประเภทใหม่ขึ้นมา ซึ่งก็คือ จุดที่กระจายตัวเป็นรูปวงกลมทึบในรูปสี่เหลี่ยม



รูปที่ 2.19 ข้อมูลจุดที่กระจายตัวกันเป็นกลุ่ม (ที่มา : Barbarà และ Chen [22])

Gionis และ Hinneburg [23] ใช้มิติความสัมพันธ์เพื่อแบ่งกลุ่มของข้อมูลจุดที่กระจายตัวกันโดยมีรูปร่างที่แตกต่างกันในเชิงมิติ ดังแสดงในรูปที่ 2.20 ซึ่งจะมีการจุดที่กระจายตัวกันเป็นรูปสี่เหลี่ยมและเส้นโค้ง โดยงานวิจัยนี้จะทำการแยกสองส่วนนี้ออกจากกันเป็น 2 ประเภท และงานวิจัยนี้ได้นำเสนอมิติความสัมพันธ์แบบเฉพาะที่ (Local-correlation Dimension) เป็นวิธีหาความสัมพันธ์โดยคำนวณจากความสัมพันธ์ที่เกิดขึ้นโดยเทียบกับจุดอื่น ๆ และวัดอัตราการเติบโตของจุดที่อยู่ใกล้เคียงโดยทำการเปลี่ยนขีดแบ่งระยะทาง (Threshold Distance) และกระทำจนครบทุกจุด จุดที่มีอัตราการเติบโตใกล้เคียงกันแสดงว่าจุดเหล่านั้นเป็นจุดที่อยู่ในประเภทเดียวกัน



รูปที่ 2.20 ข้อมูลจุดที่กระจายตัวกันโดยมีรูปร่างแตกต่างกันในเชิงมิติ (ที่มา : Gionis และ Hinneburg [23])

Xiao และคณะ [24] ใช้มิติแฟร็กทัลเพื่อจำแนกประเภทการบันทึกคลื่นไฟฟ้าของกล้ามเนื้อ (Electromyography-EMG) เพื่อจำแนกข้อมูลที่มีอยู่ 2 ประเภท คือ การหงายแขนท่อนปลาย (Forearm Supination-FS) และการคว่ำแขนท่อนปลาย (Forearm Pronation-FP) โดยคำนวณมิติแฟร็กทัลด้วยมิติความสัมพันธ์ แล้วทดสอบกับข้อมูลที่มีการผ่านตัวกรองความถี่ผ่านต่ำ (Low-pass Filter) แต่สำหรับข้อมูลดิบที่ไม่มีการกรองข้อมูล ผลที่ได้จากการคำนวณยังคงมีความซ้ำซ้อนกันระหว่างข้อมูลที่ต่างประเภทกัน และสำหรับข้อมูลที่ยังไม่มีการกรองความถี่ต่ำ ข้อมูลจะกระจายตัวกันโดยไม่สามารถแบ่งแยกจากกันได้เลย

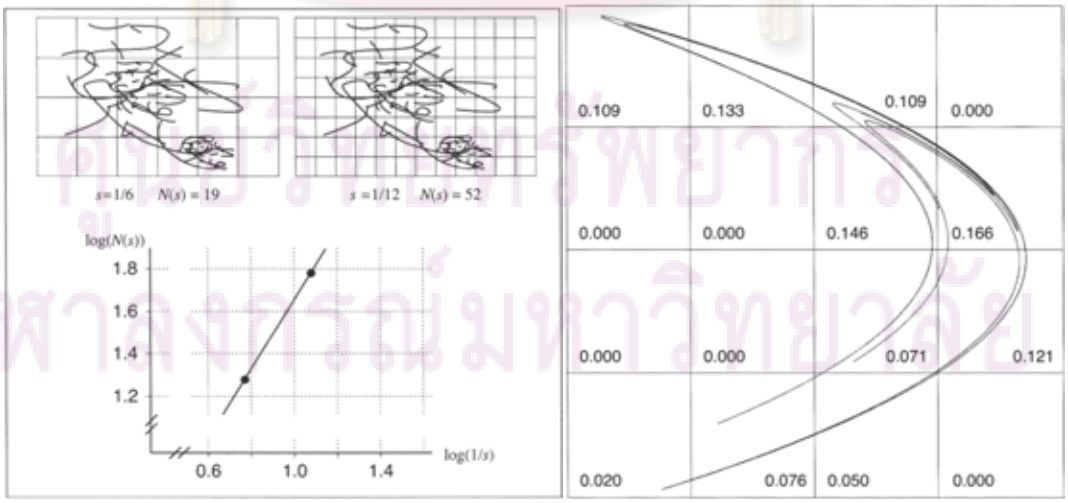
บทที่ 3

การแทนข้อมูลแบบแฟร็กทัล

สำหรับบทที่ 3 ผู้วิจัยได้นำเสนอแนวทางใหม่สำหรับการลดขนาดข้อมูลอนุกรมเวลาด้วยการแทนข้อมูลแบบแฟร็กทัล โดยเริ่มต้นจะกล่าวถึงแนวคิดสำหรับการนำมิติแฟร็กทัลมาพัฒนากับชนิดข้อมูลอนุกรมเวลา แนวทางเพื่อประยุกต์มิติแฟร็กทัลโดยนำไปใช้สำหรับการแทนข้อมูลแบบแฟร็กทัลกับข้อมูลอนุกรมเวลา ต่อมาบรรยายถึงระบบโดยรวมของการแทนข้อมูลแบบแฟร็กทัล และในส่วนสุดท้ายอธิบายการทำงานของมิติเส้นขอบที่นำไปพัฒนาสำหรับการแทนข้อมูลแบบแฟร็กทัลในสองแนวทาง คือ มิติเส้นขอบตามความยาวที่เท่ากัน และมิติเส้นขอบตามความกว้างที่เท่ากัน

3.1 มิติแฟร็กทัล

งานวิจัยโดยทั่วไปของทฤษฎีมิติแฟร็กทัลไม่ได้ถูกนำมาประยุกต์อย่างกว้างขวางกับงานทำเหมืองข้อมูลกันมากนัก ซึ่งจะเห็นได้จากงานวิจัยจำนวนไม่มากที่กล่าวในบทที่ 2 มิติแฟร็กทัลนิยมนำมาใช้สำหรับคำนวณหาค่ามิติของข้อมูลเป็นหลัก ชนิดข้อมูลที่นำมาใช้หาค่ามิติโดยส่วนมากคือ ข้อมูลรูปภาพ ซึ่งจะเห็นได้จากแนวคิดของมิติคล้ายคลึงตัวเอง โดยเป็นการนับจำนวนโครงสร้างที่คล้ายกันกับโครงสร้างเริ่มต้น เช่น ฝุ่นแคนทอร์ เส้นโค้งคอช เป็นต้น หรือมิติแฟร็กทัลอื่น ๆ ที่ไม่ได้กล่าวถึงในบทที่ 2 ส่วนมิติแฟร็กทัลที่นิยมนำมาประยุกต์กับข้อมูลรูปภาพคือ มิตินับช่อง (Box-counting Dimension) [10] และมิติสารสนเทศ (Information Dimension) [10] โดยทั้งสองวิธีนี้จะนำเสนอโดยการตีช่องสี่เหลี่ยมขนาดเท่า ๆ กันให้กับข้อมูลที่อยู่บนระนาบ ดังแสดงในรูปที่ 3.1



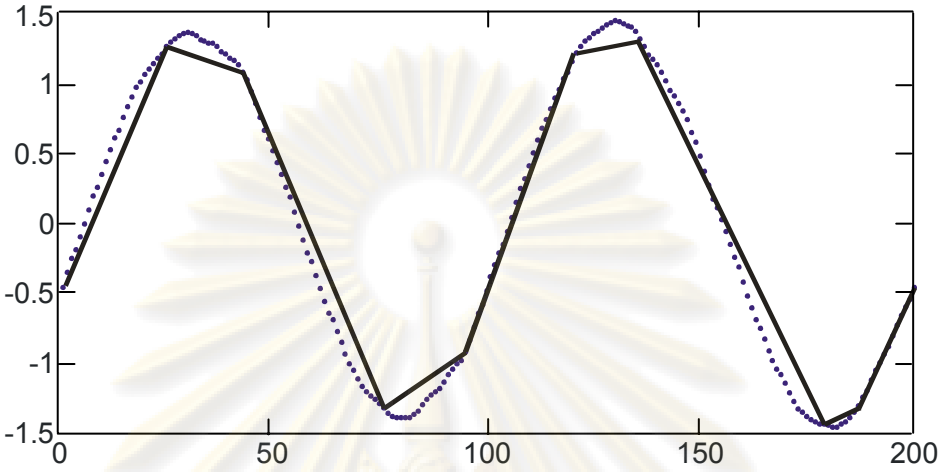
รูปที่ 3.1 วิธีการคำนวณค่ามิตินับช่อง (ซ้าย) และมิติสารสนเทศ (ขวา) (ที่มา : Peitgen[10])

จากรูปที่ 3.1 แสดงวิธีคำนวณมิตินับช่อง (ซ้าย) โดยตีช่องสี่เหลี่ยมให้มีขนาดเท่ากันทุกช่องลงบนข้อมูลและนับจำนวนช่องที่พบข้อมูล โดยกำหนดให้ช่องที่มีข้อมูลผ่าน มีค่าเท่ากับ 1 ถ้าช่องใดไม่มีข้อมูลผ่านให้มีค่าเท่ากับ 0 แล้วเก็บค่าจำนวนช่องที่พบข้อมูลทั้งหมด $N(s)$ ไว้ สำหรับคำนวณหาค่ามิตินับช่อง ต่อมากระทำซ้ำแบบเดิมตามหลักการของมิติแฟร็กทัล คือ ซอยช่องให้ละเอียดมากขึ้น โดยครั้งแรกแบ่งด้วยมาตราส่วนเท่ากับ 1/6 ในครั้งต่อไปแบ่งด้วยมาตราส่วนเท่ากับ 1/12 เป็นต้น แล้วนำค่าแต่ละครั้งที่คำนวณได้ในแต่ละมาตราส่วนของค่าจำนวนช่องที่พบข้อมูลทั้งหมด $N(s)$ และมาตราส่วนที่ทำการแบ่งในแต่ละรอบ s ไปคำนวณหาความชัน โดยค่าความชันที่ได้คือมิตินับช่อง

สำหรับรูปที่ 3.1 (ขวา) เป็นวิธีคำนวณมิติสารสนเทศ โดยมีหลักการคำนวณคล้ายกันกับมิตินับช่อง คือ แบ่งช่องให้มีขนาดเท่ากันบนข้อมูลรูปภาพ แต่วิธีนี้จะเพิ่มการกำหนดน้ำหนักเข้าไปในแต่ละช่อง โดยถ้าในช่องใดมีปริมาณของข้อมูลจำนวนมากกว่าจะถูกกำหนดให้มีความสำคัญมากขึ้น โดยกำหนดให้มีค่าน้ำหนักที่มาก หรือถ้าช่องใดมีปริมาณของข้อมูลน้อยจะกำหนดให้มีค่าน้ำหนักน้อยลง โดยผลรวมของค่าน้ำหนักทุกช่องมีค่าเท่ากับ 1 หน่วย แล้วกระทำในแนวทางเดียวกับมิตินับช่อง เพื่อคำนวณหาค่ามิติสารสนเทศ โดยวิธีการทั้งสองมีความสัมพันธ์กันในเชิงทฤษฎี คือ ค่าของมิติสารสนเทศจะเป็นขีดแบ่งล่างของค่ามิตินับช่อง เนื่องจาก มิตินับช่องจะนับจำนวนของแต่ละช่องที่พบข้อมูลเท่ากับ 1 หน่วย แต่มิติสารสนเทศจะให้ผลรวมค่าน้ำหนักของทุกช่องเท่ากับ 1 หน่วย ทำให้ค่ามิตินับช่องมีค่ามากกว่ามิติสารสนเทศเสมอ จากแนวคิดของวิธีทั้งสองถูกออกแบบเพื่อคำนวณหามิติของข้อมูลที่มีพื้นที่ในแนวระนาบหรือข้อมูลรูปภาพเป็นหลัก ดังนั้น การนำวิธีดังกล่าวมาประยุกต์ใช้ให้มีความเหมาะสมกับข้อมูลอนุกรมเวลาที่มีคุณลักษณะเป็นเลขจำนวนจริงและมีลำดับ โดยใช้หลักการตีช่องของวิธีทั้งสอง จึงเป็นไปได้ยากที่จะนำไปพัฒนาใช้สำหรับข้อมูลอนุกรมเวลา

ในบางมิติแฟร็กทัลนำมาพัฒนาเพื่อใช้สำหรับคำนวณกับข้อมูลที่สามารถวัดระยะทางโดยรอบของข้อมูล คือ มิติเส้นขอบ แนวคิดของวิธีนี้กล่าวไว้ในบทที่ 2 ซึ่งเห็นได้ว่าการวัดระยะทางตามแนวเส้นขอบของข้อมูลสำหรับมิติเส้นขอบมีความเป็นไปได้เพื่อนำมาประยุกต์ใช้กับข้อมูลอนุกรมเวลาซึ่งสามารถหาระยะทางจากจุดหนึ่งไปยังจุดหนึ่ง โดยใช้คุณสมบัติของข้อมูลอนุกรมเวลาที่มีลำดับต่อเนื่องกัน การคำนวณจะเริ่มจากจุดแรกของข้อมูลอนุกรมเวลา จนกระทั่งถึงจุดสุดท้าย แล้ววัดระยะทางตามแนวเส้นที่ลากทั้งหมด และเก็บค่าระยะทางทั้งหมดไว้เพื่อคำนวณหาค่ามิติเส้นขอบสำหรับข้อมูลอนุกรมเวลา ดังแสดงในรูปที่ 3.2 ซึ่งแสดงการลากเส้นระหว่างจุด จากจุดเริ่มต้นถึงจุดปลายสุดของข้อมูลอนุกรมเวลา โดยข้อมูลชนิดเดียวกัน น่าจะได้ผลลัพธ์ของระยะทางเส้นขอบโดยรวมในแต่ละรอบมีแนวโน้มที่ใกล้เคียงกัน ดังนั้น เมื่อคำนวณหาค่าความชันของมิติเส้นขอบ จะทำให้ข้อมูลอนุกรมเวลาชนิดเดียวกันให้ค่าจำนวนจริงที่ใกล้เคียงกัน ซึ่งค่าจำนวนจริงดังกล่าวสามารถแทนที่ข้อมูลอนุกรมเวลาขนาดใหญ่ให้เหลือเพียง 1 ค่า และนำไปประยุกต์ใช้กับการทำเหมืองข้อมูลอนุกรมเวลา ซึ่งเมื่อ

เปรียบเทียบกับงานวิจัยของการลดขนาดข้อมูลอนุกรมเวลาอื่น ๆ ยังไม่มีวิธีใดเลยที่สามารถลดขนาดข้อมูลอนุกรมเวลาให้เหลือเพียงตัวเลขไม่กี่จำนวน และให้ผลของความแม่นยำที่ดีเพียงพอ



รูปที่ 3.2 การลากเส้นตามแนวขอบของข้อมูลอนุกรมเวลาสำหรับมิติแฟร็กทัล

เมื่อผู้วิจัยได้ทำการทดลองในเบื้องต้นด้วยมิติแฟร็กทัล โดยลดขนาดของข้อมูลอนุกรมเวลาด้วยมิติเส้นขอบ ซึ่งจะได้ผลลัพธ์เป็นค่าจำนวนจริงเพียง 1 ค่า สำหรับแต่ละข้อมูลอนุกรมเวลา โดยผลของความแม่นยำที่ได้ค่อนข้างต่ำเมื่อเปรียบเทียบกับผลของการวัดความคล้ายคลึงแบบอื่น ๆ เนื่องจาก ช่วงของมิติสำหรับข้อมูลอนุกรมเวลาอยู่ระหว่างจุด (0 มิติ) และเส้นโค้ง (1 มิติ) ซึ่งจะส่งผลให้ค่าจำนวนจริงที่ได้ มีการซ้อนกันของข้อมูลต่างประเภทกัน ยกตัวอย่างเช่น มีข้อมูลอนุกรมเวลาจำนวน 4 อนุกรม และมี 2 ประเภท ซึ่งเมื่อคำนวณค่ามิติแฟร็กทัลของข้อมูลในประเภทที่ 1 ได้ผลลัพธ์เท่ากับ 0.5438 และ 0.5834 และค่ามิติแฟร็กทัลของข้อมูลในประเภทที่สองคือ 0.5812 และ 0.59123 ซึ่งจะเห็นว่า ถ้าทำการเรียงลำดับข้อมูลจะได้เป็น 0.5438 (ประเภทที่หนึ่ง) 0.5812 (ประเภทที่สอง) 0.5834 (ประเภทที่หนึ่ง) และ 0.59123 (ประเภทที่สอง) จากผลดังกล่าวทำให้เกิดการซ้อนกันของข้อมูลอนุกรมเวลาที่ต่างชนิดกัน ซึ่งส่งผลให้ความแม่นยำลดลง แต่ประเด็นต่อมา เมื่อผู้วิจัยเปรียบเทียบผลลัพธ์ของค่ามิติแฟร็กทัลกับทุกข้อมูล พบว่า ข้อมูลอนุกรมเวลาที่มีชนิดเดียวกันให้ผลลัพธ์ของค่าจำนวนจริงใกล้เคียงกัน และมีความเป็นไปได้ที่จะพัฒนามิติแฟร็กทัลให้มีประสิทธิภาพมากยิ่งขึ้น ในส่วนของผลการทดลองมิติแฟร็กทัลกับข้อมูลอนุกรมเวลา โดยใช้มิติเส้นขอบสำหรับการลดขนาดข้อมูลให้เหลือเพียงเลขจำนวนจริง 1 ค่า ที่กล่าวมาข้างต้นจะนำเสนอในบทที่ 4

สำหรับมิติเส้นขอบในงานวิจัยนี้ได้ประยุกต์การคำนวณออกเป็นสองแนวทาง ซึ่งแต่ละวิธีถูกเรียกว่า มิติเส้นขอบตามความยาวที่เท่ากัน (Equi-length Compass Dimension)

และมิติเส้นขอบตามความกว้างที่เท่ากัน (Equi-width Compass Dimension) ซึ่งจะอธิบายวิธีการคำนวณในหัวข้อถัดไป

สำหรับมิติแฟร็กทัลที่มีความเป็นไปได้เพื่อประยุกต์กับข้อมูลอนุกรมเวลาอีกวิธีหนึ่ง คือ มิติความสัมพันธ์ วิธีนี้ถูกพัฒนาเพื่อใช้กับข้อมูลที่มีลักษณะเป็นจุด โดยคำนวณหาความสัมพันธ์จากระยะทางระหว่างจุดทุกจุดในปริภูมิและเปรียบเทียบกับขีดแบ่งระยะทางในแต่ละรอบ เนื่องจาก ข้อมูลอนุกรมเวลามีคุณลักษณะเป็นจุด ดังนั้น จึงสามารถคำนวณหาระยะทางระหว่างจุดได้เช่นกัน ซึ่งข้อมูลชนิดเดียวกันน่าจะมีการกระจายตัวของจุดที่คล้ายคลึงกัน ทำให้ได้ผลลัพธ์ของค่าอินทิกรัลความสัมพันธ์มีแนวโน้มใกล้เคียงกัน ดังนั้น เมื่อนำค่าอินทิกรัลความสัมพันธ์และขอบเขตระยะทางมาคำนวณหาค่าความชัน ค่ามิติความสัมพันธ์ที่ได้น่าจะสามารถจำแนกข้อมูลอนุกรมเวลาชนิดเดียวกันได้จากการแทนด้วยตัวเลขจำนวนจริงเช่นเดียวกันกับมิติเส้นขอบ

จากการทดลองเพื่อหาค่าของมิติความสัมพันธ์สำหรับข้อมูลอนุกรมเวลา ซึ่งผลลัพธ์ของค่าจำนวนจริงที่ได้ในแต่ละข้อมูลอนุกรมเวลา มีค่าใกล้เคียงกันสำหรับข้อมูลประเภทเดียวกันเหมือนมิติเส้นขอบ จึงทำให้เกิดประเด็นต่อมาคือ ถ้าหากนำค่าจำนวนจริงแต่ละวิธีของมิติแฟร็กทัลที่สามารถจำแนกข้อมูลอนุกรมเวลาด้วยค่าจำนวนจริงที่ให้ผลใกล้เคียงกันกับข้อมูลอนุกรมเวลาชนิดเดียวกัน แล้วนำมาช่วยกันจำแนกประเภทของข้อมูลอนุกรมเวลา น่าจะทำให้ประสิทธิภาพของความแม่นยำเพิ่มสูงขึ้น โดยข้อมูลอนุกรมเวลาแต่ละตัวเมื่อลดขนาดข้อมูลจะแสดงด้วยค่าจำนวนจริงที่มากกว่าหนึ่งค่า ซึ่งวิธีการดังกล่าวเป็นวิธีที่นำเสนอในงานวิจัยนี้ คือ วิธีการลดขนาดข้อมูลอนุกรมเวลาด้วยการแทนข้อมูลแบบแฟร็กทัล

แต่ในทางปฏิบัติ มิติความสัมพันธ์เป็นวิธีที่ใช้เวลาในการประมวลผลค่อนข้างมาก $O(n^2)$ เมื่อเปรียบเทียบกับมิติเส้นขอบที่ใช้เวลาเพียง $O(n)$ ทำให้มิติความสัมพันธ์ส่งผลให้การคำนวณช้าลงมาก และจากการทดลองซึ่งจะแสดงต่อไปในบทที่ 4 เมื่อใช้เฉพาะมิติเส้นขอบตามความยาวที่เท่ากัน และมิติเส้นขอบตามความกว้างที่เท่ากัน เพื่อแทนที่ข้อมูลอนุกรมเวลาใด ๆ ให้เหลือเลขจำนวนจริง 2 ค่า สำหรับการแทนข้อมูลแบบแฟร็กทัล ผลของความแม่นยำยังคงใกล้เคียงกับวิธีที่ดีที่สุดในแต่ละชุดข้อมูล และเวลาที่ใช้ลดลงอย่างมากเมื่อเปรียบเทียบกับการนำมิติความสัมพันธ์มาใช้ร่วมกัน

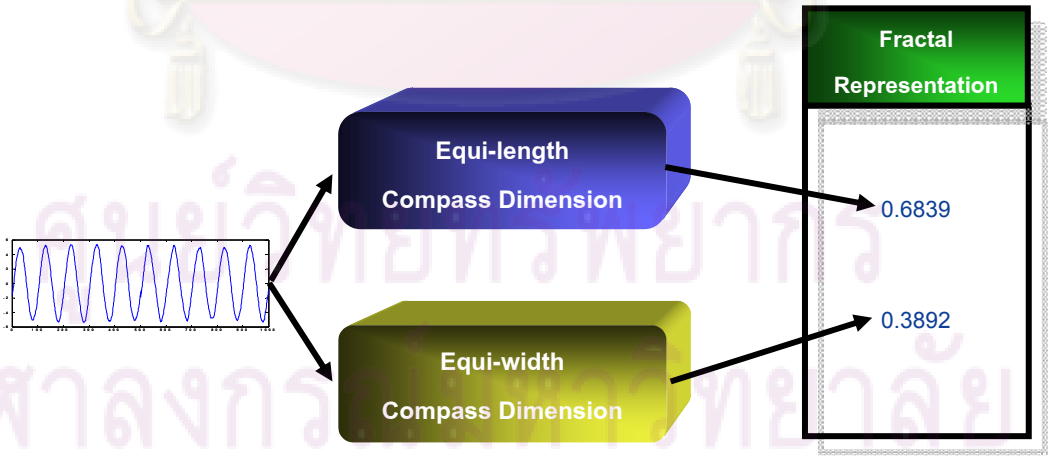
จากคุณสมบัติของมิติแฟร็กทัลที่สำคัญที่สุดคือ ความคล้ายคลึงตัวเอง ดังนั้น การทดสอบกับข้อมูลอนุกรมเวลาในงานวิจัยนี้ จึงมุ่งเป้าไปที่ข้อมูลอนุกรมเวลาขนาดใหญ่ และมีโครงสร้างภายในของข้อมูลที่ซ้ำ ๆ กัน ซึ่งการเกิดรูปแบบที่คล้ายกันภายในข้อมูลอนุกรมเวลาจะทำให้วิธีการคำนวณมิติเส้นขอบ ได้ค่ามิติของข้อมูลอนุกรมเวลาที่เหมาะสม ซึ่งวิธีการนำมิติเส้นขอบมาประยุกต์ใช้กับข้อมูลอนุกรมเวลาจะกล่าวในหัวข้อต่อไป

3.2 การแทนข้อมูลแบบแฟร็กทัล (Fractal Representation)

งานวิจัยนี้นำเสนอการลดขนาดข้อมูลอนุกรมเวลาโดยใช้หลักการของมิติแฟร็กทัล ซึ่งเป็นวิธีสำหรับคำนวณค่ามิติข้อมูลที่เป็นเลขจำนวนจริงหนึ่งค่า โดยอยู่บนสมมติฐานที่ว่า ผลลัพธ์ของค่ามิติที่คำนวณได้มีความเป็นไปได้ที่จะดึงคุณลักษณะของข้อมูลอนุกรมเวลาที่มีโครงสร้างภายในเกิดขึ้นซ้ำ ๆ กัน ซึ่งค่ามิติแฟร็กทัลที่คำนวณกับข้อมูลอนุกรมเวลาประเภทเดียวกันได้ผลลัพธ์เป็นเลขจำนวนจริงใกล้เคียงกัน และผลลัพธ์ของมิติแฟร็กทัลจะทำให้ความยาวของข้อมูลอนุกรมเวลาถูกลดลงอย่างมาก สำหรับค่ามิติแฟร็กทัลของข้อมูลอนุกรมเวลาที่ได้ในงานวิจัยนี้ไม่ได้มุ่งเน้นให้เป็นค่ามิติที่ถูกต้องสำหรับข้อมูลอนุกรมเวลา แต่สนใจเฉพาะส่วนของค่าจำนวนจริงที่คำนวณได้นั้นสามารถจำแนกข้อมูลอนุกรมเวลาชนิดเดียวกันได้ผลความแม่นยำที่มีประสิทธิภาพ

3.2.1 การแทนข้อมูลแบบแฟร็กทัล (Fractal Representation System)

การแทนข้อมูลแบบแฟร็กทัลเป็นการลดขนาดข้อมูลอนุกรมเวลาให้เหลือเลขจำนวนจริงเพียง 2 ค่า ซึ่งมาจากสองวิธีของมิติแฟร็กทัลสำหรับแต่ละข้อมูลอนุกรมเวลา และได้รับประสิทธิภาพของความแม่นยำใกล้เคียงกับวิธีอื่น ๆ เช่น การวัดระยะทางแบบยุคลิด วิธีไดนามิกไทม์วอร์ปปิง และวิธีซีดีเอ็ม และการคำนวณกับข้อมูลอนุกรมเวลาจะส่งผลให้มีความเร็วที่มีประสิทธิภาพ โดยสองวิธีของมิติแฟร็กทัลนำเสนอด้วยมิติเส้นขอบในสองแนวทาง คือ มิติเส้นขอบตามความยาวที่เท่ากัน (Equi-length Compass Dimension) และมิติเส้นขอบตามความกว้างที่เท่ากัน (Equi-width Compass Dimension) ซึ่งการนำทั้งสองวิธีดังกล่าวไปประยุกต์ใช้กับข้อมูลอนุกรมเวลาจะบรรยายในหัวข้อถัดไป



รูปที่ 3.3 การแทนข้อมูลแบบแฟร็กทัล

สำหรับการแทนข้อมูลแบบแฟร็กทัล แสดงได้ดังรูปที่ 3.3 โดยเป็นการขั้นตอนสำหรับการแทนข้อมูลแบบแฟร็กทัลกับข้อมูลอนุกรมเวลาหนึ่งตัว โดยเมื่อนำข้อมูลอนุกรมเวลามาคำนวณด้วยมิติเส้นขอบตามความยาวที่เท่ากัน และมิติเส้นขอบตามความกว้างที่เท่ากัน ในแต่ละวิธีจะได้ผลลัพธ์เป็นเลขจำนวนจริงค่าหนึ่ง ซึ่งจากการคำนวณด้วยมิติเส้นขอบทั้งสองสามารถแทนข้อมูลอนุกรมเวลาเดิมได้ด้วยตัวเลขเพียงสองค่าเท่านั้น ผลลัพธ์ของข้อมูลอนุกรมเวลาที่ได้สำหรับการลดขนาดด้วยมิติเส้นขอบตามความยาวที่เท่ากันเท่ากับ 0.6839 และมิติเส้นขอบตามความกว้างที่เท่ากันเท่ากับ 0.3892 จะเห็นได้ วิธีทั้งสองจะให้ค่าจำนวนจริงไม่เท่ากัน ถึงแม้ว่าการคำนวณจะวัดจากระยะทางตามหลักการของมิติเส้นขอบ แต่แนวทางการคำนวณที่แตกต่างกันจึงส่งผลผลให้แต่ละมิติได้รับคุณลักษณะที่แตกต่างกัน ซึ่งจากตัวเลขที่มีความแตกต่างกันนี่เอง จึงส่งผลให้การนำเลขจำนวนจริงทั้งสองค่ามาแทนหนึ่งข้อมูลอนุกรมเวลาให้ผลความแม่นยำที่สูงขึ้น

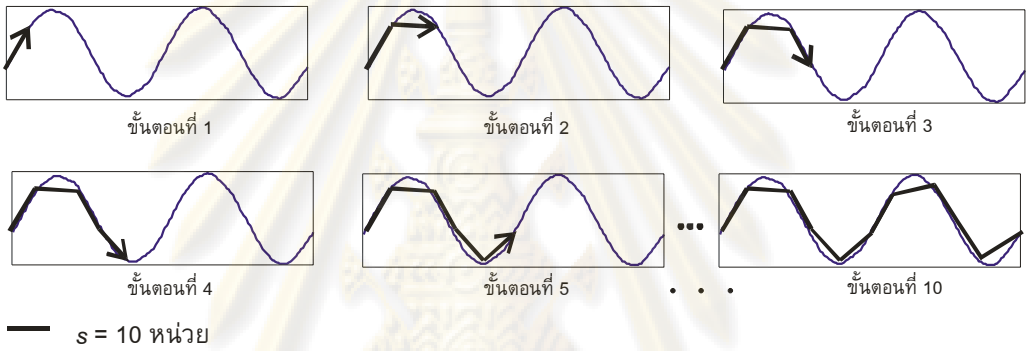
3.2.2 มิติเส้นขอบ (Compass Dimension)

มิติเส้นขอบสำหรับงานวิจัยนี้ถูกพัฒนาเพื่อประยุกต์ใช้สำหรับข้อมูลอนุกรมเวลาเป็นหลัก ซึ่งค่ามิติเส้นขอบที่ได้ คือ เลขจำนวนจริงหนึ่งค่าสำหรับข้อมูลอนุกรมเวลา โดยส่วนมากมิติเส้นขอบถูกนำไปใช้คำนวณหามิติกับข้อมูลที่สามารถวัดระยะทางรอบวัตถุนั้นได้ ซึ่งจะเห็นได้ว่า ข้อมูลอนุกรมเวลาที่เป็นข้อมูลจุดและเรียงตัวกันอย่างเป็นลำดับ การลากเส้นจากจุดหนึ่งไปอีกจุดหนึ่งภายในข้อมูลอนุกรมเวลาจะได้ระยะทางระหว่างจุดเช่นกัน จากจุดนี้เองผู้วิจัยจึงได้พัฒนามิติเส้นขอบเพื่อประยุกต์ใช้กับข้อมูลอนุกรมเวลา โดยมีวิธีการคำนวณมิติเส้นขอบแบ่งเป็น 2 วิธี คือ มิติเส้นขอบตามความยาวที่เท่ากัน โดยลากเส้นที่มีขนาดความยาวคงที่จากจุดเริ่มต้นไปยังจุดที่วัดดระยะห่างได้เท่ากับความยาวที่กำหนด แล้วกระทำจนถึงจุดสุดท้ายของข้อมูลอนุกรมเวลา และมิติเส้นขอบตามความกว้างที่เท่ากัน โดยทำการแบ่งช่วงตามระยะห่างในแนวแกนนอนที่เท่ากัน แล้วลากเส้นภายในแต่ละช่วงนั้น ซึ่งผลของค่ามิติที่เป็นจำนวนจริงสำหรับทั้งสองวิธีนี้จะถูกกำหนดให้เป็นเลขจำนวนจริง 2 ค่า ที่ถูกลดขนาดด้วยการแทนข้อมูลแบบแฟร็กทัล สำหรับวิธีการคำนวณของทั้งสองวิธีอธิบายได้ดังนี้

3.2.2.1 มิติเส้นขอบตามความยาวที่เท่ากัน (Equi-length Compass Dimension)

มิติเส้นขอบเป็นการค้นหาความคล้ายคลึงตัวเองจากระยะทางโดยรอบของข้อมูลและใช้มาตราส่วนจากความยาวที่มีขนาดเท่ากันหรือเส้นขอบเพื่อใช้ในการลากเส้น โดยจะทำการปรับระดับของเส้นขอบที่แบ่งไว้ให้มีความละเอียดมากยิ่งขึ้นสำหรับแต่ละรอบการคำนวณ ซึ่งการลากเส้นให้มีขนาดเท่ากันเพื่อวัดระยะทางรอบข้อมูลเป็นแนวทางสำหรับการนำมิติเส้นขอบมาประยุกต์ใช้เพื่อหาค่ามิติของข้อมูลอนุกรมเวลา ซึ่งได้ค่าที่สามารถจำแนกข้อมูลอนุกรมเวลากลุ่มเดียวกันได้ สำหรับในส่วนนี้จะกล่าวถึงแนวทางเริ่มต้นในการพัฒนามิติเส้นขอบกับข้อมูลอนุกรมเวลา โดยจะใช้หลักการคล้ายคลึงกับมิติเส้นขอบเดิมคือ การกำหนดความยาวเส้น

ขอบให้มีขนาดคงที่ในแต่ละรอบ เพื่อลากเส้นระหว่างจุดภายในข้ออมูลอนุกรมเวลา ดังแสดงในรูปที่ 3.4 เป็นแนวทางสำหรับการคำนวณด้วยมิติเส้นขอบในหนึ่งรอบด้วยความยาวที่กำหนดไว้คือ *s* เท่ากับ 10 หน่วย โดยเริ่มลากเส้นจากจุดแรกของข้อมูลอนุกรมเวลา แล้วทำการวัดความยาวในระยะกระจัดของข้อมูลอนุกรมเวลาจนได้ความยาวของเส้นเท่ากับความยาวที่แบ่งไว้ในแต่ละรอบ ซึ่งแสดงในขั้นตอนที่ 1 ต่อมาลากเส้นจากจุดปลายในครั้งที่แล้วและวัดความยาวในระยะกระจัดจนกระทั่งได้ความยาวเท่ากับ 20 หน่วย อีกครั้ง ซึ่งแสดงในขั้นตอนที่ 2 และกระทำซ้ำแบบเดิมจนถึงขั้นตอนที่ 9 จนกระทั่งถึงจุดปลายสุดของข้อมูลอนุกรมเวลาจึงหยุดการคำนวณ ซึ่งแสดงในขั้นตอนที่ 10 ในที่สุด จะได้ความยาวรวมที่ทำการลากในทุกขั้นตอนหรือเส้นตรงสีดำทึบทั้งหมดในขั้นตอนที่ 10 โดยแนวทางเริ่มต้นสำหรับแนวคิดที่กล่าวมาถูกเรียกว่า มิติเส้นขอบที่มีความยาวเท่ากัน



รูปที่ 3.4 แนวทางสำหรับการคำนวณด้วยมิติเส้นขอบตามความยาวที่เท่ากัน

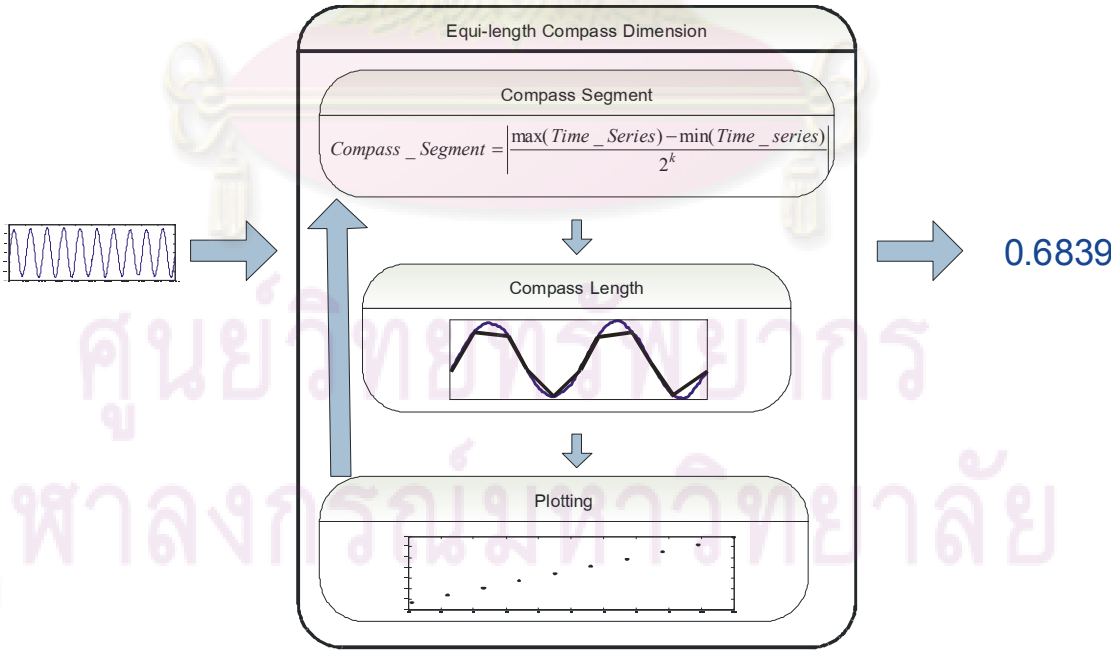
จากที่กล่าวมาข้างต้นเป็นแนวทางแรกเริ่มสำหรับการนำมิติเส้นขอบมาประยุกต์ใช้กับข้อมูลอนุกรมเวลา ในส่วนนี้จะอธิบายถึงการพัฒนาเพื่อนำมิติเส้นขอบตามความยาวที่เท่ากันไปประยุกต์ใช้กับข้อมูลอนุกรมเวลา ซึ่งจะได้ค่าจำนวนจริง 1 ค่า และเป็นขนาดข้อมูลที่ถูกลดลงมาสำหรับการแทนข้อมูลแบบแฟร็กทัลที่จะแทนแต่ละตัวของข้อมูลอนุกรมเวลา โดยส่วนต่อไปจะกล่าวคร่าว ๆ ถึงวิธีการคำนวณมิติเส้นขอบตามความยาวที่เท่ากันและโครงสร้างโดยรวมของมิติเส้นขอบตามความยาวที่เท่ากัน ก่อนที่จะลงรายละเอียดในส่วนของรหัสเทียม

มิติเส้นขอบตามความยาวที่เท่ากันสำหรับข้อมูลอนุกรมเวลาเป็นการกำหนดความยาวคงที่ค่าหนึ่งเพื่อนำมาเปรียบเทียบกับระยะทางระหว่างจุดของข้อมูลอนุกรมเวลา ซึ่งหากระยะทางระหว่างจุดมากกว่าความยาวที่กำหนดแล้วจะลากเส้นขอบระหว่างจุดเพื่อคำนวณหาความยาวโดยรวมของระยะทางทั้งหมด สำหรับงานวิจัยนี้เรียกความยาวขนาดคงที่ว่า เซกเมนต์เส้นขอบ (Compass Segment) โดยในแต่ละรอบสามารถคำนวณหาค่าเซกเมนต์เส้นขอบได้ดังแสดงในสมการที่ 3.1 แสดงสูตรสำหรับคำนวณหาค่าเซกเมนต์เส้นขอบ ในรอบที่

k โดยเป็นค่าสัมบูรณ์ของการลบกันระหว่างค่ามากที่สุด $\max(Time\_Series)$ กับค่าน้อยที่สุด $\min(Time\_series)$ ภายในข้อมูลอนุกรมเวลาและแบ่งครึ่งค่าความยาวอย่างต่อเนื่องในแต่ละรอบของการกระทำซ้ำ *k*

$$Compass\_Segment(k) = \left| \frac{\max(Time\_Series) - \min(Time\_series)}{2^k} \right| \quad (3.1)$$

สำหรับโครงสร้างโดยรวมในการคำนวณด้วยมิติเส้นขอบตามความยาวที่เท่ากันแสดงได้ดังรูปที่ 3.5 เริ่มต้นจากการนำข้อมูลรับเข้า ซึ่งเป็นข้อมูลอนุกรมเวลาเข้าสู่กระบวนการคำนวณมิติเส้นขอบตามความยาวที่เท่ากัน ต่อมาคำนวณความยาวเซกเมนต์ตามสมการที่ 3.1 เซกเมนต์เส้นขอบนี้จะเป็นความยาวอ้างอิงที่ใช้ในการลากเส้นจากจุดเริ่มต้นของข้อมูลอนุกรมเวลาถึงจุดปลายสุดของข้อมูลอนุกรมเวลา โดยแนวทางในการลากเส้นขอบจะใช้เฉพาะระยะทางในแนวแกนตั้งเท่านั้น ซึ่งผลลัพธ์ของการลากเส้นขอบในแต่ละรอบคือ ความยาวเส้นขอบโดยรวมที่ได้จากแต่ละครั้งสำหรับการลากเส้นระหว่างจุดและนำความยาวทั้งหมดมารวมกันแล้วนำค่าเซกเมนต์เส้นขอบและความยาวเส้นขอบจะถูกกำหนดจุดบนมาตราส่วนลอการิทึม ในรอบแรก ต่อมาลดขนาดความยาวของเซกเมนต์เส้นขอบให้เหลือครึ่งหนึ่งของความยาวในครั้งแรก และกระทำซ้ำในแนวทางเดิมอย่างต่อเนื่อง เมื่อกระทำซ้ำจนถึงจุดที่เหมาะสมแล้ว ซึ่งจำนวนรอบที่จะกระทำจนกระทั่งหยุดการคำนวณสำหรับมิติเส้นขอบนี้อธิบายไว้ในส่วนของรหัสเทียม ในที่สุด คำนวณหาความชันของข้อมูลจากจุดทั้งหมดที่กำหนดไว้บนกราฟ ดังแสดงในสมการที่ 3.2



รูปที่ 3.5 โครงสร้างโดยรวมสำหรับการคำนวณด้วยมิติเส้นขอบตามความยาวที่เท่ากัน

สมการที่ 3.2 แสดงสูตรคำนวณหาค่ามิติเส้นขอบตามความยาวที่เท่ากัน โดยเป็นการใส่ลิมิตเมื่อค่าเซกเมนต์เส้นขอบลู่เข้าสู่ 0 ซึ่งเป็นการคำนวณหาค่าความชันระหว่างความยาวเส้นขอบ $Compass\_Length$ และเซกเมนต์เส้นขอบ $Compass\_Segment$ ผลลัพธ์ของความชันที่ได้คือ ค่ามิติเส้นขอบตามความยาวที่เท่ากัน D_{ELC} สำหรับเหตุผลที่กลับเศษของเซกเมนต์เส้นขอบเพื่อไม่ให้ค่าจำนวนจริงที่ได้มีค่าติดลบ โดยค่าความชันที่คำนวณได้สำหรับมิติเส้นขอบตามความยาวที่เท่ากันกับข้อมูลอนุกรมเวลาหนึ่งตัว คือ เลขจำนวนจริงหนึ่งค่าสำหรับข้อมูลอนุกรมเวลาที่ใช้การแทนข้อมูลแบบแฟร็กทัล

$$D_{ELC} = \lim_{Compass\_Segment \to 0} \frac{\ln(Compass\_Length)}{\ln(1/Compass\_Segment)} \tag{3.2}$$

โดยขั้นตอนการคำนวณด้วยมิติเส้นขอบตามความยาวที่เท่ากัน สำหรับงานวิจัยนี้สามารถแสดงเป็นรหัสเทียมได้ดังรูปที่ 3.6

ในบรรทัดที่ 1 ข้อมูลอนุกรมเวลา *TS* จะถูกปรับระดับด้วยคะแนน Z ก่อนการคำนวณ ในบรรทัดที่ 2 กำหนดให้รอบการทำซ้ำลู่เข้าสู่อนันต์ สำหรับบรรทัดที่ 4 ถึง 32 เป็นการคำนวณเพื่อหาความยาวรวมของเส้นขอบ `Sum_Compass` ในแต่ละรอบ โดยเปรียบเทียบกับความยาวในแต่ละเซกเมนต์เส้นขอบ `Compass_Segment` ในบรรทัดที่ 5 เป็นการคำนวณค่า `Compass_Segment` ซึ่งเป็นค่าที่ใช้ในการอ้างอิงกับระยะห่างระหว่างจุดภายในข้อมูลอนุกรมเวลา ซึ่งถ้าหากค่า `Compass_Segment` มีค่าน้อยกว่าระยะทางระหว่างจุด จะทำการเลือกสองจุดนั้นเพื่อลากเส้นขอบ ซึ่งสูตรการคำนวณหาค่า `Compass_Segment` แสดงได้ดังสมการที่ 3.1 ในบรรทัดที่ 6 และ 7 ทำการเก็บจุดเริ่มต้น *Fixed_Point* และจุดต่อไป *Next_Point* สำหรับวัดระยะทางระหว่างจุด ในบรรทัดที่ 8 เก็บจำนวนครั้งของการลากเส้น `Count_Calculate` ในแต่ละรอบการคำนวณกับค่า `Compass_Segment` ซึ่งกำหนดไว้เพื่อใช้สำหรับหยุดการคำนวณทั้งระบบ

สำหรับบรรทัดที่ 9 ถึง 26 เป็นการลากเส้นขอบจากจุดเริ่มต้นต่อเนื่องไปจนถึงจุดสุดท้าย โดยเปรียบเทียบกับค่า `Compass_Segment` ที่กำหนดไว้ในแต่ละรอบ ในบรรทัดที่ 10 ถึง 17 มีไว้เพื่อตรวจสอบว่าจุดที่ต้องการลากเส้นเกินกว่าค่าความยาวของข้อมูลอนุกรมเวลาหรือไม่ ถ้าไม่มากกว่าให้ทำต่อในบรรทัดที่ 11 โดยคำนวณค่าสัมบูรณ์ของการวัดระยะทางจากจุด `Fixed_Point` ถึง `Next_Point` ซึ่งระยะทางที่ได้ เรียกว่า ความยาวเส้นขอบ `Compass_Length` โดยคำนวณระยะทางเฉพาะแนวแกนตั้งเท่านั้น และไม่คำนวณความยาวแนวแกนนอน ซึ่งแตกต่างจากแนวทางเริ่มต้นที่แสดงได้ดังรูปที่ 3.4 เนื่องจาก เมื่อช่วงในแนวแกนตั้งถูกปรับระดับคะแนนด้วย *Z* จะทำให้ความกว้างของข้อมูลแคบลงมาก การนำค่าในแนวแกนนอน ซึ่งมีระยะห่างเท่ากับหนึ่งมาคำนวณจะส่งผลต่อความแม่นยำ ทำให้ความยาวเส้นขอบไปอ้างอิงจากระยะห่างในแนวแกนนอนมากกว่า ซึ่งไม่เหมาะสมในการนำมาคำนวณ โดย

แนวแกนนอนจะนำมาใช้สำหรับการคิดลำดับเพื่อวัดระยะทางระหว่างจุดเท่านั้น สำหรับบรรทัดที่ 12 ถึง 14 ถ้าความยาวของจุด *Fixed_Point* มากกว่าความยาวของข้อมูลอนุกรมเวลาจะหยุดการคำนวณในรอบนี้ แต่ถ้าค่า *Next_Point* มากกว่าความยาวของข้อมูลอนุกรมเวลาจะทำการปรับให้มีความยาวเท่ากันกับข้อมูลอนุกรมเวลา ซึ่งเป็นการกระทำรอบสุดท้ายในแต่ละค่า *Compass_Segment* ในบรรทัดที่ 18 ถึง 25 เป็นการเปรียบเทียบค่า *Compass_Length* และค่า *Compass_Segment* โดยถ้าความยาวเส้นขอบมากกว่าเซกเมนต์เส้นขอบ จะทำการรวมค่าความยาวเส้นขอบที่คำนวณได้กับความยาวที่ลากก่อนหน้านี้คือค่า *Sum_Compass* และทำการปรับค่า *Fixed_Point* เป็นจุดของการลากครั้งก่อน และ *Next_Point* จะถูกปรับให้เป็นตำแหน่งต่อไป แต่ถ้าความยาวเส้นขอบน้อยกว่าเซกเมนต์เส้นขอบ ค่าของ *Fixed_Point* จะอยู่ที่ตำแหน่งเดิม แต่ *Next_Point* จะกำหนดเพิ่มขึ้นหนึ่งจุด แล้ววัดระยะทางจนกว่าจะมากกว่าค่าเซกเมนต์เส้นขอบ

สำหรับบรรทัดที่ 2 ถึง 28 จะเก็บคู่ของความยาวเส้นขอบรวม *ListCompFunc* และเซกเมนต์เส้นขอบในแต่ละรอบ *ListCompSegment* เพื่อคำนวณหาความชันในบรรทัดที่ 32 ซึ่งค่าความชันที่ได้คือ ค่ามิติเส้นขอบตามความยาวที่เท่ากัน *ResultELC* และบรรทัดที่ 28 ถึง 30 เป็นการตรวจสอบว่าเกิดการลากเส้นระหว่างจุดกับทุก ๆ จุดแล้วหรือยัง ถ้าเกิดขึ้นแล้วจะหยุดการคำนวณทั้งระบบ เนื่องจาก การคำนวณครั้งต่อไปจะได้ค่าความยาวเส้นขอบเท่าเดิมทุกครั้ง

Algorithm 1 : Equi-length_Compass_Dimension(*TS*)

```
 1:  TS_Normalized ← Z_SCORE_NORMALIZATION(TS)
 2:  Loop_of_Repeating ← Infinite
 3:
 4:  Foreach Loop_of_Repeating do
 5:    Compass_Segment ←
       SCALE_FACTOR(max(TS_Normailzed),min(TS_Normailzed))
 6:    Fixed_Point ←  1
 7:    Next_Point ← Fixed_Point + 1
 8:    Count_Calculate ← 0
 9:    Foreach LengthTS do
10:      If Next_point ≤ LengthTS then
11:         Compass_Length ←
            DISTANCE_of_AMPLITUDE(Fixed_Point, Next_Point)
12:      ElseIf Fixed_Point ≥ LengthTS
13:         Break
14:      ElseIf Next_Point ≥ LengthTS
15:         Next_Point ← LengthTS
16:         Compass_Length ←
            DISTANCE_of_AMPLITUDE(Fixed_Point, Next_Point)
```

```
17:         EndIf
18:         If Compass_Length ≥ Compass_Segment
19:            Sum_Compass ← Sum_Compass + Compass_Length
20:            Fixed_Point ← Next_Point
21:            Next_Point ← Fixed_Point + 1
22:            Count_Calculate ← Count_Calculate + 1
23:         Else
24:            Next_Point ← Next_Point + 1
25:         EndIf
26:      EndFor
27:      ListCompFunc ← Sum_Compass
28:      ListCompSegment ← Compass_Segment
29:      If Count_Calculate ≥ LengthTS
30:         Break
31:      EndIf
32:   EndFor
33:   ResultELC ← Regression(ListCompFunc, ListCompSegment)
34:   Return ResultELC
```

รูปที่ 3.6 รหัสเทียมสำหรับมิติเส้นขอบตามความยาวที่เท่ากันของข้อมูลอนุกรมเวลา

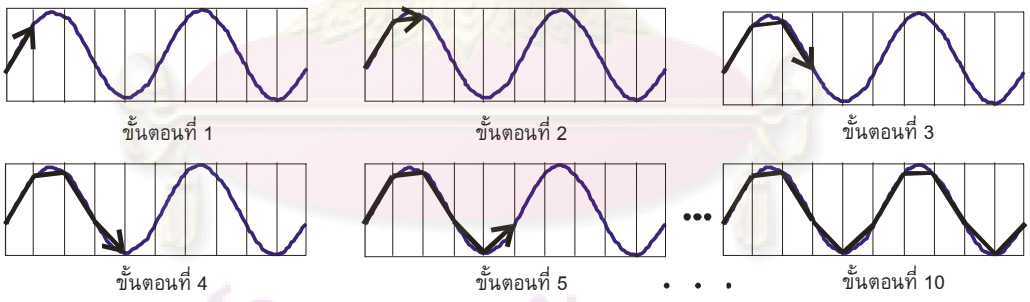
จากมิติเส้นขอบตามความยาวที่เท่ากัน ซึ่งแสดงด้วยรหัสเทียมดังรูปที่ 3.6 จะเห็นว่า จากทฤษฎีที่นำเสนอของมิติเส้นขอบในการคำนวณจะพบพารามิเตอร์ที่ต้องทำการปรับด้วยมือหลายส่วน เช่น รอบของการทำซ้ำ ค่าเริ่มต้นของเซกเมนต์เส้นขอบ หรือจุดหยุดของการคำนวณ ซึ่งมิติเส้นขอบตามความยาวที่เท่ากันสำหรับงานวิจัยนี้สามารถคำนวณเพื่อหาค่าจำนวนจริงที่แสดงคุณลักษณะของข้อมูลอนุกรมเวลาได้อย่างอัตโนมัติ แต่งานวิจัยนี้ไม่ได้เน้นไปที่การเลือกพารามิเตอร์ต่าง ๆ ให้ได้ผลของความแม่นยำดีที่สุด ซึ่งวิธีที่ผู้วิจัยนำเสนอเป็นแนวทางในการพัฒนาต่อไปสำหรับการเพิ่มประสิทธิภาพของมิติแฟร็กทัลให้ได้รับผลของความแม่นยำที่ดีที่สุด

3.2.2.2 มิติเส้นขอบตามความกว้างที่เท่ากัน (Equi-width Compass Dimension)

โดยทั่วไปมิติเส้นขอบเป็นวิธีวัดระยะทางโดยลากเส้นให้มีความยาวคงที่ตามขอบของวัตถุ ด้วยค่าหนึ่งที่ถูกกำหนดในแต่ละรอบของการคำนวณ ซึ่งวิธีนี้ถูกนำไปประยุกต์กับข้อมูลอนุกรมเวลาด้วยมิติเส้นขอบตามความยาวที่เท่ากัน ต่อมาผู้วิจัยจึงได้ค้นหาแนวคิดใหม่ซึ่งน่าจะนำมาใช้สำหรับข้อมูลอนุกรมเวลาได้เช่นกัน และเป็นค่าจำนวนจริงเลขที่สองสำหรับการแทนข้อมูลแบบแฟร็กทัล โดยวิธีการนี้ใช้คุณสมบัติของข้อมูลอนุกรมเวลาที่มีลำดับต่อเนื่องกันมาประยุกต์เพื่อใช้ในการลากเส้นแทนการกำหนดค่าความยาวที่ตายตัวสำหรับการคำนวณมิติเส้นขอบ ซึ่งเรียกวิธีนี้ว่า มิติเส้นขอบตามความกว้างที่เท่ากัน โดยเป็นการกำหนดช่วงของแกน

นอนให้มีขนาดที่เท่ากันแล้วลากเส้นตามระยะกระจัดภายในช่วงที่แบ่งไว้ เรียกค่าที่ใช้สำหรับการแบ่งช่วงที่เท่า ๆ กันว่า ไทม์สไลซ์ (Time Slice)

จากรูปที่ 3.7 แสดงแนวทางในการคำนวณมิติเส้นขอบตามความยาวที่เท่ากันสำหรับหนึ่งรอบการคำนวณ เริ่มต้น กำหนดค่าไทม์สไลซ์เท่ากับ 20 หน่วย ซึ่งการแบ่งช่วงภายในข้อมูลอนุกรมเวลาจะเริ่มจากจุดแรกแล้วนับไปอีก 20 จุด ซึ่งจะเห็นการแบ่งช่วงจากขั้นตอนที่ 1 ยกตัวอย่างเช่น ถ้าความยาวของข้อมูลอนุกรมเวลาใด ๆ มีค่าเท่ากับ 200 หน่วย การกำหนดค่าไทม์สไลซ์เท่ากับ 20 หน่วย จะทำให้เกิดการแบ่งคาบแนวแกนนอนที่เท่า ๆ กันได้ ทั้งหมด 10 คาบ โดยคาบที่ 1 อยู่ระหว่าง จุดที่ 1 ของข้อมูลอนุกรมเวลากับ จุดที่ 21 คาบที่ 2 อยู่ระหว่าง จุดที่ 21 ถึง 41 จนกระทั่งถึงคาบที่ 10 อยู่ระหว่าง จุดที่ 181 ถึง 200 ซึ่งจะเห็นว่าถึงแม้ว่าคาบสุดท้ายจะไม่สามารถนับจุดได้ครบ 20 จุด แต่จะลากเส้นในช่วงสุดท้ายระหว่างจุดเท่าที่ลากได้จนกระทั่งถึงจุดปลายสุดของข้อมูลอนุกรมเวลา สำหรับการลากเส้นเพื่อหาความยาวเส้นขอบโดยรวม เมื่อกำหนดค่าเริ่มต้นของไทม์สไลซ์ แล้วลากเส้นจากจุดแรกของข้อมูลอนุกรมเวลาจนกระทั่งถึงจุดที่ 21 โดยวัดความยาวในระยะกระจัดตามขั้นตอนที่ 1 ต่อมาขั้นตอนที่ 2 ลากจากจุดปลายของเส้นก่อนหน้าจนกระทั่งถึงตำแหน่งปลายในคาบถัดไป และกระทำซ้ำอย่างต่อเนื่องจนกระทั่งถึงคาบสุดท้าย ในขั้นตอนที่ 10 ในที่สุด รวมความยาวของเส้นสีดำที่ทำการลากในขั้นตอนที่ 10 ซึ่งความยาวที่ได้ก็จะเป็นความยาวรวมของเส้นขอบ ที่ค่าไทม์สไลซ์เท่ากับ 20 หน่วย ซึ่งจากที่กล่าวมาเป็นแนวทางเริ่มต้นของผู้วิจัยสำหรับการพัฒนามิติเส้นขอบตามความกว้างที่เท่ากันสำหรับข้อมูลอนุกรมเวลา

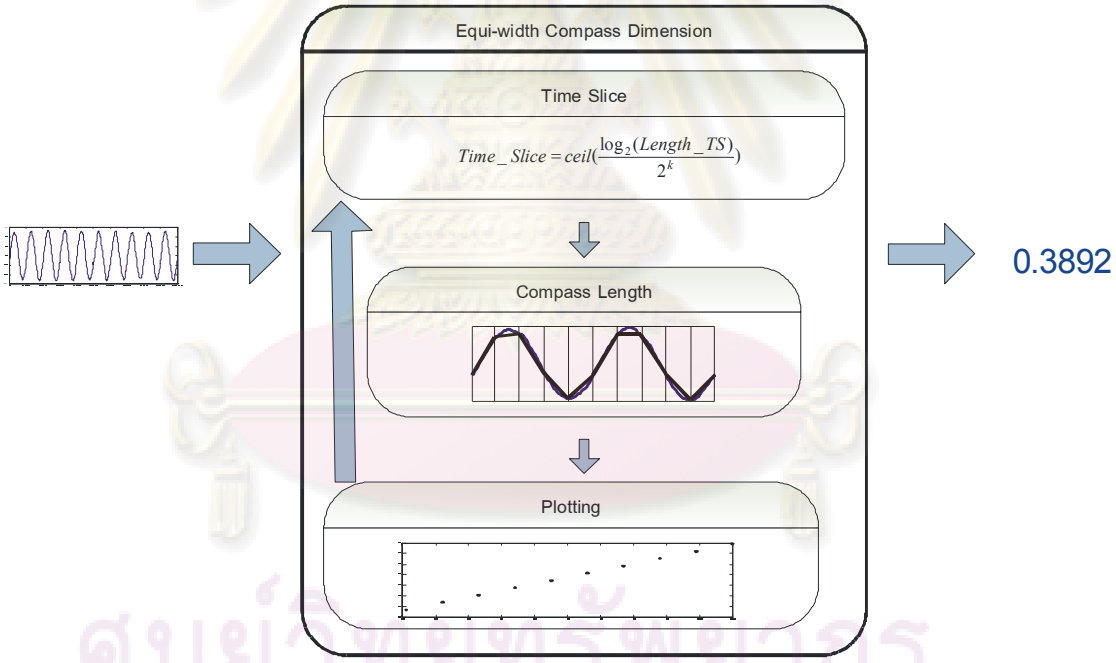


รูปที่ 3.7 แนวทางสำหรับการคำนวณด้วยมิติเส้นขอบตามความกว้างที่เท่ากัน

ในส่วนนี้กล่าวถึง สูตรการคำนวณค่าไทม์สไลซ์และโครงสร้างโดยรวมของมิติเส้นขอบตามความกว้างที่เท่ากันสำหรับงานวิจัยนี้ ซึ่งผลลัพธ์ของมิตินี้เป็นค่าจำนวนจริงเลขที่สองสำหรับข้อมูลอนุกรมเวลาที่ทำการลดขนาดข้อมูลแล้ว โดยส่วนของรายละเอียดแต่ละขั้นตอนจะกล่าวในส่วนของรหัสเทียมต่อไป

มิติเส้นขอบตามความกว้างที่เท่ากันเป็นการกำหนดช่วงความกว้างหรือคาบของข้อมูลในแนวแกนนอนที่มีขนาดเท่ากันหรือค่าไทม์สไลซ์ แล้ววัดระยะทางในตำแน่งของจุดเริ่มต้นและจุดปลายของแต่ละคาบนั้น ๆ สำหรับสูตรการคำนวณค่าไทม์สไลซ์ในแต่ละรอบแสดงดังสมการที่ 3.3 ซึ่งเป็นการนำความยาวของข้อมูลอนุกรมเวลา $Length\_TS$ มาลดขนาดลงโดยการใส่ลอการิทึมฐาน 2 เพราะว่า ถ้าหากใช้ช่วงของไทม์สไลซ์ที่มากเกินไปจะทำให้การลากเส้นเกิดการผิดรูปและส่งผลต่อการคำนวณหาค่ามิติแฟร็กทัล ต่อมานำผลลัพธ์ที่ได้จากการใส่ลอการิทึมฐาน 2 แล้วหารด้วย 2 อย่างต่อเนื่องตามรอบการคำนวณ ***k*** ซึ่งจะได้ค่าไทม์สไลซ์ $Time\_Slice$ ในรอบที่ ***k*** โดยจะทำการปัดเศษขึ้นทุกครั้ง เนื่องจาก ช่วงของข้อมูลอนุกรมเวลาเป็นเลขจำนวนเต็มเรียงลำดับกัน และการปัดขึ้นเพื่อจำกัดการคำนวณรอบสุดท้ายคือ ระยะห่างระหว่างช่วงของข้อมูลอนุกรมเวลามีค่าได้น้อยที่สุดเท่ากับ 1 ดังนั้น การปัดเศษขึ้นจะช่วยแก้ปัญหาที่เกิดขึ้นกับเหตุผลดังกล่าว

$$Time\_Slice = ceil(\frac{\log_2(Length\_TS)}{2^k}) \tag{3.3}$$



รูปที่ 3.8 โครงสร้างโดยรวมสำหรับการคำนวณด้วยมิติเส้นขอบตามความกว้างที่เท่ากัน

โครงสร้างโดยรวมสำหรับการคำนวณด้วยมิติเส้นขอบตามความกว้างที่เท่ากันแสดงได้ดังรูปที่ 3.8 เริ่มต้นจากการนำข้อมูลอนุกรมเวลาซึ่งเป็นข้อมูลรับเข้า แล้วผ่านเข้าไปคำนวณในมิติเส้นขอบตามความกว้างที่เท่ากัน ต่อมาคำนวณค่าไทม์สไลซ์ตามสมการที่ 3.3 ซึ่งค่าไทม์สไลซ์ที่คำนวณได้ในรอบแรกใช้สำหรับกำหนดช่วงเพื่อลากเส้นระหว่างคู่จุดตามตำแหน่งที่แบ่งไว้ของไทม์สไลซ์ ซึ่งจะแบ่งช่วงไปจนถึงปลายสุดของข้อมูลอนุกรมเวลา แล้วทำการ

ลากเส้นเพื่อหาระยะทางระหว่างจุด โดยสนใจเฉพาะระยะห่างในแนวแกนตั้งเท่านั้น แล้วกระทำซ้ำอย่างต่อเนื่อง จนกระทั่งถึงจุดสุดท้ายของข้อมูลอนุกรมเวลา ซึ่งความยาวรวมที่ได้ของการลากเส้นทั้งหมดคือความยาวเส้นขอบและค่าไทม์สไลซ์ในรอบแรก และถูกวาดลงบนมาตราส่วนลอการิทึม ต่อมาค่าไทม์สไลซ์จะลดลงเป็นครึ่งหนึ่งของค่าก่อนหน้า และกระทำซ้ำอย่างต่อเนื่องเมื่อกระทำซ้ำจนถึงช่วงที่เหมาะสมแล้ว ในที่สุด หาค่าความชันของจุดทั้งหมดที่วาดบนกราฟ ดังแสดงในสมการที่ 3.4 เป็นสูตรการหาความชันของคู่ลำดับระหว่างความยาวเส้นขอบ $Compass\_Length$ และไทม์สไลซ์ $Time\_Slice$ ซึ่งความชันที่ได้คือค่ามิติเส้นขอบตามความกว้างที่เท่ากัน D_{EWC}

$$D_{EWC} = \lim_{Time\_Slice \to 1} \frac{\ln(Compass\_Length)}{\ln(1/Time\_Slice)} \tag{3.4}$$

โดยขั้นตอนการคำนวณด้วยมิติเส้นขอบตามความกว้างที่เท่ากัน สำหรับงานวิจัยนี้สามารถแสดงเป็นรหัสเทียมได้ดังแสดงในรูปที่ 3.9

ในบรรทัดที่ 1 ข้อมูลอนุกรมเวลา `TS` จะถูกปรับระดับคะแนนด้วย Z ในบรรทัดที่ 2 กำหนดการกระทำซ้ำลู่เข้าสู่อนันต์ สำหรับบรรทัดที่ 4 ถึง 24 เป็นการกำหนดค่าเริ่มต้นของไทม์สไลดซ์แล้วลากเส้นภายในช่วงตามค่าไทม์สไลด์ ซึ่งในแต่ละรอบการคำนวณจะวัดระยะทางโดยรวมของเส้นขอบทั้งหมด *Sum_Compass* โดยอ้างอิงกับแต่ละค่าของไทม์สไลซ์ `Time_Slice` และกระทำซ้ำอย่างต่อเนื่องจนกว่าจะพบจุดหยุดที่เหมาะสม ในบรรทัดที่ 5 คำนวณหาค่าไทม์สไลซ์ในแต่ละรอบ ดังแสดงในสมการที่ 3.3 ในบรรทัดที่ 6 กำหนดจุดเริ่มต้นของการคำนวณ `Fixed_Point` ซึ่งก็คือตำแหน่งแรกของข้อมูลอนุกรมเวลา

สำหรับบรรทัดที่ 7 ถึง 17 เป็นการคำนวณหาค่า *Sum_Compass* ในแต่ละรอบโดยขึ้นอยู่กับค่าไทม์สไลซ์ ในบรรทัดที่ 7 คำนวณจำนวนรอบที่เป็นไปได้ในการลากเส้นแต่ละรอบ โดยนำความยาวของข้อมูลอนุกรมเวลาหารกับค่าของไทม์สไลซ์แล้วปัดเศษขึ้น เพราะว่าเมื่อผลของการหารมีส่วนของเลขทศนิยม ในส่วนของเลขทศนิยมนี้ คือ การคำนวณรอบสุดท้ายซึ่งจะไม่ได้ขนาดของช่วงเท่ากับค่าไทม์สไลซ์ โดยจะนำส่วนระยะทางที่เหลือมาคำนวณด้วย ในบรรทัดที่ 8 กำหนดจุดต่อไปตามค่าไทม์สไลซ์เพื่อลากเส้นจากจุด `Fixed_Point` ไปยังจุด `Next_Point` สำหรับบรรทัดที่ 9 ถึง 17 ถ้าจุด `Next_point` อยู่ในตำแหน่งที่มีค่าน้อยกว่าหรือเท่ากับความยาวของข้อมูลอนุกรมเวลา ให้ลากเส้นระหว่างจุด `Fixed_Point` ไปยังจุด `Next_Point` ซึ่งจะได้ระยะทางระหว่างจุดคือความยาวเส้นขอบ `Compass_Length` โดยระยะทางที่คำนวณจะวัดจากแนวแกนตั้งเท่านั้น แต่ระยะทางในแนวแกนนอนไม่นำมาคำนวณด้วย ซึ่งมีเหตุผลเดียวกันกับมิติเส้นขอบตามความยาวที่เท่ากันที่ได้อธิบายในรหัสเทียมก่อนหน้านี้ ในบรรทัดที่ 11 นำค่า `Compass_Length` ที่คำนวณได้มารวมกันกับ `Sum_Compass` เพื่อหาความยาวรวม ต่อมาถ้าจุด `Fixed_Point` มีค่ามากกว่าความยาวของข้อมูลอนุกรม

เวลา ทำให้ไม่สามารถคำนวณค่าระยะทางได้อีกแล้ว การคำนวณความยาวรวมของเส้นขอบในรอบนี้จะหยุด แต่ถ้าจุด *Fixed_Point* ยังมีค่าน้อยกว่าความยาวของข้อมูลอนุกรมเวลา และค่า *Next_Point* มีความยาวมากกว่าข้อมูลอนุกรมเวลา จะกำหนดให้ค่า *Next_Point* มีค่าเท่ากับความยาวของข้อมูลอนุกรมเวลา ซึ่งจะเป็นการลากเส้นในครั้งสุดท้ายสำหรับการคำนวณในรอบนี้ และจะคำนวณในครั้งต่อไปกับค่าไทม์สไลซ์ที่ลดลงครึ่งหนึ่ง

ในบรรทัดที่ 19 เมื่อลากเส้นขอบจากจุดเริ่มต้นถึงจุดสุดท้ายของข้อมูลอนุกรมเวลา ค่า *Sum_Compass* และ *Time_Slice* จะถูกเก็บค่าไว้ในตัวแปร *ListCompFunc* และ *ListTime_Slice* ตามลำดับ ซึ่งค่าของ *ListCompFunc* และ *ListTime_Slice* ที่คำนวณได้ทั้งหมดจะนำมาวาดบนกราฟเพื่อหาความชัน โดยค่าความชันที่ได้ คือ ค่ามิติเส้นขอบตามความกว้างที่เท่ากัน *ResultEWC* สำหรับในบรรทัดที่ 21 ถึง 23 ถูกกำหนดเพื่อเป็นตัวหยุดการคำนวณของโปรแกรมทั้งหมดสำหรับข้อมูลอนุกรมเวลาหนึ่งตัว โดยจะเกิดขึ้นก็ต่อเมื่อค่าไทม์สไลซ์ในรอบการคำนวณใด ๆ มีค่าเท่ากับหนึ่ง แสดงว่าการลากเส้นขอบได้ลากเส้นระหว่างจุดครบทุกจุดภายในข้อมูลอนุกรมเวลา ซึ่งการคำนวณในครั้งต่อไปจะได้ผลลัพธ์เท่าเดิมเสมอ

Algorithm 2 : Equi-width_Compass_Dimension(*TS*)

```
 1:   TS_Normalized ← Z_SCORE_NORMALIZATION(TS)
 2:   Loop_of_Repeating ← Infinite
 3:
 4:   Foreach Loop_of_Repeating do
 5:     Time_Slice ← SCALE_FACTOR(Length_TS)
 6:     Fixed_Point ← 1
 7:     Foreach Ceil(Length_TS/Time_Slice) do
 8:       Next_Point ← Fixed_Point + Time_Slice
 9:       If Next_point ≤ LengthTS then
10:         Compass_Length ←
            DISTANCE_of_AMPLITUDE(Fixed_Point, Next_Point)
11:         Sum_Compass ← Sum_Compass + Compass_Length
12:       ElseIf Fixed_Point ≥ LengthTS
13:         Break
14:       ElseIf Next_Point ≥ LengthTS
15:         Next_Point ← LengthTS
16:         Compass_Length ←
            DISTANCE_of_AMPLITUDE(Fixed_Point, Next_Point)
17:       EndIf
18:     EndFor
19:     ListCompFunc ← Sum_Compass
20:     ListTime_Slice ← Time_Slice
21:     If Time_Slice = 1
```

```
22:         Break
23:       EndIf
24:     EndFor
25:     ResultEWC ← Regression(ListCompFunc, ListTime_Slice)
26:     Return ResultEWC
```

รูปที่ 3.9 รหัสเทียมสำหรับมิติเส้นขอบตามความกว้างที่เท่ากันของข้อมูลอนุกรมเวลา

สำหรับมิติเส้นขอบตามความกว้างที่เท่ากันสำหรับงานวิจัยนี้ ได้นำเสนอวิธีคำนวณมิติแฟร็กทัลแบบอัตโนมัติเช่นเดียวกับมิติเส้นขอบตามความกว้างที่เท่ากัน ซึ่งวิธีที่ผู้วิจัยนำเสนอเป็นแนวทางในการพัฒนาต่อไปสำหรับการเพิ่มประสิทธิภาพของมิติเส้นขอบตามความกว้างที่เท่ากันให้ได้รับผลของความแม่นยำที่ดีที่สุด

จากผลลัพธ์ที่ได้จากการคำนวณด้วยมิติเส้นขอบตามความยาวที่เท่ากันและมิติเส้นขอบตามความกว้างที่เท่ากันสามารถแทนที่ข้อมูลอนุกรมเดิมที่มีขนาดใหญ่ ให้เหลือเพียง 2 เลขจำนวนจริงเท่านั้น และเห็นได้ว่าวิธีนี้ยังนำเสนอการคำนวณแบบอัตโนมัติกับมิติแฟร็กทัลโดยเน้นไปที่การพัฒนาสำหรับข้อมูลอนุกรมเวลาเป็นหลัก

บทที่ 4

การทดลองและวิเคราะห์ผล

สำหรับบทที่ 4 จะเริ่มจากอธิบายประเภทและชุดข้อมูลที่นำมาใช้สำหรับงานวิจัยนี้ ต่อมากล่าวถึงการทดลองเพื่อวิเคราะห์ประสิทธิภาพของมิติแฟร็กทัลสำหรับการนำมาประยุกต์กับข้อมูลอนุกรมเวลา ในส่วนของการประเมินคุณภาพสำหรับงานวิจัยนี้จะแบ่งการทดลองออกเป็น 2 ส่วนหลัก ๆ คือ วิธีการจำแนกข้อมูลแบบเพื่อนบ้านใกล้ที่สุดอันดับที่หนึ่งด้วยวิธีทดสอบแบบการนำออกหนึ่ง และการทดสอบจากชุดข้อมูลฝึกหัดและข้อมูลทดสอบ ซึ่งวิธีดังกล่าวสามารถแสดงถึงประสิทธิภาพทั้งความแม่นยำและความเร็วของแต่ละวิธีได้อย่างชัดเจนและเป็นวิธีที่หลาย ๆ งานวิจัยได้นำมาใช้ประเมินผลเช่นกัน

4.1 ประเภทและชุดข้อมูลสำหรับงานวิจัย

สำหรับข้อมูลอนุกรมเวลาในงานวิจัยนี้เน้นไปที่ข้อมูลขนาดความยาวตั้งแต่ 1,000 จุดขึ้นไป รวมทั้งในงานวิจัยนี้ได้นำข้อมูลอนุกรมเวลาขนาดสั้นมาทดสอบอีกเช่นกัน เพื่อแสดงว่าการแทนข้อมูลแบบแฟร็กทัลมีประสิทธิภาพหรือไม่ เมื่อทดสอบกับข้อมูลอนุกรมเวลาขนาดสั้นแม้ว่าจะไม่อยู่บนสมมติฐานที่วางไว้ของมิติแฟร็กทัล โดยในแต่ละชุดข้อมูลจะมีความยาวของข้อมูลอนุกรมเวลาเท่ากันทุกตัว โดยประกอบไปด้วยข้อมูลอนุกรมเวลาหลากหลายประเภทและจะถูกสุ่มหยิบมารวมกัน แล้วในแต่ละชุดข้อมูลจะทำการปรับด้วยคะแนน Z สำหรับแต่ละประเภทของข้อมูลอนุกรมเวลาที่นำมาทดลองในงานวิจัยนี้ได้มาจากหน่วยเก็บถาวรของมหาวิทยาลัยแคลิฟอร์เนีย [25, 26] และฟิซิโอเน็ต (PhysioNet) [27] เป็นหลัก ซึ่งข้อมูลอนุกรมที่ได้มาในแต่ละประเภทจะมีความยาวไม่เท่ากัน บางประเภทของข้อมูลอนุกรมเวลามีขนาดสั้นน้อยกว่า 1,000 จุด ซึ่งการสร้างชุดข้อมูลให้มีขนาดใหญ่มากกว่า 1,000 จุด ที่ได้นำเสนอในวัตถุประสงค์การวิจัยกับข้อมูลขนาดสั้น จึงต้องมีการนำความยาวสั้น ๆ หลาย ๆ ตัวมาต่อกันให้ได้เท่ากับความยาวที่ต้องการ หรือบางประเภทมีขนาดประมาณหลักหมื่นจุดจนกระทั่งหลักล้านจุด ซึ่งจะต้องมีการตัดข้อมูลดังกล่าวให้เหลือขนาดของข้อมูลตามที่ต้องการ จากที่กล่าวมาว่างานวิจัยนี้ใช้ความยาวในแต่ละชุดข้อมูลที่มีขนาดเท่ากัน ทำให้ต้องมีการเปลี่ยนแปลงความยาวของข้อมูลดิบให้มีขนาดที่เหมาะสมในแต่ละชุดข้อมูล ดังนั้น ข้อมูลอนุกรมเวลาที่นำมาทดลองได้มาจาก 3 วิธีหลัก ๆ ดังนี้

- ข้อมูลดิบ

  ข้อมูลอนุกรมเวลาประเภทนี้มีความยาวตามขนาดที่ต้องการอยู่แล้ว ดังนั้น จึงไม่มีการเปลี่ยนแปลงความยาวของข้อมูลดิบ ซึ่งข้อมูลแบบนี้จะพบได้ค่อนข้างน้อย

- วิธีสร้างข้อมูลอนุกรมเวลาโดยการแบ่งช่วงเท่า ๆ กัน

ข้อมูลอนุกรมเวลาที่มีความยาวมากจะถูกนำมาแบ่งให้มีขนาดเท่า ๆ กัน โดยข้อมูลอนุกรมเวลาตัวใดที่มีความยาวมากกว่าความยาวที่ต้องการ จะถูกแบ่งโดยนับความยาวจากจุดเริ่มต้นจนถึงขนาดที่ต้องการ เช่น จากจุดที่ 1 ถึง 1,000 แล้วทำการแบ่งข้อมูลภายในช่วงนั้น จะได้ข้อมูลใหม่หนึ่งตัวที่มีความยาว 1,000 จุด ยกตัวอย่างเช่น ข้อมูลอนุกรมเวลาขนาดความยาวเริ่มต้นเท่ากับ 20,000 จุด แต่ต้องการข้อมูลความยาวขนาด 1,000 จุด ดังนั้น จะได้จำนวนข้อมูลเมื่อผ่านการแบ่งเท่ากับ 20 ตัว แต่ละตัวมีความยาวเท่ากับ 1,000 จุด ซึ่งข้อมูลแบบนี้จะพบได้มากที่สุด

- วิธีสร้างข้อมูลอนุกรมเวลาโดยการต่อกันระหว่างข้อมูล

การสร้างข้อมูลอนุกรมเวลาด้วยการต่อกันจะใช้กับข้อมูลที่มีขนาดสั้นกว่าความยาวที่ต้องการ โดยนำข้อมูลอนุกรมเวลาขนาดสั้นสองเส้น และมีประเภทเดียวกันมาต่อกันจากจุดปลายของข้อมูลอนุกรมเวลาตัวแรกมาต่อกับจุดเริ่มต้นของข้อมูลอนุกรมเวลาอีกตัว ยกตัวอย่างเช่น ข้อมูลอนุกรมเวลาสองเส้นมีความยาวเท่ากับ 500 จุด และต้องการความยาว 1,000 จุด โดยจะนำข้อมูลอนุกรมตัวแรกมาอยู่ในตำแหน่งที่ 1 ถึงตำแหน่งที่ 500 และข้อมูลตัวที่สองอยู่ในตำแหน่งที่ 501 ถึง 1,000 ซึ่งเป็นการนำมาวางต่อกันไปเรื่อย ๆ จนกระทั่งได้ความยาวที่ต้องการ

ในหัวข้อถัดไปจะกล่าวถึงรายละเอียดในแต่ชุดข้อมูลสำหรับงานวิจัยนี้ โดยมีการนำชุดข้อมูลแต่ละชุดมาทดสอบเพื่อประเมินประสิทธิภาพสำหรับวิธีต่าง ๆ โดยแบ่งได้เป็น 2 แบบ ดังนี้

4.1.1 ชุดข้อมูลอนุกรมเวลาที่นำไปทดสอบกับการจำแนกข้อมูลแบบเพื่อนบ้านใกล้ที่สุดอันดับที่หนึ่งด้วยวิธีทดสอบแบบการนำออกหนึ่ง

ชุดข้อมูลอนุกรมเวลาของหัวข้อนี้จะนำมาจากข้อมูลอนุกรมเวลาที่สร้างโดยวิธีที่กล่าวไว้ ซึ่งในแต่ละชุดข้อมูลจะมีความยาวที่แตกต่างกันตั้งแต่หลักร้อยจนถึงหลักพัน โดยชุดข้อมูลเหล่านี้จะนำไปใช้ทดสอบสำหรับการวัดประสิทธิภาพด้วยวิธีการจำแนกข้อมูลแบบเพื่อนบ้านใกล้ที่สุดอันดับที่หนึ่งและวิธีทดสอบแบบการนำออกหนึ่งจะเป็นวิธีที่ไม่มีการเอนเอียงในการสร้างชุดข้อมูล เนื่องจาก ข้อมูลทุกตัวต้องมีการเปรียบเทียบกัน และเป็นวิธีที่งานวิจัยต่าง ๆ นิยมนำมาวัดประสิทธิภาพทั้งความแม่นยำและความเร็วอีกเช่นกัน โดยชุดข้อมูลที่ใช้ทดสอบด้วยวิธีนี้จะมีทั้งหมด 9 ชุด ดังนี้

4.1.1.1 ชุดข้อมูลความยาว 1,000 จุด

- ชุดข้อมูลที่หนึ่ง

ชุดข้อมูลที่หนึ่งสร้างขึ้นเพื่อวิเคราะห์ประสิทธิภาพในด้านความเร็วกับจำนวนข้อมูลที่มีปริมาณมากเป็นหลัก โดยชุดข้อมูลนี้มีจำนวนประเภททั้งหมด 5 ประเภท และจำนวนข้อมูล 10,000 อนุกรม โดยแต่ละประเภทจะมีจำนวนเท่า ๆ กันคือ 2,000 อนุกรม ทำให้การทดสอบในแต่ละประเภทจะไม่มีความเอนเอียงไปที่ประเภทใดประเภทหนึ่งมากเกินไป สำหรับประเภทและชุดข้อมูลที่หนึ่งแสดงได้ดังตารางที่ 4.1 และตัวอย่างรูปภาพของแต่ละประเภทแสดงได้ดังภาคผนวก ก

ตารางที่ 4.1 ชุดข้อมูลที่หนึ่งมีจำนวนข้อมูลอนุกรมเวลา 10,000 อนุกรม 5 ประเภท

ประเภทข้อมูล	**จำนวน ข้อมูลอนุกรมเวลา**
ข้อมูลมัลลัท (Mallat Technometrics Dataset)	2,000
ข้อมูลคลื่นหัวใจจากภาวะหยุดหายใจชั่วขณะ (Electrocardiogram Dataset from Apnea Detectors)	2,000
ข้อมูลปริมาณแสง (Labeled Light Curves Dataset)	2,000
ข้อมูลคลื่นสมอง (Electroencepphalogram Dataset)	2,000
ข้อมูลคลื่นหัวใจจากผู้ป่วยลมบ้าหมู (Electrocardiogram Dataset from Partial Epilepsy)	2,000

- ชุดข้อมูลที่สอง

ชุดข้อมูลที่สองเป็นข้อมูลดิบที่ได้มาจากหน่วยเก็บถาวรของมหาวิทยาลัยแคลิฟอร์เนีย [25, 26] โดยตรง ซึ่งชุดข้อมูลนี้จะมีประเภทข้อมูลที่แตกต่างกันถึง 18 ประเภท และแต่ละประเภทมีจำนวนข้อมูลอนุกรมเวลาเพียง 2 ตัวเท่านั้น ซึ่งจะทำให้การจับคู่เพื่อค้นหาข้อมูลประเภทเดียวกันทำได้ยากยิ่งขึ้น สำหรับประเภทและชุดข้อมูลที่สองแสดงได้ดังตารางที่ 4.2 และตัวอย่างรูปภาพของแต่ละประเภทแสดงได้ดังภาคผนวก ก

ตารางที่ 4.2 ชุดข้อมูลที่สองมีจำนวนข้อมูลอนุกรมเวลา 36 อนุกรม 18 ประเภท

ประเภทข้อมูล	จำนวน ข้อมูลอนุกรมเวลา
ข้อมูลการเคลื่อนไหวของกล้ามเนื้อจากผู้ป่วยกระดูกหัก (Motor Current Dataset-Broken Bars)	2
ข้อมูลการเคลื่อนไหวของกล้ามเนื้อจากคนปกติ (Motor Current Dataset-Healthy)	2
ข้อมูลข้อมูลปืน-Ann (Ann Gun Dataset)	2
ข้อมูลข้อมูลปืน-Keogh (Keogh Gun Dataset)	2
ข้อมูลความต้องการพลังงานของชาวอิตาลี (Power Demand Dataset-italian)	2
ข้อมูลความต้องการพลังงานของชาวฮอลแลนด์ (Power Demand Dataset-Dutch)	2
ข้อมูลคลื่นหัวใจของทารกในครรภ์ (Foetal Electrocardiogram Dataset)	2
ข้อมูลความแห้งในอากาศ (Dryer Dataset)	2
ข้อมูลทะเลสาบ (Great Lakes Dataset)	2
ข้อมูลสัญญาณระดับน้ำทะเล (Buoy Sensor Dataset)	2
ข้อมูลคลื่นหัวใจโคสคีแบบช้า (Koski Electrocardiogram Data-Slow)	2
ข้อมูลคลื่นหัวใจโคสคีแบบเร็ว (Koski Electrocardiogram Data-Fast)	2
ข้อมูลแลกเปลี่ยนเงินตรา (Exchange Dataset)	2
ข้อมูลจุดบอดบนดวงอาทิตย์ (Sunspot Dataset)	2
ข้อมูลความร้อนของเตาหลอม (Furnace Dataset)	2
ข้อมูลอัตราการหมุน (Reel Dataset)	2
ข้อมูลบอลลูน (Balloon Dataset)	2
ข้อมูลปริมาณความชื้น (Evaporator Dataset)	2

4.1.1.2 ชุดข้อมูลความยาว 2,000 จุด

- ชุดข้อมูลที่สาม

ชุดข้อมูลที่สามสร้างขึ้นมาเพื่อวิเคราะห์ว่าหากมีจำนวนประเภทที่ค่อนข้างมาก และมีจำนวนข้อมูลในแต่ละประเภทที่แตกต่างกัน แล้ววัดประสิทธิภาพด้านความแม่นยำและความเร็วของการแทนข้อมูลแบบแฟร็กทัลเมื่อเปรียบเทียบกับวิธีอื่น ๆ สำหรับชุดข้อมูลที่สามมีจำนวนข้อมูลอนุกรมทั้งหมด 880 อนุกรม แบ่งประเภทได้เป็น 14 ประเภท โดยประเภทและจำนวนข้อมูลในชุดข้อมูลที่สาม ดังแสดงในตารางที่ 4.3 และตัวอย่างรูปภาพของแต่ละประเภทแสดงได้ดังภาคผนวก ก

ตารางที่ 4.3 ชุดข้อมูลที่สามมีจำนวนข้อมูลอนุกรมเวลา 880 อนุกรม 14 ประเภท

ประเภทข้อมูล	**จำนวนข้อมูลอนุกรมเวลา**
ข้อมูลการรับลูกบอลของหุ่นยนต์ (Field Dataset)	45
ข้อมูลการเล่นฟุตบอลของหุ่นยนต์ (Playing Soccer Dataset)	36
ข้อมูลคลื่นหัวใจโคสคี (Koski Electrocardiogram Dataset)	72
ข้อมูลปากกา (Pen Dataset)	34
ข้อมูลรูปร่าง (Shape Mixed Bag Dataset)	129
ข้อมูลพรม (Lab Carpet Dataset)	45
ข้อมูลอัตราส่วนแนวตั้งและแนวนอนของภาพ (Aspect Ratio Dataset)	13
ข้อมูลคลื่นหัวใจจากการวัดความเครียดของผู้ขับรถยนต์ (Electrocardiogram Dataset from Stress Recognition in Automobile Drivers)	41
ข้อมูลการเคลื่อนที่ของเท้าจากการวัดความเครียดของผู้ขับรถยนต์ (Foot Galvanic Skin Resistance Dataset from Stress Recognition in Automobile Drivers)	41
ข้อมูลเอชอาร์จากการวัดความเครียดของผู้ขับรถยนต์ (HR Dataset from Stress Recognition in Automobile Drivers)	41

ข้อมูลการหายใจจากการวัดความเครียดของผู้ขับรถยนต์ (Respiration Dataset from Stress Recognition in Automobile Drivers)	41
ข้อมูลโชคชะตา (Fortune Dataset)	64
ข้อมูลสองรูปแบบ (Two Pattern Dataset)	83
ข้อมูลปริมาณการส่งข้อมูล (Packet Dataset)	180

4.1.1.3 ชุดข้อมูลความยาว 3,000 จุด

- ชุดข้อมูลที่สี่

ชุดข้อมูลที่สี่สร้างเพื่อวิเคราะห์กับจำนวนประเภทที่ค่อนข้างมากกับจำนวนข้อมูลอนุกรมเวลาของแต่ละประเภทเท่ากันทั้งหมด ซึ่งการทดสอบจะมีการเปรียบเทียบจำนวนข้อมูลที่เหมาะสมกันในแต่ละประเภท สำหรับชุดข้อมูลที่สี่มีจำนวนข้อมูลอนุกรมเวลาเท่ากับ 900 อนุกรม และจำนวนประเภทของข้อมูลเท่ากับ 10 ประเภท โดยประเภทและจำนวนของชุดข้อมูลที่สี่แสดงได้ดังตารางที่ 4.4 และตัวอย่างของรูปภาพในแต่ละประเภทแสดงได้ดังภาคผนวก ก

ตารางที่ 4.4 ชุดข้อมูลที่สี่มีจำนวนข้อมูลอนุกรมเวลา 900 อนุกรม 10 ประเภท

ประเภทข้อมูล	**จำนวน ข้อมูลอนุกรมเวลา**
ข้อมูลติกไวส์ (Tickwise Dataset)	90
ข้อมูลกังฟู (Kungfu Dataset)	90
ข้อมูลการเคลื่อนไหวของกล้ามเนื้อ (Motor Current Dataset)	90
ข้อมูลสลิป (Slip Dataset)	90
ข้อมูลเควเตอร์ (Quater Dataset)	90
ข้อมูลความสว่าง (Light Mesurements Dataset)	90
ข้อมูลแรงดันไฟ (Voltage Measurements Dataset)	90
ข้อมูลคลื่นสมอง (Electroencepphalogram Dataset)	90
ข้อมูลคลื่นสมองจากผู้ป่วยขณะหลับ (Electroencepphalogram Dataset from MIT-BIH Polysomnographic Database)	90

ข้อมูลการหายใจจากผู้ป่วยขณะหลับ (Respiration Dataset from MIT-BIH Polysomnographic Database)	90

4.1.1.4 ชุดข้อมูลความยาว 500 จุด

- ชุดข้อมูลที่ห้า

ชุดข้อมูลที่ห้าสร้างขึ้นเพื่อวิเคราะห์ประสิทธิภาพสำหรับการแทนข้อมูลแบบแฟร็กทัล เมื่อทดสอบกับข้อมูลอนุกรมเวลาขนาดสั้นที่มีความยาวเท่ากับ 500 จุด ซึ่งชุดข้อมูลนี้จะอยู่นอกสมมติฐานสำหรับงานวิจัยนี้ โดยชุดข้อมูลนี้มีจำนวนประเภททั้งหมด 4 ประเภท และจำนวนข้อมูล 600 อนุกรม สำหรับประเภทและชุดข้อมูลที่หนึ่งแสดงได้ดังตารางที่ 4.5 และตัวอย่างรูปภาพของแต่ละประเภทแสดงได้ดังภาคผนวก ก

ตารางที่ 4.5 ชุดข้อมูลที่ห้ามีจำนวนข้อมูลอนุกรมเวลา 600 อนุกรม 4 ประเภท

ประเภทข้อมูล	**จำนวน ข้อมูลอนุกรมเวลา**
ข้อมูลเบิร์สทิน (Burstin Dataset)	100
ข้อมูลปริมารความชื้น (Evaporator Dataset)	72
ข้อมูลความต้องการไฟฟ้า (Power Demand Dataset)	70
ข้อมูลอีอาร์พี (ERP Dataset)	358

4.1.1.5 ชุดข้อมูลความยาว 200 จุด

- ชุดข้อมูลที่หก

ชุดข้อมูลที่หกสร้างขึ้นเพื่อวิเคราะห์ประสิทธิภาพสำหรับข้อมูลอนุกรมเวลาขนาดสั้น ซึ่งเป็นชุดข้อมูลที่ไม่อยู่บนสมมติฐานสำหรับการแทนข้อมูลแบบแฟร็กทัลอีกเช่นกัน โดยชุดข้อมูลนี้มีความยาวของข้อมูลอนุกรมเวลาเท่ากับ 200 จุด จำนวนประเภททั้งหมดเท่ากับ 6 ประเภท และมีจำนวนข้อมูลเท่ากับ 210 อนุกรม สำหรับประเภทและชุดข้อมูลที่หนึ่งแสดงได้ดังตารางที่ 4.6 และตัวอย่างรูปภาพของแต่ละประเภทแสดงได้ดังภาคผนวก ก

ตารางที่ 4.6 ชุดข้อมูลที่หกมีจำนวนข้อมูลอนุกรมเวลา 210 อนุกรม 6 ประเภท

ประเภทข้อมูล	จำนวน ข้อมูลอนุกรมเวลา
ข้อมูลการเคลื่อนไหวของเหลว (Fluid Dynamic Dataset)	50
ข้อมูลบอลลูน (Balloon Dataset)	20
ข้อมูลการประทุ (Burst Dataset)	46
ข้อมูลแผ่นดินไหว (Earthquake Dataset)	20
ข้อมูลคลื่นใต้เสียง (Infra-sound Beam Dataset)	40
ข้อมูลหน่วยความจำ(Memory Dataset)	34

4.1.2 ชุดข้อมูลอนุกรมเวลาที่นำไปทดสอบโดยแบ่งเป็นข้อมูลฝึกหัด และข้อมูลทดสอบ

สำหรับชุดข้อมูลในหัวข้อนี้ถูกแบ่งข้อมูลออกเป็นสองแบบคือ ข้อมูลฝึกหัด และข้อมูลทดสอบ โดยที่จำนวนในแต่ละชุดข้อมูลของข้อมูลฝึกหัดจะมีจำนวนมากกว่าข้อมูลทดสอบค่อนข้างมาก ซึ่งชุดข้อมูลที่มีข้อมูลฝึกหัดเป็นจำนวนมากและมีข้อมูลทดสอบเพียงไม่กี่จำนวนจะพบได้ค่อนข้างมากสำหรับงานเหมืองข้อมูลโดยทั่วไป โดยจะทดสอบกับข้อมูลอนุกรมที่มีความยาว 3 แบบ คือ หนึ่งพัน สองพัน และสามพันจุด รวมทั้งในชุดข้อมูลที่แปดจะมีความแตกต่างจากชุดข้อมูลกลุ่มอื่น ๆ คือ ข้อมูลภายในชุดข้อมูลจะเป็นข้อมูลคลื่นหัวใจเท่านั้น โดยจะมีความแตกต่างกันสำหรับผู้ป่วยแต่ละประเภท แต่คนละชุดข้อมูลที่นำเสนอในหัวข้อนี้มีทั้งหมด 3 ชุด ดังนี้

4.1.2.1 ชุดข้อมูลความยาว 1,000 จุด

- ชุดข้อมูลที่เจ็ด

ชุดข้อมูลที่เจ็ดสร้างขึ้นเพื่อวิเคราะห์กับลักษณะของชุดข้อมูลที่พบได้โดยทั่วไป โดยมีจำนวนข้อมูลฝึกหัดมากกว่าข้อมูลทดสอบ และมีจำนวนข้อมูลในแต่ละประเภทไม่เท่ากันทั้งข้อมูลฝึกหัดและข้อมูลทดสอบ สำหรับชุดข้อมูลที่เจ็ดมีจำนวนข้อมูลอนุกรมเวลาที่เป็นข้อมูลฝึกหัดเท่ากับ 463 อนุกรม และข้อมูลทดสอบเท่ากับ 47 อนุกรม โดยชุดข้อมูลนี้มีประเภทของข้อมูลเท่ากับ 8 ประเภท ดังแสดงในตารางที่ 4.7 และตัวอย่างของรูปภาพในแต่ละประเภทแสดงได้ดังภาคผนวก ก

ตารางที่ 4.7 ชุดข้อมูลที่เจ็ดมีจำนวนข้อมูลฝึกหัดเท่ากับ 463 อนุกรม และจำนวนข้อมูลทดสอบเท่ากับ 47 อนุกรม โดยมีประเภททั้งหมด 8 ประเภท

ประเภทข้อมูล	จำนวนข้อมูลฝึกหัด	จำนวนข้อมูลทดสอบ
ข้อมูลข้อมูลปืน-Ann (Ann Gun Dataset)	40	4
ข้อมูลการกระตุ้นของกล้ามเนื้อ (Muscle Activation Dataset)	20	9
ข้อมูลการยืนของหุ่นยนต์ (Standing Dataset)	25	5
ข้อมูลการเดินของหุ่นยนต์ (Wall Dataset)	40	8
ข้อมูลความต้องการพลังงาน (Power Demand Dataset)	30	5
ข้อมูลการหายใจ (Respiration Dataset)	196	7
ข้อมูลสปอท (Spot Exrates Dataset)	20	4
ข้อมูลท่าทาง (Posture Centroid Dataset)	92	5

4.1.2.2 ชุดข้อมูลความยาว 2,000 จุด

- ชุดข้อมูลที่แปด

สำหรับข้อมูลชุดที่แปดประกอบไปด้วยข้อมูลคลื่นหัวใจทั้งหมด โดยแต่ละประเภทจะเป็นอาการของผู้ป่วยที่แตกต่างกัน โดยชุดข้อมูลนี้มีจำนวนข้อมูลทดสอบเท่ากับ 80 อนุกรม และข้อมูลฝึกหัดเท่ากับ 800 อนุกรม ซึ่งมีจำนวนประเภทเท่ากับ 4 ประเภท ดังแสดงในตารางที่ 4.8 และตัวอย่างของรูปภาพในแต่ละประเภทแสดงได้ดังภาคผนวก ก

ตารางที่ 4.8 ชุดข้อมูลที่แปดมีจำนวนข้อมูลฝึกหัดเท่ากับ 800 อนุกรม และจำนวนข้อมูลทดสอบเท่ากับ 80 อนุกรม โดยมีประเภททั้งหมด 4 ประเภท

ประเภทข้อมูล	จำนวนข้อมูลฝึกหัด	จำนวนข้อมูลทดสอบ
ข้อมูลคลื่นหัวใจจากผู้ป่วยภาวะหัวใจล้มเหลวประเภทที่หนึ่ง (Electrocardiogram Dataset from The BIDMC Congestive Heart Failure Database)	200	20

ข้อมูลคลื่นหัวใจจากผู้ป่วยภาวะหัวใจล้มเหลวประเภทที่สอง (Electrocardiogram Dataset from The BIDMC Congestive Heart Failure Database)	200	20
ข้อมูลคลื่นหัวใจจากผู้ป่วยขณะนอนหงาย (Electrocardiogram Dataset from Fantasia Database)	200	20
ข้อมูลคลื่นหัวใจจากผู้ป่วยขณะหลับ (Electrocardiogram Dataset from MIT-BIH Polysomnographic Database)	200	20

4.1.2.3 ชุดข้อมูลความยาว 3,000 จุด

- ชุดข้อมูลที่เก้า

สำหรับข้อมูลชุดที่เก้าสร้างเพื่อวิเคราะห์ประสิทธิภาพในด้านความเร็วเป็นหลัก สำหรับชุดข้อมูลที่มีข้อมูลฝึกหัดจำนวนมากเมื่อเปรียบเทียบกับข้อมูลทดสอบ สำหรับชุดข้อมูลที่เก้ามีจำนวนข้อมูลฝึกหัดเท่ากับ 6,163 อนุกรม และข้อมูลทดสอบเท่ากับ 300 อนุกรม โดยชุดข้อมูลนี้มีจำนวนประเภทเท่ากับ 6 ประเภท ดังแสดงในตารางที่ 4.9 และตัวอย่างของรูปภาพในแต่ละประเภทแสดงได้ดังภาคผนวก ก

ตารางที่ 4.9 ชุดข้อมูลที่เก้ามีจำนวนข้อมูลฝึกหัดเท่ากับ 6,163 อนุกรม และจำนวนข้อมูลทดสอบเท่ากับ 300 อนุกรม โดยมีประเภททั้งหมด 6 ประเภท

ประเภทข้อมูล	**จำนวน ข้อมูลฝึกหัด**	**จำนวน ข้อมูลทดสอบ**
ข้อมูลความสัมพันธ์ต่อเหตุการณ์ของสมอง (Event-related Brain Potentials Dataset)	958	50
ข้อมูลหุ้น (Stock Dataset)	900	50
ข้อมูลโชคชะตา (Fortune Dataset)	753	50
ข้อมูลคลื่นหัวใจจากผู้ป่วยขณะหลับ (Electrocardiogram Dataset from MIT-BIH Polysomnographic Database)	500	50

ข้อมูลบีพีจากผู้ป่วยขณะหลับ (BP Dataset from MIT-BIH Polysomnographic Database)	500	50
ข้อมูลคลื่นสมอง (Electroencepphalogram Dataset)	2,552	50

4.2 การทดลองเพื่อวิเคราะห์ประสิทธิภาพเพื่อนำมิติแฟร็กทัลมาพัฒนากับข้อมูลอนุกรมเวลา

ในส่วนนี้จะกล่าวถึง การทดลองเพื่อวิเคราะห์ประสิทธิภาพในการนำมิติแฟร็กทัลมาประยุกต์ใช้กับการลดขนาดของข้อมูล ซึ่งเป็นแนวทางเพื่อนำมาพัฒนากับประเภทข้อมูลอนุกรมเวลา โดยก่อนที่จะแสดงผลลัพธ์ของการทดลองกับแต่ละชุดข้อมูล จะกล่าวถึงความเป็นไปได้ที่จะนำมิติแฟร็กทัลมาพัฒนากับข้อมูลอนุกรมเวลา ปัญหาในด้านของความเร็วที่เกิดขึ้นจากการนำมิติความสัมพันธ์มาใช้ร่วมกันในการแทนข้อมูลแบบแฟร็กทัล และสุดท้ายจะกล่าวถึงการนำค่าสัมประสิทธิ์ความสัมพันธ์มาเป็นเกณฑ์วัดสำหรับการเลือกมิติแฟร็กทัลที่เหมาะสมสำหรับข้อมูลอนุกรมเวลา

4.2.1 การทดลองเพื่อวิเคราะห์ประสิทธิภาพของมิติแฟร็กทัลสำหรับการทำเหมืองข้อมูลด้วยเลขจำนวนจริงหนึ่งค่า

จากที่กล่าวไว้ในบทที่ 3 เมื่อผู้วิจัยได้ทดสอบข้อมูลอนุกรมเวลากับมิติแฟร็กทัลด้วยการลดขนาดของข้อมูลให้เหลือเพียงเลขจำนวนจริง 1 ค่า สำหรับข้อมูลอนุกรมเวลา 1 อนุกรม ซึ่งผลลัพธ์ที่เกิดขึ้นจะส่งผลให้ค่าจำนวนจริงของข้อมูลอนุกรมเวลาประเภทเดียวกัน ถูกคร่อมด้วยเลขจำนวนจริงของข้อมูลอนุกรมเวลาประเภทอื่น ทำให้เกิดความผิดพลาดสำหรับการจำแนกข้อมูล ในส่วนนี้จะแสดงถึงความเป็นไปได้ในการนำมิติแฟร็กทัลมาพัฒนาให้มีประสิทธิภาพมากขึ้น โดยจากการสังเกตค่าของเลขจำนวนจริงสำหรับข้อมูลอนุกรมเวลาประเภทเดียวกัน จะได้ผลลัพธ์ที่ไม่แตกต่างกันมากนัก โดยได้ทำการทดลองกับชุดข้อมูลที่สาม ซึ่งชุดข้อมูลนี้เป็นข้อมูลดิบที่ได้จากหน่วยเก็บถาวรโดยตรง และแต่ละประเภทจะมีข้อมูลอนุกรมเวลาประเภทเดียวกันจำนวนสองข้อมูล ซึ่งผลลัพธ์ของค่ามิติแฟร็กทัลในชุดข้อมูลที่สองด้วยการลดขนาดข้อมูลจากมิติเส้นขอบตามความกว้างที่เท่ากันแสดงได้ในตารางที่ 4.10 โดยคอลัมน์แรกแสดงประเภทของข้อมูลอนุกรมเวลา คอลัมน์ที่สองแสดงผลลัพธ์เมื่อผ่านการคำนวณด้วยมิติเส้นขอบ โดยในคอลัมน์นี้จะมีเลขจำนวนจริงสองค่า โดยเป็นค่ามิติแฟร็กทัลของข้อมูลประเภทเดียวกัน แล้วนำค่าจำนวนจริงทั้งสองมาเปรียบเทียบผลต่างกัน เพื่อแสดงค่าความแตกต่างสำหรับข้อมูลประเภทเดียวกัน ซึ่งแสดงในคอลัมน์ที่ 3

ตารางที่ 4.10 เปรียบเทียบผลต่างของค่าจำนวนจริงที่ผ่านการคำนวณด้วยมิติเส้นขอบตามความกว้างที่เท่ากันสำหรับชุดข้อมูลที่สอง

ประเภทข้อมูล	ค่ามิติแฟร็กทัลของข้อมูลประเภทเดียวกัน		ผลต่าง
ข้อมูลการเคลื่อนไหวของกล้ามเนื้อจากผู้ป่วยกระดูกหัก (Motor Current Dataset-Broken Bars)	0.01651	0.01390	0.00260
ข้อมูลการเคลื่อนไหวของกล้ามเนื้อจากคนปกติ (Motor Current Dataset-Healthy)	0.00805	0.00808	0.00003
ข้อมูลข้อมูลปืน-Ann (Ann Gun Dataset)	0.07862	0.04098	0.037641
ข้อมูลข้อมูลปืน-Keogh (Keogh Gun Dataset)	0.02063	0.03833	0.01770
ข้อมูลความต้องการพลังงานอิตาลี (Power Demand Dataset-italian)	0.54954	0.47144	0.07812
ข้อมูลความต้องการพลังงานฮอลแลนด์ (Power Demand Dataset-Dutch)	0.30392	0.29051	0.01340
ข้อมูลคลื่นหัวใจของทารกในครรภ์ (Foetal Electrocardiogram Dataset)	0.35710	0.39760	0.04049
ข้อมูลความแห้ง (Dryer Dataset)	0.00780	0.01133	0.00353
ข้อมูลทะเลสาบ (Great Lakes Dataset)	0.28023	0.23985	0.04037
ข้อมูลสัญญาณระดับน้ำทะเล (Buoy Sensor Dataset)	0.62329	0.567689	0.05560
ข้อมูลคลื่นหัวใจโคสคีแบบช้า (Koski Electrocardiogram Data-Slow)	0.61856	0.60061	0.01795
ข้อมูลคลื่นหัวใจโคสคีแบบเร็ว (Koski Electrocardiogram Data-Fast)	0.56061	0.57843	0.01782
ข้อมูลแลกเปลี่ยนเงินตรา (Exchange Dataset)	0.48129	0.44691	0.03437
ข้อมูลจุดบอดบนดวงอาทิตย์ (Sunspot Dataset)	0.74272	0.77334	0.03061
ข้อมูลเตาหลอม (Furnace Dataset)	0.99474	0.98417	0.01057
ข้อมูลการหมุน (Reel Dataset)	0.08196	0.06415	0.01781

ข้อมูลบอลลูน (Balloon Dataset)	0.88057	0.87126	0.00931
ข้อมูลความชื้น (Evaporator Dataset)	0.99334	0.97563	0.01771

จากตารางที่ 4.10 จะเห็นได้ว่าผลต่างของค่ามิติแฟร็กทัลในประเภทข้อมูลเดียวกันจะมีค่าใกล้เคียงกันและค่าเฉลี่ยของผลต่างจะมีค่าประมาณ 0.02 หน่วย โดยที่ข้อมูลบางประเภทมีผลต่างกันเพียง 0.0003 หน่วยเท่านั้น ซึ่งจากจุดนี้เองมิติแฟร็กทัลจึงน่าจะมีความเป็นไปได้ในการพัฒนาให้มีประสิทธิภาพเพิ่มมากยิ่งขึ้น และได้ผลของความแม่นยำสำหรับชุดข้อมูลที่สองกับมิติเส้นขอบตามความกว้างที่เท่ากันด้วยวิธีการจำแนกข้อมูลแบบเพื่อนบ้านใกล้ที่สุดอันดับที่หนึ่ง และทดสอบแบบการนำออกหนึ่งมีค่าเท่ากับ 33.33 เปอร์เซ็นต์ ต่อมาเมื่อนำชุดข้อมูลที่สองมาทดสอบกับมิติเส้นขอบตามความยาวที่เท่ากันแสดงได้ดังตารางที่ 4.11

ตารางที่ 4.11 เปรียบเทียบผลต่างของค่าจำนวนจริงที่ผ่านการคำนวณด้วยมิติเส้นขอบตามความยาวที่เท่ากันสำหรับชุดข้อมูลที่สอง

ประเภทข้อมูล	**ค่ามิติแฟร็กทัลของข้อมูลประเภทเดียวกัน**		**ผลต่าง**
ข้อมูลการเคลื่อนไหวของกล้ามเนื้อจากผู้ป่วยกระดูกหัก (Motor Current Dataset-Broken Bars)	0.04091	0.02949	0.01142
ข้อมูลการเคลื่อนไหวของกล้ามเนื้อจากคนปกติ (Motor Current Dataset-Healthy)	0.04001	0.04459	0.00457
ข้อมูลข้อมูลปืน-Ann (Ann Gun Dataset)	0.04933	0.01801	0.03131
ข้อมูลข้อมูลปืน-Keogh (Keogh Gun Dataset)	0.01108	0.01834	0.00726
ข้อมูลความต้องการพลังงานอิตาลี (Power Demand Dataset-italian)	0.73465	0.13230	0.60235
ข้อมูลความต้องการพลังงานฮอลแลนด์ (Power Demand Dataset-Dutch)	0.06067	0.06891	0.00825
ข้อมูลคลื่นหัวใจของทารกในครรภ์ (Foetal Electrocardiogram Dataset)	0.13828	0.13748	0.00080
ข้อมูลความแห้ง (Dryer Dataset)	0.07457	0.16441	0.08985
ข้อมูลทะเลสาบ (Great Lakes Dataset)	0.45603	0.48701	0.03098

ข้อมูลสัญญาณระดับน้ำทะเล (Buoy Sensor Dataset)	0.23958	0.19669	0.04288
ข้อมูลคลื่นหัวใจโคสคีแบบช้า (Koski Electrocardiogram Data-Slow)	0.07401	0.09385	0.01985
ข้อมูลคลื่นหัวใจโคสคีแบบเร็ว (Koski Electrocardiogram Data-Fast)	0.06960	0.07166	0.00206
ข้อมูลแลกเปลี่ยนเงินตรา (Exchange Dataset)	0.36620	0.35185	0.01435
ข้อมูลจุดบอดบนดวงอาทิตย์ (Sunspot Dataset)	0.26858	0.27793	0.00935
ข้อมูลเตาหลอม (Furnace Dataset)	0.01563	0.00073	0.01490
ข้อมูลการหมุน (Reel Dataset)	0.04266	0.06619	0.02353
ข้อมูลบอลลูน (Balloon Dataset)	0.17715	0.17580	0.00134
ข้อมูลความชื้น (Evaporator Dataset)	0.00184	0.00409	0.00225

จากตารางที่ 4.11 จะพบว่า ค่าจำนวนจริงเมื่อผ่านการคำนวณด้วยมิติเส้นขอบตามความยาวที่เท่ากันจะให้ผลลัพธ์ที่ใกล้เคียงอีกเช่นกัน โดยมีค่าเฉลี่ยความแตกต่างประมาณ 0.05 หน่วย และได้ผลความแม่นยำเท่ากับ 33.33 เปอร์เซ็นต์ ซึ่งผู้วิจัยจึงเล็งเห็นว่า เลขจำนวนจริงที่ใกล้เคียงกันจากทั้งสองวิธี ถ้านำวิธีทั้งสองมาใช้ร่วมกันจากหนึ่งเลขจำนวนจริงสำหรับข้อมูลอนุกรมเวลาหนึ่งตัวเป็นเลขจำนวนจริงสองค่าสำหรับแต่ละข้อมูลอนุกรมเวลา เมื่อคำนวณหาผลของความแม่นยำด้วยวิธีเดิมจะได้เท่ากับ 63.89 เปอร์เซ็นต์ ซึ่งเป็นไปตามที่ตั้งสมมติฐานไว้ว่า หากนำผลลัพธ์ของแต่ละมิติมาช่วยกันแทนข้อมูลอนุกรมเวลาที่มีจำนวนจริงเพิ่มขึ้นจะทำให้ผลความแม่นยำที่ได้เพิ่มมากขึ้นตามไปด้วย

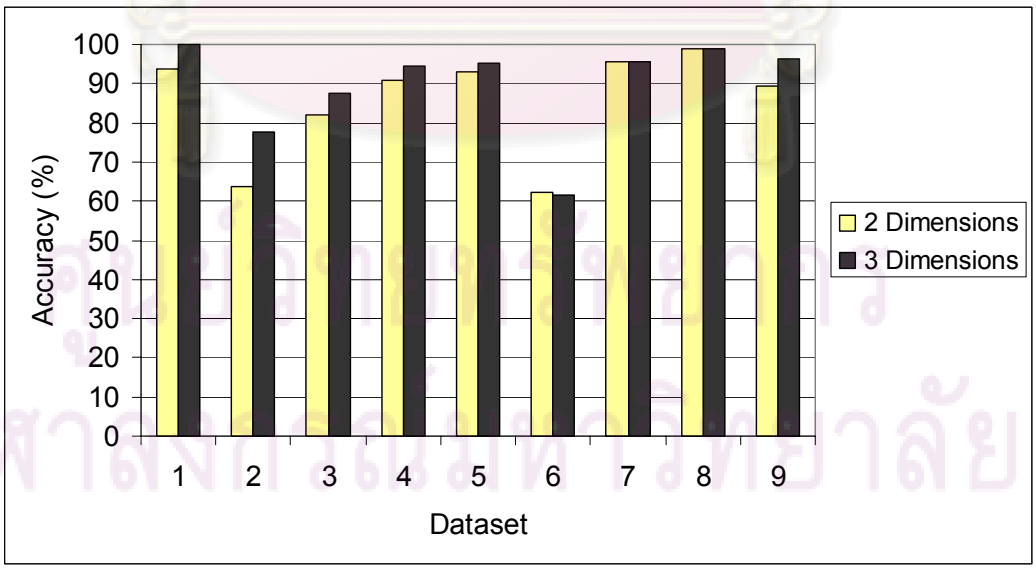
4.2.2 การทดลองเพื่อวิเคราะห์ประสิทธิภาพของมิติความสัมพันธ์

ในส่วนนี้จะกล่าวถึงปัญหาที่เกิดขึ้นจากการใช้มิติความสัมพันธ์มาพัฒนากับการแทนข้อมูลแบบแฟร็กทัล ในส่วนแรกจะกล่าวถึงแนวทางในการคำนวณสำหรับมิติความสัมพันธ์กับข้อมูลอนุกรมเวลาที่ได้ทดลองไว้ ซึ่งจะอธิบายแนวทางอย่างคร่าว ๆ โดยเริ่มต้นจาก กำหนดขีดแบ่งระยะทาง *r* สำหรับข้อมูลอนุกรมเวลา โดยใช้สูตรการคำนวณเดียวกับมิติเส้นขอบตามความยาวที่เท่ากัน ต่อมาทำการคำนวณหาความสัมพันธ์ที่เกิดขึ้นภายในข้อมูลอนุกรมเวลาเพื่อหาค่าอินทิกรัลความสัมพันธ์ โดยคำนวณหาระยะทางของค่าสัมบูรณ์จากค่าจำนวนจริงของจุดหนึ่งเปรียบเทียบกับทุกจุดภายในข้อมูลโดยการลบกัน ซึ่งการคำนวณด้วยวิธีนี้จะใช้ค่าเฉพาะแนวแกนตั้งเช่นกัน และเปรียบเทียบกับขีดแบ่งระยะทางโดยค่าใดมีค่าน้อยกว่าหรือเท่ากับจะให้ค่าเท่ากับ 1 แล้วกระซ้ำทำกับทุก ๆ จุดภายในข้อมูลอนุกรมเวลาจะได้ผลลัพธ์ของค่าอินทิกรัลความสัมพันธ์สำหรับขีดแบ่งระยะทางเริ่มต้น ต่อมาลดขนาด

ของขีดแบ่งระยะทางให้เป็นครึ่งหนึ่งของค่าก่อนหน้า แล้วคำนวณหาค่าอินทิกรัลความสัมพันธ์อีกครั้งกับขีดแบ่งระยะทางใหม่ ซึ่งจะมีจุดหยุดการคำนวณก็ต่อเมื่อ ค่าอินทิกรัลความสัมพันธ์ไม่มีการเปลี่ยนแปลงเมื่อเปรียบเทียบกับรอบก่อนหน้าหรือมีค่าเท่ากับ 0 หน่วย และในที่สุดแต่ละขีดแบ่งระยะทาง และค่าอินทิกรัลความสัมพันธ์จะถูกนำมาหาความชัน ซึ่งค่าจำนวนจริงที่ได้คือ มิติความสัมพันธ์

ซึ่งจะเห็นว่า แต่ละรอบการคำนวณของอินทิกรัลความสัมพันธ์จะใช้เวลาประมาณ $O(n^2)$ ดังนั้น เพื่อลดเวลาการคำนวณของมิติความสัมพันธ์จะใช้การค้นหาแบบไบนารี (Binary Search) เพื่อลดเวลาให้เหลือประมาณ $O(n\log(n))$ โดยทำการเรียงลำดับค่าภายในของข้อมูลอนุกรมเวลาก่อน ซึ่งสามารถทำได้เนื่องจาก วิธีการคำนวณมิติความสัมพันธ์จะไม่ใช้คุณสมบัติการมีลำดับของข้อมูลอนุกรมเวลา แล้วใช้การค้นหาแบบไบนารีเพื่อหาตำแหน่งของจุดที่มากกว่าขีดแบ่งระยะทาง ซึ่งจุดทั้งหมดที่อยู่ภายในช่วงของตำแหน่งที่คำนวณหามาได้จะถูกเก็บไว้ แล้วกระทำซ้ำจนกระทั่งครบทุกจุด ผลรวมของค่าความสัมพันธ์ที่คำนวณได้ในแต่ละจุดจะนำมารวมกัน ซึ่งก็คือค่าอินทิกรัลความสัมพันธ์ในแต่ละรอบของแต่ละขีดแบ่งระยะทาง

เมื่อทดลองเปรียบเทียบประสิทธิภาพของคามเร็วและความแม่นยำกับการแทนข้อมูลแบบแฟร็กทัลสำหรับจำนวนจริงสองค่า ซึ่งมาจาก มิติเส้นขอบตามความกว้างที่เท่ากันและมิติเส้นขอบตามความยาวที่เท่ากัน เปรียบเทียบกับผลของความแม่นยำและเวลาด้วยการแทนข้อมูลแบบแฟร็กทัลสำหรับจำนวนจริงสามค่า ซึ่งมีมิติความสัมพันธ์เพิ่มขึ้นมา ซึ่งผลของความแม่นยำเมื่อทำการเปรียบเทียบกับทุกชุดข้อมูลของงานวิจัยแสดงได้ดังรูปที่ 4.1 และความเร็วในการประมวลผลสำหรับทุกชุดข้อมูลของงานวิจัยแสดงได้ดังตารางที่ 4.12



รูปที่ 4.1 ผลความแม่นยำเมื่อเปรียบเทียบระหว่างการแทนข้อมูลแบบแฟร็กทัลแบบเลขจำนวนจริงสองค่า และเลขจำนวนจริงสามค่า

ตารางที่ 4.12 ผลของเวลาเมื่อเปรียบเทียบระหว่างการแทนข้อมูลแบบแฟร็กทัลแบบเลขจำนวนจริงสองค่า และเลขจำนวนจริงสามค่า

ลำดับชุดข้อมูล	ของจำนวนจริงสองค่า (วินาที)	เวลาของจำนวนจริงสามค่า (วินาที)
ชุดข้อมูลที่หนึ่ง	894.235	7172.987
ชุดข้อมูลที่สอง	2.131	26.396
ชุดข้อมูลที่สาม	66.951	1105.1
ชุดข้อมูลที่สี่	133.241	2389.705
ชุดข้อมูลที่ห้า	15.282	175.998
ชุดข้อมูลที่หก	2.893	29.126
ชุดข้อมูลที่เจ็ด	2.648	32.96
ชุดข้อมูลที่แปด	5.098	85.237
ชุดข้อมูลที่เก้า	53.655	633.68

จากผลการทดลองจะเห็นได้ว่า จากปัญหาที่กล่าวไว้สำหรับการนำมิติความสัมพันธ์มาเพิ่มจากสองมิติเป็นสามมิติจะส่งผลต่อเวลาในการคำนวณค่อนข้างมาก จากผลของเวลาในตารางที่ 4.12 เกือบทุกชุดข้อมูลของเวลาในการคำนวณด้วยเลขจำนวนจริงสามค่าจะใช้เวลาคำนวณมากกว่าถึง 10 เท่า เมื่อเปรียบเทียบกับเลขจำนวนจริงสามค่า นอกจากชุดข้อมูลที่หนึ่ง แต่เวลาสำหรับชุดข้อมูลที่หนึ่งก็ยังใช้มากกว่าประมาณ 8 เท่า และสำหรับชุดข้อมูลที่สี่เลขจำนวนจริงสามค่าใช้เวลาในการคำนวณมากกว่าถึง 18 เท่า และเมื่อวัดจากผลของความแม่นยำ จากรูปที่ 4.1 โดยส่วนมากจะมีค่าไม่แตกต่างกันมากนัก ในบางชุดข้อมูลจะได้ผลของควาแม่นยำเท่ากัน ซึ่งแสดงถึงมิติความสัมพันธ์ไม่ได้ช่วยเพิ่มประสิทธิภาพในการคำนวณเลย และถึงแม้ว่าชุดข้อมูลที่สองจะมีความแม่นยำแตกต่างกันประมาณ 13 เปอร์เซ็นต์ แต่เมื่อหาค่าเฉลี่ยของผลต่างกับทุกชุดข้อมูลแล้ว จะแตกต่างกันเพียง 4.13 เปอร์เซ็นต์เท่านั้น ซึ่งเมื่อเปรียบเทียบกับเวลาในการคำนวณเพิ่มขึ้นมาจากมิติความสัมพันธ์ประมาณ 10 เท่า ดังนั้น การเลือกเพียงแค่เลขจำนวนจริงสองค่าก็เพียงพอสำหรับผลการทดลองที่มีประสิทธิภาพทั้งในด้านเวลาและความแม่นยำ

4.2.3 การทดลองเพื่อวิเคราะห์ความสัมพันธ์ของแต่ละมิติแฟร็กทัล

ในงานวิจัยนี้ได้นำมิติแฟร็กทัลมาประยุกต์ให้มีความเหมาะสมกับข้อมูลอนุกรมเวลา ซึ่งได้เลือกวิธีมิติเส้นขอบตามความยาวที่เท่ากันและมิติเส้นขอบตามความกว้างที่เท่ากัน

โดยทั้งสองวิธีถูกประยุกต์มาจากหลักการของมิติเส้นขอบ ซึ่งจะเห็นได้ว่า การเลือกมิติเส้นขอบมาประยุกต์ใช้กับข้อมูลอนุกรมเวลาในสองแนวทาง เมื่อลดขนาดข้อมูลอนุกรมเวลาด้วยวิธีทั้งสอง ซึ่งผลลัพธ์ที่ได้ในชุดข้อมูลชนิดเดียวกันอาจจะเกิดความสัมพันธ์ของตัวเลขในทิศทางเดียวกันหรือไม่ แล้วถ้าเลือกมิติแฟร็กทัลชนิดอื่นที่มีค่าความสัมพันธ์ของตัวเลขที่น้อยกว่า มาจับคู่ร่วมกันน่าจะทำให้ประสิทธิภาพความแม่นยำเพิ่มมากขึ้น หรือไม่เป็นปัจจัยในด้านความสัมพันธ์ ซึ่งในส่วนนี้จะใช้การคำนวณหาความสัมพันธ์ด้วยสัมประสิทธ์ความสัมพันธ์ (Correlation Coefficient) โดยเปรียบเทียบกับ 3 วิธีคือ มิติเส้นขอบตามความยาวที่เท่ากัน มิติเส้นขอบตามความกว้างที่เท่ากัน และมิติความสัมพันธ์

จากที่ได้ศึกษามิติแฟร็กทัลมิติเส้นขอบและมิติความสัมพันธ์จะมีแนวโน้มที่เหมาะสม เพื่อนำมาพัฒนากับข้อมูลอนุกรมเวลาได้ตามคุณลักษณะของข้อมูลอนุกรมเวลา เช่น มิติเส้นขอบเป็นวิธีการลากเส้นตามแนวขอบของวัตถุ ซึ่งข้อมูลอนุกรมเวลาที่มีลักษณะเป็นจุดและมีลำดับอยู่แล้วทำให้สามารถลากเส้นระหว่างจุดกันได้ หรือมิติความสัมพันธ์เป็นวิธีการคำนวณหาความสัมพันธ์ระหว่างจุดภายในปริภูมิ ซึ่งข้อมูลอนุกรมเวลาที่มีลักษณะเป็นจุด และสามารถวัดระยะทางระหว่างจุดเพื่อหาความสัมพันธ์ในแต่ละข้อมูล แต่ในบางวิธี เช่น มิตินับช่องเป็นการตีช่องที่มีขนาดเท่ากันบนข้อมูล โดยมิตินับช่องจะนับข้อมูลจำนวนช่องที่มีข้อมูลผ่าน ซึ่งข้อมูลอนุกรมเวลาเป็นข้อมูลจุดจำนวนมากกระจายตัวกัน ทำให้การนับรวมหลาย ๆ จุดของข้อมูลอนุกรมเวลาให้มีค่าเพียง 1 หน่วย ไม่น่าจะเป็นวิธีที่เหมาะสมสำหรับการค้นหาความคล้ายคลึง หรือมิติสารสนเทศจะมีแนวทางการคำนวณคล้ายกับมิตินับช่องแต่จะมีการกำหนดน้ำหนักเข้ามาแทนซึ่งจะเหมาะสมกว่ามิตินับช่อง แต่การกำหนดเพียงน้ำหนักให้กับแต่ละช่องก็น่าจะยังไม่ละเอียดเพียงพอเมื่อเปรียบเทียบกับการคำนวณของมิติความสัมพันธ์ ซึ่งในทางทฤษฎีของมิติแฟร็กทัลทั้งมิตินับช่อง มิติสารสนเทศ และมิติความสัมพันธ์จะมีความสัมพันธ์กัน โดยมิติความสัมพันธ์จะคำนวณระยะทางระหว่างทุก ๆ จุดภายในปริภูมิ ซึ่งน่าจะนำมาประยุกต์มากกว่าการกำหนดช่องแบบตายตัว โดยทั้งมิตินับช่องและมิติสารสนเทศจะนำมาใช้กับข้อมูลรูปภาพเป็นหลัก และมิติความคล้ายคลึงเป็นทฤษฎีสำหรับอธิบายความเป็นแฟร็กทัล ซึ่งวิธีนี้จะใช้กับข้อมูลที่สามารถปรับไปที่ระดับใดก็ตามจะยังพบบางส่วนหรือทั้งหมดที่ยังคล้ายกับข้อมูลตั้งตนอยู่ ซึ่งไม่น่าจะเหมาะสมสำหรับการนำมาใช้กับข้อมูลอนุกรมเวลา

สัมประสิทธิ์ความสัมพันธ์เป็นตัววัดค่าความสัมพันธ์ที่เกิดขึ้นกับข้อมูลสองกลุ่ม ซึ่งมีทิศทางการเปลี่ยนแปลงค่าของภาพรวมทั้งหมดมีความแตกต่างกันมากน้อยเพียงใด ซึ่งเปรียบเทียบกันในเชิงตัวเลข โดยผลลัพธ์ที่ได้จะอยู่ระหว่าง 0 ถึง 1 โดยถ้าค่าเข้าใกล้หนึ่งมากแสดงว่าค่าตัวเลขสองกลุ่มมีความสัมพันธ์กันมาก แต่ถ้าค่าลู่เข้าใกล้ 0 แสดงว่าค่าตัวเลขทั้งสองกลุ่มไม่มีแนวโน้วที่คล้ายกัน ถ้าหาสำหรับสูตรการคำนวณด้วยสัมประสิทธิ์ความสัมพันธ์แสดงได้ดังสมการที่ 4.1

$$Correlation\_Coefficient = \left| \frac{\sum XY - \frac{(\sum X)(\sum Y)}{n}}{\sqrt{\left[(\sum X^2) - \frac{(\sum X)^2}{n}\right] \bullet \left[(\sum Y^2) - \frac{(\sum Y)^2}{n}\right]}} \right| \quad (4.1)$$

โดยที่ X และ Y คือกลุ่มค่าที่คำนวณได้จากหนึ่งวิธีของมิติแฟร็กทัลสำหรับหนึ่งชุดข้อมูล และ n คือจำนวนทั้งหมดของชุดข้อมูลนั้น ๆ

ในส่วนต่อไปจะทำการคำนวณหาผลลัพธ์ของค่าสัมประสิทธิ์ความสัมพันธ์สำหรับการคำนวณเพื่อหาความสัมพันธ์ที่เกิดขึ้นกับทุกชุดข้อมูลของงานวิจัย โดยเปรียบเทียบกันระหว่างมิติเส้นขอบตามความกว้างที่เท่ากัน มิติเส้นขอบตามความยาวที่เท่ากัน และมิติความสัมพันธ์ ซึ่งจะแบ่งเป็นสองตาราง สำหรับตารางที่หนึ่งจะเป็นการทดสอบกับการจำแนกข้อมูลแบบเพื่อนบ้านใกล้ที่สุดอันดับที่หนึ่งด้วยวิธีทดสอบแบบการนำออกหนึ่ง ดังแสดงในตารางที่ 4.13 และตารางที่สองจะเป็นการทดสอบกับชุดข้อมูลทดสอบและข้อมูลฝึกหัด โดยจะทำการคำนวณค่าสัมประสิทธิ์ความสัมพันธ์ทั้งข้อมูลทดสอบและข้อมูลฝึกหัด ดังแสดงในตารางที่ 4.14

ตารางที่ 4.13 ค่าของสัมประสิทธิ์ความสัมพันธ์จากการเปรียบเทียบแต่ละคู่จากสามของมิติแฟร็กทัลคือ มิติเส้นขอบตามความกว้างที่เท่ากัน มิติเส้นขอบตามความยาวที่เท่ากัน และมิติความสัมพันธ์ สำหรับชุดข้อมูลที่หนึ่งถึงหก

ลำดับชุดข้อมูล	ค่าความสัมพันธ์ของมิติเส้นขอบตามความกว้างและตามความยาวที่เท่ากัน	ค่าความสัมพันธ์ของมิติเส้นขอบตามความกว้างที่เท่ากันและมิติความสัมพันธ์	ค่าความสัมพันธ์ของมิติเส้นขอบตามความยาวที่เท่ากันและมิติความสัมพันธ์
ชุดข้อมูลที่หนึ่ง	0.706125	0.632636	0.637554
ชุดข้อมูลที่สอง	0.118691	0.162824	0.187922
ชุดข้อมูลที่สาม	0.706109	0.160668	0.160197
ชุดข้อมูลที่สี่	0.160944	0.486005	0.31442
ชุดข้อมูลที่ห้า	0.421796	0.040576	0.343309
ชุดข้อมูลที่หก	0.598191	0.630095	0.481519

ตารางที่ 4.14 ค่าของสัมประสิทธิ์ความสัมพันธ์จากการเปรียบเทียบแต่ละคู่จากสามของมิติแฟร็กทัลคือ มิติเส้นขอบตามความกว้างที่เท่ากัน มิติเส้นขอบตามความยาวที่เท่ากัน และมิติความสัมพันธ์ สำหรับชุดข้อมูลที่เจ็ดถึงเก้า

ลำดับชุดข้อมูล	ประเภทชุดข้อมูล	ค่าความสัมพันธ์ของมิติเส้นขอบตามความกว้างและตามความยาวที่เท่ากัน	ค่าความสัมพันธ์ของมิติเส้นขอบตามความกว้างที่เท่ากันและมิติความสัมพันธ์	ค่าความสัมพันธ์ของมิติเส้นขอบตามความยาวที่เท่ากันและมิติความสัมพันธ์
ชุดข้อมูลที่เจ็ด	ข้อมูลฝึกหัด	0.57133	0.24056	0.400964
	ข้อมูลทดสอบ	0.579402	0.295913	0.745563
ชุดข้อมูลที่แปด	ข้อมูลฝึกหัด	0.958367	0.064814	0.032699
	ข้อมูลทดสอบ	0.965795	0.06055	0.077073
ชุดข้อมูลที่เก้า	ข้อมูลฝึกหัด	0.245535	0.22328	0.050767
	ข้อมูลทดสอบ	0.230701	0.311442	0.013005

จากตารางที่ 4.13 และ 4.14 ในคอลัมน์ที่หนึ่งถึงสามแสดงผลลัพธ์ของค่าสัมประสิทธิ์ความสัมพันธ์ โดยในคอลัมน์ที่หนึ่งเป็นการเปรียบเทียบกันระหว่างมิติเส้นขอบตามความกว้างที่เท่ากัน และมิติเส้นขอบตามความยาวที่เท่ากัน ซึ่งเป็นสองวิธีที่งานวิจัยนี้นำมาใช้ คอลัมน์ที่สองเปรียบเทียบระหว่างมิติเส้นขอบตามความกว้างที่เท่ากับและมิติความสัมพันธ์ และคอลัมน์ที่สามเปรียบเทียบระหว่างมิติเส้นขอบตามความยาวที่เท่ากันและมิติความสัมพันธ์ ซึ่งจะพบว่า เมื่อเปรียบเทียบค่ามิติเส้นขอบสำหรับงานวิจัยนี้ ถึงแม้ว่าชุดข้อมูลที่แปดได้ค่าความสัมพันธ์ที่เข้าใกล้หนึ่งมาก แต่ชุดข้อมูลอื่น ๆ ก็จะมีการกระจายตัวกันตั้งแต่ 0.1 ถึง 0.7 หน่วย ซึ่งแสดงได้ว่าวิธีทั้งสองถึงแม้ว่าจะใช้หลักการของมิติเส้นขอบ แต่ผลของค่าความสัมพันธ์ที่ได้รับจะมีค่าที่แตกต่างกันในแต่ละชุดข้อมูล ซึ่งคล้ายกับคอลัมน์ที่สอง และคอลัมน์ที่สาม ที่มีค่าที่กระจายกันระหว่าง 0 ถึง 0.8 หน่วย

ซึ่งในลำดับต่อไปจะทำการทดสอบว่า ค่าสัมประสิทธิ์ความสัมพันธ์ที่ได้จะมีความเกี่ยวข้องกับผลของความแม่นยำหรือไม่ โดยที่ถ้าแต่ละคู่ของค่ามิติแฟร็กทัลสำหรับชุดข้อมูลใด ๆ ที่มีค่าสัมประสิทธิ์ความสัมพันธ์ค่อนข้างมาก และให้ผลของความแม่นยำเมื่อแยกการคำนวณออกจากกันในแต่ละวิธีและได้รับผลความแม่นยำที่สูงใกล้เคียงกัน แต่เมื่อนำคู่ที่ให้ผลของค่าสัมประสิทธิ์ความสัมพันธ์ที่ลู่เข้าสู่ศูนย์หรือน้อยกว่าแทน จะส่งผลให้ผลความแม่นยำสูงขึ้นหรือไม่ ซึ่งผลความแม่นยำสำหรับทุกชุดข้อมูลสำหรับแต่ละคู่ของทั้งสามวิธีสำหรับ

มิติแฟร็กทัลแสดงได้ดังตารางที่ 4.15 และผลความแม่นยำสำหรับทุกชุดข้อมูลสำหรับแต่ละวิธีของมิติแฟร็กทัลแสดงได้ดังตารางที่ 4.16

ตารางที่ 4.15 ผลของความแม่นยำจากการเปรียบเทียบแต่ละคู่จากสามของมิติแฟร็กทัลคือ มิติเส้นขอบตามความกว้างที่เท่ากัน มิติเส้นขอบตามความยาวที่เท่ากัน และมิติความสัมพันธ์สำหรับชุดข้อมูลที่หนึ่งถึงสิบ

ลำดับชุดข้อมูล	ผลความแม่นยำของมิติเส้นขอบตามความกว้างและความยาวที่เท่ากัน (เปอร์เซ็นต์)	ผลควาแม่นยำของมิติเส้นขอบตามความกว้างที่เท่ากันและมิติความสัมพันธ์ (เปอร์เซ็นต์)	ผลควาแม่นยำของมิติเส้นขอบตามความยาวที่เท่ากันและมิติความสัมพันธ์ (เปอร์เซ็นต์)
ชุดข้อมูลที่หนึ่ง	94.58	95.28	90.57
ชุดข้อมูลที่สอง	63.88889	69.44444	44.44444
ชุดข้อมูลที่สาม	82.08092	74.21965	66.12717
ชุดข้อมูลที่สี่	91	92.66667	84.55556
ชุดข้อมูลที่ห้า	93.16667	76.83333	81
ชุดข้อมูลที่หก	62.38095	54.7619	34.28571
ชุดข้อมูลที่เจ็ด	95.74468	76.59574	76.59574
ชุดข้อมูลที่แปด	98.75	97.5	96.25
ชุดข้อมูลที่เก้า	89.33333	91.33333	71.33333

ตารางที่ 4.16 ผลความแม่นยำของแต่ละค่ามิติแฟร็กทัลสำหรับแต่ละวิธีกับทุกชุดข้อมูล

ลำดับชุดข้อมูล	ผลความแม่นยำของมิติเส้นขอบตามความกว้างที่เท่ากัน (เปอร์เซ็นต์)	ผลความแม่นยำของมิติเส้นขอบตามความยาวที่เท่ากัน (เปอร์เซ็นต์)	ผลความแม่นยำของมิติความสัมพันธ์ (เปอร์เซ็นต์)
ชุดข้อมูลที่หนึ่ง	70.20	63.45	73.23
ชุดข้อมูลที่สอง	33.33	33.3333	13.8889
ชุดข้อมูลที่สาม	54.34	43.815	25.4335

ชุดข้อมูลที่สี่	67.11	64.333	62.777
ชุดข้อมูลที่ห้า	67.33	73.666	66
ชุดข้อมูลที่หก	50.95	23.333	31.9048
ชุดข้อมูลที่เจ็ด	59.57	44.68	59.57
ชุดข้อมูลที่แปด	91.25	70	50
ชุดข้อมูลที่เก้า	69.00	39.333	52.667

จากตารางที่ 4.15 และ 4.16 พบว่า สำหรับชุดข้อมูลที่หนึ่ง ค่าความสัมพันธ์ของคอลัมน์ที่หนึ่งจะให้ค่ามากที่สุด และให้ผลความแม่นยำเท่ากับ 94.58 เปอร์เซ็นต์ โดยเมื่อคำนวณผลความแม่นยำโดยใช้เพียงหนึ่งมิติแฟร็กทัลได้ผลความแม่นยำสำหรับมิติเส้นขอบตามความกว้างที่เท่ากันเท่ากับ 70.20 เปอร์เซ็นต์ และมิติเส้นขอบอีกวิธีได้เท่ากับ 63.45 เปอร์เซ็นต์ แต่เมื่อเปรียบเทียบกับค่าความสัมพันธ์ในคอลัมน์ที่สามที่มีค่าน้อยกว่า โดยที่มิติความสัมพันธ์ให้ผลความแม่นยำเท่ากับ 74.23 ซึ่งได้ค่ามากกว่าอีกทั้งสองวิธี แต่เมื่อใช้ค่าสัมประสิทธิ์ความสัมพันธ์มาเลือกคู่ขอมิติแฟร็กทัล ซึ่งจะเลือกคู่มิติแฟร็กทัลที่ให้ค่าสัมประสิทธิ์ความสัมพันธ์ที่น้อยกว่า โดยที่แต่ละวิธีให้ผลความแม่นยำใกล้เคียงกัน แต่เมื่อเปรียบเทียบผลความแม่นยำของคู่มิติแฟร็กทัลในตารางที่ 4.15 ค่าสัมประสิทธิ์ความสัมพันธ์น้อยกว่ากลับให้ผลลัพธ์ของความแม่นยำลดลง ชุดข้อมูลที่ห้า ค่าสัมประสิทธิ์ความสัมพันธ์ของคอลัมน์ที่สองจะลู่เข้าสู่ศูนย์ แต่กลับได้ผลความแม่นยำที่น้อยกว่าคู่ที่ให้ค่าความสัมพันธ์มากกว่า และในชุดข้อมูลที่เจ็ด ก็จะเป็นในแนวทางเดียวกัน โดยคู่ที่ให้ค่าสัมประสิทธิ์ความสัมพันธ์ที่น้อยกว่า แต่ให้ผลความแม่นยำสำหรับการคำนวณแยกในแต่ละวิธี ก็ยังคงทำให้ผลความแม่นยำน้อยกว่าเช่นกัน ซึ่งจะเห็นได้ว่า การเลือกแต่ละสำหรับมิติแฟร็กทัลด้วยค่าของสัมประสิทธิ์ความสัมพันธ์ไม่น่าจะมีความเกี่ยวข้องกับผลของความแม่นยำสำหรับมิติแฟร็กทัล ซึ่งประเด็นที่สำคัญสำหรับการแทนข้อมูลแบบแฟร็กทัลน่าจะอยู่ที่หลักการคำนวณของแต่ละวิธีที่สามารถให้ค่าของเลขจำนวนจริงที่ใกล้เคียงกันมากที่สุดสำหรับข้อมูลประเภทเดียวกัน ซึ่งจะส่งผลให้มีประสิทธิภาพความแม่นยำที่สูงขึ้น

4.3 การทดลองเกี่ยวกับการวิเคราะห์ประสิทธิภาพของการแทนข้อมูลแบบแฟร็กทัล

ในหัวข้อนี้เป็นการวิเคราะห์ประสิทธิภาพสำหรับการแทนข้อมูลแบบแฟร็กทัล เพื่อประเมินผลของงานวิจัยนี้ โดยเริ่มจากอธิบายวิธีการทดลองสำหรับงานวิจัยอื่น ๆ ที่ใช้ในการเปรียบเทียบ สำหรับการค้นหาความคล้ายคลึง ได้แก่ การวัดระยะทางแบบยุคลิด การวัดระยะทางแบบไดนามิกไทม์วอร์ปปิง และวิธีซีดีเอ็ม การลดขนาดของข้อมูลอนุกรมเวลา ได้แก่ การแทนข้อมูลแบบแซค และการแทนข้อมูลแบบคลิป และสุดท้ายจะแสดงผลการทดลองเพื่อวัดประสิทธิภาพของความเร็วและความแม่นยำของแต่ละวิธีเปรียบเทียบกับการแทนข้อมูลแบบ

แฟร็กทัล โดยแบ่งการทดลองออกเป็นสองส่วนคือ ทดสอบกับวิธีการจำแนกข้อมูลแบบเพื่อนบ้านใกล้ที่สุดอันดับที่หนึ่งด้วยการทดสอบแบบการนำออกหนึ่ง และทดสอบกับชุดข้อมูลฝึกหัดและข้อมูลทดสอบ

4.3.1 วิธีการทดลองกับงานวิจัยอื่น ๆ ที่นำมาเปรียบเทียบ

ในส่วนนี้จะอธิบายถึงขั้นตอนการทำงานของแต่ละวิธีที่งานวิจัยนี้ได้นำมาเปรียบเทียบกับการแทนข้อมูลแบบแฟร็กทัล โดยจะเปรียบเทียบประสิทธิภาพในฝั่งของการลดขนาดของข้อมูลอนุกรมเวลา ได้แก่ การแทนข้อมูลแบบแซค และการแทนข้อมูลแบบคลิป ซึ่งทั้งสองงานวิจัยมีแนวทางเพื่อลดขนาดของข้อมูลเหมือนกับการแทนข้อมูลแบบแฟร็กทัล รวมทั้งเปรียบเทียบกับการค้นหาความคล้ายคลึง ได้แก่ การวัดระยะทางแบบยุคลิด การวัดระยะทางแบบไดนามิกไทม์วอร์ปปิง และวิธีซีดีเอ็ม การนำวิธีเหล่ามาเปรียบเทียบกับงานวิจัยในด้านการลดขนาดข้อมูลจะเป็นความท้าทายในด้านของประสิทธิภาพความแม่นยำ เนื่องจากข้อมูลอนุกรมเวลาในการคำนวณกับการค้นหาความคล้ายคลึงจะใช้ข้อมูลดิบ ซึ่งจะคงคุณลักษณะของข้อมูลอนุกรมเวลาประเภทเดียวกันได้อย่างชัดเจน ซึ่งโดยส่วนมากงานวิจัยเหล่านี้จะเน้นไปที่ผลของความแม่นยำเป็นหลัก

แต่ในส่วนของการลดขนาดข้อมูล เมื่อทำการลดขนาดข้อมูลด้วยวิธีใด ๆ จะทำให้คุณลักษณะของข้อมูลอนุกรมเวลาเปลี่ยนแปลงไปตามแต่ละแนวคิดของงานวิจัย ยกตัวอย่างเช่น การแทนข้อมูลแบบแซค ข้อมูลอนุกรมเวลาจะถูกเก็บค่าด้วยตัวแทนใหม่คือ ตัวอักษร โดยจำนวนของตัวอักษณส่วนมากจะไม่เกิน 10 ตัวอักษร และมีความยาวของข้อมูลอนุกรมเวลาที่น้อยกว่าหรือเท่ากับความยาวเดิม ซึ่งขึ้นกับความเหมาะสมที่ให้ผลลัพธ์ที่มีคุณภาพในแต่ละชุดข้อมูล การแทนข้อมูลแบบคลิป ข้อมูลอนุกรมเวลาจะถูกเปลี่ยนจากเลขจำนวนจริงเป็นตัวเลขเพียงสองค่าคือ 0 หรือ 1 แต่ความยาวของข้อมูลอนุกรมเวลาจะไม่มีการเปลี่ยนแปลง แต่การคำนวณด้วยฟังก์ชันเอ็กออร์จะช่วยลดเวลาในการคำนวณ หรือการแทนข้อมูลแบบแฟร็กทัลเป็นวิธีที่สามารถลดขนาดของข้อมูลอนุกรมเวลาที่ความยาวใด ๆ ให้เหลือเพียงเลขจำนวนจริงสองค่า ซึ่งจะเห็นได้ว่า การลดขนาดของข้อมูลจะมีจุดเด่นในด้านของความเร็วเป็นหลัก ซึ่งเวลาในการคำนวณจะน้อยกว่าการวัดความคล้ายคลึงค่อนข้างมาก

4.3.1.1 การค้นหาความคล้ายคลึง

การค้นหาความคล้ายคลึงของทั้งสามวิธีจะถูกนำมาเปรียบเทียบกับการแทนข้อมูลแบบแฟร็กทัลในส่วนของประสิทธิภาพความเร็วและผลความแม่นยำ ซึ่งคาดหวังว่างานวิจัยนี้น่าจะให้ผลความแม่นยำใกล้เคียงกับวิธีที่ดีที่สุดสำหรับแต่ละชุดข้อมูล และในส่วนของความเร็วในการคำนวณ ถ้าหากจำนวนข้อมูลในชุดข้อมูลมีปริมาณเพิ่มมากขึ้น การแทนข้อมูลแบบแฟร็กทัลน่าจะใช้เวลาน้อยกว่ามากเมื่อเปรียบเทียบกับการค้นหาความคล้ายคลึง

1. *การวัดระยะทางแบบยุคลิด* จะใช้การคำนวณโดยปกติ คือ การวัดระยะทางระหว่างจุดต่อจุดยกกำลังสอง และนำผลรวมของความแตกต่างในแต่ละจุดมารวมกัน ซึ่งจะลดเวลาการคำนวณลงหากไม่มีการถอดรากที่สอง แล้วเปรียบเทียบระยะทางกับข้อมูลอนุกรมเวลาอื่น ๆ สำหรับการทดสอบแบบนำออกหนึ่ง และเปรียบเทียบกับข้อมูลฝึกหัดสำหรับชุดข้อมูลฝึกหัดและข้อมูลทดสอบ ถ้าคู่ใดให้ค่าระยะทางน้อยที่สุดแสดงว่าข้อมูลทดสอบเป็นประเภทเดียวกันกับข้อมูลอนุกรมเวลาที่จับคู่ได้

2. *การวัดระยะทางแบบไดนามิกไทม์วอร์ปปิง* ในส่วนของวิธีนี้ใช้การคำนวณเพื่อหาระยะทางจากจุดหนึ่งเทียบกับทุกจุดแล้วหาระยะทางสั้นที่สุด โดยได้เพิ่มให้ไดนามิกไทม์วอร์ปปิงมีประสิทธิภาพมากยิ่งขึ้นด้วยการใช้เงื่อนไขบังคับโดยรวมแบบซาโก-ชิบะ และฟังก์ชันขอบเขตล่างซึ่งวิธีทั้งสองได้กล่าวไว้ในงานวิจัยที่เกี่ยวข้อง ในส่วนของเงื่อนไขบังคับโดยรวมแบบซาโก-ชิบะจะทำการทดสอบเพื่อหาเปอร์เซ็นความกว้างของแถบให้มีความเหมาะสมสำหรับแต่ละชุดข้อมูล โดยการทดสอบแบบนำออกหนึ่งและการทดลองกับข้อมูลฝึกหัดและข้อมูลทดสอบจะเลือกผลของความแม่นยำที่ดีที่สุดสำหรับการปรับความกว้างของแถบตั้งแต่ 1 จนถึง 10 เปอร์เซ็นต์

3. *ซีดีเอ็ม* เป็นวิธีที่นำข้อมูลอนุกรมเวลามาบีบอัดด้วยแฟ้มข้อมูลแบบซิบ โดยนำข้อมูลอนุกรมเวลาสองตัวมาต่อกันแล้วทำการบีบอัดซึ่งจะได้ขนาดของแฟ้มข้อมูลซึ่งเป็นตัวเลขมาหนึ่งค่า และหารกับขนาดของไฟล์ที่แต่ละตัวถูกบีบอัดและนำขนาดของข้อมูลแต่ละตัวมาบวกกัน ซึ่งถ้าผลลัพธ์ที่ได้เข้าใกล้ 1 แสดงว่าข้อมูลอนุกรมเวลาสองตัวไม่มีความสัมพันธ์แต่ถ้าค่าที่ได้น้อยกว่า 1 แสดงว่าข้อมูลอนุกรมเวลาทั้งสองมีความคล้ายคลึงกันมากขึ้น ในส่วนของแนวทางสำหรับคำนวณของวิธีซีดีเอ็มที่ได้กล่าวไว้ในงานวิจัยจะทำการแปลงข้อมูลอนุกรมเวลาโดยใช้การแทนข้อมูลแบบแซคก่อนที่จะคำนวณกับวิธีซีดีเอ็ม โดยในส่วนของการทดลองกับการทดสอบแบบนำออกหนึ่งจะทำการเลือกผลของความแม่นยำที่ดีที่สุดในแต่ละชุดข้อมูลสำหรับการแทนข้อมูลแบบแซค โดยงานวิจัยนี้ได้นำเสนอด้วยการกำหนดตัวอักษร โดยเพิ่มจำนวนตัวอักษรสำหรับการแทนข้อมูลแบบแซคเริ่มต้นจาก 3 ตัวอักษรจนกระทั่งถึง 10 ตัวอักษร และใช้ความยาวเท่ากับความยาวเดิมของข้อมูลอนุกรมเวลาในแต่ละชุดข้อมูล เนื่องจากการลดขนาดข้อมูลของแซคจะส่งผลต่อความแม่นยำที่ลดต่ำลงเมื่อทำการบีบอัดข้อมูล และการทดลองกับข้อมูลฝึกหัดและข้อมูลทดสอบจะทำในแนวทางเดียวกันกับการทดสอบแบบนำออกหนึ่ง

4.3.1.2 การลดขนาดข้อมูลอนุกรมเวลา

การเปรียบเทียบกับวิธีการลดขนาดข้อมูลอนุกรมเวลาจะถูกนำมาเปรียบเทียบกับทั้งประสิทธิภาพความแม่นยำและความเร็วอีกเช่นกัน โดยคาดว่าการแทนข้อมูลแบบแฟร็กทัลน่าจะให้ผลของความแม่นยำสูงกว่าทั้งสองวิธี รวมถึงประสิทธิภาพในด้านความเร็วการแทนข้อมูลแบบแฟร็กทัลน่าจะใช้เวลาน้อยกว่าทั้งสองวิธี

1. *การแทนข้อมูลแบบแซค* เป็นการลดขนาดข้อมูลโดยเริ่มต้นจากแปลงข้อมูลอนุกรมเวลาโดยใช้วิธีพีเอเอก่อน ซึ่งต้องมีการกำหนดพารามิเตอร์ของความยาวที่เหมาะสมสำหรับแต่ละชุดข้อมูล ต่อมานำผลลัพธ์ที่ได้จากวิธีพีเอเอมาผ่านตารางเกาส์เซียน ซึ่งจะมีการกำหนดจำนวนตัวอักษรที่เหมาะสมสำหรับแต่ละชุดข้อมูล โดยที่งานวิจัยนี้ได้นำเสนอว่าจำนวนตัวอักษรประมาณ 3 ถึง 10 ตัวอักษร ก็เพียงพอสำหรับการนำไปใช้งานได้ ในส่วนของการทดลองกับการทดสอบแบบนำออกหนึ่งจะเลือกความแม่นยำที่ดีที่สุดสำหรับแต่ละชุดข้อมูล โดยจะทำการเลือกพารามิเตอร์ของขนาดความยาวและจำนวนตัวอักษร โดยจำนวนตัวอักษรจะคำนวณตั้งแต่จำนวน 3 ถึง 10 ตัวอักษร ในส่วนของการลดขนาดความยาว จะเลือกทุกความยาวที่เป็นไปได้ ซึ่งจากวิธีของแซคจะคำนวณจากความยาวของข้อมูลอนุกรมเวลาหารด้วยความยาวที่ลดลง แล้วต้องหารกันลงตัว ยกตัวอย่างเช่น ข้อมูลอนุกรมเวลาความยาวเท่ากับ 1,000 จุด ถ้าลดความยาวเท่ากับ 500 จุด จะสามารถทำได้ แต่ถ้าลดความยาวลงเหลือ 400 จุด ไม่สามารถทำได้ เนื่องจากหารไม่ลงตัว โดยจะเลือกขนาดเริ่มต้นที่ลดลงน้อยที่สุดเท่ากับ 10 จุด ซึ่งถ้าความยาวเท่ากับ 1,000 จุด จะมีเลขที่หารลงตัวเท่ากับ 11 จำนวน

2. *การแทนข้อมูลแบบคลิป* สำหรับในส่วนของวิธีนี้ เป็นการแทนข้อมูลจากข้อมูลอนุกรมเวลาที่เป็นเลขจำนวนจริง แล้วทำการเปลี่ยนให้เป็นเลขไบนารีเพียง 0 และ 1 โดยเริ่มจากคำนวณหาค่าเฉลี่ยสำหรับแต่ละข้อมูลอนุกรมเวลา ซึ่งหากจุดใดของข้อมูลอนุกรมเวลามีค่ามากกว่าหรือเท่ากับค่าเฉลี่ยจะมีค่าเท่ากับ 1 นอกจากนี้จะมีค่าเท่ากับ 0 ในส่วนการเปรียบเทียบระหว่างคู่ข้อมูลอนุกรมเวลาจะใช้ฟังก์ชันเอ็กออร์ และนำผลลัพธ์ที่ได้ทั้งหมดมารวมกัน ซึ่งถ้าข้อมูลอนุกรมเวลาคู่ใดคล้ายคลึงกันก็จะให้ผลลัพธ์ที่ได้เข้าใกล้ 0 โดยในการทดสอบด้วยการทดสอบแบบนำออกหนึ่งจะทำการเปรียบเทียบข้อมูลทดสอบที่ดึงออกมากับข้อมูลอนุกรมเวลาอื่น ๆ และการทดลองกับชุดข้อมูลฝึกหัดและข้อมูลทดสอบจะเปรียบเทียบกันระหว่างข้อมูลทดสอบหนึ่งตัวและข้อมูลฝึกหัดทุกตัว ถ้าคู่ใด

ให้ผลลัพธ์มีค่าน้อยที่สุดแสดงว่ามีความคล้ายคลึงกันมากกว่าสำหรับการแทนข้อมูลแบบคลิป

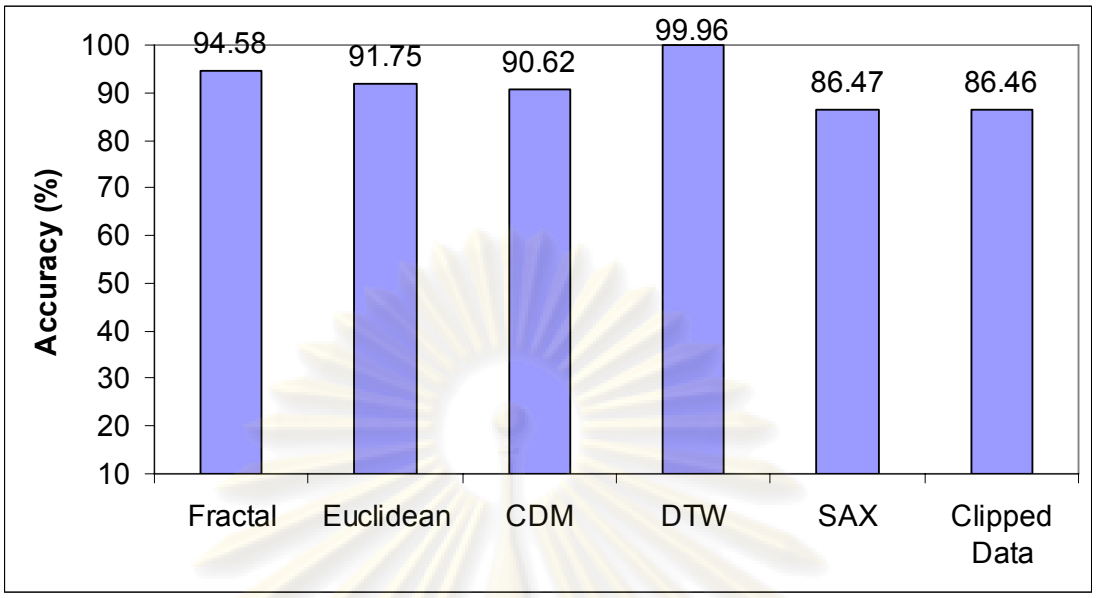
4.3.2 การทดลองเพื่อวิเคราะห์ความแม่นยำและเวลา ด้วยการจำแนกข้อมูลแบบเพื่อนบ้านใกล้ที่สุดอันดับที่หนึ่ง และทดสอบแบบการนำออกหนึ่งของวิธีที่นำเสนอเมื่อเปรียบเทียบกับวิธีอื่น ๆ

สำหรับการทดลองในส่วนนี้จะทำการวิเคราะห์ทั้งประสิทธิภาพในด้านความแม่นยำในการจำแนกข้อมูลและความเร็วในการประมวลด้วยวิธีการจำแนกข้อมูลแบบเพื่อนบ้านใกล้ที่สุดอันดับที่หนึ่ง และทดสอบแบบการนำออกหนึ่ง ซึ่งวิธีการทดสอบแบบนี้จะมีข้อดีคือเป็นวิธีการทดสอบที่ไม่ค่อยจะมีความเอนเอียงกับแต่ละวิธีที่นำมาทดสอบ เนื่องจาก การคำนวณเพื่อวัดประสิทธิภาพกับงานวิจัยสำหรับชุดข้อมูลใด ๆ จะเกิดการเปรียบเทียบกันระหว่างข้อมูลทดสอบที่ดึงออกมาหนึ่งตัวภายในชุดข้อมูลกับข้อมูลที่เหลือทั้งหมดภายในชุดข้อมูล ซึ่งต้องมีการกระทำซ้ำตามจำนวนข้อมูลในแต่ละชุดข้อมูล ทำให้ไม่มีความเอนเอียงในการสร้างชุดข้อมูลสำหรับจำนวนข้อมูลทดสอบหรือจำนวนข้อมูลฝึกหัด และวิธีนี้ยังเป็นการทดสอบที่มีถูกใช้ในงานวิจัยจำนวนมากเพื่อประเมินประสิทธิภาพของการทำเหมืองข้อมูลอีกด้วย

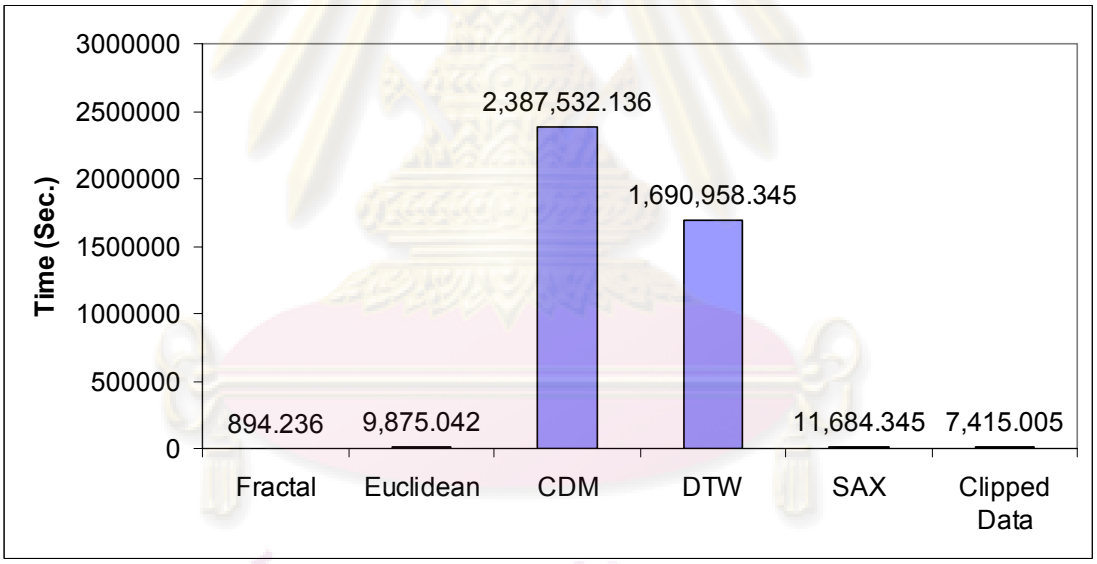
สำหรับผลการทดสอบในแต่ละชุดข้อมูล จะแสดงเป็นสองส่วน คือ ประสิทธิภาพของความแม่นยำ และประสิทธิภาพของความเร็ว โดยจะนำงานวิจัยอื่น ๆ ทั้งหมดมาเปรียบเทียบกับการแทนข้อมูลแบบแฟร็กทัล โดยการทดสอบด้วยวิธีนี้จะทดลองกับชุดข้อมูลที่หนึ่งจนถึงชุดข้อมูลที่หก ซึ่งชุดข้อมูลที่หนึ่งถึงสี่เป็นชุดข้อมูลที่มีความยาวตั้งแต่ 1,000 จุดขึ้นไป ซึ่งเป็นความยาวที่กล่าวไว้ว่าน่าจะเหมาะสมสำหรับการแทนข้อมูลแบบแฟร็กทัล และชุดข้อมูลที่ห้าและหกเป็นชุดข้อมูลที่มีความยาวน้อยกว่า 1,000 จุด ซึ่งจะทำการทดสอบว่าการแทนข้อมูลแบบแฟร็กทัลสามารถให้ผลลัพธ์ได้ดีมากน้อยเพียงใด เมื่อนำมาเปรียบเทียบกับงานวิจัยอื่น ๆ

4.3.2.1 ผลการทดสอบสำหรับชุดข้อมูลที่หนึ่ง ความยาวเท่ากับ 1,000 จุด

ชุดข้อมูลที่หนึ่งจะเน้นไปที่ปริมาณของข้อมูลที่มีจำนวนมาก ชุดข้อมูลนี้จะแสดงถึงประสิทธิภาพของเวลาที่ใช้ในการประมวลผลได้เป็นอย่างดี ซึ่งมีความแตกต่างกับข้อมูลขนาดเล็กที่แต่ละงานวิจัยจะใช้เวลาในการประมวลผลไม่แตกต่างกันมากนัก โดยผลการทดลองของการแทนข้อมูลแบบแฟร็กทัลเมื่อเปรียบเทียบกับวิธีอื่น ๆ เพื่อวัดประสิทธิภาพในด้านความแม่นยำสำหรับชุดข้อมูลที่หนึ่งแสดงดังรูปที่ 4.2 และประสิทธิภาพด้านความเร็วแสดงได้ดังรูปที่ 4.3



รูปที่ 4.2 ผลการทดสอบประสิทธิภาพความแม่นยำสำหรับชุดข้อมูลที่หนึ่ง



รูปที่ 4.3 ผลการทดสอบประสิทธิภาพความเร็วสำหรับชุดข้อมูลที่หนึ่ง

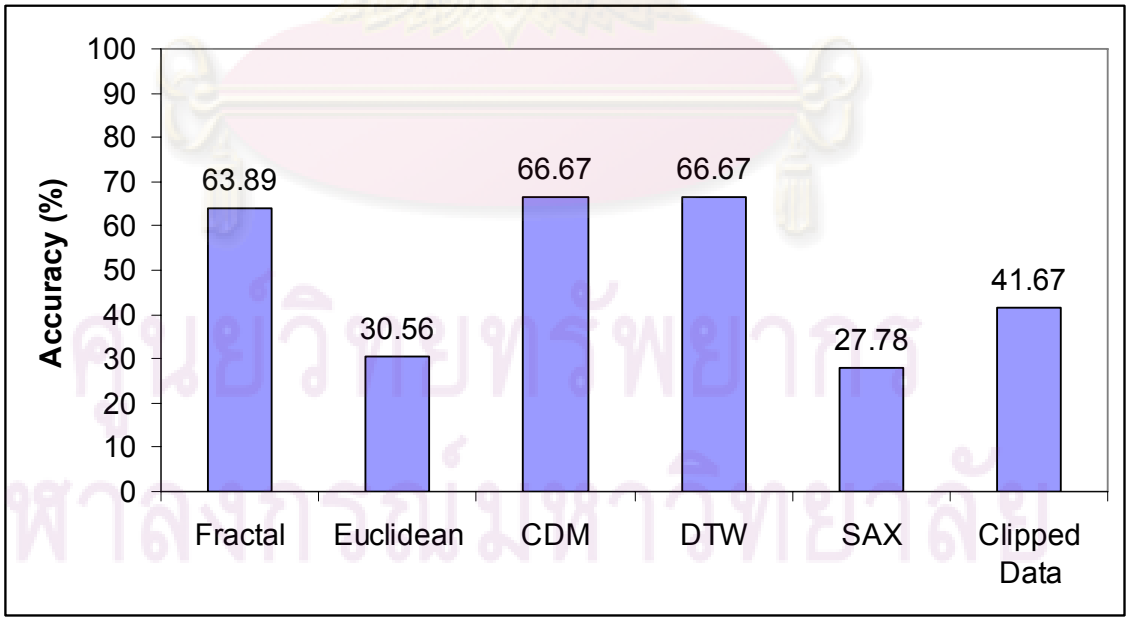
จากผลการทดลองในชุดข้อมูลที่หนึ่งสำหรับประสิทธิภาพในด้านของผลความแม่นยำกับการแทนข้อมูลแบบแฟร็กทัลสามารถเอาชนะการลดขนาดข้อมูลทั้งการแทนข้อมูลแบบแซคและการแทนข้อมูลแบบคลิปรวม และยังให้ผลความแม่นยำมากกว่าทั้งการวัดระยะทางแบบยุคลิด และซีดีเอ็ม แต่ได้รับผลความแม่นยำน้อยกว่าไม่มากสำหรับไดนามิกไทม์วอร์ปปิง ซึ่งเป็นวิธีที่นิยมนำมาพัฒนาในงานเหมืองข้อมูลอนุกรมเวลาที่สุด รวมถึงประสิทธิภาพโดยรวมที่สูงกว่าวิธีอื่น ๆ ในปัจจุบัน แต่วิธีนี้ได้ใช้การคำนวณกับข้อมูลดิบ ซึ่งแตกต่างกับงานวิจัยนี้ใน

การวัดความคล้ายคลึงของข้อมูลอนุกรมเวลา เนื่องจาก การลดขนาดข้อมูลต้องเกิดการสูญเสียคุณลักษณะที่สำคัญของข้อมูลอนุกรมเวลา เพื่อลดขนาดข้อมูลให้เหลือเพียงเลขจำนวนจริงสองค่า ทำให้ผลความแม่นยำโดยส่วนมากกับการเปรียบเทียบวิธีที่ใช้ข้อมูลดิบ และข้อมูลที่มีการลดขนาดแล้ว การนำข้อมูลดิบมาใช้จะส่งให้ผลความแม่นยำที่สูงกว่า

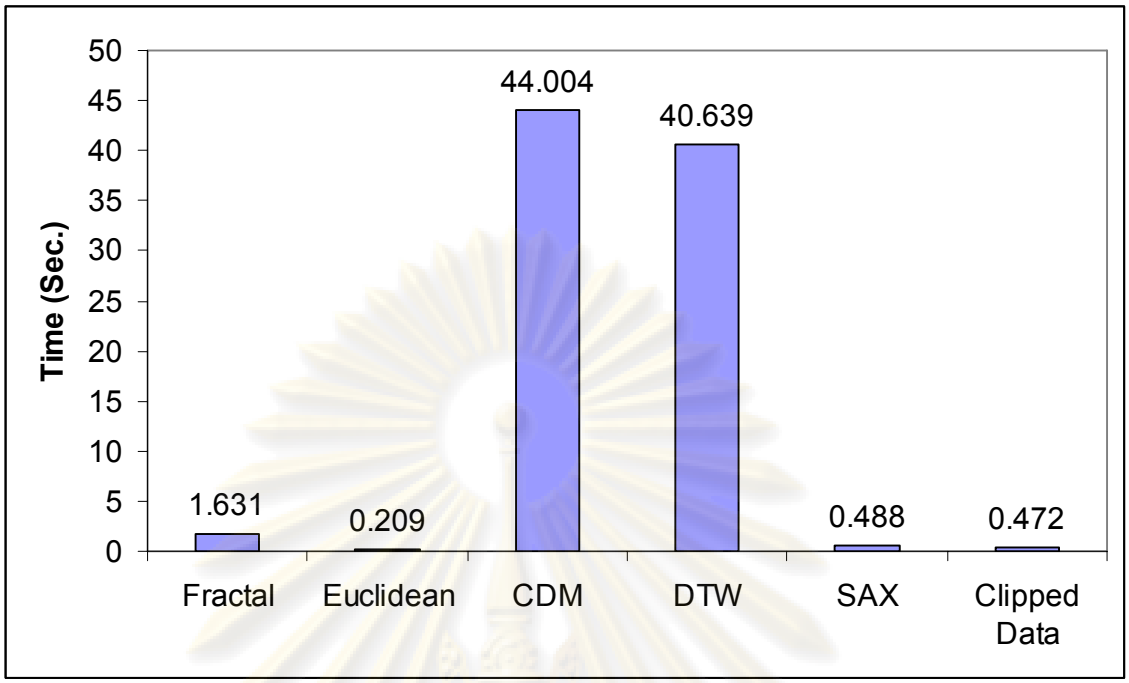
ในส่วนของเวลาที่ใช้ในการคำนวณ ซึ่งงานวิจัยนี้ใช้เวลาในการคำนวณน้อยกว่างานวิจัยอื่น ๆ ค่อนข้างมาก รวมทั้งเวลาที่ใช้ในการคำนวณยังน้อยกว่าไดนามิกไทม์วอร์ปปิงประมาณ 2,000 เท่า สำหรับชุดข้อมูลนี้ ซึ่งจะเห็นได้ว่า ผลความแม่นยำที่ได้ยังคงใกล้เคียงกับไดนามิกไทม์วอร์ปปิงมากกว่าการลดขนาดด้วยวิธีอื่น ๆ และยังใช้เวลาน้อยกว่าทุกวิธีที่นำมาเปรียบเทียบสำหรับงานวิจัยนี้

4.3.2.2 ผลการทดสอบสำหรับชุดข้อมูลที่สอง ความยาวเท่ากับ 1,000 จุด

สำหรับชุดข้อมูลที่สองเป็นชุดข้อมูลที่มีประเภทต่าง ๆ เป็นจำนวนมากจำนวน 18 ประเภท และแต่ละประเภทจะมีข้อมูลชนิดเดียวกันเป็นเพียงหนึ่งตัวเท่านั้น ทำให้ชุดข้อมูลนี้จะมีความยากในการค้นหาข้อมูลประเภทเดียวกันมากกว่าชุดข้อมูลอื่น ๆ ในด้านของความแม่นยำ รวมทั้งชุดข้อมูลที่สองเป็นข้อมูลดิบที่ได้มาจากหน่วยเก็บถาวรโดยตรง ซึ่งไม่ได้เกิดจากการสร้างด้วยการตัดหรือต่อกันเพื่อให้ได้ความยาวของข้อมูลตามต้องการเหมือนชุดข้อมูลอื่น ๆ โดยผลการทดลองสำหรับชุดข้อมูลที่สอง แสดงประสิทธิภาพในด้านความแม่นยำและความเร็วดังแสดงในรูปที่ 4.4 และ 4.5 ตามลำดับ



รูปที่ 4.4 ผลการทดสอบประสิทธิภาพความแม่นยำสำหรับชุดข้อมูลที่สอง

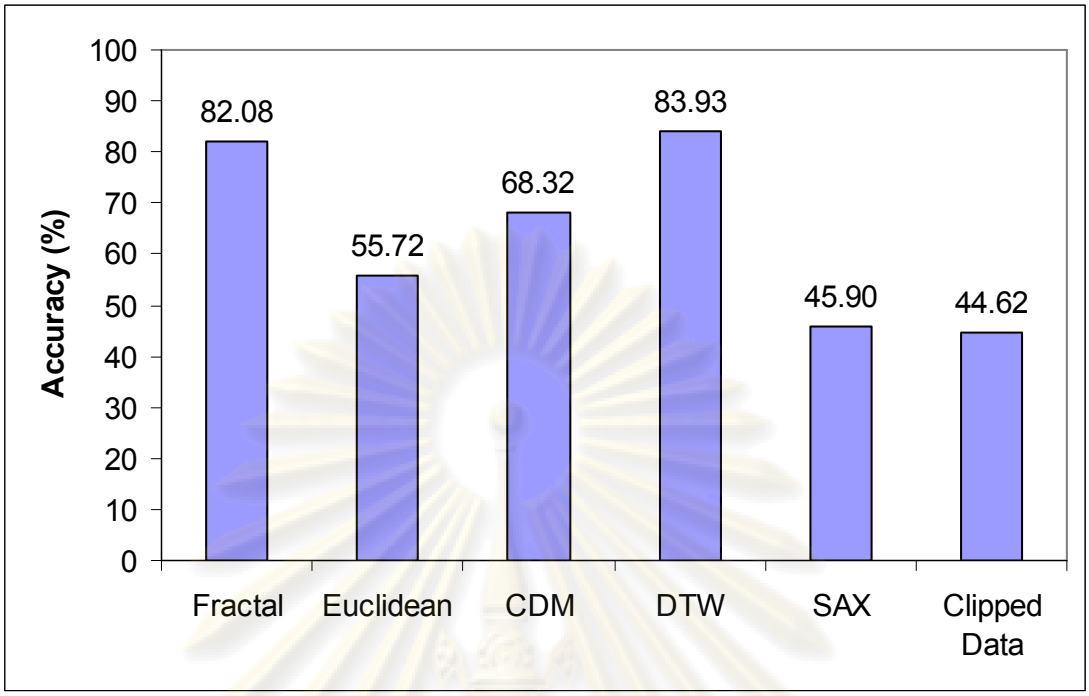


รูปที่ 4.5 ผลการทดสอบประสิทธิภาพความเร็วสำหรับชุดข้อมูลที่สอง

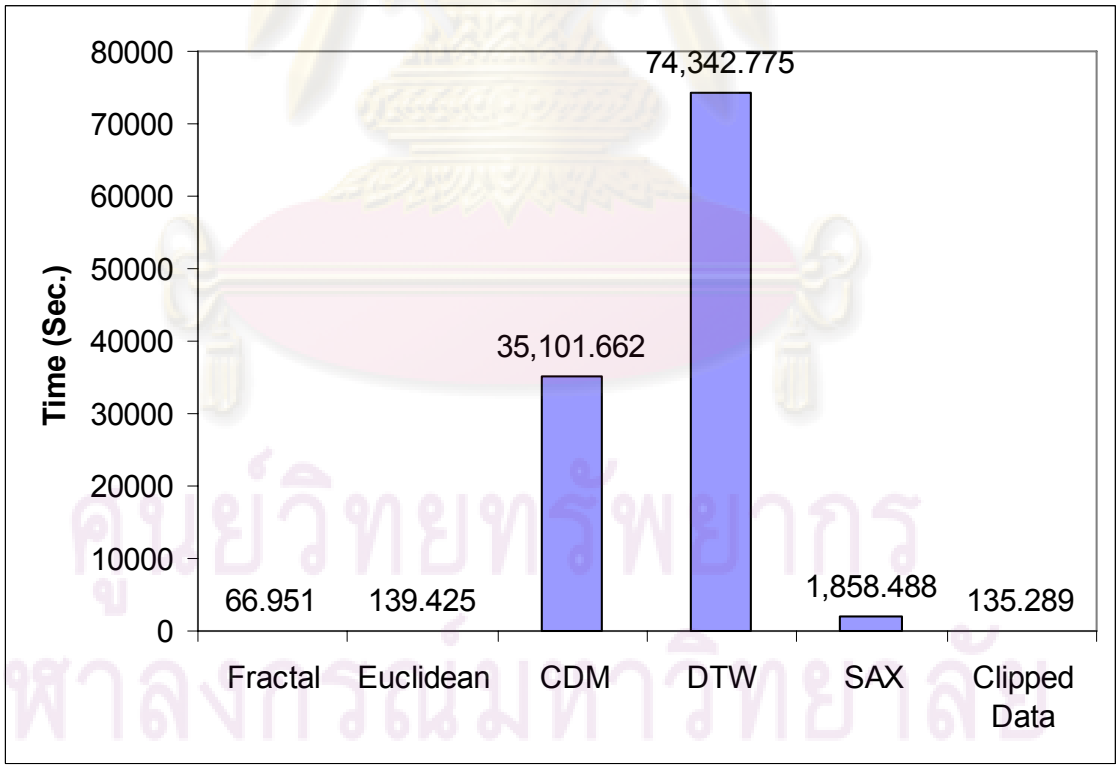
สำหรับผลการทดลองในชุดข้อมูลที่สองจะทำการทดสอบกับชุดข้อมูลที่ในแต่ละประเภทจะมีจำนวนไม่มาก ทำให้เวลาที่ใช้สำหรับการแทนข้อมูลแบบแฟร็กทัลใช้เวลามากกว่าการวัดระยะทางแบบยุคลิด และการลดขนาดข้อมูลทั้งสอง แต่ในทางปฏิบัติเวลาที่ใช้ในการคำนวณในงานวิจัยนี้ใช้เพียง 1.631 วินาที ซึ่งจะไม่มีความแตกต่างกันมากนักสำหรับเวลาที่น้อยกว่าเมื่อเปรียบเทียบกับทั้งสามวิธี สำหรับส่วนของผลความแม่นยำ ในงานวิจัยนี้สามารถเอาชนะทั้งสามวิธีที่กล่าวมาค่อนข้างมากซึ่งให้ผลที่ดีกว่าการวัดระยะทางแบบยุคลิดและการแทนข้อมูลแบบแซคถึง 30 เปอร์เซ็นต์ รวมทั้งให้ผลความแม่นยำที่ต่างจากไดนามิกไทม์วอร์ปปิงและซีดีเอ็มเพียง 3 เปอร์เซ็นต์เท่านั้น ซึ่งวิธีทั้งสองใช้ข้อมูลดิบในการคำนวณ และแต่เวลาที่ใช้ในการคำนวณแตกต่างกันมากถึง 25 เท่า

4.3.2.3 ผลการทดสอบสำหรับชุดข้อมูลที่สาม ความยาวเท่ากับ 2,000 จุด

สำหรับชุดข้อมูลที่สามเป็นชุดข้อมูลที่มีประเภทต่าง ๆ เป็นจำนวนมากอีกเหมือนกับชุดข้อมูลที่สอง ซึ่งมีจำนวนประเภทเท่ากับ 14 ประเภท และมีจำนวนข้อมูลเท่ากับ 865 อนุกรม โดยในแต่ละประเภทจะมีจำนวนข้อมูลไม่เท่ากัน ซึ่งจะพบได้ทั่วไปในทางปฏิบัติสำหรับงานวิจัยในด้านการทำเหมืองข้อมูล สำหรับรูปที่ 4.6 และ 4.7 แสดงประสิทธิภาพในด้านความแม่นยำและความเร็ว ตามลำดับ



รูปที่ 4.6 ผลการทดสอบประสิทธิภาพความแม่นยำสำหรับชุดข้อมูลที่สาม

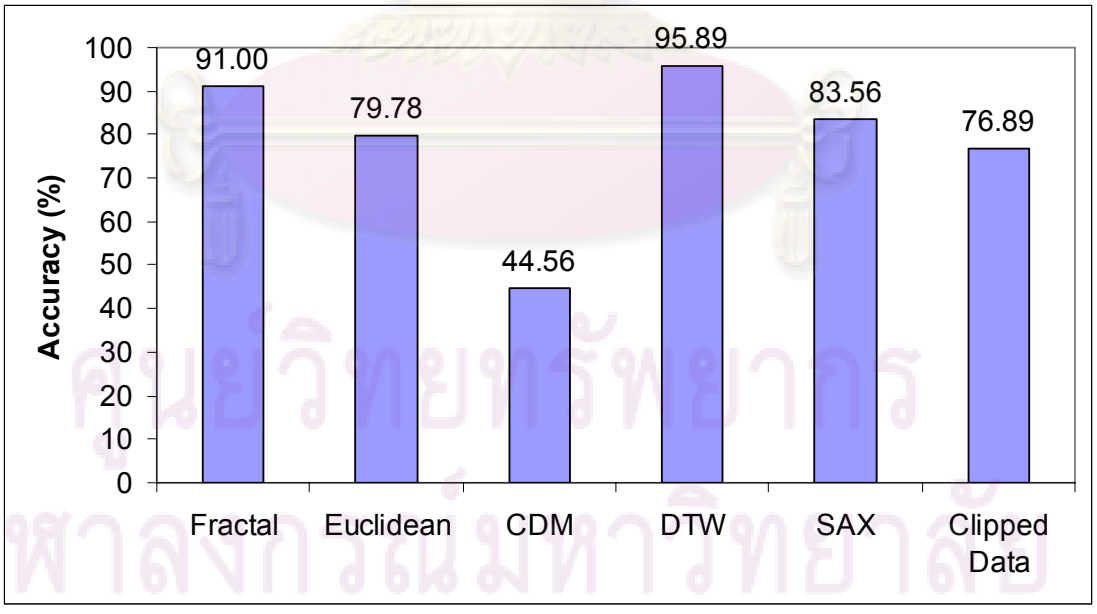


รูปที่ 4.7 ผลการทดสอบประสิทธิภาพความเร็วสำหรับชุดข้อมูลที่สาม

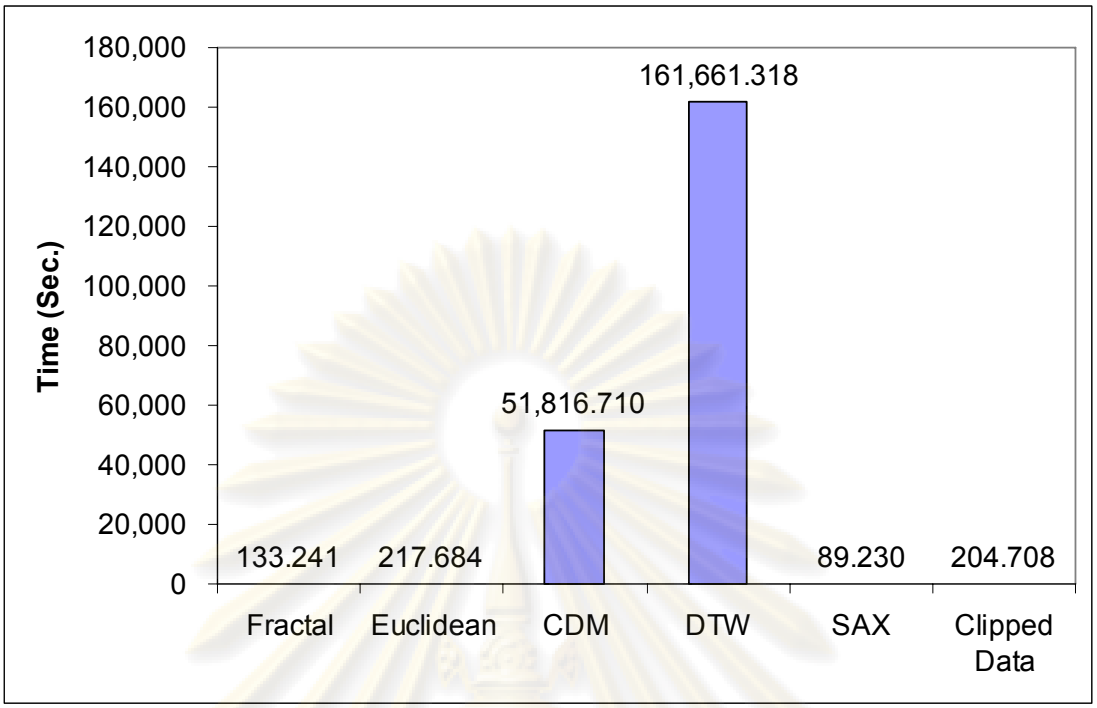
สำหรับการทดลองในชุดข้อมูลที่สาม การแทนข้อมูลแบบแฟร็กทัลให้ผลความแม่นยำที่มากกว่าในหลายวิธี โดยชุดข้อมูลนี้จะมีประเภทของข้อมูลจำนวนมากเหมือนกับชุดข้อมูลที่สอง ซึ่งถ้าหากจำนวนของประเภทข้อมูลมีมากขึ้นจะส่งผลต่อความแม่นยำกับทุกวิธีค่อนข้างมาก ซึ่งอาจจะทำให้มีโอกาสจับกลุ่มผิดได้ง่ายขึ้น แต่การแทนข้อมูลแบบแฟร็กทัลก็ยังคงให้ผลความแม่นยำใกล้เคียงกับวิธีที่ใช้ข้อมูลดิบในการคำนวณเช่นเดิม รวมถึงใช้เวลาในการคำนวณน้อยที่สุดอีกเช่นกัน ในส่วนของการแทนข้อมูลแบบแซคจะใช้เวลาในการคำนวณค่อนข้างมาก ซึ่งมากกว่างานวิจัยนี้ถึง 30 เท่า เนื่องจาก การลดขนาดข้อมูลของแซคทำได้ไม่ดีนัก ซึ่งสามารถลดขนาดข้อมูลได้เท่ากับ 400 จุด ซึ่งเป็นชุดข้อมูลที่ไม่สามารถลดขนาดได้มากนักเมื่อเปรียบเทียบกับชุดข้อมูลอื่นที่ลดขนาดข้อมูลให้เหลือประมาณ 25 ถึง 200 จุด ทำให้เวลาที่ใช้ในการคำนวณค่อนข้างมาก

4.3.2.4 ผลการทดสอบสำหรับชุดข้อมูลที่สี่ ความยาวเท่ากับ 3,000 จุด

ในชุดข้อมูลที่สี่จะเพิ่มความยาวของข้อมูลอนุกรมให้มากขึ้น เท่ากับ 3,000 จุด มีจำนวนประเภทเท่ากับ 9 ประเภท และจำนวนข้อมูลเท่ากับ 900 ข้อมูล ซึ่งแต่ละประเภทจะมีจำนวนข้อมูลเท่ากันคือ 90 ข้อมูล ซึ่งชุดข้อมูลนี้สร้างขึ้นมาเพื่อให้มีความเอนเอียงน้อยที่สุด โดยที่แต่ละประเภทจะมีจำนวนข้อมูลเท่ากันหมด ทำให้การเปรียบเทียบกับข้อมูลภายในจะอ้างอิงไปกับข้อมูลบางประเภทมากเกินไป ผลการทดลองในด้านความแม่นยำและความเร็วแสดงได้ดังรูปที่ 4.8 และ 4.9 ตามลำดับ



รูปที่ 4.8 ผลการทดสอบประสิทธิภาพความแม่นยำสำหรับชุดข้อมูลที่สี่



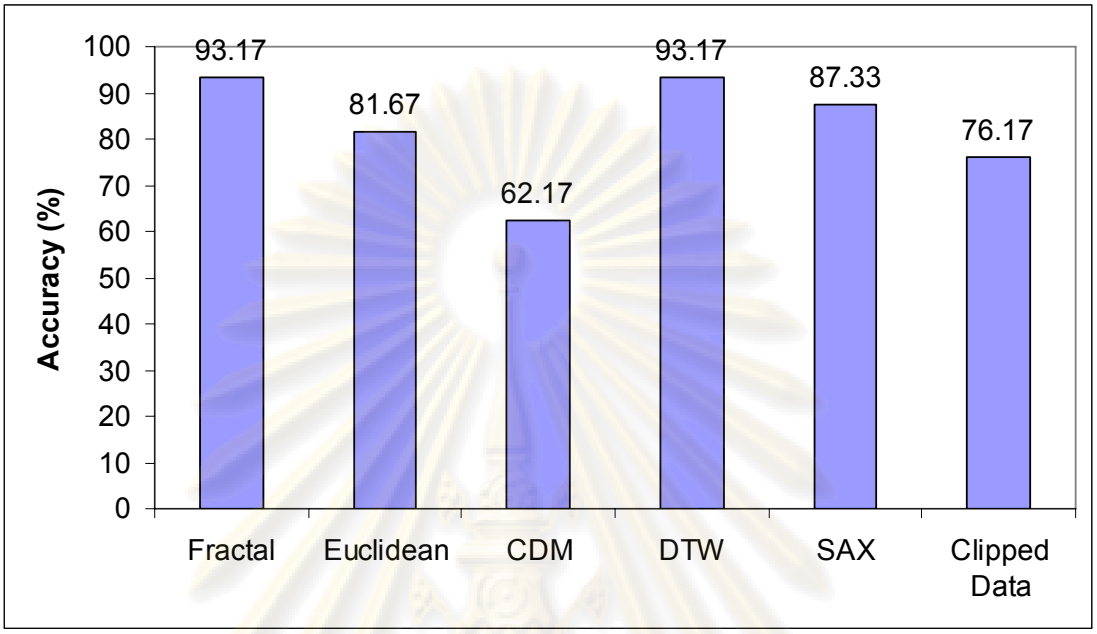
รูปที่ 4.9 ผลการทดสอบประสิทธิภาพความเร็วสำหรับชุดข้อมูลที่สี่

สำหรับผลการทดลองในชุดข้อมูลที่สี่ เป็นชุดข้อมูลที่มีขนาดยาวที่สุดสำหรับงานวิจัยนี้ ซึ่งการแทนข้อมูลแบบแฟร็กทัลยังคงให้ผลลัพธ์ใกล้เคียงกับไดนามิกไทม์วอร์ปปิงมากกว่าวิธีอื่น ๆ อีกเช่นกัน โดยเวลาที่ใช้จะมากกว่าการแทนข้อมูลแบบแซค เนื่องจาก วิธีแซคสามารถลดขนาดข้อมูลได้มากกว่าชุดข้อมูลที่สามซึ่งเหลือความยาวเท่ากับ 40 จุด ทำให้ประสิทธิภาพของความแม่นยำสูงที่สุด ทำให้เวลาในการคำนวณค่อนข้างต่ำ สำหรับเวลาในการคำนวณสำหรับชุดข้อมูลนี้จะใช้ค่อนข้างสูง ซึ่งอาจเป็นสาเหตุมาจากฟังก์ชันขอบเขตล่างทำงานได้ไม่ดีนักสำหรับชุดข้อมูลนี้ทำให้มีการคำนวณไดนามิกไทม์วอร์ปปิง ในหลาย ๆ ชุดข้อมูล รวมทั้งค่าของเงื่อนไขบังคับโดยรวมเท่ากับ 10 เปอร์เซ็นต์ ซึ่งจากการทดลองโดยส่วนมากสำหรับข้อมูลอนุกรมเวลาขนาดใหญ่จะอยู่ระหว่าง 1 ถึง 5 เปอร์เซ็นต์เท่านั้น และเป็นอีกสาเหตุหนึ่งที่จะส่งผลต่อเวลา

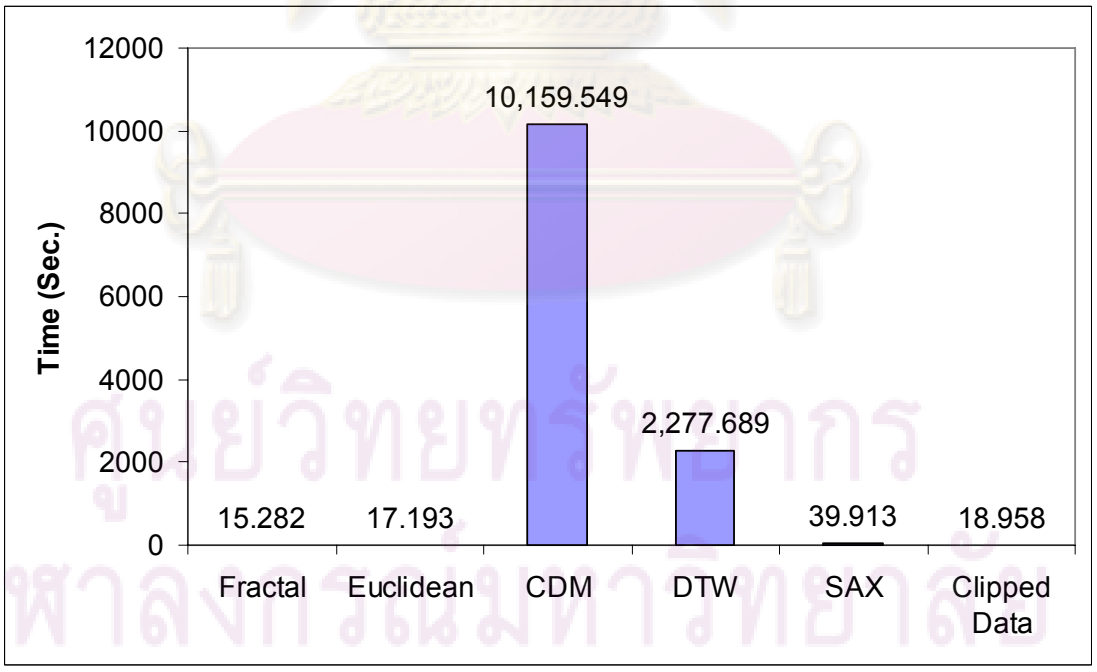
4.3.2.5 ผลการทดสอบสำหรับชุดข้อมูลที่ห้า ความยาวเท่ากับ 500 จุด

สำหรับชุดข้อมูลที่ห้าจะเลือกข้อมูลอนุกรมเวลาที่มีขนาดสั้นเท่ากับ 500 จุด ซึ่งอยู่นอกเหนือจากวัตถุประสงค์ของงานวิจัยที่ได้ตั้งไว้ ซึ่งชุดข้อมูลนี้จะใช้วัดประสิทธภาพของการแทนข้อมูลแบบแฟร็กทัลกับข้อมูลอนุกรมเวลาขนาดสั้น ถึงแม้ว่าวิธีที่เหมาะสมกับข้อมูลขนาดสั้นมากที่สุดคือ ไดนามิกไทม์วอร์ปปิง แต่ก็เลือกชุดข้อมูลนี้เพื่อทดสอบว่างานวิจัยนี้จะมีประสิทธิภาพมากใด เมื่อเปรียบเทียบกับวิธีอื่น ๆ โดยมีจำนวนข้อมูลเท่ากับ 600 อนุกรม และ

จำนวนประเภทเท่ากับ 4 ประเภท สำหรับการทดลองเพื่อวัดประสิทธิภาพของความแม่นยำและเวลาแสดงได้ดังรูปที่ 4.10 และ 4.11 ตามลำดับ



รูปที่ 4.10 ผลการทดสอบประสิทธิภาพความแม่นยำสำหรับชุดข้อมูลที่ห้า

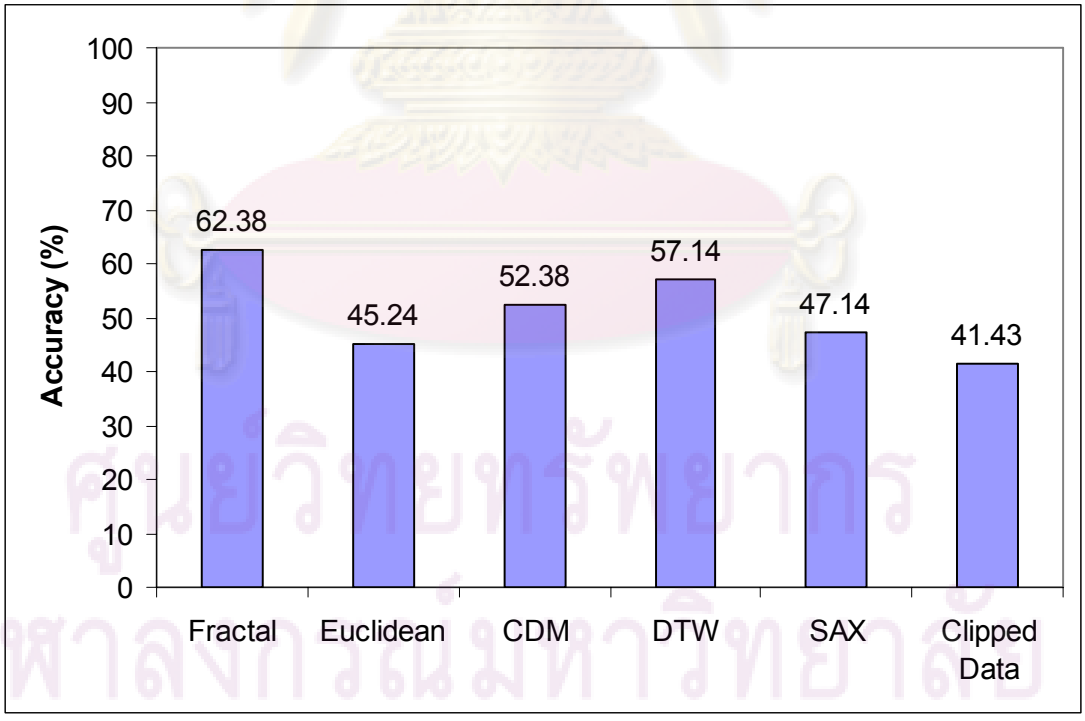


รูปที่ 4.11 ผลการทดสอบประสิทธิภาพความเร็วสำหรับชุดข้อมูลที่ห้า

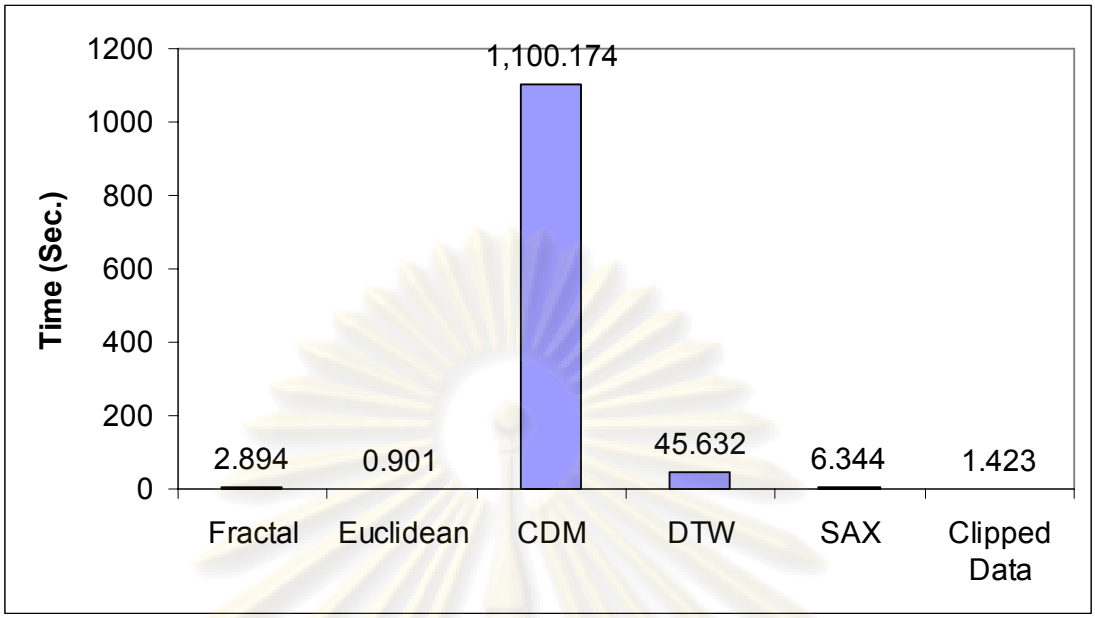
สำหรับผลการทดลองในชุดข้อมูลที่ห้าเป็นชุดข้อมูลที่มีความยาวของข้อมูลอนุกรมเวลาขนาดสั้น ซึ่งความยาวน้อยกว่า 1,000 จุด จะอยู่นอกขอบเขตสำหรับงานวิจัยนี้ โดยผลความแม่นยำที่คำนวณได้จะเท่ากับไดนามิกไทม์วอร์ปปิงและมากกว่าวิธีอื่น ๆ รวมทั้งใช้เวลาน้อยกว่าประมาณสิบเท่าสำหรับไดนามิกไทม์วอร์ปปิง ซึ่งจะเห็นได้ว่า การแทนข้อมูลแบบแฟร็กทัลก็ยังให้ผลความแม่นยำที่มีประสิทธิภาพ ถึงแม้ว่าจะทดลองกับข้อมูลอนุกรมเวลาขนาดสั้นก็ตาม ในส่วนของซีดีเอ็มจะใช้เวลาในการคำนวณจะค่อนข้างสูง เนื่องจาก การคำนวณจะต้องมีการติดต่อกับอินพุต/เอาต์พุต ซึ่งต้องเสียไปกับการอ่านไฟล์หรือการบีบอัดไฟล์ โดยจะส่งผลให้เวลาที่ใช้ในการคำนวณเพิ่มขึ้นค่อนข้างมาก ถึงแม้ว่าจะเป็นข้อมูลอนุกรมเวลาขนาดสั้นแล้วก็ตาม

4.3.2.6 ผลการทดสอบสำหรับชุดข้อมูลที่หก ความยาวเท่ากับ 200 จุด

สำหรับชุดข้อมูลที่หกจะมีขนาดของข้อมูลที่สั้นเท่ากับ 200 จุด ซึ่งจะเป็นความท้าทายในการนำมาทดสอบกับการแทนข้อมูลแบบแฟร็กทัล เพื่อแสดงขอบเขตการทำงานของงานวิจัยนี้กับข้อมูลอนุกรมเวลาขนาดสั้น โดยชุดข้อมูลที่หกมีจำนวนข้อมูลเท่ากับ 210 อนุกรม และมีจำนวนประเภทเท่ากับ 6 ประเภท สำหรับการทดลองเพื่อวัดประสิทธิภาพของความแม่นยำและเวลาแสดงได้ดังรูปที่ 4.12 และ 4.13 ตามลำดับ



รูปที่ 4.12 ผลการทดสอบประสิทธิภาพความแม่นยำสำหรับชุดข้อมูลที่หก



รูปที่ 4.13 ผลการทดสอบประสิทธิภาพความเร็วสำหรับชุดข้อมูลที่หก

สำหรับผลการทดลองในชุดข้อมูลที่หกเป็นชุดข้อมูลอนุกรมเวลาขนาดสั้นที่มีจำนวนข้อมูลไม่มากนักทำให้เวลาที่ใช้ในการคำนวณสำหรับการแทนข้อมูลแบบแฟร็กทัลมากกว่าการวัดระยะทางแบบยุคลิดและการแทนข้อมูลแบบคลิป ในส่วนของการแทนข้อมูลแบบแซคสามารถลดขนาดข้อมูลจากความยาว 200 จุด เหลือ 50 จุด ซึ่งจะลดขนาดข้อมูลได้ไม่มากเท่าไหร่ทำให้ส่งผลกับเวลาในการคำนวณ โดยชุดข้อมูลนี้แสดงให้เห็นว่าการแทนข้อมูลแบบแฟร็กทัลสามารถให้ผลความแม่นยำที่สูงกว่าไดนามิกไทม์วอร์ปปิง จะเห็นได้ว่า การแทนข้อมูลแบบแฟร็กทัลยังสามารถนำมาใช้กับข้อมูลอนุกรมเวลาขนาดสั้นได้มีประสิทธิภาพอีกเช่นกัน

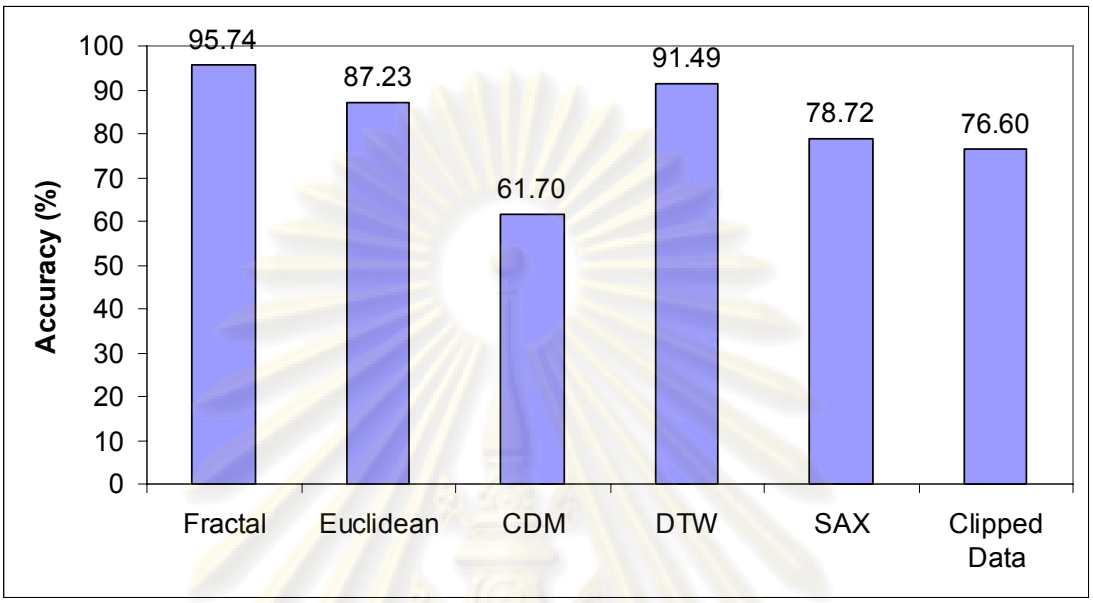
4.3.3 การทดลองเพื่อวิเคราะห์ความแม่นยำและความเร็วจากชุดข้อมูลฝึกหัดและข้อมูลทดสอบของวิธีที่นำเสนอเมื่อเปรียบเทียบกับวิธีอื่น ๆ

การทดลองกับชุดข้อมูลที่แบ่งข้อมูลออกเป็นข้อมูลฝึกหัดและข้อมูลทดสอบรวมทั้งมีข้อมูลฝึกหัดปริมาณมากกว่าข้อมูลทดสอบ จะพบได้ทั่วไปในทางปฏิบัติสำหรับงานเหมืองข้อมูล ในหัวข้อนี้จะทดสอบกับชุดข้อมูลทั้งหมด 4 ชุดข้อมูล ดังนี้

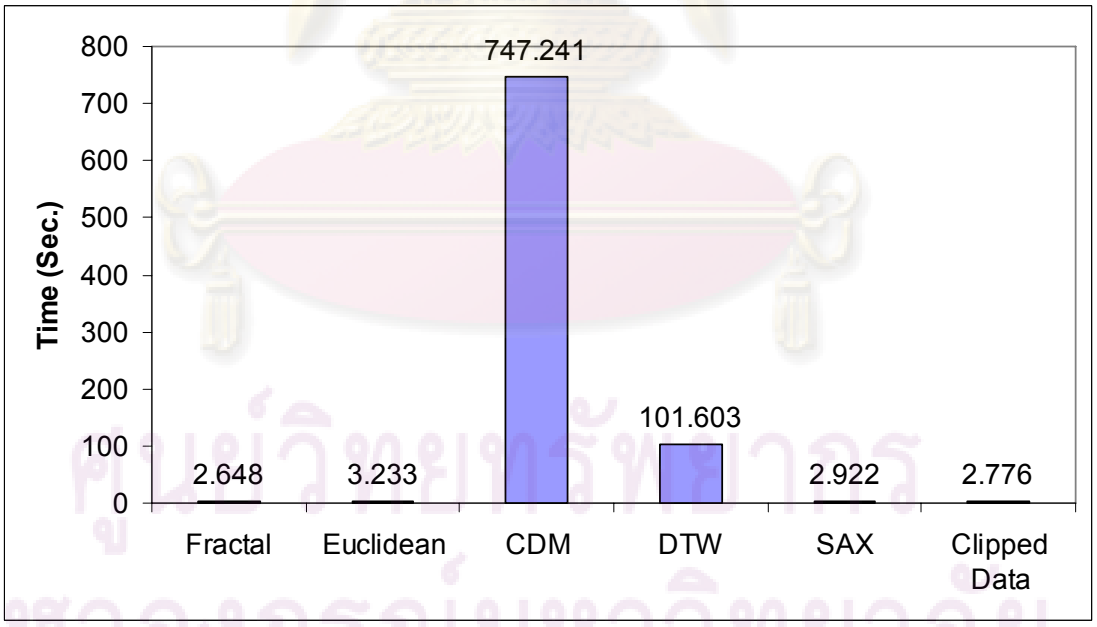
4.3.3.1 ผลการทดสอบสำหรับชุดข้อมูลที่เจ็ด ความยาวเท่ากับ 1,000 จุด

สำหรับชุดข้อมูลที่เจ็ดจะพิจารณาถึงข้อมูลทดสอบที่มีจำนวนในแต่ละประเภทไม่เท่ากัน โดยในประเภทต่าง ๆ จะมีจำนวนข้อมูลไม่มากนัก และในส่วนของข้อมูลฝึกหัดจะมีทั้งหมด 463 อนุกรม โดยมีจำนวนข้อมูลในแต่ละประเภทไม่เท่ากันเช่นกัน ซึ่งข้อมูลในลักษณะนี้พบได้ทั่วไปเช่นกันในงานทำเหมืองข้อมูล สำหรับจำนวนประเภทมีจำนวนเท่ากับ 8 ประเภท

ในส่วนของผลการทดลองสำหรับประสิทธิภาพในด้านความแม่นยำและเวลาของชุดข้อมูลที่เจ็ด แสดงได้ดังรูปที่ 4.14 และ 4.15 ตามลำดับ



รูปที่ 4.14 ผลการทดสอบประสิทธิภาพความแม่นยำสำหรับชุดข้อมูลที่เจ็ด



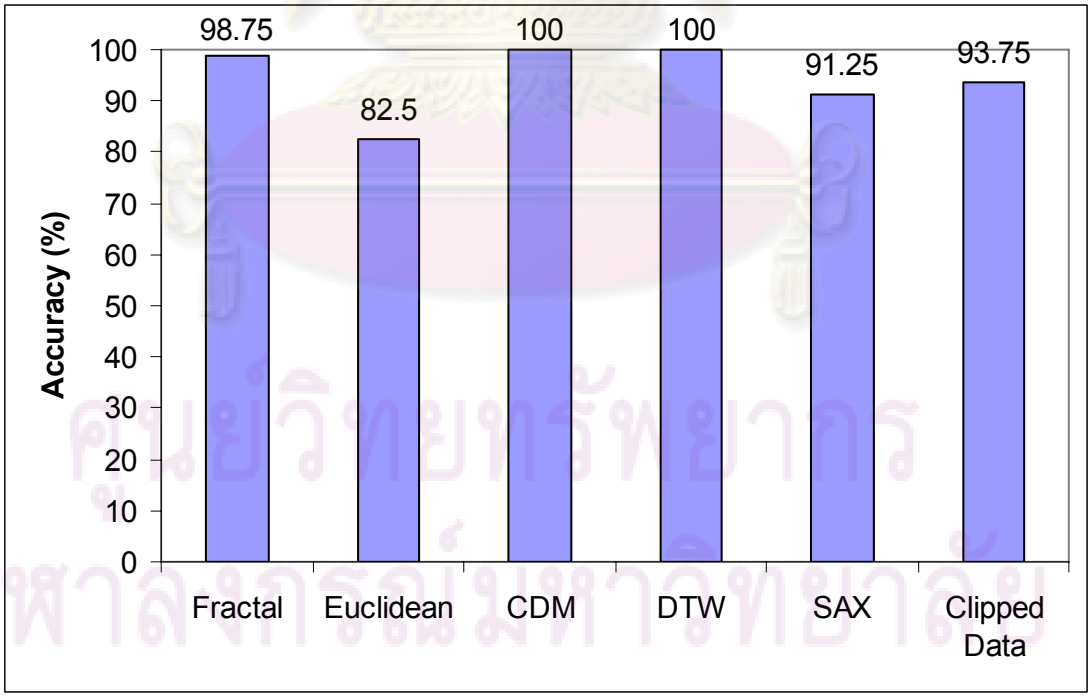
รูปที่ 4.15 ผลการทดสอบประสิทธิภาพความเร็วสำหรับชุดข้อมูลที่เจ็ด

สำหรับผลการทดลองของชุดข้อมูลที่หกเป็นชุดข้อมูลที่การแทนข้อมูลแบบแฟร็กทัลสามารถเอาชนะทั้งในด้านผลความแม่นยำและเวลาได้กับทุกวิธี ถึงแม้ว่าการแทน

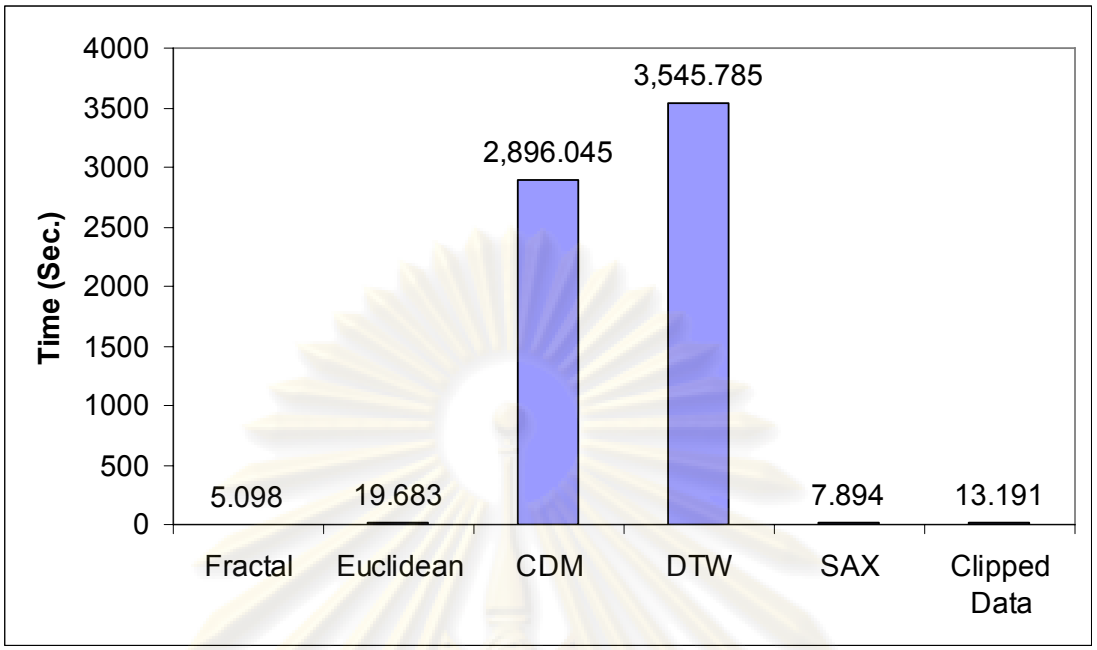
ข้อมูลแบบแซคจะสามารถลดขนาดข้อมูลในชุดข้อมูลนี้จากความยาว 1,000 จุด เหลือเพียง 25 จุด ก็ตาม แต่ชุดข้อมูลประเภทนี้จะทำให้การแทนข้อมูลแบบแฟร็กทัลเสียเวลาในการลดขนาดข้อมูลเพียงข้อมูลฝึกหัด ทำให้เวลาในการคำนวณน้อยกว่าวิธีอื่น ๆ ในส่วนของเวลาการคำนวณจากไดนามิกไทม์วอร์ปปิงที่ใช้เวลาค่อนข้างต่ำ เมื่อเปรียบเทียบกับซีดีเอ็ม เนื่องจาก ในการคำนวณของชุดข้อมูลนี้ใช้ค่าเงื่อนไขบังคับโดยรวมเท่ากับ 1 เปอร์เซ็นต์ ซึ่งเป็นค่าที่น้อยที่สุดสำหรับทดลองทั้งหมด ทำให้การคำนวณในแต่ละคู่ของข้อมูลอนุกรมเวลาจะถูกวอร์ปเพียงแค่ 10 จุด เพื่อคำนวณหาระยะทาง และยังมีฟังก์ชันขอบเขตล่างช่วยทำให้ประสิทธิภาพในด้านของความเร็วเพิ่มมากขึ้นอีกด้วย

4.3.3.2 ผลการทดสอบสำหรับชุดข้อมูลที่แปด ความยาวเท่ากับ 2,000 จุด

สำหรับชุดข้อมูลที่แปดเป็นชุดข้อมูลที่มีความยาวเพิ่มขึ้นเป็น 2,000 จุด โดยชุดข้อมูลนี้จะมีจำนวนข้อมูลฝึกหัดเท่ากับ 800 อนุกรม ข้อมูลทดสอบเท่ากับ 80 อนุกรม และมีจำนวนประเภทเท่ากับ 4 ประเภท โดยจำนวนในแต่ละประเภททั้งข้อมูลฝึกหัดและข้อมูลทดสอบจะมีค่าเท่ากัน โดยมีจำนวนเท่ากับ 200 อนุกรม และ 20 อนุกรม ตามลำดับ รวมทั้งชุดข้อมูลนี้ยังเลือกประเภทของข้อมูลเฉพาะคลื่นหัวใจจากผู้ป่วยแต่ละโรค ซึ่งแตกต่างจากชุดข้อมูลอื่น ๆ ที่มีการคละกันระหว่างประเภทข้อมูลที่ไม่เหมือนกัน สำหรับประสิทธิภาพในด้านความแม่นยำและเวลาแสดงได้ดังรูปที่ 4.16 และ 4.17 ตามลำดับ



รูปที่ 4.16 ผลการทดสอบประสิทธิภาพความแม่นยำสำหรับชุดข้อมูลที่แปด



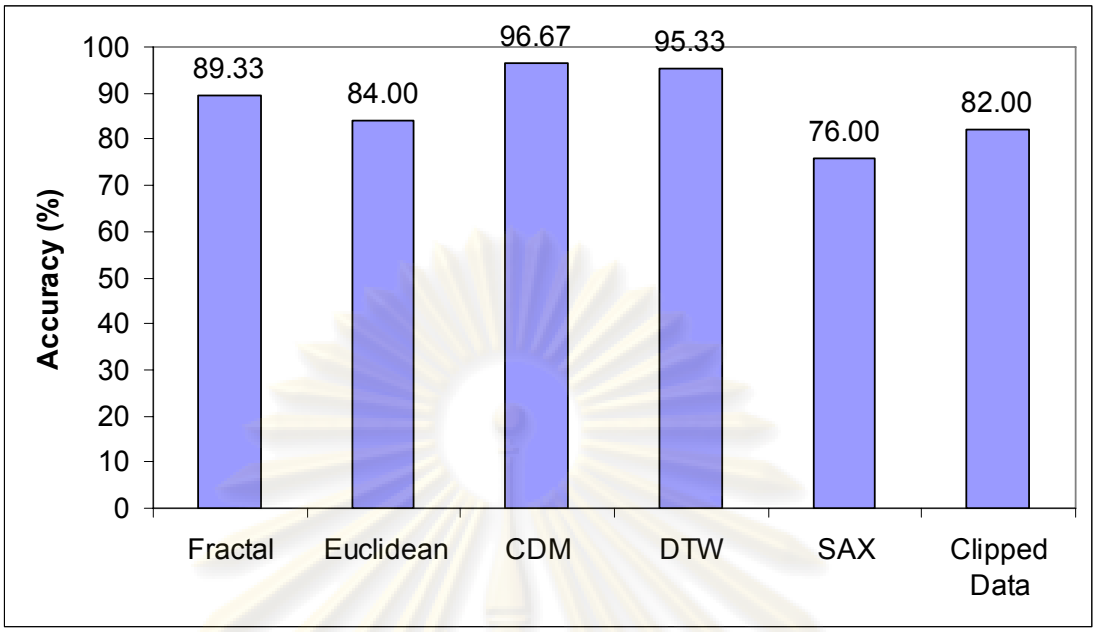
รูปที่ 4.17 ผลการทดสอบประสิทธิภาพความเร็วสำหรับชุดข้อมูลที่แปด

สำหรับผลการทดลองในชุดข้อมูลที่แปด สำหรับผลความแม่นยำยังคงให้ผลที่ใกล้เคียงกับทั้งไดนามิกไทม์วอร์ปปิงและซีดีเอ็ม และมียังให้ประสิทธิภาพของความแม่นยำที่สูงกว่าการลดขนาดข้อมูลอื่น ๆ รวมถึงเวลาที่ใช้ในการคำนวณยังน้อยที่สุดอีกเช่นกัน ซึ่งเวลาที่ใช้ในการคำนวณสามารถเอาชนะไดนามิกไทม์วอร์ปปิงและซีดีเอ็มได้มากถึงประมาณ 700 เท่า และ 550 เท่า ตามลำดับ

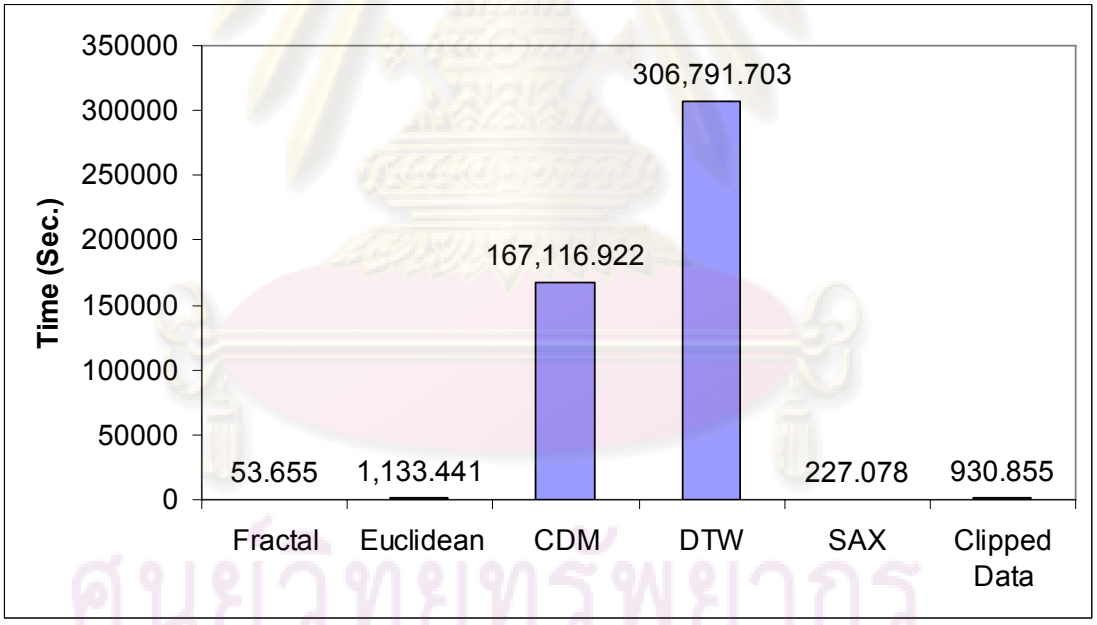
ซึ่งจะเห็นได้ว่า จากการทดลองที่ผ่านมาจะเป็นไปในทางเดียวกันคือ เมื่อปริมาณข้อมูลเพิ่มมากขึ้น เวลาที่ใช้สำหรับการแทนข้อมูลแบบแฟร็กทัลจะน้อยกว่าวิธีอื่น ๆ ตามไปด้วย เนื่องจาก ในการทดสอบจะทำการคำนวณค่าแฟร็กทัลในสถานะออนไลน์เพียงแค่ข้อมูลทดสอบ ในส่วนของข้อมูลฝึกหัดสามารถคำนวณเก็บไว้ก่อนในสถานะออฟไลน์ ซึ่งจะเป็นข้อดีสำหรับการลดขนาดของข้อมูล ซึ่งจะเห็นได้ชัดเจนที่สุดในชุดข้อมูลที่เก้าที่มีข้อมูลฝึกหัดเป็นจำนวนมาก

4.3.3.3 ผลการทดสอบสำหรับชุดข้อมูลที่เก้า ความยาวเท่ากับ 3,000 จุด

ชุดข้อมูลที่เก้าเน้นไปที่การเปรียบเทียบประสิทธิภาพในด้านเวลาเป็นหลัก สำหรับชุดข้อมูลทดสอบและข้อมูลฝึกหัด โดยมีจำนวนของข้อมูลทดสอบเท่ากับ 300 อนุกรม และจำนวนข้อมูลฝึกหัดเท่ากับ 6,153 ข้อมูล และมีจำนวนประเภทเท่ากับ 6 ประเภท ในส่วนของผลการทดลองสำหรับประสิทธิภาพในด้านความแม่นยำและความเร็วของชุดข้อมูลที่เก้าแสดงได้ดังรูปที่ 4.18 และ 4.19 ตามลำดับ



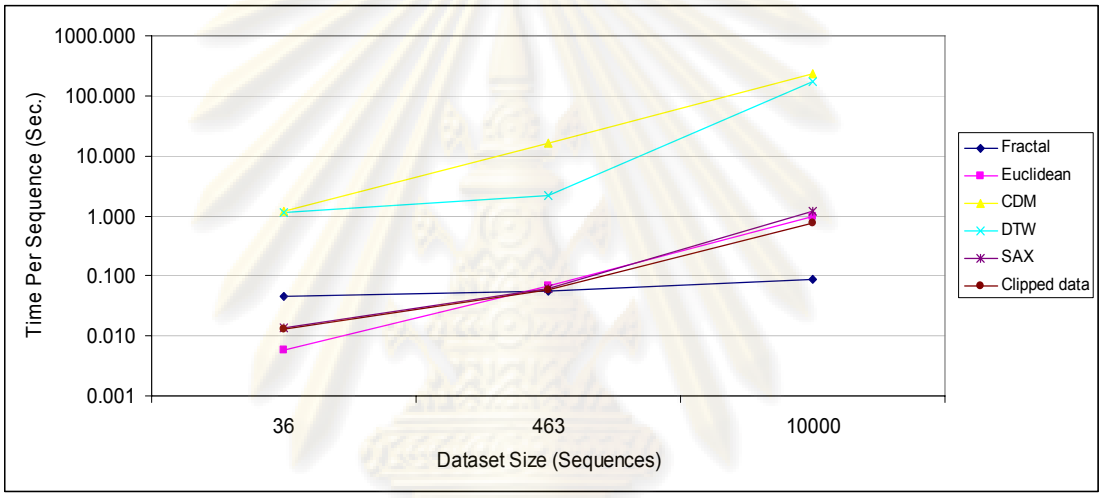
รูปที่ 4.18 ผลการทดสอบประสิทธิภาพความแม่นยำสำหรับชุดข้อมูลที่เก้า



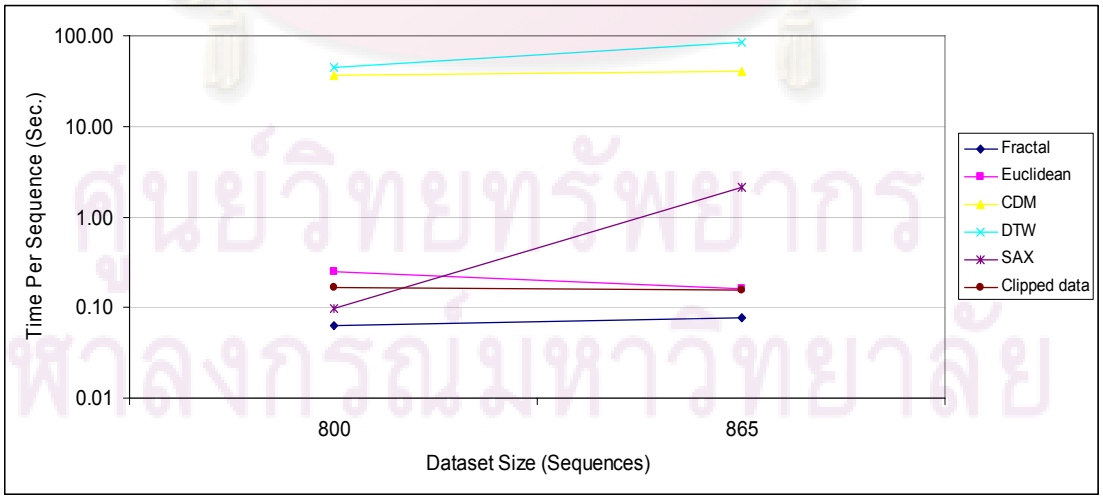
รูปที่ 4.19 ผลการทดสอบประสิทธิภาพความเร็วสำหรับชุดข้อมูลที่เก้า

สำหรับชุดข้อมูลสุดท้ายของงานวิจัยนี้ จะทดสอบประสิทธิภาพในด้านของเวลาสำหรับข้อมูลฝึกหัดจำนวนมาก ๆ ซึ่งเวลาที่ใช้ในการคำนวณจะน้อยกว่าการวัดระยะทางแบบยุคลิดถึง 20 เท่า โดยผลความแม่นยำของการแทนข้อมูลแบบแฟร็กทัลก็ยังคงให้ผลที่มากกว่าการลดขนาดข้อมูลอื่น ๆ อีกเช่นกัน

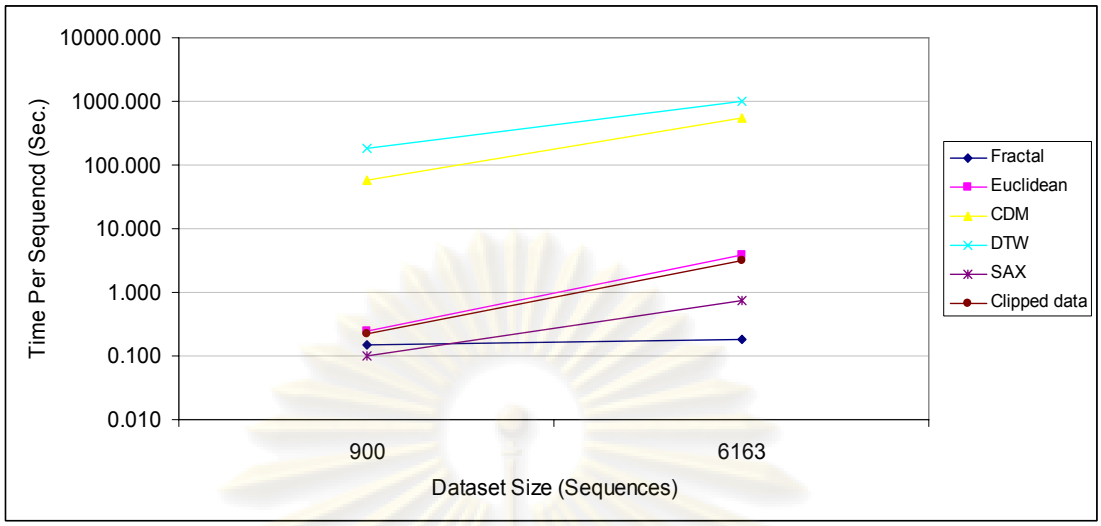
ในส่วนนี้เป็นการเปรียบเทียบเวลาในการคำนวณสำหรับข้อมูลอนุกรมเวลาขนาดใหญ่ที่มีความยาวเท่ากับ 1,000 2,000 และ 3,000 จุด โดยนำเวลาในการคำนวณจากแต่ละชุดข้อมูลที่ได้ทดลองไว้แล้วมาวาดบนกราฟเพื่อดูแนวโน้วของความเร็วสำหรับแต่ละวิธี ซึ่งจะพบว่า เวลาในการคำนวณของการแทนข้อมูลแบบแฟร็กทัลเวลาในการคำนวณมีการเปลี่ยนแปลงที่ช้ากว่าวิธีอื่น ๆ ทั้งหมด ในส่วนของแกนตั้งจะแสดงเวลาในการคำนวณต่อหนึ่งอนุกรมที่คำนวณได้จากแต่ละชุดข้อมูล สำหรับเวลาในการคำนวณของข้อมูลอนุกรมเวลาความยาวเท่ากับ 1,000 จุดแสดงได้ในตารางที่ 4.20 ข้อมูลอนุกรมเวลาความยาวเท่ากับ 2,000 จุดแสดงได้ในตารางที่ 4.21 และข้อมูลอนุกรมเวลาความยาวเท่ากับ 3,000 จุดแสดงได้ในตารางที่ 4.22



รูปที่ 4.20 เวลาในการคำนวณสำหรับข้อมูลอนุกรมเวลาความยาวเท่ากับ 1,000 จุด



รูปที่ 4.21 เวลาในการคำนวณสำหรับข้อมูลอนุกรมเวลาความยาวเท่ากับ 2,000 จุด



รูปที่ 4.22 เวลาในการคำนวณสำหรับข้อมูลอนุกรมเวลาความยาวเท่ากับ 3,000 จุด

จากการทดลองสำหรับทุกชุดข้อมูลอนุกรมเวลา การแทนข้อมูลแบบแฟร็กทัลจะมีประสิทธิภาพในด้านเวลามากยิ่งขึ้น ก็ต่อเมื่อจำนวนข้อมูลมีปริมาณมากขึ้นจะทำให้เวลาที่ใช้ในการคำนวณน้อยกว่าวิธีอื่น ๆ มาก สำหรับข้อมูลจำนวนไม่มากงานวิจัยนี้ก็ยังคงใช้เวลาที่ค่อนข้างต่ำ ถึงแม้ว่าจะมีบางวิธีที่ใช้เวลาน้อยกว่า แต่ความแตกต่างของเวลาในระดับวินาทีในทางปฏิบัติจะไม่เห็นถึงความต่างของเวลาที่เกิดขึ้น ในส่วนของผลความแม่นยำได้รับผลที่ใกล้เคียงกับการวัดความคล้ายคลึงที่ใช้ข้อมูลดิบในการคำนวณ และสามารถเอาชนะผลความแม่นยำกับทุกชุดข้อมูลของการวัดระยะทางแบบยุคลิด การแทนข้อมูลแบบแซค และการแทนข้อมูลแบบคลิป รวมทั้งในบางชุดข้อมูลได้รับผลความแม่นยำที่เท่ากันกับไดนามิกไทม์วอร์ปปิง และในชุดข้อมูลที่หกและเจ็ดได้รับผลที่ดีกว่าทุกวิธีด้วย ซึ่งเมื่อทดลองกับการปรับขนาดความยาวของข้อมูลอนุกรมเวลาในระดับตั้งแต่ 200 จนถึง 3,000 ก็ยังคงได้รับผลที่มีประสิทธิภาพรวมถึงข้อมูลอนุกรมเวลาขนาดสั้นด้วย

แต่สำหรับงานวิจัยนี้จะพบปัญหาหลัก ๆ กับข้อมูลในบางประเภทคือ ยังไม่สามารถจำแนกข้อมูลอนุกรมเวลาชนิดเดียวกัน แต่ต่างประเภทกัน โดยมีความแตกต่างกันในโครงสร้างเพียงเล็กน้อย ซึ่งมิติเส้นขอบที่เป็นการประมาณค่าแบบหยาบจะค้นหาส่วนของความแตกต่างนั้นได้ไม่ดีนัก แต่สำหรับข้อมูลชุดที่แปด ได้ทดลองกับข้อมูลชนิดเดียวกันคือ ข้อมูลคลื่นหัวใจ สำหรับผู้ป่วยในแต่ละโรค ก็ยังสามารถให้ผลความแม่นยำที่สูงเกือบ 100 เปอร์เซ็นต์ ซึ่งจะเห็นได้ว่าข้อมูลประเภทนี้ที่แต่ละประเภทจะมีความแตกต่างกันเพียงเล็กน้อย แต่จะเกิดขึ้นหลาย ๆ คาบ ซ้ำ ๆ กัน ก็สามารถนำมาพัฒนากับการแทนข้อมูลแบบแฟร็กทัล ซึ่งเป็นประเด็นหลักที่ตั้งสมติฐานไว้สำหรับการนำมิติแฟร็กทัลมาพัฒนากับข้อมูลประเภทดังกล่าว

บทที่ 5

สรุปผลการวิจัย อภิปรายผล และข้อเสนอแนะ

สำหรับงานวิจัยนี้ได้พัฒนาการลดขนาดของข้อมูลอนุกรมเวลาโดยเน้นไปที่การลดขนาดของข้อมูลอนุกรมเวลาให้เหลือน้อยที่สุดและเท่ากันทุกตัวด้วยแนวคิดของมิติแฟร็กทัล ซึ่งสามารถลดขนาดของข้อมูลอนุกรมเวลาใด ๆ ให้เหลือเพียงเลขจำนวนจริงสองค่าเท่านั้น โดยในเรียกงานวิจัยนี้ว่าการแทนข้อมูลแบบแฟร็กทัล ซึ่งผลการทดสอบเมื่อเปรียบเทียบกับงานวิจัยอื่น ๆ ได้ผลลัพธ์ที่มีประสิทธิภาพความแม่นยำ และความเร็วโดยเฉพาะข้อมูลขนาดใหญ่ จะใช้เวลาในการคำนวณน้อยกว่าวิธีอื่นมาก ซึ่งแสดงผลการทดสอบทั้งหมดไว้ในบทที่ 4 โดยภาพรวมทั้งหมดสำหรับงานวิจัยนี้สามารถสรุปได้ดังนี้

5.1 สรุปผลการวิจัย

การแทนข้อมูลแบบแฟร็กทัลที่พัฒนาให้เหมาะสมกับข้อมูลอนุกรมเวลาที่ได้กล่าวไว้ทั้งหมด จะเห็นได้ว่า งานวิจัยนี้สามารถให้ผลของประสิทธิภาพความแม่นยำและความเร็วได้อย่างมีประสิทธิภาพกับทุกชุดข้อมูลที่ทดสอบ ถึงแม้ว่าในบางชุดข้อมูลประสิทธิภาพของความแม่นยำไม่สามารถชนะไดนามิกไทม์วอร์ปปิงได้ แต่วิธีการลดขนาดของข้อมูลจะต้องสูญเสียคุณลักษณะของข้อมูลอนุกรมเวลาให้เหลือขนาดที่ลดลงมาก ซึ่งการนำไปเปรียบเทียบกับวิธีที่ใช้ข้อมูลดิบในการคำนวณจะเป็นงานที่ยาก แต่จะพบว่าความแม่นยำที่ได้ในการคำนวณด้วยการแทนข้อมูลแบบแฟร็กทัลก็จะได้ผลลัพธ์ใกล้เคียงกับไดนามิกไทม์วอร์ปปิง และในบางผลการทดลองสามารถเอาชนะกับทุกวิธีที่นำมาเปรียบเทียบ

ในส่วนของงานวิจัยที่นำมาเปรียบเทียบอื่น ๆ เช่น การวัดระยะทางแบบยุคลิด งานวิจัยนี้สามารถชนะในด้านผลความแม่นยำได้ทั้งหมด แต่ในด้านความเร็ว ถ้าจำนวนข้อมูลมีปริมาณไม่มากนัก การวัดระยะทางแบบยุคลิดจะได้ผลลัพธ์ที่มีประสิทธิภาพกว่างานวิจัยนี้ แต่ในทางปฏิบัติจะพบว่าเวลาในการคำนวณจริง ๆ จะอยู่ในระดับวินาที ซึ่งแสดงได้ดังผลการทดลองกับข้อมูลชุดที่สอง ซึ่งจะไม่แตกต่างกันมากนัก แต่ถ้าเปรียบเทียบกับข้อมูลอนุกรมเวลาที่มีปริมาณมาก ๆ การแทนข้อมูลแบบแฟร็กทัลสามารถเอาชนะได้มากถึง 22 เท่า ในชุดข้อมูลที่เก้า ซึ่งคล้ายกับผลลัพธ์ของการแทนข้อมูลแบบคลิป สำหรับวิธีซีดีเอ็ม จะเห็นว่า ผลความแม่นยำสำหรับงานวิจัยนี้ไม่สูงมากนัก เนื่องจากข้อมูลอนุกรมเวลาที่ถูกปรับระดับคะแนนด้วย Z จะทำให้ระดับของทุกข้อมูลอนุกรมเวลาอยู่ในระดับเดียวกัน เมื่อกระทำการลดขนาดด้วยแซคจะส่งผลต่อความแม่นยำ และการทดลองได้เลือกผลความแม่นยำที่ดีที่สุดสำหรับจำนวนตัวอักษรตั้งแต่ 3 ถึง 10 ตัวอักษร สำหรับการทดลองกับการแทนข้อมูลแบบแซค งานวิจัยนี้สามารถเอาชนะทั้งในผลความแม่นยำและความเร็วเมื่อทำการเลือกผลที่ดีที่สุดในด้านความ

แม่นยำ โดยการปรับความยาวของของข้อมูลอนุกรมเวลา โดยลดจากความยาวเดิมทีละ 100 จุด จนกระทั่งเหลือความยาวน้อยสุดเท่ากับ 100 จุดและปรับจำนวนตัวอักษรตั้งแต่ 3 ถึง 10 ตัวอักษร ซึ่งจากการทดลองทั้งหมดพอจะสรุปได้ว่า การแทนข้อมูลแบบแฟร็กทัลสามารถให้ผลลัพธ์ในด้านความแม่นยำได้อย่างมีประสิทธิภาพเมื่อเปรียบเทียบกับงานวิจัยอื่น ๆ และผลในด้านความเร็วจะมีประสิทธิภาพมากขึ้นอย่างต่อเนื่องเมื่อปริมาณข้อมูลเพิ่มมากขึ้น และสำหรับข้อมูลจำนวนไม่มาก เช่น ชุดข้อมูลที่สองและหก เวลาในทางปฏิบัติจริงจะไม่มีความแตกต่างกันมากนักกับเวลาที่แตกต่างกันในหลักวินาทีเมื่อเปรียบเทียบกับวิธีอื่น ๆ

5.2 ข้อเสนอแนะ

ข้อเสนอแนะสำหรับแนวทางต่อไปในงานวิจัยเพื่อนำมิติแฟร็กทัลมาพัฒนากับข้อมูลอนุกรมเวลา โดยประเด็นหลักที่ต้องการพัฒนาสำหรับงานวิจัยนี้คือการลดขนาดของข้อมูลอนกรมเวลาให้ได้น้อยที่สุด คือ แต่ละตัวของข้อมูลอนุกรมเวลาสามารถลดลงเหลือเพียงเลขจำนวนจริงเลขเดียวเท่านั้น และยังได้รับผลประสิทธิภาพทั้งด้านความแม่นยำและความเร็วได้ดีเมื่อเปรียบเทียบกับงานวิจัยอื่น ซึ่งน่าจะเป็นงานวิจัยท้ายสุดสำหรับการลดขนาดของข้อมูล โดยงานวิจัยนี้ได้ทดสอบกับการลดขนาดของข้อมูลด้วยมิติแฟร็กทัลให้เหลือเพียง 1 ค่า ซึ่งแสดงในบทที่ 4 แต่ประสิทธิภาพที่ได้เมื่อเปรียบเทียบกับ 2 เลขจำนวนจริง ยังค่อนข้างต่ำ ซึ่งการลดขนาดของข้อมูลอนุกรมเวลาให้เหลือเลขจำนวนจริงเพียงเลขเดียว การจำแนกข้อมูลจะทำได้โดยการเรียงลำดับข้อมูลจากมากไปน้อยเท่านั้น ซึ่งข้อมูลที่อยู่ติดกันน่าจะจำแนกได้เป็นประเภทเดียวกัน โดยไม่จำเป็นต้องใช้วิธีการทำเหมืองข้อมูลอื่น ๆ เลย ซึ่งปัจจุบันจำเป็นต้องใช้การวัดความคล้ายคลึงด้วยการวัดระยะทางแบบยุคลิด เพื่อค้นหาข้อมูลประเภทเดียวกัน

สำหรับการพัฒนาอีกแนวทางหนึ่งคือ เพิ่มประสิทธิภาพการแทนข้อมูลแบบแฟร็กทัลเดิมให้มีประสิทธิภาพเพิ่มมากขึ้น ซึ่งมีหลายประเด็นที่น่าสนใจในการทดลองต่อไป ยกตัวอย่างเช่น ค่าตัวแปรต่าง ๆ ที่งานวิจัยนี้ได้นำเสนอไว้ตามสมการที่กล่าวไว้ คือ จำนวนรอบการทำซ้ำในการลากเส้นขอบ จุดหยุดการทำงานในแต่ละข้อมูลอนุกรมเวลาคือจุดใด หรือค่าไทม์สไลด์และเซกเมนต์เส้นขอบสำหรับแต่ละมิติเส้นขอบที่เหมาะสมกว่าการคำนวณที่ใช้ในปัจจุบัน ซึ่งถ้าสามารถหาค่าของแต่ละพารามิเตอร์ที่เหมาะที่สุดได้ จะทำให้ประสิทธิภาพของการแทนข้อมูลแบบแฟร็กทัลเพิ่มมากขึ้น โดยผู้วิจัยได้เคยทดลองปรับพารามิเตอร์ต่าง ๆ ของชุดข้อมูลที่สาม แล้วพบว่าการแทนข้อมูลแบบแฟร็กทัลสามารถให้ผลได้ดีที่สุด โดยได้รับผลการทดลองที่สูงกว่าวิธีในปัจจุบันมากขึ้นกว่า 20 เปอร์เซ็นต์ สำหรับการทดสอบกับการแทนข้อมูลแบบแฟร็กทัลด้วยจำนวนจริง 3 ค่า

ในที่สุดแล้วจะเห็นได้ว่า มิติแฟร็กทัลเป็นวิธีที่มีประสิทธิภาพในการค้นหารูปแบบที่มีลักษณะซ้ำ ๆ ภายในข้อมูลอนุกรมเวลา หรือสามารถค้นหาความคล้ายคลึงตัวเองได้

อย่างเหมาะสม ซึ่งงานวิจัยนี้เป็นงานวิจัยแรก ๆ ในการนำมิติแฟร็กทัลมาพัฒนากับข้อมูลอนุกรมเวลา และมีประสิทธิภาพสูงเมื่อเปรียบเทียบกับงานวิจัยอื่น ๆ

รายการอ้างอิง

[1] Ratanamahatana, C.A., and Keogh, E. (2005). Three Myths about Dynamic Time Warping. Proceedings of SIAM International Conference on Data Mining (SDM) 2005, pp. 506–510. Newport Beach, CA: SIAM.

[2] Keogh, E. (2002). Exact Indexing of Dynamic Time Warping. Proceedings of 28th International Conference on Very Large Data Bases, pp. 406–417. Hong Kong, China: VLDB Endowment.

[3] Ratanamahatana, C.A., and Keogh, E. (2004). Everything You Know about Dynamic Time Warping is Wrong. 3rd Workshop on Mining Temporal and Sequential Data, in conjunction with 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD) 2004, Seattle, WA, USA.

[4] Keogh, E., Lonardi, S., and Ratanamahatana, C.A. (2004). Towards Parameter-Free Data Mining. Proceedings of 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 206–215. Seattle, WA, USA Association for Computing Machinery

[5] Yi, B.K., and Faloutsos, C. (2000). Fast Time Sequence Indexing for Arbitrary LP-Norms. VLDB International Conference, pp. 385–394. Cairo, Egypt: Morgan Kaufmann Publishers.

[6] Lin, J., Keogh, E., Lonardi, S., and Chiu, B. (2003). A Symbolic Representation of Time Series, with Implications for Streaming Algorithms. Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, pp. 2–11. San Diego, CL, USA: Association for Computing Machinery

[7] Keogh, E., Lin, J., and Fu, A. (2005). HOT SAX: Efficiently Finding The Most Unusual Time Series Subsequence: Algorithms and Applications. Proceedings of 5th IEEE International Conference on Data Mining (ICDM), pp. 226–233. : IEEE Computer Society

[8] Bagnall, A., and Janacek, G. (2005). Clustering Time Series with Clipped Data Machine Learning Journal 58: 151–178.

[9] Bourke, P. (1991-2004). Fractals, Chaos [Online]. Available from: http://local.wasp.uwa.edu.au/~pbourke/fractals/index.html 2008]

[10] Peitgen, H.O., Jurgens, H., and Saupe, D. (2003). Chaos and Fractals New Frontiers of Science.

[11] Grabbe, J.O. (1999–2003). Chaos & Fractals in Financial Markets, Part 1-8 [Online]. Available from: http://www.aci.net/Kalliste/chaos_index.htm

[12] Miqel. (2006–2007). Naturally Occurring Fractals [Online]. Available from: http://www.miqel.com/fractals_math_patterns/visual-math-natural-fractals.html

[13] Bourke, P. (2007). Self Similarity [Online]. Available from: http://local.wasp.uwa.edu.au/~pbourke/fractals/selfsimilar/ 2008]

[14] Cantor, G. (1872). Cantor Dust [Online]. Available from: http://www.daviddarling.info/encyclopedia/C/Cantor_dust.html 2008]

[15] Jelinek, H.F., Jones, C.L., and Warfel, M.D. (1998). Complexity International [Online]. Available from: http://journal-ci.csse.monash.edu.au/ci_louise/vol06/jelinek/jelinek.html 2008]

[16] Clayton, K. (1996). Basic Concepts in Nonlinear Dynamics and Chaos [Online]. Available from: http://www.vanderbilt.edu/AnS/psychology/cogsci/chaos/workshop/Workshop.html 2009]

[17] Dickau, R.M. (1995-2002). First Steps Toward the Sierpinski Carpet [Online]. Available from: http://www.prairienet.org/~pops/carpet.html 2008]

[18] Grassberger, P., and Procaccia, I. (1983). Measuring the Strangeness of Strange Attractors. Physica D: Nonlinear Phenomena, pp. 189–208.

[19] Prokoph, A. (1999). Fractal, Multifractal and Sliding Window Correlation Dimension Analysis of Sedimentary Time Series. Computers & Geosciences 25, pp. 1009–1021 Tarrytown, NY, USA Pergamon Press.

[20] Ashkenazy, Y. (1999). The Use of Generalized Information Dimension in Measuring Fractal Dimension of Time Series. Physica A: Statistical Mechanics and its Applications, pp. 427–447.

[21] Guerrero, A., and Smith, L.A. (2003). Towards Coherent Estimation of Correlation Dimension. Physics Letters A, pp. 373–379. Elsevier.

[22] Barbará, D., and Chen, P. (2000). Using the Fractal Dimension to Cluster Datasets. Proceedings of 2000 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 260–264. Boston, MA: Association for Computing Machinery

[23] Gionis, A., Hinneburg, A., Papadimitriou, S., and Tsaparas, P. (2005). Dimension Induced Clustering. International Conference on Knowledge Discovery and Data

Mining (KDD), pp. 51–60. Chicago, LN, USA Association for Computing Machinery

[24] Xiao, H., Zhi-Zhong, W., and Xiao-mei, R. (May 30, 2005). Classification of surface EMG signal with fractal dimension. Journal of Zhejiang University SCIENCE (JZUS)

[25] Keogh, E., Shelton, C., and Moerchen, F. (2007). Workshop and Challenge on Time Series Classification [Online]. SIGKDD Available from: http://www.cs.ucr.edu/~eamonn/SIGKDD2007TimeSeries.html

[26] Keogh, E. (2006). The UCR Time Series Data Mining Archive [Online]. Available from: http://www.cs.ucr.edu/~eamonn/TSDMA/datasets.html

[27] Goldberger, A., Amaral, L., Glass, L., Gausdorff, J., Ivanov, P., Mark, R., et al. (2000). PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. Circulation Journal: 215–220.

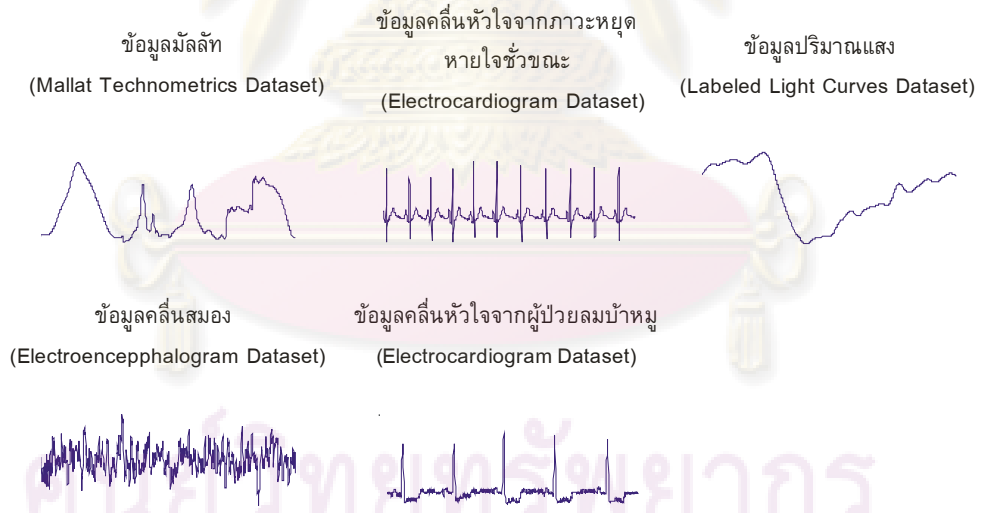
ภาคผนวก

ภาคผนวก ก

สำหรับตัวอย่างประเภทของชุดข้อมูลที่ใช้ในการทดลองซึ่งกล่าวไว้ในบทที่4 จะนำมาแสดงอยู่ในส่วนนี้ โดยจะแจกแจงตัวอย่างในแต่ละชุดข้อมูล โดยนำมาแสดงเป็น 1 ตัวอย่างของข้อมูลอนุกรมเวลาต่อ 1 ประเภท ซึ่งข้อมูลแต่ละตัวได้มาจากหน่วยเก็บถาวรของมหาวิทยาลัยแคลิฟอร์เนีย [25, 26] และฟิซิโอเน็ต (PhysioNet) [27] โดยมีทั้งหมด 10 ชุดข้อมูล โดยตั้งแต่ ชุดข้อมูลที่ 1 ถึง 6 จะนำไปใช้สำหรับวิธีการจำแนกข้อมูลแบบเพื่อนบ้านใกล้ที่สุดอันดับที่หนึ่ง และทดสอบแบบการนำออกหนึ่ง และชุดข้อมูลที่ 7 ถึง 10 จะนำเป็นข้อมูลที่ถูกแบ่งเป็นชุดข้อมูลฝึกหัดและข้อมูลทดสอบ

ก.1 ชุดข้อมูลที่หนึ่ง

ข้อมูลชุดที่หนึ่งมีจำนวนประเภทของข้อมูลอนุกรมเวลาเท่ากับ 5 ประเภท และมีจำนวนข้อมูลทั้งหมดเท่ากับ 10,000 อนุกรม โดยแต่ละประเภทจะถูกแบ่งจำนวนข้อมูลเท่า ๆ กัน คือ ประเภทละ 2,000 อนุกรม และข้อมูลอนุกรมเวลามีความยาวเท่ากับ 1,000 จุด ดังแสดงในรูปที่ ก.1

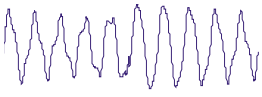


รูปที่ ก.1 ข้อมูลอนุกรมเวลาของแต่ละประเภทสำหรับชุดข้อมูลที่หนึ่ง

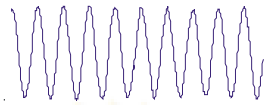
ก.2 ชุดข้อมูลที่สอง

ข้อมูลชุดที่สองมีจำนวนประเภทของข้อมูลอนุกรมเวลาเท่ากับ 18 ประเภท และมีจำนวนข้อมูลทั้งหมดเท่ากับ 36 อนุกรม โดยแต่ละประเภทจะถูกแบ่งจำนวนข้อมูลเท่า ๆ กัน คือ ประเภทละ 2 อนุกรม และมีความยาวเท่ากันคือ 1,000 จุด ดังแสดงในรูปที่ ก.2

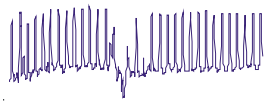
ข้อมูลการเคลื่อนไหวของกล้ามเนื้อจากผู้ป่วยกระดูกหัก
(Motor Current Dataset-Broken Bars)



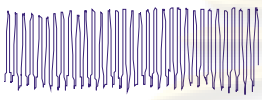
ข้อมูลการเคลื่อนไหวของกล้ามเนื้อจากคนปกติ
(Motor Current Dataset-Healthy)



ข้อมูลปืน-Ann
(Ann Gun Dataset)

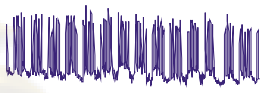


ข้อมูลปืน-Keogh
(Keogh Gun Dataset)

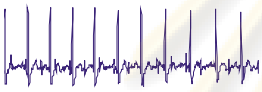


ข้อมูลความต้องการพลังงานของชาวอิตาลี
(Power Demand Dataset-italian)

ข้อมูลความต้องการพลังงานของชาวฮอลแลนด์
(Power Demand Dataset-Dutch)

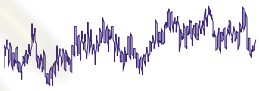


ข้อมูลคลื่นหัวใจของทารกในครรภ์
(Foetal Electrocardiogram Dataset)

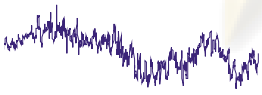


ข้อมูลความแห้งในอากาศ
(Dryer Dataset)

ข้อมูลทะเลสาบ
(Great Lakes Dataset)

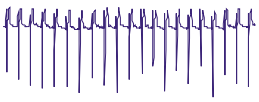


ข้อมูลสัญญาณระดับน้ำทะเล
(Buoy Sensor Dataset)



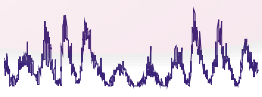
ข้อมูลคลื่นหัวใจโคสตีแบบช้า
(Koski Electrocardiogram Data-Slow)

ข้อมูลคลื่นหัวใจโคสตีแบบเร็ว
(Koski Electrocardiogram Data-Fast)



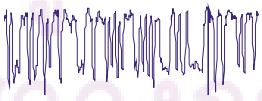
ข้อมูลแลกเปลี่ยนเงินตรา
(Exchange Dataset)

ข้อมูลจุดบอดบนดวงอาทิตย์
(Sunspot Dataset)



ข้อมูลความร้อนของเตาหลอม
(Furnace Dataset)

ข้อมูลอัตราการหมุน
(Reel Dataset)



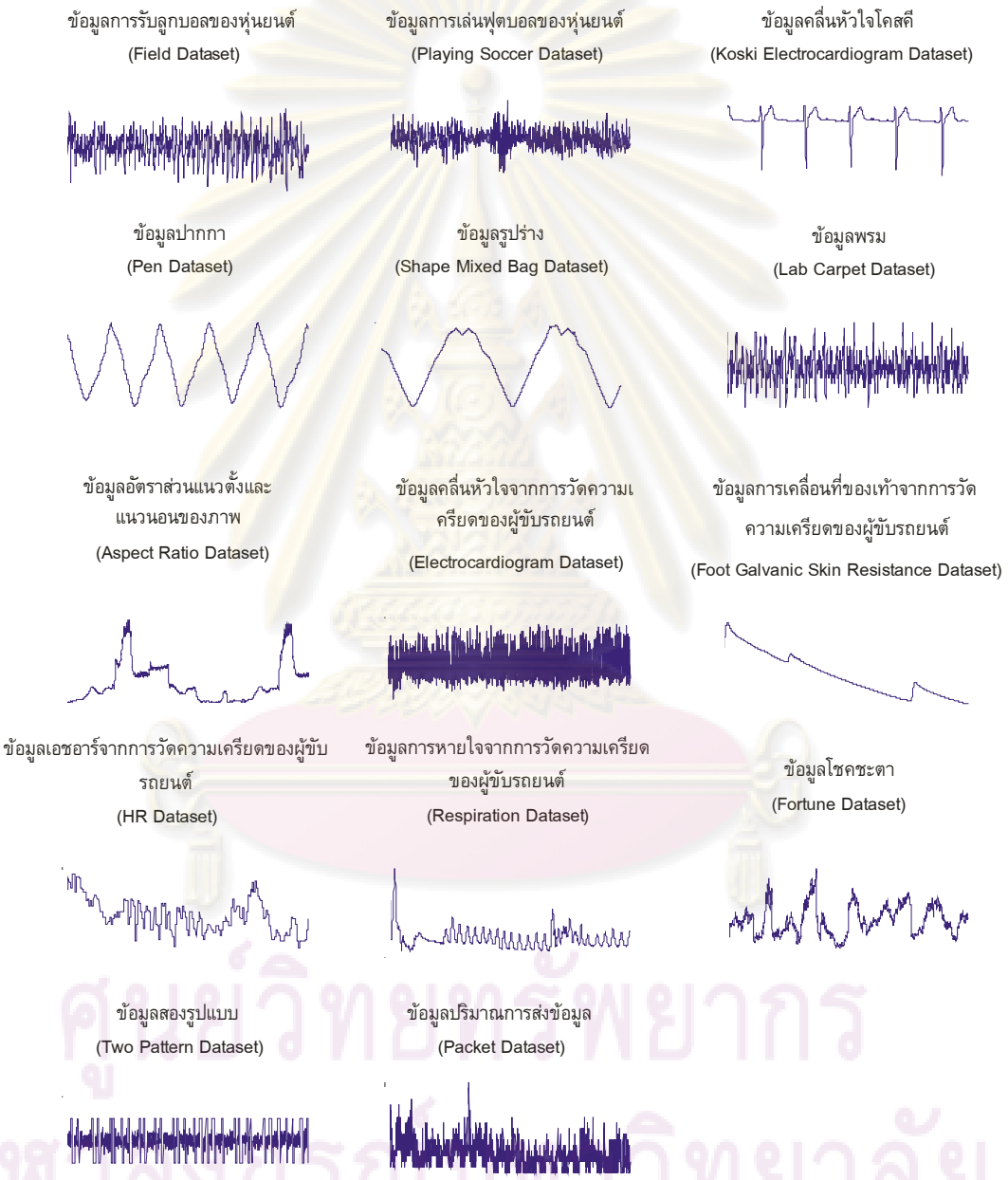
ข้อมูลบอลลูน
(Balloon Dataset)

ข้อมูลปริมาณความชื้น
(Evaporator Dataset)

รูปที่ ก.2 ข้อมูลอนุกรมเวลาของแต่ละประเภทสำหรับชุดข้อมูลที่สอง

ก.3 ชุดข้อมูลที่สาม

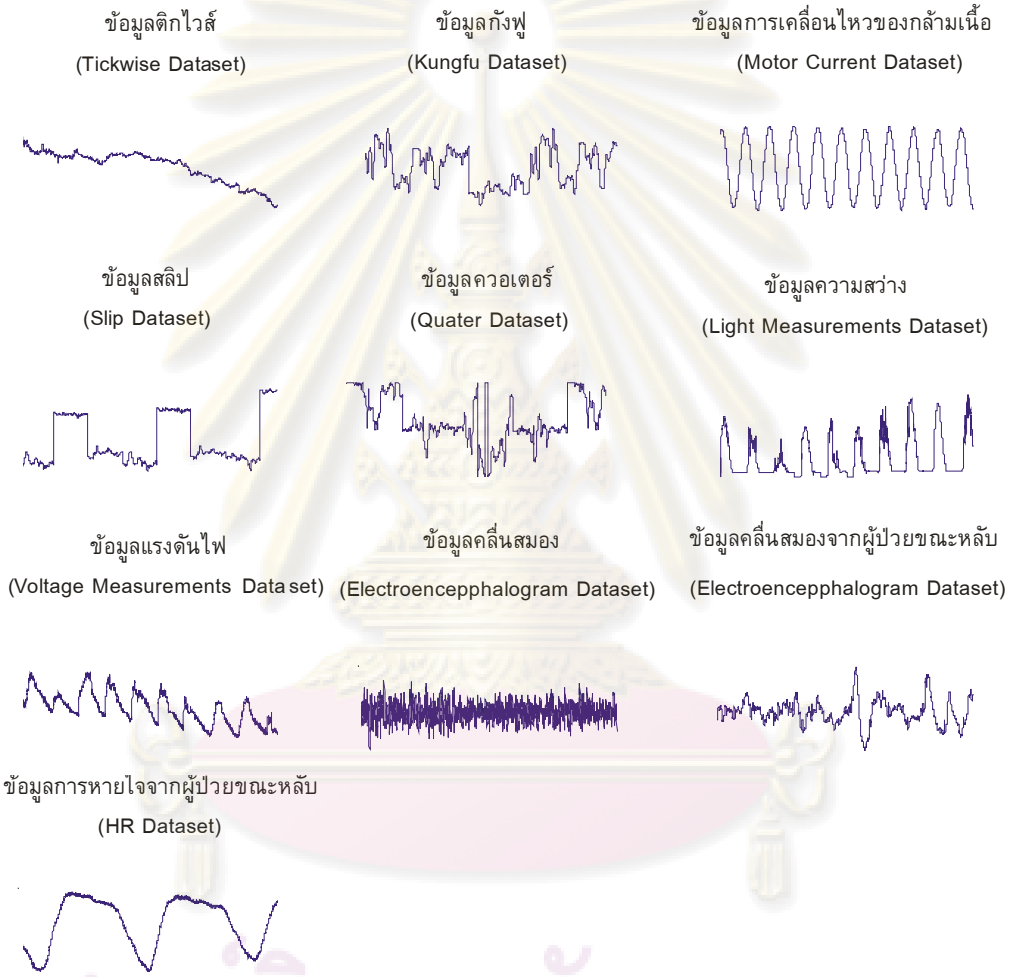
ข้อมูลชุดที่สามมีจำนวนประเภทของข้อมูลอนุกรมเวลาเท่ากับ 14 ประเภท และมีจำนวนข้อมูลทั้งหมดเท่ากับ 880 อนุกรม โดยแต่ละประเภทจะมีจำนวนข้อมูลไม่เท่ากัน โดยแต่ละตัวของข้อมูลอนุกรมเวลามีความยาวเท่ากับ 2,000 จุด ดังแสดงในรูปที่ ก.3



รูปที่ ก.3 ข้อมูลอนุกรมเวลาของแต่ละประเภทสำหรับชุดข้อมูลที่สาม

ก.4 ชุดข้อมูลที่สี่

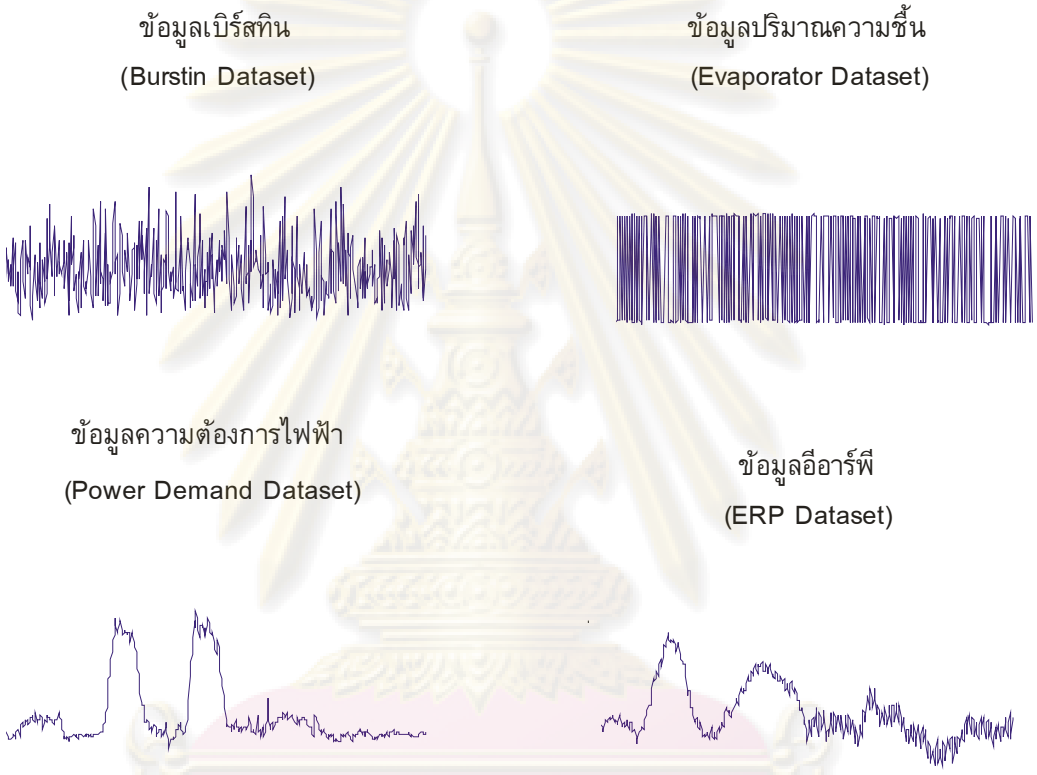
ข้อมูลชุดที่สี่มีจำนวนประเภทของข้อมูลอนุกรมเวลาเท่ากับ 10 ประเภท และมีจำนวนข้อมูลทั้งหมดเท่ากับ 900 อนุกรม โดยแต่ละประเภทจะถูกแบ่งจำนวนข้อมูลเท่า ๆ กัน คือ ประเภทละ 90 อนุกรม ซึ่งข้อมูลอนุกรมเวลาแต่ละตัวจะมีขนาดเท่ากับ 3,000 จุด ดังแสดงในรูปที่ ก.4



รูปที่ ก.4 ข้อมูลอนุกรมเวลาของแต่ละประเภทสำหรับชุดข้อมูลที่สี่

ก.5 ชุดข้อมูลที่ห้า

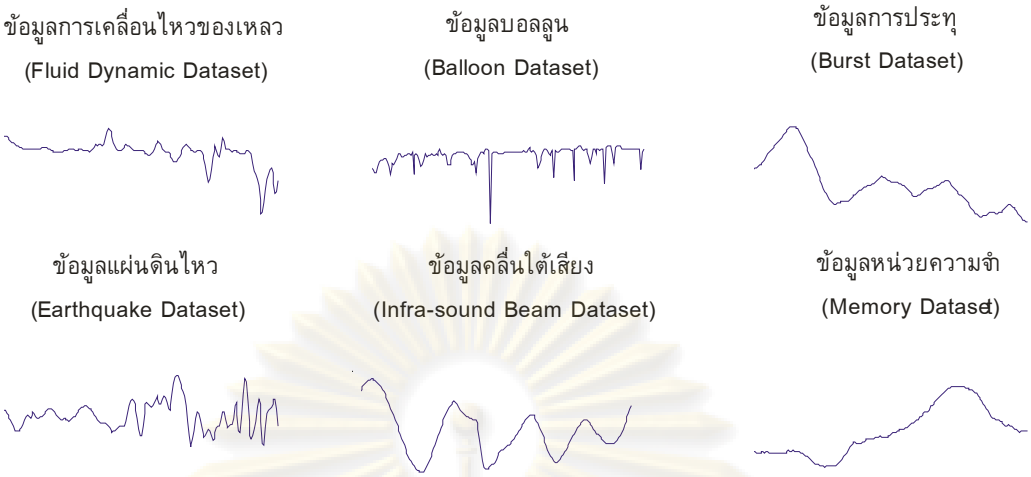
ข้อมูลชุดที่ห้าเป็นชุดข้อมูลขนาดสั้น โดยมีจำนวนประเภทของข้อมูลอนุกรมเวลาเท่ากับ 4 ประเภท และมีจำนวนข้อมูลทั้งหมดเท่ากับ 600 อนุกรม โดยแต่ละประเภทจะมีจำนวนข้อมูลไม่เท่ากัน ซึ่งแต่ละตัวของข้อมูลอนุกรมเวลามีความยาวเท่ากับ 500 จุด ดังแสดงในรูปที่ ก.5



รูปที่ ก.5 ข้อมูลอนุกรมเวลาของแต่ละประเภทสำหรับชุดข้อมูลที่ห้า

ก.6 ชุดข้อมูลที่หก

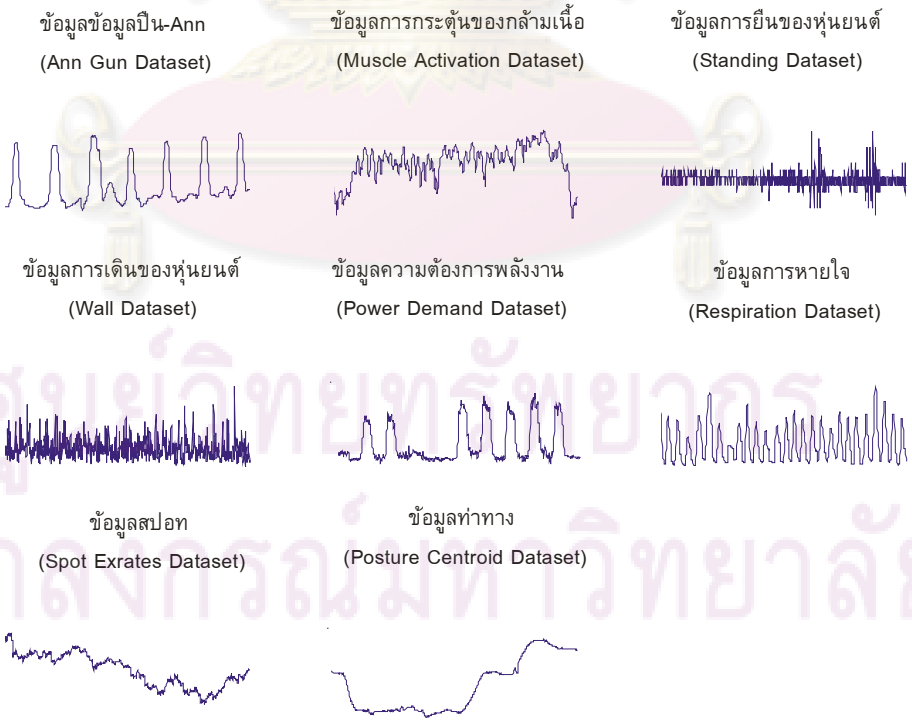
ข้อมูลชุดที่หกเป็นชุดข้อมูลขนาดสั้น โดยมีจำนวนประเภทของข้อมูลอนุกรมเวลาเท่ากับ 6 ประเภท และมีจำนวนข้อมูลทั้งหมดเท่ากับ 210 อนุกรม โดยแต่ละประเภทจะมีจำนวนข้อมูลไม่เท่ากัน ซึ่งแต่ละตัวของข้อมูลอนุกรมเวลามีความยาวเท่ากับ 200 จุด ดังแสดงในรูปที่ ก.6



รูปที่ ก.6 ข้อมูลอนุกรมเวลาของแต่ละประเภทสำหรับชุดข้อมูลที่หก

ก.7 ชุดข้อมูลที่เจ็ด

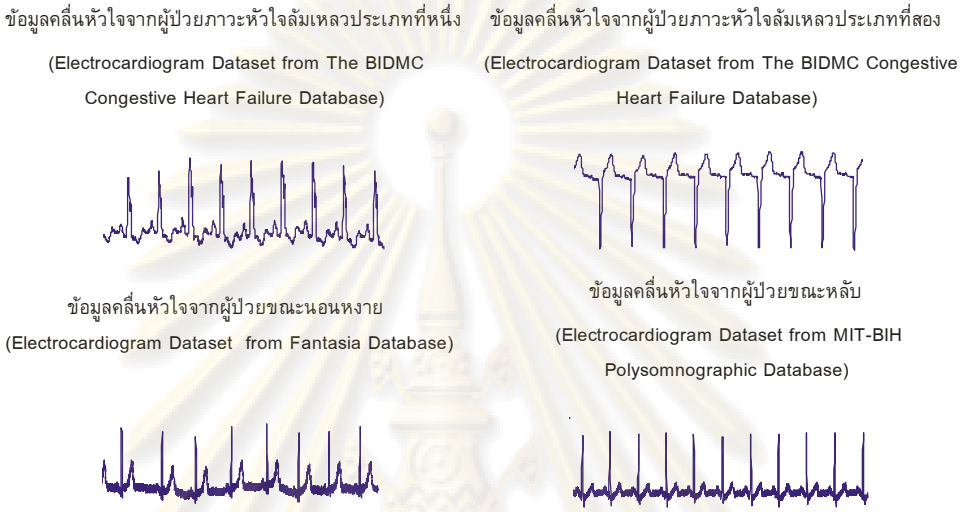
ข้อมูลชุดที่เจ็ดมีจำนวนประเภทของข้อมูลเท่ากับ 8 ประเภท และมีจำนวนข้อมูลฝึกหัดเท่ากับ 463 อนุกรม และข้อมูลทดสอบเท่ากับ 47 อนุกรม โดยจำนวนในแต่ละประเภทของข้อมูลฝึกหัดและข้อมูลทดสอบจะไม่เท่ากัน โดยมีความยาว 1,000 จุด ดังแสดงในรูปที่ ก.7



รูปที่ ก.7 ข้อมูลอนุกรมเวลาของแต่ละประเภทสำหรับชุดข้อมูลที่เจ็ด

ก.8 ชุดข้อมูลที่แปด

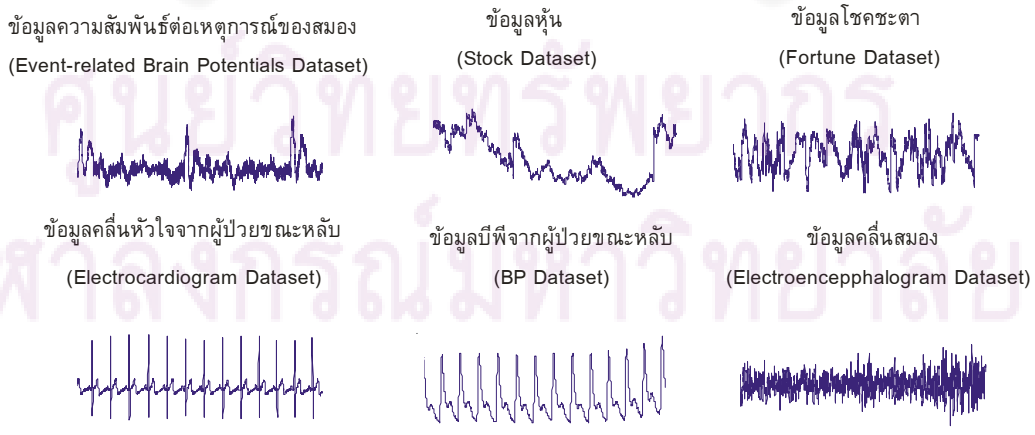
ข้อมูลชุดที่เจ็ดมีจำนวนข้อมูลฝึกหัดเท่ากับ 463 อนุกรม และจำนวนประเภทของข้อมูลเท่ากับ 4 ประเภท ซึ่งข้อมูลทดสอบเท่ากับ 47 อนุกรม โดยจำนวนในแต่ละประเภทของข้อมูลฝึกหัดและข้อมูลทดสอบจะไม่เท่ากัน และมีความยาว 2,000 จุด ดังแสดงในรูปที่ ก.8



รูปที่ ก.8 ข้อมูลอนุกรมเวลาของแต่ละประเภทสำหรับชุดข้อมูลที่แปด

ก.9 ชุดข้อมูลที่เก้า

ข้อมูลชุดที่เจ็ดมีจำนวนประเภทของข้อมูลเท่ากับ 6 ประเภท และมีจำนวนข้อมูลฝึกหัดเท่ากับ 6,163 อนุกรม และข้อมูลทดสอบเท่ากับ 300 อนุกรม โดยในแต่ละประเภทมีจำนวนของข้อมูลฝึกหัดไม่เท่ากัน โดยมีความยาว 3,000 จุด ดังแสดงในรูปที่ ก.9



รูปที่ ก.9 ข้อมูลอนุกรมเวลาของแต่ละประเภทสำหรับชุดข้อมูลที่เก้า

ภาคผนวก ข

บทความทางวิชาการเรื่อง “Efficient Time Series Mining using Fractal Representation” โดยพจน์ สัจจิพานนท์ และโชติรัตน์ รัตนามหัทธนะ ในงานประชุมวิชาการนานาชาติ ครั้งที่ 3 “The 2008 International Conference on Convergence and Hybrid Information Technology (ICCIT)” ซึ่งจัดขึ้น ณ เมืองปูซาน ประเทศเกาหลีใต้ ระหว่างวันที่ 11 ถึง 13 พฤศจิกายน 2551

Efficient Time Series Mining using Fractal Representation

Poat Sajjipanon Chotirat Ann Ratanamahatana

Department of Computer Engineering, Chulalongkorn University
Phayathai Rd., Pathumwan, Bangkok 10330 Thailand
{g50psj,ann}@cp.eng.chula.ac.th

ABSTRACT

As time series mining has become more prevalent and attracted much research interest, recent goals and efforts have been shifted toward scalability issue. One of the successful solutions is finding suitable representation of the data via dimensionality reduction. In this work, we introduce a novel fractal representation for time series data, which uses merely three real values to represent any time series. One of its unique advantages is that this representation does capture the pattern and self similarity within, thus representing the time series' global structure using only a few values, producing incredible speedup in similarity search. We demonstrate the utility of our proposed method on clustering problems, which are shown to be scalable to much larger datasets.

KEYWORDS

Time Series Mining, Fractal Representation, Fractal Dimension, Clustering

1. INTRODUCTION

As similarity search has become one of the most prevalent tasks for time series mining, two main focal points to the research interests are directed to accuracy and speed. Several similarity measures have been proposed to help improve the accuracy of time series mining, including Dynamic Time Warping (DTW) [1] and a more recent Compression-Based Dissimilarity Measure (CDM) [2]. Alternatively, the speedup has been improved through indexing, dimensionality reduction/data representation, among many others. The dimensionality reduction technique for time series data aims to find a small representation from the data structure. At present, a large number of dimensionality reduction techniques have been proposed, including Piecewise Aggregate Approximation (PAA) [3], Symbolic Aggregate Approximation (SAX) [4, 5], and clipped data representation [6], most of which require that the number of desired dimension must be specified beforehand. However, determining this appropriate number of dimension is a relatively difficult task since multi-resolution scheme is usually allowed, depending on the desired reconstruction error of each time series and the accuracy of the overall data mining tasks.

Ideally, if there exists a precise one-dimensional representation of the data, everything will be effortless and straightforward; all we need is to sort those numbers at hands, and we can retrieve any nearest neighbors with ease, without any need to calculate pairwise similarity among all the objects. Therefore, the smaller the dimension, the easier the data manipulation. Since time series data can be seen as an n-dimensional data, where n is the length of the time series, past research attempts have been focusing on how to approximate or reduce its dimensionality. Most of these methods have been used to facilitate the similarity search by pruning off unnecessary data that guarantee not to be the solution.

In this paper, we propose a novel representation of the time series data based on fractal dimension, which can transform any time series data to a 3-value representation that reflects the pattern and self-similarity within each data sequence. Such a small representation can achieve a large speedup on the data mining tasks, and our proposed fractal representation could actually achieve better accuracy than manipulating on the raw data directly. We demonstrate the utility of our work on clustering problems, comparing to the results from other existing methods, such as Euclidean Distance, Dynamic Time Warping (DTW), and Compression-Based Dissimilarity Measure (CDM).

The rest of this paper is organized as follows. Section 2 briefly describes related work on dimensionality reduction techniques. Section 3 discusses background material of fractal dimension, and then introduces our proposed work in Section 4 on how to exploit fractal dimension with time series data. Section 5 contains our experimental evaluation and

discussion, following by conclusion and suggestion for future work in Section 6.

2. RELATED WORK

Recent research has been focusing on how to speed up time series data mining tasks. Many research groups, including Yi and Faloutsos [3], Lin and Keogh et al. [4, 5], and Bagnall and Janacek [6], among many others, have proposed various dimensionality reduction techniques for time series data, which have been demonstrated to work well, in terms of speeding up the data mining tasks by pruning off the sequences that guarantee not to be the answer, while the accuracy still remains unchanged.

Others have been focusing on how to make time series data mining more accurate, mostly concentrating on the distance measure [1]. Dynamic Time Warping (DTW) distance measure has been accepted as one of the best distance measures for time series data, particularly for relatively short time series. For longer ones, some have proposed a compression-based algorithm [2], inspired by Kolmogorov Complexity theory, to measure similarity among the data sequences, while being able to capture the overall structure of the data. Vlachos et al. [7] transform long time series into periodogram using autocorrelation values, which has been shown to be effective for clustering problems.

Fractal dimension, on the other hand, is not widely known in the data mining community. Most research has applied it to images and data points. For example, Barbara and Chen [8] have used fractal dimension to distinguish different groups of data. Gionis and Hinneburg [9] use fractal dimension for separating data that has been mixed between lines and planes, and Xiao et al. [10] use fractal dimension to classify surface of EMG signals.

3. FRACTAL DIMENSION

Before getting into the notion of fractal dimension, we would like to first describe the general idea of the topological dimension which lies on the Euclidean space, i.e., points have zero dimension, lines and curves have one dimension, planes have two dimensions, and cubes have three dimensions. These dimensions are represented by integer values. However, dimensions can also be real values, which can be determined by Fractal dimension [11, 12]. Fractal is a deterministic dynamic system which is part of chaos theory inspired by the self-similarity property. This particular property is the heart of fractal dimension which is calculated from repeating patterns. A fractal dimension can be seen as an irregular geometric object with an infinite nesting of structure at all scales. Many types of fractal dimensions have been proposed, such as Hausdorff dimension, box-counting dimension, compass dimension, and correlation dimension. In practice, the last three are widely used, partly due to their ease of implementation. Mostly, these fractal dimensions are closely associated with image data, but in this work, we will demonstrate that they can also work well with time series data.

Figure 1 illustrates a typical fractal dimension framework. Input data is applied repeatedly to a fractal function with various parameters. Each set of parameter will produce one point on a log-log plot. After all the iterations, a straight line is fitted to this plot, and finally the fractal dimension is determined by the slope of this fitted line.

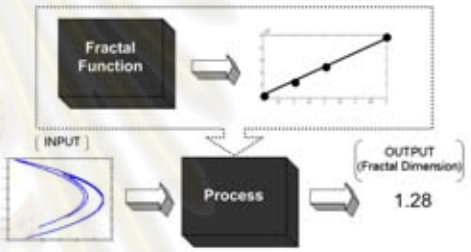


Figure 1. General framework of fractal dimension calculation.

3.1. Compass Dimension

Compass dimension [11] is a method for calculating the dimension by using a compass or a ruler to travel along the perimeter of input data. First, we select an arbitrary length of the line segment (compass length), and then use this line to traverse from the starting position of the input data until it reaches the ending position of the data, as shown in Figure 2. The ruler length s here is 10 units, and the overall estimate of its perimeter is 5*10 = 50 units in length.

The overall length of all the line segments becomes one of the outputs of this compass function. The subsequent iterations are repeated in similar fashion, but with shorter line segment, generally half the previous length. After a few iterations when the line segment's length approaches zero, it terminates. At this point, we will have a plot between the length of the line segment (x-axis) and the compass function output corresponding to that line segment (y-axis), in a log-log scale. Finally, the compass dimension (D_c) is determined by the slope of the best-fitted line to those points, shown in the following equation.

$$D_C = \lim_{s \to 0} \frac{\log N(s)}{\log s} \quad (1)$$

where, s is the length of the line segment, and $N(s)$ is the sum of the line segment lengths from the start of the curve to the end, composed by these line segments of length s. This compass dimension will be closer to 1 if the input is very smooth, e.g., virtually an arc of a circle, or a straight line. In general, the "rougher" the input signal/perimeter, the steeper the slope, and the larger the fractal dimension.

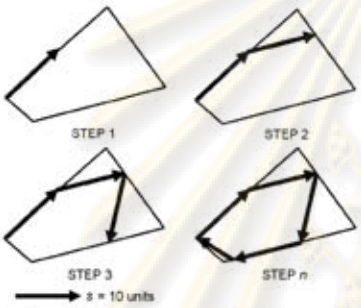


Figure 2. Part of the compass dimension calculation steps, using some fixed-length line segment to traverse around the image.

3.2. Correlation Dimension

In 1983, Grassberger and Procaccia [13] proposed a new method for fractal dimension calculation, which is still widely used nowadays. Correlation dimension [14-16] is a measure of the dimensionality of the space occupied by a set of random points. It can also be seen as the probability that any two randomly selected points will be within certain distance from each other. This correlation dimension will observe how this probability changes as the distance decreases. This correlation dimension is calculated from the correlation integral [13], which is estimated by the following equation.

$$C_{C2}(r) = \lim_{N \to \infty} \frac{1}{N} \sum_{i=1}^{N} \frac{1}{N} \sum_{j=1}^{N} \Theta(r - |\vec{x}_i - \vec{x}_j|) \quad (2)$$

where, N is the number of points, $\vec{x}$ is a point of space, r is a threshold value which is decreased in each step. $\Theta(x)$ is the Heaviside step function, and is defined in equation (3).

$$\Theta(x) = \begin{cases} 0 & \text{when } x \le 0 \\ 1 & \text{when } x > 0 \end{cases} \quad (3)$$

Figure 3 shows parts of the calculation steps for correlation integral. The initial data point is denoted by a solid dot; a star denotes a current point of interest at a particular iteration, and a solid square denotes the data point that has already been processed. Step 0 illustrates the initial state of the data. In step 1, one data point is selected (indicated by a star), and we count the total number of objects that fall within the distance r (in step 1, we get 3 as a result). The processed data point is marked, and we repeat with different data point. In step 2, we get 2 as a result, and 0 in step n.

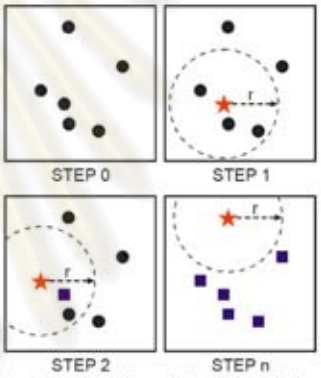


Figure 3. An example of the correlation integral calculation steps, with a threshold distance r.

After we obtain the correlation integral, $C_{C2}(r)$, we repeat with different r values, i.e. half the previous value. Finally, we will have a log-log plot between r values in the x-axis and the correlation integral, $C_{C2}(r)$, in the y-axis. The slope of the best-fitted line is the correlation dimension (D_{C2}), defined by equation (4) below.

$$D_{C2} = \lim_{r \to 0} \frac{\log C_{C2}(r)}{\log r} \quad (4)$$

As mentioned earlier, many other types of fractal dimension exist, but this correlation dimension has the advantage that it is quite simple and easy to implement, while giving the results in accordance with other dimension calculations.

4. OUR PROPOSED WORK

In this section, we will introduce our fractal representation for time series data. In particular, we utilize both the compass dimension and correlation dimension, and obtain 3 real values that nicely represent the self-similarity within the data; one from compass dimension, one from our modified compass dimension, and another one from correlation dimension.

4.1. Equi-Length Compass Dimension

This equi-length compass dimension is identical to the method described in Section 3.1, except that the starting and ending points are not the same since time series data do not form a closed figure. Figure 4 shows an example of the equi-length compass dimension steps, where the ruler's length is set to 15 units, and the total estimated length of this time series is 150 units in length.

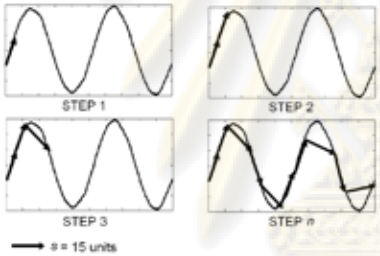


Figure 4. An equi-length compass dimension, where the length of the line segment, *s*, is set to 15 units.

We first assign an initial length of the line segment, *s*, and start fitting the line from the starting position of the time series, traversing until the ending position. The lengths of all the line segments are summed up for this particular *s* value. Subsequent iterations are done in a similar fashion, with decreasing *s* value in half. In the end, we obtain equi-length compass dimension from the slope of the best-fitted line of the points in the log-log scale, as described in Section 3.1.

4.2. Equi-Width Compass Dimension

We slightly modify the original compass dimension, which has shown to work especially well for time series data. Since time series data is plotted on a time axis (*x*-axis), we can utilize this fact and divide time series into equal time slices. This way, instead of having line segments with equal length, we will have variable line segment lengths. Each iteration will have the time slice duration reduced by half. Figure 5 illustrates our Equi-Width Compass dimension calculation.

From Figure 5, we first assign an initial time slice duration, and then try to fit a line to each time series segment from the beginning until the end of the time series. The lengths of all the line segments are summed up, and the next iteration begins with time slice duration reduced by half. After the time slice is reduced down to only 1 data point, the algorithm terminates. Finally, the equi-width compass dimension is determined from the slope of the best-fitted line of the points in the log-log scale, similar to the equi-length version.

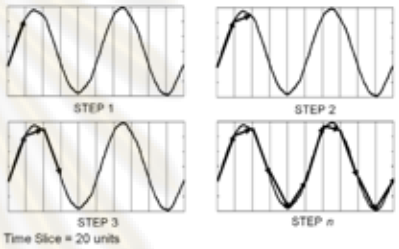


Figure 5. An equi-width compass dimension, where the time slice is set to 20 units.

4.3. Correlation Dimension

For correlation dimension, we simply treat time series data as a sequence of data points, and then apply the correlation dimension algorithm as described in Section 3.2. We first assign an initial *r* value and seek the correlation within the data from the correlation integral. Subsequent iterations are repeated with *r* values reduced by half. Finally, the log-log plot between *r* values and the correlation integrals is obtained, and the final correlation dimension is determined by the slope of the best-fitted line.

At this point, we obtain 3 real values that represent each time series data in a unique way, which also give a new identity to the time series data, with a massive reduction to the original data. We will demonstrate the utility and effectiveness of our proposed fractal representation in the next section.

5. EXPERIMENTS

We evaluate our proposed method by experimenting on clustering problems, testing on both accuracy and speed. However, to make the evaluation as fairest as possible, we compare the accuracy of our proposed method with the raw data (without any dimensionality reduction), using the ubiquitous Euclidean distance

metric, and the Dynamic Time Warping (DTW) distance measure that has proven to work exceptionally well for relatively short time series data. To evaluate the efficiency, we compare the clustering speed with the Compression-Based Dissimilarity Measure (CDM) [2], which has been demonstrate to work well with longer time series data, while capturing the global structure of the data.

5.1. Datasets

Since long time series data with labels are difficult to obtain, we assemble and build our datasets for experiments from the benchmark datasets obtained from the UCR Time Series Data Mining Archive [17], all of which are z-normalized, as described below.

- The first dataset is obtained directly from the archive; it is pairwise time series data, consisting of 36 time series (with 18 classes) of 1,000 data points in length. Adjacent two items belong to the same class, e.g., 1 and 2, 3 and 4, 5 and 6, etc. This dataset contains a mixture of various types of time series data, some are random walk, some are heart signals, some are synthetic data, etc., as shown in Figure 6.
- The second dataset is a 13-class dataset with 65 time series sequences, each with 2,000 data points in length. Each class consists of 5 sequences. We pick various datasets from the archive at random, and if the data sequence is too short, we concatenate the data together until we obtain the length of 2,000 data points. This dataset also contains a good mixture of different types of signals (posture, stemgen, starlightcurves, tickwise, mallet, kung fu, etc.)

Note that all experiments are performed in Matlab®, so the reported running time would be much faster if implemented in other lower-level programming languages, such as C++, Java, etc.

5.2. Hierarchical Clustering

We test our proposed method on a hierarchical clustering problem, comparing with Euclidean distance, DTW distance, and CDM. The result of our method is shown as a dendrogram in Figure 6 and the summary of the accuracy and running time of other algorithms are shown in Table 1. Due to the nature of the dendrogram and its visualization limitation, its interpretation is best with relatively small datasets; we therefore provide the resulting dendrogram on the first dataset only. According to Table 1, our proposed fractal representation can correctly cluster all the data, while Euclidean and DTW perform poorly. Even though CDM is able to recognize global structure of the data and also gets 100% accuracy, our proposed method is running about 3 orders of magnitude faster since all the fractal dimension calculation can be done offline, while most parts of the CDM algorithm cannot.

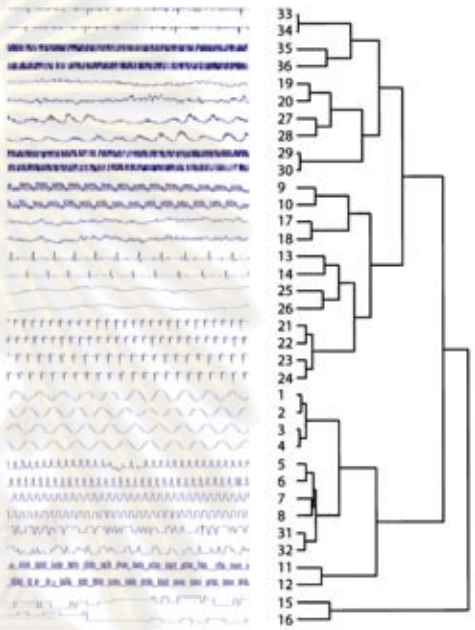


Figure 6. Clustering result using the proposed fractal representation. All 18 pairs of data are clustered correctly.

Table 1. Clustering result of the first dataset, reporting accuracy and running time. Our proposed method outperforms others in both accuracy and time complexity metrics.

Algorithms	Accuracy (%)	Time (sec)
Euclidean	22.22	0.132
DTW	66.67	191.013
CDM	**100**	48.236
Fractal Rep.	**100**	**0.051**

5.3. Partitional Clustering

We also test our proposed method on partitional clustering problems, using 1-nearest-neighbor method. We report the accuracy and running time, comparing our algorithm to Euclidean distance, DTW distance, and CDM. Results of all 2 datasets are shown in Table 2 and Table 3.

Table 2. Clustering results for the first dataset.

Algorithms	Accuracy (%)	Time (sec)
Euclidean	30.56	0.204
DTW	63.88	184.878
CDM	63.88	44.862
Fractal Rep.	**91.67**	**0.018**

Table 3. Clustering results for the second dataset.

Algorithms	Accuracy (%)	Time (sec)
Euclidean	44.62	0.665
DTW	63.08	232.258
CDM	**95.38**	187.508
Fractal Rep.	**95.38**	**0.057**

Our fractal representation performs well, especially in the running time. We could achieve as high as 4 orders of magnitude speedup, comparing to DTW. In a few cases, we get the same accuracy as CDM, but our running time is about 3,000 times faster. We would like to reemphasize that the reason why most DTW experiments here do not perform very well is the fact that DTW is generally good for shape recognition for relatively short time series. It can only capture the local structure of the data, not the global one.

6. CONCLUSION

In this paper, we introduce a novel fractal representation that has been demonstrated to be both effective and efficient for time series data. Their main advantage is that this representation can be done offline, providing great speedup for the mining tasks since each raw time series data is reduced down to only 3 numbers. In addition, various fractal dimensions are proven to be effective in capturing the overall structure of the time series data. For our future work, it would be ideal if we could develop a good representation that uses only one number to describe each time series, which would provide an ideal speedup to any data mining tasks.

7. REFFERENCES

[1] C. A. Ratanamahatana, and E. Keogh, "Three Myths about Dynamic Time Warping", *in proceedings of SIAM International Conference on Data Mining (SDM '05)* Newport Beach, CA, 2005, pp. 506-510.

[2] E. Keogh, S. Lonardi, and C. A. Ratanamahatana, "Towards Parameter-Free Data Mining", *in Proc. of SIGKDD*, 2004.

[3] B.Yi, and C.Faloutsos, "Fast Time Sequence Indexing for Arbitary LP-Norms", *in VLDB International Conference* Cairo, Egypt, September 2000.

[4] J. Lin, E. Keogh, S. Lonardi, B. Chiu, "A Symbolic Representation of Time Series, with Implications for Streaming Algorithms", *in proceedings of 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, 2003, pp. 460-469.

[5] E. Keogh, J. Lin, A. Fu, "HOT SAX: Finding the Most Unusual Time Series Subsequence: Algorithms and Applications", *in Proc. of the 5th IEEE International Conference on Data Mining (ICDM 2005)*, 2005.

[6] A. Bagnall, and G. Janacek, "Clustering Time Series with Clipped Data", *Machine Learning Journal,* pp. 151-178, 2005.

[7] M. Vlachos, P.Yu, V. Castelli, "On Periodicity Detection and Structural Periodic Similarity", *in Proc. of SIAM International Conf. on Data Mining (SDM)* Newport Beach, CA, 2005.

[8] D. Barbará, and P. Chen, "Using the Fractal Dimension to Cluster Datasets", *in Proceedings of 2000 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* Boston, MA, August 2000, pp. 260-264.

[9] A. Gionis, A. Hinneburg, S. Papadimitriou and P. Tsaparas, "Dimension Induced Clustering", *in Proceedings of LWA'2005*, 2005, pp. 109-110.

[10] H. Xiao, W. Zhi-Zhong, R. Xiao-mei, "Classification of surface EMG signal with fractal dimension", *Journal of Zhejiang University SCIENCE (JZUS),* May 30, 2005.

[11] H.O. Peitgen, H. Jurgens, D. Saupe, *Chaos and Fractals New Frontiers of Science*, 2003.

[12] J. O. Grabbe, "Chaos & Fractals in Financial Markets, Part 1-8", *in Chaos & Fractals in Financial Markets*, 1999-2003.

[13] P. Grassberger, and I. Procaccia, "Measuring the strangeness of strange attractors", *in Physica D*, 1983, pp. 189-208.

[14] A. Prokoph, "Fractal, multifractal and sliding window correlation dimension analysis of sedimentary time series", *in Computers & Geosciences 25*. vol. 25: Pergamon Press, Inc. Tarrytown, NY, USA 1999.

[15] A. Guerrero, L.A. Smith, "Towards coherent estimation of correlation dimension", *in Physics Letters A 318*, 2003.

[16] Y. Ashkenazy, "The use of generalized information dimension in measuring fractal dimension of time series", *in Physica A 271* 1999, pp. 427–447.

[17] E. Keogh, "The UCR Time Series Data Mining Archive", *in* http://www.cs.ucr.edu/~eamonn/TSDMA/datasets.html Riverside CA. University of California, Computer Science & Engineering Department, 2006.

ภาคผนวก ค

บทความทางวิชาการเรื่อง “A Novel Fractal Representation for Dimensionality Reduction of Large Time Series Data” โดย พจน์ สัจจิพานนท์ และโชติรัตน์ รัตนามหัทธนะ ในงานประชุมวิชาการนานาชาติ ครั้งที่ 13 “The Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)” ซึ่งจัดขึ้น ณ กรุงเทพมหานคร ประเทศไทย ระหว่างวันที่ 27 ถึง 30 เมษายน 2552

A Novel Fractal Representation for Dimensionality Reduction of Large Time Series Data

Poat Sajjipanon and Chotirat Ann Ratanamahatana

Department of Computer Engineering, Chulalongkorn University
254 Phyathai Road, Patumwan, Bangkok Thailand, 10330
{g50psj,ann}@cp.eng.chula.ac.th

Abstract. Recent research has attempted to speed up time series data mining tasks which focus on dimensionality reduction, indexing, and lower bounding function, among many others. For large time series data, current dimensionality reduction techniques cannot reduce the total dimensions of time series data by a large margin without losing their global characteristics. In this paper, we introduce a novel Fractal Representation which uses merely three real values to represent a whole time series data sequence. Moreover, our proposed representation can be efficiently used under Euclidean distance. We demonstrate effectiveness and utility of our novel Fractal Representation on classification problems and our proposed method outperforms existing methods in terms of speed performance and accuracy. Our results reconfirm that this representation can effectively represent global characteristics of the data, especially in larger time series data.

Keywords: Time Series Mining, Fractal Representation, Fractal Dimension.

1 Introduction

The ultimate goal to improve time series mining tasks is both speedup and increasing accuracy. Unfortunately, there is a tradeoff between accuracy and computational cost; for example, Euclidean distance is very fast, but the accuracy may be low when comparing with other similarity measures. On the other hand, Dynamic Time Warping (DTW) distance measure [1] is relatively slow, but it has been widely accepted as one of the best distance measures for time series data, particularly for relative short time series. Although this method consumes a large amount of computation time, time complexity can be reduced by combining DTW distance with other algorithms, especially bounding function [2]. High computational cost for DTW distance is clearly inappropriate for large time series data; therefore, a recent Compression-Based Dissimilarity Measure [3] has been proposed for similarity measure between two time series data based on file compression, which focuses mainly on large time series data. However, CDM is I/O bounded, which affects the computational time.

Alternatively, the speedup has been improved through indexing, dimensionality reduction/data representation, among many others. For large time series data, the

T. Theeramunkong et al. (Eds.): PAKDD 2009, LNAI 5476, pp. 989–996, 2009.

current dimensionality reduction techniques cannot reduce the dimension by a large margin without losing their global characteristics. Most of these methods are used for pruning off time series data that are not the answer, such as Symbolic Aggregate Approximation (SAX) [4], and clipped data representation [5].

The ideal case for dimensionality reduction is to represent time series data with only one dimension, while preserving original data's characteristics. Once these resulted dimensions are sorted, we can compare two time series using constant time. Consequently, this research work focuses on reducing the number of dimensions of large time series data.

In this work, we introduce a novel representation, Fractal Representation, for large time series data based on fractal dimension, using merely three real values to represent the internal structure of a time series data. In our experiment, Fractal Representation can accomplish a large speedup in a wide range of data mining tasks. In addition, we demonstrate the superiority of our proposed method over the well-known distance measures, i.e., Euclidean distance, DTW distance, and CDM, on classification problems. The results have demonstrated that our representation can accurately classify the data, especially in longer time series, since Fractal Representation can effectively represent global characteristics of the data.

The rest of this paper is organized as follows. In section 2, we review related work on time series mining and dimensionality reduction techniques. Section 3 describes background knowledge of the fractal dimension. After that, our proposed Fractal Representation is introduced in section 4. Section 5 contains our experimental evaluation and discussion, and, finally, section 6 draws some conclusions and gives suggestions for future work.

2 Background and Related Work

Generally, the dimension of any data, such as a point, a line, a plane, a cube, is represented by integer value. This dimension is called topological dimension. However, the dimension, i.e., Fractal dimension [8], can be a real value which considers the local structure of the data by seeking self similarity within. Both fractal theory and fractal dimension are based on chaos theory, which is a non linear and deterministic dynamic system. Fractal theory's key characteristic is the self similarity; the shapes of the data still look similar at any scaling levels. In the other words, a fractal dimension is the dimension of an irregular geometric object with an infinite nesting of structure at all scales. To calculate the fractal dimension, many techniques, e.g., compass dimension, box-counting dimension, information dimension, and correlation dimension, have been proposed. Generally, framework of fractal dimension is commonly calculated iteratively in different levels of local structure, as shown in Fig. 1.

Concept of fractal dimension is not commonly applied for the dimensionality reduction on data mining tasks. More specifically, fractal dimension is typically used for determining the dimension of a picture. Some research applies fractal dimension for data points and time series data, i.e., Barbara and Chen [6] applied fractal dimension for data points classification, and Xiao et al. [7] used fractal dimension to classify EMG signals. However, the results of fractal dimension still cannot distinguish the classes very effectively.

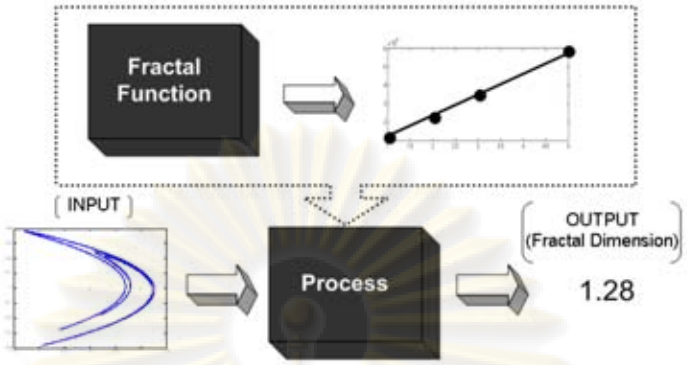


Fig. 1. General framework of fractal dimension calculation

2.1 Compass Dimension

The idea of compass dimension [8] is to estimate the dimension by seeking self similarity using a compass segment or a ruler to travel along the perimeter of input data based on distance. For example, in the first step, the compass segment with length s is assigned, and this compass segment traverses from the starting position to the ending position of the input data. The overall image perimeter length is then estimated from the compass function. Next, a new compass segment's length is assigned as half of the previous length, and the compass function calculation is repeated. Each coordinate of the compass segment and the compass function output is plotted on log-log scale, and the slope of the best-fitted line for these points is the compass dimension (D_c), shown in equation (1).

$$D_C = \lim_{s \to 0} \frac{\ln N(s)}{\ln(1/s)}. \tag{1}$$

Where s is the compass length, and $N(s)$ is the sum of the compass segments' lengths s from any starting position to the ending position.

2.2 Correlation Dimension

Correlation dimension [8, 9], proposed by Grassberger and Procaccia, measures the self-similarity by calculating correlation among all data points in space. Correlation of the data points, also called Correlation Integral, is measured as distances between a fixed point and every other point. When the distance is less than a threshold distance r, it is counted as '1'. More specifically, Correlation Integral is defined as the total number of a pair of data points, which has distance less than the threshold distance r. This correlation integral is estimated by the following equation.

$$C_{CI}(r) = \lim_{N \to \infty} \frac{1}{N} \sum_{i=1}^{N} \frac{1}{N} \sum_{j=1}^{N} \Theta(r - |\vec{x}_i - \vec{x}_j|). \tag{2}$$

where N is the total number of points, $\vec{x}$ is a data point of space, and r is a threshold distance. $\Theta(x)$ is the Heaviside step function, and is defined in equation (3).

$$\Theta(x) = \begin{cases} 0 & \text{when } x \leq 0 \\ 1 & \text{when } x > 0 \end{cases} \tag{3}$$

To calculate the correlation dimension, the initial threshold distance r is first assigned to an arbitrary value. Next, the correlation integral $C_{CI}(r)$ is calculated, and then r is set to half of its previous value. Correlation integral is repeatedly calculated. Finally, each coordinate of correlation integral $C_{CI}(r)$ (y-axis) and scale factor r (x-axis) is plotted on a log-log diagram. The slope of the best-fitted line is the correlation dimension (D_{C2}), defined by an equation below.

$$D_{C2} = \lim_{r \to 0} \frac{\ln C_{CI}(r)}{\ln r}. \tag{4}$$

3 Our Proposed Method

In this section, we introduce a novel Fractal Representation which uses merely three real values to represent each time series sequence efficiently, according to our proposed fractal dimension approaches below.

Although fractal dimension can significantly reduce dimensions of time series, a bad set of parameters does affect the accuracy. Hence, our proposed work also includes automatic parameter tuning, which is demonstrated to achieve higher accuracy than those of existing methods, especially for z-normalized large time series data.

3.1 Equi-width Compass Dimension

Equi-width compass dimension is extended from the original compass dimension to be applied specifically to time series data. We divide a sequence on x-axis into equal widths, called a time slice, and then traverse a line within each time slice. As shown in Fig. 2, a time series length is 200 units, and the time slice is 20 units; therefore, time series is equally divided to 10 periods, and then travel along from period 1 to 10. The compass function is based on y-axis values only, and the time slice in each iteration is assigned according to equation (5).

$$Time\_Slice(k) = ceil\left(\frac{\log_2(L)}{2^k}\right) \tag{5}$$

where $Time\_Slice(k)$ is a period assigned in each iteration k, k is the number of the current iteration, and L is length of time series data.

From equation (5), in iteration 1 ($k = 1$), the time slice is initialized by $\log_2$ of time series length because large time slice leads to an inaccurate approximation. The time slice is divided into half, until the time slice equals 1; then it terminates. Finally, values of compass function and time slice are then plotted on graph, and then the equi-width compass dimension is determined by the slope of the best-fitted line of the points in the log-log scale, shown in equation (1).

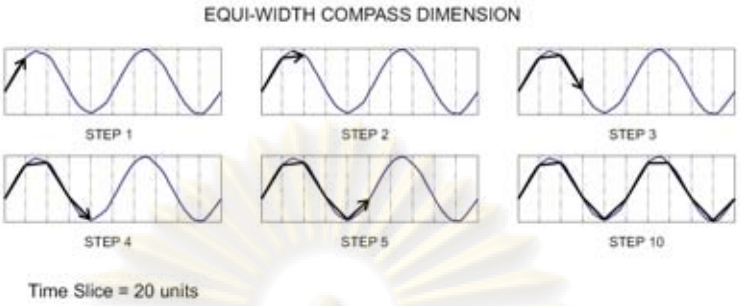


Fig. 2. An example of equi-width compass dimension with a time slice of 20 units

3.2 Equi-length Compass Dimension

Equi-length compass dimension resembles the original compass dimension. A line with a scaling factor s is the compass segment which uses a fixed length to travel around time series. Total summation of all the compass segment length is a compass function value of each iteration, as shown in Fig. 3. The length of compass segment for each iteration $s(k)$ of equi-length compass dimension is shown in equation (6).

$$s(k) = \left| \frac{\max(ts) - \min(ts)}{2^k} \right| \tag{6}$$

where $s(k)$ is the length of compass segment for each iteration k, $\max(ts)$ is a maximum value of time series data, and $\min(ts)$ is a minimum value of time series data.

From equation (6), we use the difference between $\max(ts)$ and $\min(ts)$ value to initialize the compass segment. For each iteration, $s(k)$ is divided into half of previous length. When every data point of the time slice are traversed in any particular $s(k)$, it terminates. We obtain equi-length compass dimension from the slope of the best-fitted line of the points in the log-log scale, as shown in equation (1).

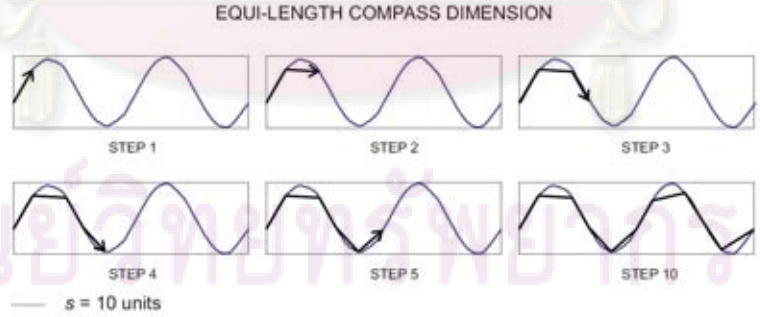


Fig. 3. An example of equi-length compass dimension with a scaling factor s

3.3 Correlation Dimension

Correlation dimension for time series data resembles the original method. We use correlation integral, as in equation (2), to calculate distance on a y-axis between a

fixed point and every other points of time series data. When distance is less than threshold distance r, correlation integral accumulates number of correlation in each point, and then the threshold distance r is reduced to half. Until correlation integral is zero or invariant values, it terminates. Finally, the log-log plot between $r(k)$ values and the correlation integrals is obtained, and correlation dimension is determined by the slope of the best-fitted line. The threshold distance r for each iteration $r(k)$ of correlation dimension is shown in equation (7).

$$r(k) = \left| \frac{\max(ts) - \min(ts)}{2^k} \right| \tag{7}$$

where $r(k)$ is a threshold distance of each iteration k, $\max(ts)$ is maximum value of a time series data, and $\min(ts)$ is minimum value of a time series data.

Finally, our proposed method are three real values to represent a time series called Fractal Representation, which also give a new identity to the time series data, with a massive reduction. Moreover, our proposed method is automatic parameter.

4 Experiment

In this section, we demonstrate efficiency of our Fractal Representation on classification problem in terms of accuracy and computation time. We compare our proposed method with Euclidean Distance, DTW distance, and CDM.

4.1 Datasets

In our experiment, we build our datasets by generating time series data from the benchmark datasets of the UCR Time Series Data Mining Archive [10]. We test on both long and short time series data. Note that all datasets are z-normalized. A total of eight datasets are described in Table 1.

Table 1. Detail of each dataset of time series in our experiment

Datasets	Number of Data	Data Points	Number of Class	Typical Data
Dataset 1	36	1,000	18	Evaporator, furnace, balloon, reel, dryer, etc.
Dataset 2	70	1,000	7	Aspect ratio, cement, field, hooked, etc.
Dataset 3	80	2,000	8	Power demand, standing, wall, EEG, etc.
Dataset 4	65	1,000	13	ERP, koski ecg, kung fu, 4 class of ECG etc.
Dataset 5	44	2,000	11	Power data, nprs, stemgen, random walk, etc.
Dataset 6	130	3,000	13	CinC, symbols, faces, random walk, words, etc.
Dataset 7	70	128	7	Adiac, CBF, face-all, wafer, etc.
Dataset 8	60	420	6	Beef, fish, lighting-2, Olive-oil, OSU leaf, etc.

4.2 Experiment

We evaluate our proposed method, Fractal Representation, on classification problem, using 1-nearest-neighbor classifier with leaving-one-out technique in terms of both

Table 2. Classification results for comparing Fractal Representation with existing methods

Datasets	Euclidean Distance		DTW		CDM		Fractal Representation	
	Accuracy (%)	Time (sec)	Accuracy (%)	Time (sec)	Accuracy (%)	Time (sec)	Accuracy (%)	Time (sec)
Dataset 1	30.56	0.182	63.89	140.5	33.33	46.928	**75**	**0.02**
Dataset 2	75.72	0.46	88.57	685.71	55.72	156.12	**95.72**	**0.042**
Dataset 3	83.75	1.48	**90**	3742.29	81.25	318.18	88.75	**0.041**
Dataset 4	21.54	0.38	**83.08**	570.46	46.15	132.25	73.85	**0.055**
Dataset 5	31.82	0.39	70.45	1052.1	65.91	95.75	**81.82**	**0.032**
Dataset 6	66.15	3.85	80	18766.1	37.69	1111.7	**80.77**	**0.979**
Dataset 7	85.71	0.13	**100**	14.087	14.28	96.05	84.29	**0.0636**
Dataset 8	88.33	0.189	**96.67**	86.30	26.66	88.81	75	**0.0659**

accuracy and running time, and then compare our algorithm with Euclidean distance, DTW distance, and CDM. Results from 8 datasets are shown in Table 2.

From Table 2, Fractal Representation outperforms other methods in all datasets in terms of the running time. In some cases, we could achieve as much as 4 orders of speedup magnitude in wall clock time, especially when comparing with DTW. For large time series (datasets 1-6), our Fractal Representation performs well in many cases in terms of the accuracy. In dataset 3, we get the accuracy less than DTW distance, but our running time is almost 90,000 times faster. For dataset 4, our accuracy is not high, comparing to other datasets because this dataset consists of very similar ECG signals (4 classes); therefore, Fractal Representation may not differentiate those global structures very well. In general, DTW outperforms our Fractal Representation when dealing with short time series data. However, the results of Fractal Representation on both accuracy and time of short time series perform well in all cases. In addition, CDM does not seem to work very well here since CDM generally works well with unnormalized data, whereas every dataset here is z-normalized. Amplitude of each time series therefore is highly similar, and its compressions have similar file size which directly affects and degrades the classification results.

5 Conclusion

Our Fractal Representation is a novel dimensionality reduction technique that does not need extra pruning of unnecessary data, providing great speedup for the mining tasks because of the large reduction in the number of dimensions down to only three real values. Our main advantage is that this representation can be done offline. The Fractal Representation has demonstrated to be both effective and efficient for time series data under automatic parameter selection. For our future work, if we could develop representation which uses fewer values, ideally one. It will provide an ideal speedup to any data mining tasks.

Acknowledgement

This work is partially supported by the Thailand Research Fund (Grant No. MRG5080246).

References

1. Ratanamahatana, C.A., Keogh, E.: Three Myths about Dynamic Time Warping. In: Proceedings of SIAM International Conference on Data Mining (SDM 2005), pp. 506–510 (2005)
2. Keogh, E., Ratanamahatana, C.A.: Exact Indexing of Dynamic Time Warping. In: Knowledge and Information Systems (KAIS), pp. 358–386 (2005)
3. Keogh, E., Lonardi, S., Ratanamahatana, C.A.: Towards Parameter-Free Data Mining. In: Proceedings of 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2004)
4. Lin, J., Keogh, E., Lonardi, S., Chiu, B.: A Symbolic Representation of Time Series, with Lmplications for Streaming Algorithms. In: Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, pp. 460-469 (2003)
5. Bagnall, A., Janacek, G.: Clustering Time Series with Clipped Data. Machine Learning Journal, 151–178 (2005)
6. Barbará, D., Chen, P.: Using the Fractal Dimension to Cluster Datasets. In: Proceedings of 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Boston, MA, pp. 260–264 (August 2000)
7. Xiao, H., Zhi-Zhong, W., Xiao-mei, R.: Classification of surface EMG signal with fractal dimension. Journal of Zhejiang University SCIENCE (JZUS) (May 30, 2005)
8. Peitgen, H.O., Jurgens, H., Saupe, D.: Chaos and Fractals New Frontiers of Science (2003)
9. Grassberger, P., Procaccia, I.: Measuring the strangeness of strange attractors. Physica D
10. Keogh, E., Xi, X., Wei, L., Ratanamahatana, C.A.: The UCR Time Series Data Mining Archive, University of California, Computer Science & Engineering Department, Riverside CA (2006), `http://www.cs.ucr.edu/~eamonn/TSDMA/datasets.html`

ประวัติผู้เขียนวิทยานิพนธ์

นายพจน์ สัจจิพานนท์ เกิดวันที่ 12 พฤจิกายน พ.ศ. 2528 สำเร็จการศึกษาระดับมัธยมศึกษาที่โรงเรียนขอนแก่นวิทยายน จากนั้นจึงเข้าศึกษาต่อที่คณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น ในปีการศึกษา 2546 และในปีการศึกษา 2549 จึงสำเร็จการศึกษาปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ และเข้าศึกษาในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ ที่ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ปีการศึกษา 2550