การแทนอนุกรมเวลาเพื่อการจัดกลุ่มที่มีประสิทธิภาพ

นางทิพยา ถินสูงเนิน

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์
มหาวิทยาลัยเทคโนโลยีสุรนารี
ปีการศึกษา 2560

TIME SERIES REPRESENTATION FOR EFFICIENT CLUSTERING

Tippaya Thinsungnoen

A Thesis Submitted in Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy in Computer Engineering

Suranaree University of Technology

Academic Year 2017

การแทนอนุกรมเวลาเพื่อการจัดกลุ่มที่มีประสิทธิภาพ

มหาวิทยาลัยเทคโนโลยีสุรนารี อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาดุษฎีบัณฑิต

คณะกรรมการสอบวิทยานิพนธ์

(รศ. ดร.กิตติศักดิ์ เกิดประสพ)
ประธานกรรมการ

(รศ. ดร.นิตยา เกิดประสพ)
กรรมการ (อาจารย์ที่ปรึกษาวิทยานิพนธ์)

(ผศ. ดร.ปรเมศวร์ ห่อแก้ว)
กรรมการ

(ผศ. ดร.สายสุนีย์ จับโจร)
กรรมการ

(ผศ. ดร.ศุภกฤษฎิ์ นิวัฒนากูล)
กรรมการ

(ศ. ดร.สันติ แม้นศิริ)
รองอธิการบดีฝ่ายวิชาการและพัฒนาความเป็นสากล

(รศ. ร.อ. ดร.กนต์ธร ชำนิประศาสน์)
คณบดีสำนักวิชาวิศวกรรมศาสตร์

ทิพยา ถินสูงเนิน : การแทนอนุกรมเวลาเพื่อการจัดกลุ่มที่มีประสิทธิภาพ (TIME SERIES REPRESENTATION FOR EFFICIENT CLUSTERING) อาจารย์ที่ปรึกษา : รองศาสตราจารย์ ดร.นิตยา เกิดประสพ, 132 หน้า.

ข้อมูลคลื่นไฟฟ้าหัวใจและคลื่นไฟฟ้าสมองที่ได้จากการตรวจจับกระแสไฟฟ้าที่ออกมาจากหัวใจและสมองมีความรู้ซ่อนอยู่มากมายสามารถนำมาใช้เพื่อช่วยในการวินิจฉัยโรคหรือเตือนภัยก่อนโรคร้ายจะมาถึงได้ ดังนั้นหากสามารถจำแนกกลุ่มข้อมูลคลื่นไฟฟ้าหัวใจและคลื่นไฟฟ้าสมองเหล่านี้ได้ชัดเจนและแม่นยำขึ้นจะเป็นประโยชน์อย่างมากในทางการแพทย์ ข้อมูลคลื่นไฟฟ้าเหล่านี้จัดเป็นข้อมูลอนุกรมเวลาที่มักมีขนาดใหญ่ มิติสูงและซับซ้อน ซึ่งเป็นความท้าทายสำหรับการเรียนรู้ของเครื่องจักร หัวใจของความสำเร็จอย่างหนึ่งสำหรับการเรียนรู้ข้อมูลประเภทนี้คือการแทนข้อมูลที่ดี ดังนั้นงานวิจัยนี้จึงได้นำเสนอวิธีการแทนข้อมูลอนุกรมเวลาเพื่อให้เกิดประสิทธิภาพสูงขึ้นในการจัดกลุ่มสำหรับข้อมูลคลื่นไฟฟ้าหัวใจและคลื่นไฟฟ้าสมอง โดยอาศัยโครงข่ายการเข้ารหัสอัตโนมัติเชิงลึก (ดีเอเอ็น) ซึ่งเป็นวิธีที่นำเอาหลักการของ โบลทซ์มันน์แมชชีนเชิงจำกัดมาใช้ร่วมกับเทคนิคที่มีประสิทธิภาพสูงในด้านการแทนข้อมูลอย่างออโตเอ็นโคดเดอร์ งานวิจัยนี้ใช้อัลกอริทึมเชิงพันธุกรรมเพื่อหาโครงข่ายที่ดีที่สุดให้กับ ดีเอเอ็น เรียกว่า “ดีเอเอ็นจีเอ” งานวิจัยนี้จัดกลุ่มอนุกรมเวลาด้วยอัลกอริทึมพีดีซีและเค-มีนส์ โดยเปรียบเทียบประสิทธิภาพกับข้อมูลดั้งเดิมและเทคนิคการแทนข้อมูลด้วยวิธีอื่น ด้วยมาตรวัดค่าความถูกต้อง ค่าพิวริตี เวลาในการประมวลผลและประเมินค่าความเหมาะสมของกลุ่มด้วย ค่าซิลลูเอ็ต และค่าผลรวมความผิดพลาด ผลการวิจัยพบว่า ดีเอเอ็นจีเอสามารถผลิตตัวแทนอนุกรมเวลาที่ช่วยเพิ่มประสิทธิภาพการจัดกลุ่มให้ดีขึ้นได้ทั้งในอัลกอริทึมพีดีซีและเค-มีนส์ เป็นตัวแทนอนุกรมเวลามีประสิทธิภาพดีกว่าข้อมูลดั้งเดิม ดีกว่าตัวแทนข้อมูลจากเทคนิคอื่น แต่มีข้อจำกัดคือใช้เวลาในการประมวลผลสูงกว่าวิธีการอื่น เมื่อพิจารณาข้อมูลคลื่นไฟฟ้าหัวใจ ผลการวิจัยพบว่า ตัวแทนอนุกรมเวลาที่ได้ให้ประสิทธิภาพการจัดกลุ่มเพิ่มขึ้นทั้งอัลกอริทึมพีดีซีและเค-มีนส์ โดยเฉพาะในอัลกอริทึมพีดีซีให้ค่าความถูกต้อง และค่าพิวริตี เพิ่มขึ้นถึง 30% และ 23% ตามลำดับ นอกจากนี้ซิลลูเอ็ตและผลรวมความผิดพลาดยังบ่งชี้ให้เห็นว่าการจัดกลุ่มมีความเหมาะสมตามธรรมชาติของข้อมูลและกลุ่มที่แท้จริงอีกด้วย สำหรับข้อมูลคลื่นไฟฟ้าสมอง พบว่า ตัวแทนข้อมูลที่ได้ เหมาะกับการจัดกลุ่มด้วยอัลกอริทึมพีดีซีโดยสามารถเพิ่มประสิทธิภาพได้สูงมากคือให้ค่าความถูกต้อง และค่าพิวริตี เพิ่มขึ้นถึง 31% และ 61% ตามลำดับ

สาขาวิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนักศึกษา _____

ปีการศึกษา 2560 ลายมือชื่ออาจารย์ที่ปรึกษา _____

TIPPAYA THINSUNGNOEN : TIME SERIES REPRESENTATION FOR EFFICIENT CLUSTERING. THESIS ADVISOR : ASSOC. PROF. NITTAYA KERDPRASOP, Ph.D., 132 PP.

TIME SERIES ANALYSIS / DEEP LEARNING / RESTRICTED BOLTZMANN MACHINES / DEEP AUTOENCODER.

Electrocardiogram signals (ECGs) and electroencephalographic signals (EEGs) are time series detected from electrical flow of the heart and brain. Deep analysis of these data can reveal some hidden knowledge potentially useful for the accurate diagnosis or warning an early alarm for heart disease. Electrical signals are normally organized into time series data that are usually large, high dimensional, and complex in their components. Therefore, efficient analysis with the machine learning techniques is a challenging problem. One of the key successes for this kind of learning is to learn from the representative data that are carefully selected. In this research, we present a method for efficient casting of time series representatives that are to be used later for time series clustering for ECGs and EEGs. To find series representative, we propose to use Deep Autoencoder Networks (DANs), which is a technique based on Restricted Boltzmann Machines and Autoencoder. This research determines the appropriate network for DANs by using genetic algorithm called "DANGA". The signal representatives are then clustered using the PDC and k-Means algorithms. The clustering results obtained from our proposed method are compared against other time series representation techniques based on the cluster evaluation (accuracy), purity, processing time, silhouette, and the number of clusters considered from the sum of

square error (SSE). The experimental results show that our proposed method can cast for more appropriate time series representatives than others techniques with the longer processing time trade-off. The ECGs representatives yield the better performance on time series clustering with the 30% improvement in grouping accuracy and 23% increase in the purity metric. Furthermore, silhouette and SSE index indicate natural clusters. For EEGs clustering, the results yield the better performance with the 31% improvement in accuracy and 61% increase in purity.

School of Computer Engineering Student's Signature

Academic Year 2017 Advisor's Signature

กิตติกรรมประกาศ

วิทยานิพนธ์นี้สำเร็จลุล่วงด้วยดี เนื่องจากผู้วิจัยได้รับความกรุณาคำปรึกษา แนะนำ การสนับสนุน และความช่วยเหลืออย่างดียิ่ง ทั้งทางด้านวิชาการและการดำเนินงานวิจัยจากบุคลและกลุ่มบุคคลหลายท่าน ผู้วิจัยขอกราบขอบพระคุณอย่างสูง สำหรับผู้มีส่วนช่วยให้ผลงานนี้ลุล่วงด้วยดีอันได้แก่

รองศาสตราจารย์ ดร.นิตยา เกิดประสพ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่เมตตาให้การอบรม สั่งสอน ชี้แนะ ช่วยเหลือ ทั้งยังให้โอกาสในการศึกษาค้นคว้า ตลอดจนให้คำแนะนำในการเขียน และตรวจแก้ไขวิทยานิพนธ์จนเสร็จสมบูรณ์

รองศาสตราจารย์ ดร.กิตติศักดิ์ เกิดประสพ ประธานกรรมการ และคณะกรรมการทุกท่านที่กรุณาให้การแนะนำ คำปรึกษา ชี้แนะแนวทางการเขียน และช่วยตรวจทานเนื้อหาวิทยานิพนธ์จนเสร็จสมบูรณ์

มหาวิทยาลัยราชภัฏนครราชสีมาที่ได้ให้การสนับสนุนทุนการศึกษา และทุนอุดหนุนการนำเสนอและเผยแพร่ผลงานวิจัย

สถาบันวิจัยและพัฒนา มหาวิทยาลัยเทคโนโลยีสุรนารี ที่ให้ทุนสนับสนุนในการนำเสนอและเผยแพร่ผลงานวิจัย

ผู้บริหารและบุคลากรทุกท่านใน หลักสูตรวิทยาการคอมพิวเตอร์ หลักสูตรเทคโนโลยีสารสนเทศ และหลักสูตรระบบสารสนเทศเพื่อการจัดการ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏนครราชสีมา ที่ให้การสนับสนุนและให้โอกาสในการศึกษาค้นคว้าทางด้านวิชาการ

เพื่อนบัณฑิตศึกษา สาขาวิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเทคโนโลยีสุรนารี ทุกคนที่ให้คำปรึกษา กำลังใจ และแรงผลักดันในการทำงานจนวิทยานิพนธ์นี้สำเร็จลุล่วงด้วยดี

ท้ายนี้ ผู้วิจัยขอกราบขอบพระคุณบิดา มารดา ที่ให้การอุปการะอบรมเลี้ยงดู และครอบครัวที่ส่งเสริมการศึกษา และคอยให้กำลังใจ และคอยดูแลในยามที่เหน็ดเหนื่อยจนกระทั่งวิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงด้วยดี

ทิพยา ถินสูงเนิน

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

ตารางที่ **หน้า**

สารบัญรูป

รูปที่ **หน้า**

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มาของปัญหาการวิจัย

การจัดกลุ่มข้อมูล (Data Clustering) เป็นเทคนิคการค้นหาความรู้โดยอาศัยหลักการเรียนรู้ของเครื่องจักรแบบไม่มีผู้สอน (Unsupervised Learning) ซึ่งการจัดกลุ่มข้อมูลจะทำงานโดยไม่ต้องรู้กลุ่ม (คลาส) ที่แท้จริงก่อน (Shahbaba and Beheshti, 2014; Kwedlo, 2011; Rai and Singh, 2010; Roiger and Geatz, 2003; Jain et al., 1999; Kaufman and Rousseeuw, 1990) การจัดกลุ่มข้อมูลมีหลักการทำงานคือพยายามจัดข้อมูลที่มีลักษณะคล้ายคลึงกันมากให้อยู่กลุ่มเดียวกัน จัดข้อมูลที่มีความคล้ายคลึงกันน้อยหรือข้อมูลที่มีลักษณะแตกต่างกันมากให้อยู่คนละกลุ่ม (Han and Kamber, 2001) โดยอาศัยการวัดความคล้ายคลึงหรือการวัดระยะหรือค่าสัมประสิทธิ์อื่นเพื่อใช้เป็นตัวชี้วัดความคล้ายคลึงและความต่าง เทคนิคการจัดกลุ่มข้อมูลได้ถูกนำมาประยุกต์ใช้กับงานหลากหลายแขนง เช่น การจำแนกสภาพอากาศ (Weather Classification) การจำแนกเอกสาร (Document Classification) งานด้านชีวสารสนเทศศาสตร์ (Bioinformatics) พันธุศาสตร์ (Genetics) การประมวลผลภาพ (Image Processing) การรู้จำเสียง (Voice Recognition) และการวิจัยตลาด (Market Research) (Shahbaba and Beheshti, 2014) เป็นต้น นอกจากนี้การจัดกลุ่มข้อมูลยังใช้เพื่อสรุปผล อธิบายข้อมูล หรือใช้ในขั้นตอน Pre-Processing สำหรับการทำงานขั้นตอนอื่นในการทำเหมืองข้อมูลได้ (Aghabozorgi et al., 2015)

ในปัจจุบันซึ่งเป็นยุคที่พัฒนาการด้านการประมวลผลข้อมูลและการจัดเก็บมีศักยภาพสูงขึ้นจึงมักมีการจัดเก็บข้อมูลไว้เป็นช่วงระยะเวลายาวนานขึ้นพบได้จากการประยุกต์ใช้ในงานหลายแขนงที่เริ่มมีการจัดเก็บข้อมูลในรูปแบบอนุกรมเวลา (Time-series) ตัวอย่างเช่น ข้อมูลการขาย ข้อมูลราคาหุ้น ข้อมูลสภาพอากาศ การตรวจวัดทางชีวการแพทย์ เช่น ความดันโลหิต การวัดคลื่นไฟฟ้าหัวใจ หรือคลื่นไฟฟ้าสมอง ข้อมูลชีวภาพ เช่น ข้อมูลรูปภาพเพื่อการจดจำใบหน้า หรือการติดตามอนุภาคทางฟิสิกส์ เป็นต้น (Aghabozorgi et al., 2015) ลักษณะสำคัญของข้อมูลอนุกรมเวลา คือเป็นชุดข้อมูลที่ถูกจัดเก็บอย่างต่อเนื่องเป็นลำดับตามระยะเวลาในช่วงเวลาใดเวลาหนึ่ง (Brockwell and Davis, 2016; Aghabozorgi et al., 2015; Längkvist et al., 2014) สำหรับคลื่นไฟฟ้าหัวใจ (Electrocardiogram Signal: ECGs) และคลื่นไฟฟ้าสมอง (Electroencephalographic Signal:

EEGs) คือ คลื่นที่ได้จากการตรวจจับกระแสไฟฟ้าที่ออกมาจากหัวใจและสมองตามลำดับ สารสนเทศที่แฝงในคลื่นไฟฟ้าเหล่านี้สามารถนำมาใช้เพื่อช่วยในการวินิจฉัยโรคที่เกี่ยวข้องหรือเตือนภัยก่อนที่โรคร้ายจะมาถึงได้ ดังนั้นหากสามารถจำแนกกลุ่มข้อมูลคลื่นไฟฟ้าหัวใจเหล่านี้ได้ชัดเจนและแม่นยำขึ้นจะนับเป็นประโยชน์อย่างมากในทางการแพทย์ ทั้ง ECGs และ EEGs จัดเป็นข้อมูลอนุกรมเวลาที่มีขนาดใหญ่ มิติสูงและซับซ้อน ซึ่งเป็นความท้าทายสำหรับการเรียนรู้ของเครื่องจักร หัวใจของความสำเร็จอย่างหนึ่งสำหรับการเรียนรู้ของเครื่องจักรคือ การแทนข้อมูลอนุกรมเวลาที่ดี (Längkvist et al., 2014) จึงเกิดผลงานวิจัยขึ้นมากมายที่ศึกษาเกี่ยวกับการหาตัวแทนข้อมูลอนุกรมเวลา (Representation Method for Time-Series) เช่น วิธี Piecewise Aggregate Approximation (PAA) นำเสนอโดย Keogh และคณะ (2000) วิธีแบบ Adaptive Piecewise Constant Approximation (APCA) ถูกเสนอโดย Keogh และคณะ (Keogh et al., 2001) วิธีแบบ Symbolic Aggregate ApproXimation (SAX) โดย Lin และคณะ (2007) วิธีแบบ Discrete Fourier Transform (DFT) เสนอโดย Agrawal และคณะ (1993) และวิธีการแบบ Wavelet Transform (WT) หรือ Discrete Wavelet Transform (DWT) ในงานของ Graps (1995) และผลงานของ Burrus และคณะ (1997) เป็นต้น ซึ่งวิธีการที่กล่าวมานี้เป็นเพียงส่วนหนึ่งของการแทนอนุกรมเวลาที่มีประสิทธิภาพ

นอกเหนือจากนี้ยังมีเทคนิคการเรียนรู้ของเครื่องแบบอื่นที่มีประสิทธิภาพและเป็นที่ยอมรับอย่างแพร่หลายได้แก่ Deep Autoencoder Networks ซึ่งเป็นรูปแบบสถาปัตยกรรมโครงข่ายที่มีการผสานทำงานของ การเรียนรู้เชิงลึก (Deep Learning) ที่ประกอบด้วยการทำงานย่อยหลายชั้น แต่ละชั้นมีการเชื่อมโยง การแปลงและส่งผ่านสัญญาณระหว่างชั้น แบบชั้นต่อชั้น (Deng and Yu, 2014; Olshausen, 1996) ด้วยจุดมุ่งหมายเพื่อสร้างแบบจำลองแทนความหมายของข้อมูลในระดับสูง โดยการค้นหาและรวมฟีเจอร์เข้าด้วยกันในระดับอื่นอย่างอัตโนมัติ (Bengio et al., 2015; Deng and Yu, 2014; Olshausen, 1996) ตัวอย่างเช่น Restricted Boltzmann Machines (RBMs) ถูกนำมาใช้ร่วมกับเทคนิค Autoencoder ที่มีสิทธิภาพสูงในด้านการแทนข้อมูล สำหรับสถาปัตยกรรมเชิงลึกถูกนำมาใช้อย่างแพร่หลายในทางคอมพิวเตอร์วิทัศน์ เช่น การรู้จำเสียงพูด (Speech Recognition) การประมวลผลภาษาธรรมชาติ (Natural Language Processing) ชีวสารสนเทศศาสตร์ (Bioinformatics) การรู้จำเสียง (Audio Recognition) และการกรองข้อมูลในสังคมออนไลน์ (Social Network Filtering) เป็นต้น (Ciresan et al., 2012) โดยมีผู้เสนอวิธีการไว้หลายแบบ เช่น โครงข่ายประสาทเชิงลึก (Deep Neural Networks) โครงข่ายความเชื่อเชิงลึก (Deep Belief Networks) (Hinton et al., 2006) โบลทซ์มันน์แมชชีนเชิงจำกัด (Restricted Boltzmann Machines: RBMs) (Hinton, 2007) โบลทซ์มันน์แมชชีนเชิงลึก (Deep Boltzmann Machines: DBMs) (Salakhutdinov and Hinton, 2009) โครงข่ายเข้ารหัสอัตโนมัติเชิงลึก (Deep Autoencoder Networks: DANs) (Hinton

and Salakhutdinov, 2006) ซึ่งได้มีนักวิจัยนำไปประยุกต์ใช้ในงานวิจัย เช่น ในงานข้อมูลอนุกรมเวลา (Thinsungnoen et al., 2017; Ripoll et al., 2016; Gianniotis et al., 2016; Längkvist et al., 2014; Busseti et al., 2012) งานวิจัยที่นำเสนอวิธีการเรียนรู้ฟีเจอร์ (Feature) ที่มีลักษณะกระจัดกระจาย (Ranzato et al., 2008) งานวิจัยที่นำเสนอวิธีการจัดกลุ่มแบบใหม่ และการแทนข้อมูลที่เหมาะสำหรับการจัดกลุ่ม (Song et al., 2013) งานวิจัยที่นำเสนอเทคนิคในการแทนข้อมูลสำหรับใช้ในการจัดกลุ่มข้อมูลภาพทั่วไปและภาพใบหน้า (Huang et al., 2014) งานวิจัยที่นำเสนอเทคนิคแทนข้อมูลที่สามารถช่วยให้สามารถจำแนกข้อมูลอนุกรมเวลาได้ง่ายและชัดเจนมากขึ้น (Gianniotis et al., 2016) เป็นต้น จากงานวิจัยที่กล่าวมาช่วยยืนยันว่าการพัฒนาการแทนข้อมูลสำหรับข้อมูลที่มีมิติสูง เป็นงานที่มีความสำคัญช่วยให้การวิเคราะห์ หรือสร้างโมเดลในการเรียนรู้ของเครื่องจักรมีประสิทธิภาพมากขึ้นได้

ถึงแม้ DANs จะมีคุณสมบัติที่โดดเด่นคือความสามารถในการเรียนรู้เพื่อแทนข้อมูลมิติสูง แต่ก็มีข้อจำกัดในเรื่องของความยากในการกำหนดโมเดลที่ดีที่สุดหรือที่เหมาะสมที่สุดของโครงข่าย ซึ่งเทคนิคที่ให้คำตอบกับปัญหานี้ได้ดีวิธีหนึ่งคือการนำหลักการของอัลกอริทึมเชิงพันธุกรรม (Genetic Algorithms: GAs) มาช่วยหาคำตอบซึ่งพบได้จากหลายงานวิจัยที่อาศัยเทคนิคนี้ (Suksut et al., 2017; Chuentawat et al., 2017; Shen et al., 2007; Demongeot et al., 1994;) ดังนั้นงานวิจัยนี้จึงนำเสนอวิธีการสำหรับแทนข้อมูลอนุกรมเวลาเพื่อให้เกิดประสิทธิภาพสูงขึ้นในการจัดกลุ่มสำหรับข้อมูลคลื่นไฟฟ้าหัวใจและคลื่นไฟฟ้าสมอง โดยใช้ DANs ซึ่งเป็นวิธีที่นำเอาหลักการของ RBMs มาใช้ร่วมกับ Autoencoder (AE) ซึ่งเป็นเทคนิคที่มีประสิทธิภาพสูงในด้านการแทนข้อมูล โดยใช้อัลกอริทึมเชิงพันธุกรรม (Genetic Algorithms: GAs) เพื่อหาโครงข่ายที่ดีที่สุดให้กับ DANs ตัวแทนข้อมูล ECGs และ EEGs ที่ได้จะถูกนำมาจัดกลุ่มด้วยอัลกอริทึม Permutation Distribution Clustering (PDC) และ k-Means โดยเปรียบเทียบประสิทธิภาพกับข้อมูลดั้งเดิม และเทคนิคการแทนข้อมูลแบบอื่นด้วยมาตรวัด ค่าความถูกต้อง ค่าพิวริตี เวลาในการประมวลผล และประเมินค่าความเหมาะสมของกลุ่มด้วย Silhouette และค่าผลรวมความผิดพลาด (Sum of Squared Error: SSE) โดยมีวัตถุประสงค์คือ ตัวแทน ECGs และ EEGs ที่ได้จะมีประสิทธิภาพ และสามารถช่วยสนับสนุนให้การจัดกลุ่มข้อมูลอนุกรมเวลามีประสิทธิภาพยิ่งขึ้น

1.2 วัตถุประสงค์การวิจัย

1. เพื่อนำเสนออัลกอริทึมการแทนอนุกรมเวลาด้วยเทคนิคการเรียนรู้เชิงลึก (Time-Series Representation with Deep Learning Technique Algorithm: TSDL Algorithm)

2. เพื่อค้นหาโมเดลที่เหมาะสมสำหรับการเรียนรู้เชิงลึกด้วยการใช้ Genetic Algorithms (GAs) สำหรับ Deep Autoencoder Networks (DANs) ที่สามารถผลิตตัวแทนอนุกรมเวลาเพื่อการจัดกลุ่มที่มีประสิทธิภาพ เรียกว่าโมเดล *DANGA*

3. เพื่อปรับปรุงประสิทธิภาพการจัดกลุ่มข้อมูลอนุกรมเวลา ให้มีประสิทธิภาพมากยิ่งขึ้น

1.3 ข้อตกลงเบื้องต้น

ข้อมูลอนุกรมเวลาที่นำเข้าสำหรับการจัดกลุ่ม จะต้องเป็นข้อมูลแบบอนุกรมเวลาเชิงเดี่ยว

1.4 ขอบเขตของการวิจัย

งานวิจัยนี้นำเสนอวิธีการแทนข้อมูลอนุกรมเวลาเชิงเดี่ยวที่สามารถช่วยปรับปรุงประสิทธิภาพของการจัดกลุ่มให้มีความเหมาะสมยิ่งขึ้นซึ่งกำหนดขอบเขตการวิจัยไว้ดังต่อไปนี้

1. งานวิจัยนี้เป็นการศึกษาและวิจัยเทคนิคการจัดกลุ่มข้อมูลอนุกรมเวลารูปแบบ Whole Time-series Clustering สำหรับจัดกลุ่มข้อมูลอนุกรมเวลาแบบ Hierarchical Clustering และแบบ Partitioning Clustering

2. ข้อมูลอนุกรมเวลาที่ใช้สำหรับการวิจัยนี้ ประกอบด้วยชุดข้อมูลอนุกรมเวลาจำนวน 2 ชุดข้อมูล ได้แก่ ข้อมูลคลื่นไฟฟ้าหัวใจ (Electrocardiogram Signals: ECGs) และข้อมูลคลื่นไฟฟ้าสมอง (Electroencephalographic Signals: EEGs)

3. การพัฒนาอัลกอริทึมการแทนอนุกรมเวลาด้วยเทคนิคการเรียนรู้เชิงลึก (Time-Series Representation with Deep Learning Technique Algorithm: TSDL Algorithm) ใช้เทคนิค Deep Autoencoder Networks เป็นพื้นฐาน ซึ่งมีส่วนประกอบในการทำงานดังนี้

3.1 เทคนิค Deep Autoencoder Networks ซึ่งมีการปรับแต่งการเข้ารหัสข้อมูลโดยใช้ Backpropagation การ Pre-training โดยใช้ Stack ของ Restricted Boltzmann Machines กระบวนการทำงานภายใน Deep Autoencoder Networks ประกอบด้วยโครงข่าย Encoder และ Decoder

3.2 การคัดเลือกโมเดลที่เหมาะสมที่สุดของ Deep Autoencoder Networks (DANs) ด้วย Genetic Algorithms (GAs) ซึ่งโมเดลที่ได้นี้เรียกว่า *DANGA*

3.3 ผลลัพธ์ที่ต้องการคือ ตัวแทนอนุกรมเวลาจาก DANGA (Time-Series-Representative by DANGA: TSR-DANGA) ที่สามารถเพิ่มประสิทธิภาพการจัดกลุ่มข้อมูล

4. การวัดความคล้ายคลึง / ความแตกต่าง ในงานวิจัยนี้จะใช้มาตรวัดที่สอดคล้องกับเทคนิควิธีการจัดกลุ่มอนุกรมเวลาที่เลือกใช้ ประกอบด้วยสองวิธีได้แก่

4.1 การวัดความคล้ายคลึงแบบดั้งเดิมโดยทั่วไป ได้แก่ การวัดระยะทางแบบยุคลิด (Euclidean Distance)

4.2 การใช้ Squared Hellinger Distance ในการวัดระยะสำหรับการจัดกลุ่มข้อมูลด้วยอัลกอริทึม PDC

5. อัลกอริทึมการจัดกลุ่มข้อมูลอนุกรมเวลา ในงานวิจัยนี้ใช้เทคนิคแบบลำดับชั้นด้วยอัลกอริทึมการจัดกลุ่มข้อมูล PDC และเทคนิคแบบแบ่งแยกด้วยอัลกอริทึม k-Means

6. การประเมินผลการจัดกลุ่มข้อมูลอนุกรมเวลา ในงานวิจัยนี้ใช้มาตรวัด 5 มาตรวัด โดยแยกเป็น 3 กลุ่มได้แก่

6.1 ประเมินภายนอก (External Evaluation) สำหรับอนุกรมเวลาที่ทราบกลุ่มที่แท้จริงมาก่อนประกอบด้วย 2 มาตรวัด ได้แก่

1) ค่าการประเมินกลุ่ม (Cluster Evaluation) หรือความถูกต้อง (Accuracy)

2) ค่าพิวริตี (Purity) หรือค่าความบริสุทธิ์

6.2 ประเมินภายใน (Internal Evaluation) สำหรับอนุกรมเวลาที่ไม่ทราบกลุ่มที่แท้จริงมาก่อน อาศัยการประเมินความเหมาะสมของจำนวนกลุ่ม (Optimal k) โดยประเมินจาก

1) ค่าผลรวมความผิดพลาด (Sum of Squared Error: SSE)

2) ค่าซิลลูเอ็ต (Silhouette)

6.3 เวลาที่ใช้ในการประมวลผล (Processing Time)

1.5 ประโยชน์ที่ได้รับ

1. ได้อัลกอริทึมสำหรับหาตัวแทนอนุกรมเวลาที่อาศัยทำงานของเทคนิคโครงข่ายการเข้ารหัสอัตโนมัติเชิงลึก (Deep Autoencoder Networks)

2. ได้โมเดลโครงข่ายการเข้ารหัสอัตโนมัติเชิงลึกที่เหมาะสม เพื่อการผลิตตัวแทนอนุกรมเวลาที่ดีสำหรับใช้ในการจัดกลุ่มข้อมูลอนุกรมเวลา

3. ได้ตัวแทนอนุกรมเวลาที่อยู่ในรูปแบบที่สามารถนำมาใช้งานร่วมกัน และสนับสนุนเทคนิควิธีการจัดกลุ่มที่มีอยู่ได้ง่าย

4. สามารถปรับปรุงประสิทธิภาพการจัดกลุ่มข้อมูลอนุกรมเวลา ให้มีประสิทธิภาพมากยิ่งขึ้น

บทที่ 2
ปริทัศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้อง

การค้นหาความรู้ในฐานข้อมูล (Knowledge Discovery in Databases - KDD) หรือการทำเหมืองข้อมูล (Data Mining) เป็นเทคนิคเพื่อค้นหารูปแบบ (Pattern) แนวทาง และความสัมพันธ์ที่ซ่อนอยู่ในชุดข้อมูลจำนวนมาก ความรู้ที่ได้จากการทำเหมืองข้อมูลมีหลายรูปแบบ ได้แก่ กฎความสัมพันธ์ (Association Rule) โมเดลเพื่อการจำแนกประเภทข้อมูล (Classification Model) กลุ่มข้อมูล (Data Clusters) (บุญเสริม กิจศิริกุล, 2548; กฤษณะ ไวยมัย และคณะ, 2544; Wenke et al., 2001) ในส่วนนี้จะกล่าวถึงการศึกษาหลักการ หลักทฤษฎี และงานวิจัยที่เกี่ยวข้อง สำหรับนำมาประยุกต์ใช้เพื่อพัฒนางานวิจัยประกอบด้วย

1. หลักในการจัดกลุ่มข้อมูลอนุกรมเวลา
2. อัลกอริทึมสำหรับการจัดกลุ่มข้อมูลอนุกรมเวลา
3. การหาตัวแทนอนุกรมเวลา
4. สถาปัตยกรรมโครงข่ายการเรียนรู้เชิงลึก
5. อัลกอริทึมเชิงพันธุกรรม
6. การตรวจสอบและประเมินผลการจัดกลุ่ม
7. งานวิจัยที่เกี่ยวข้อง

2.1 หลักในการจัดกลุ่มข้อมูลอนุกรมเวลา

การจัดกลุ่มข้อมูลอนุกรมเวลา (Time-Series Clustering) เป็นปัญหาที่ท้าทายอีกปัญหาหนึ่งในเทคนิคการจัดกลุ่มข้อมูล เนื่องจากข้อมูลอนุกรมเวลาต้องเกี่ยวข้องกับลำดับเวลาการเกิดขึ้นของข้อมูลและมักเป็นข้อมูลที่จัดเก็บไว้เป็นระยะเวลานานจึงมักมีขนาดใหญ่ และมีมิติสูง สำหรับหลักในการจัดกลุ่มข้อมูลอนุกรมเวลา ประกอบด้วยหลายส่วนอธิบายได้ดังนี้

2.1.1 หลักการเบื้องต้นข้อมูลอนุกรมเวลาและการจัดกลุ่ม

อนุกรมเวลา (Time-Series) คือ เซตของข้อมูลเชิงปริมาณที่จัดเก็บในช่วงเวลาหนึ่ง ตัวอย่างเช่น ดัชนีตลาดหลักทรัพย์ในแต่ละวันเมื่อปิดทำการซื้อขาย รายได้ประชาชาติ รายไตรมาสรายรับในแต่ละปีของบริษัทแห่งหนึ่ง คลื่นไฟฟ้าหัวใจ หรือ คลื่นไฟฟ้าสมองขณะทำกิจกรรมอย่างต่อเนื่องในช่วงเวลาหนึ่ง เป็นต้น (สำนักนโยบายการออมและการลงทุน, 2548)

ข้อมูลอนุกรมเวลา (Time-Series Data) คือ ชุดของข้อมูลที่เก็บรวบรวมตามระยะเวลาเป็นช่วงอย่างต่อเนื่องกันตามลำดับ (Aghabozorgi et al., 2015; Längkvist et al., 2014) เช่น ข้อมูลยอดขายสินค้า ข้อมูลน้ำท่า ข้อมูลการตรวจวัดทางชีวะการแพทย์ เป็นต้น ข้อมูลอนุกรมเวลาอาจมีคาบเวลาที่เป็นข้อมูลรายนาที รายชั่วโมง รายวัน รายเดือน รายไตรมาส หรือรายปี ทั้งนี้ขึ้นอยู่กับความเหมาะสมในการนำไปใช้ (สำนักนโยบายการออมและการลงทุน, 2548)

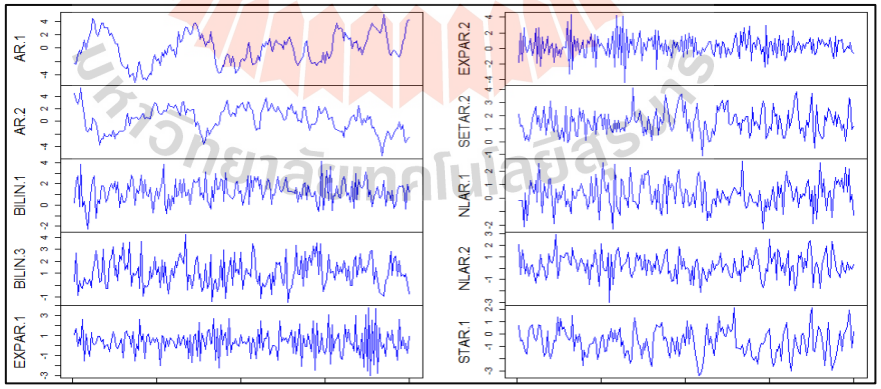
2.1.2 ประเภทการจัดกลุ่มข้อมูลอนุกรมเวลา

การจัดกลุ่มข้อมูลอนุกรมเวลาแบ่งเป็น 2 ประเภท ได้แก่ Whole Time-Series Clustering และ Subsequence Time-Series Clustering (Keogh and Lin, 2005) ดังนี้

(1) Whole Time-Series Clustering

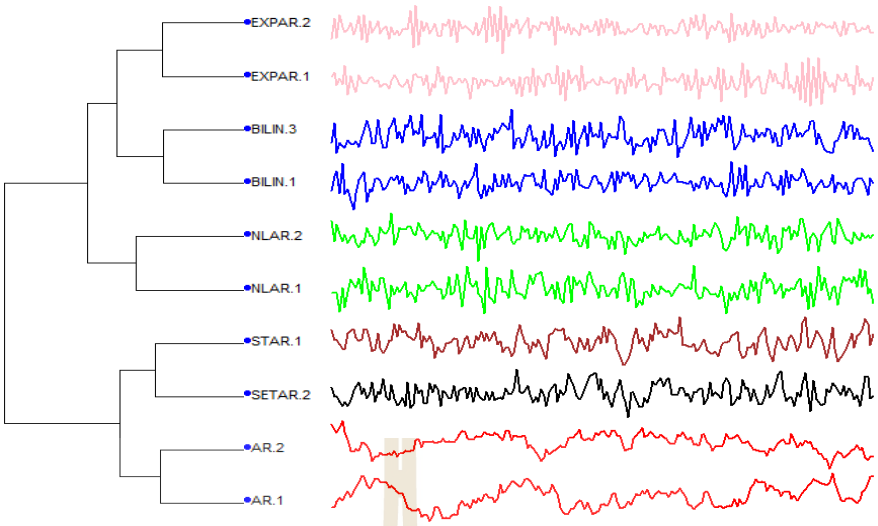
การจัดกลุ่มข้อมูลอนุกรมเวลาแบบ Whole Time-Series คือกระบวนการเรียนรู้เพื่อจัดอนุกรมเวลาที่มีความคล้ายคลึงกันมากให้อยู่ในกลุ่มเดียวกัน โดยมุ่งเน้นที่จะวัดความคล้ายคลึงของอนุกรมเวลาตลอดทั้งสาย (Aghabozorgi et al., 2015; Keogh and Lin, 2005) ตัวอย่างการจัดกลุ่มอนุกรมเวลาแบบ Whole Time-Series ในข้อมูลสังเคราะห์แสดงดังต่อไปนี้

ตัวอย่าง การจัดกลุ่มข้อมูลอนุกรมเวลาแบบ Whole Time Series กับข้อมูลอนุกรมเวลาสังเคราะห์ที่ได้จาก 6 โมเดล (กำหนดเป็น 6 กลุ่ม) ที่ต่างกัน จำนวน 10 อนุกรม โดยแต่ละอนุกรมมีขนาด 200 จุดเวลา (Time Points) ข้อมูลที่ใช้มีดังนี้ กลุ่มที่ 1 มีอนุกรม AR.1 และ AR.2 กลุ่มที่ 2 มี BILIN.1 และ BILIN.3 กลุ่มที่ 3 มี EXPAR.1 และ EXPAR.2 กลุ่มที่ 4 มีSETAR.2 กลุ่มที่ 5 มี NLAR.1 และ NLAR.2 และกลุ่มที่ 6 มี STAR.1 แสดงดังรูปที่ 2.1



รูปที่ 2.1 ตัวอย่างข้อมูลอนุกรมเวลาสังเคราะห์จำนวน 10 อนุกรม

เมื่อนำข้อมูลดั้งเดิม (Raw Data) มาจัดกลุ่มแบบ Whole Time-Series ด้วยอัลกอริทึมการจัดกลุ่มแบบลำดับชั้น (จากตัวอย่างเป็นอัลกอริทึม PDC) ผลลัพธ์การจัดกลุ่มจะแสดงกราฟของอนุกรมเวลาแยกสีตามกลุ่มจริง ดังรูปที่ 2.2

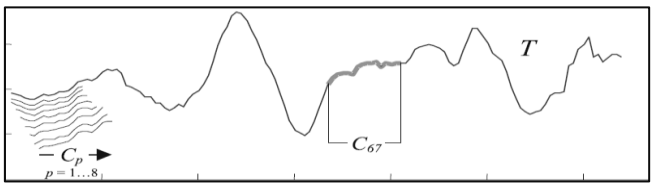


รูปที่ 2.2 ตัวอย่างการจัดกลุ่มข้อมูลอนุกรมเวลาแบบ Whole Time-Series

(2) Subsequence Time-Series Clustering

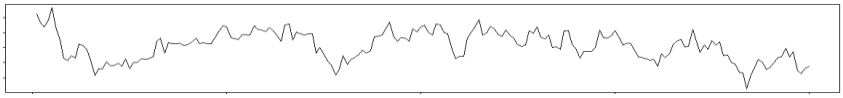
การจัดกลุ่มข้อมูลอนุกรมเวลาแบบ Subsequence Time-Series คือการจัดกลุ่มอนุกรมเวลาโดยพิจารณาที่ความคล้ายคลึงกันในลำดับย่อยของอนุกรมเวลา ด้วยการแยกอนุกรมให้เป็นลำดับย่อยที่มีขนาดเท่ากันด้วยเทคนิคบางอย่าง เช่น Sliding Window (Aghabozorgi et al., 2015; Keogh and Lin, 2005)

แนวคิดการใช้ Sliding Window อธิบายได้ดังนี้ สำหรับอนุกรมเวลา T ขนาด m = 128 มิติ (หรือคาบเวลา หรืองวดเวลา หรือจุดเวลา) แยกเป็นลำดับย่อยขนาดเท่ากัน w = 16 มิติ จะได้ตำแหน่งเริ่มต้นของลำดับย่อยแต่ละอนุกรมคือ P = 1, 2, …, p อนุกรมเวลาย่อยที่ได้คือ C_P ดังนั้นเมื่อ S คือเมตริกซ์ที่บรรจุชุดข้อมูลย่อยที่ได้จากเทคนิค Sliding Window แล้ว เมตริกซ์ S จะมีจำนวนข้อมูล = (m – w + 1) (Keogh and Lin, 2005) แสดงแนวคิดดังรูปที่ 2.3



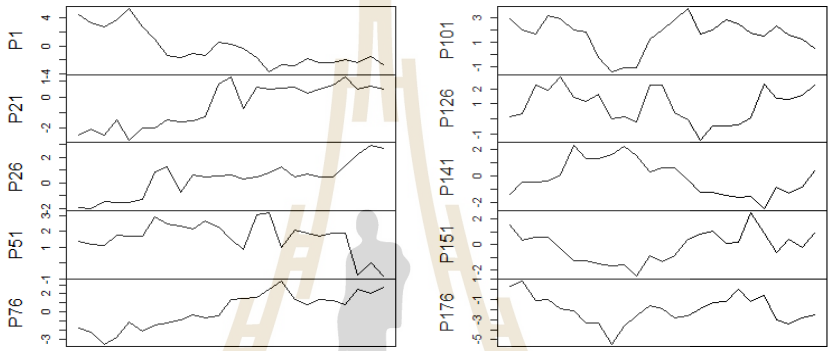
รูปที่ 2.3 แนวคิดการแยก Subsequence Time-Series ด้วย Sliding Window
(ที่มา: Keogh and Lin, 2005)

ตัวอย่าง การจัดกลุ่มอนุกรมเวลาแบบ Subsequence Time-Series กับข้อมูลอนุกรมเวลาสังเคราะห์ AR.2 ขนาดเวลายาว 200 จุดเวลา ข้อมูลมีลักษณะดังรูปที่ 2.4



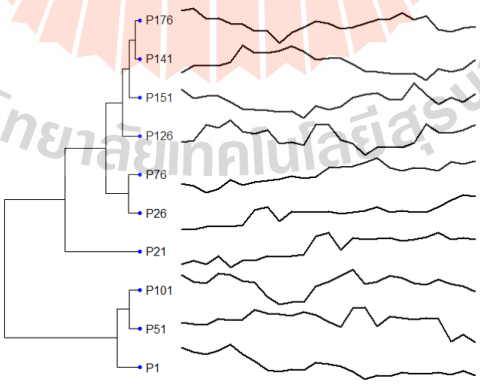
รูปที่ 2.4 อนุกรมเวลาสังเคราะห์ AR.2 สำหรับการจัดกลุ่มแบบ Subsequence Time-Series

เมื่อนำอนุกรม AR.2 มาแยกเป็นอนุกรมย่อยขนาด w = 25 มิติ ด้วยวิธี Sliding Window จะได้อนุกรมย่อยทั้งหมด = (200 – 25 + 1) = 176 ชุด เพื่อให้เห็นภาพการจัดกลุ่มได้ชัดเจน จะนำอนุกรมย่อยมาใช้เพียง 10 อนุกรมได้แก่ อนุกรมที่จุดเวลา P1, P21, P26, P51, P76, P101, P126, P141, P151 และ P176 ดังรูปที่ 2.5



รูปที่ 2.5 Subsequence Time-Series ของอนุกรมเวลาสังเคราะห์ AR.2

เมื่อนำข้อมูลอนุกรมเวลาย่อย มาจัดกลุ่มด้วยอัลกอริทึมการจัดกลุ่มแบบลำดับชั้น ผลลัพธ์การจัดกลุ่มจะแสดงกราฟของอนุกรมเวลา ดังรูปที่ 2.6



รูปที่ 2.6 ตัวอย่างการจัดกลุ่มข้อมูลอนุกรมเวลาแบบ Subsequence Time-Series

การวิเคราะห์ Subsequence Time-Series มักใช้ตรวจสอบหารูปแบบย่อยที่อาจเป็นสัญญาณบ่งชี้ความผิดปกติ เช่น ในคลื่นไฟฟ้าหัวใจ หรือตรวจจับลักษณะของรูปแบบที่มักเกิดขึ้นบ่อย ๆ ในอนุกรมเวลานั้น ๆ เป็นต้น

2.1.3 หลักการจัดกลุ่มอนุกรมเวลาแบบ Whole Time-Series Clustering

การจัดกลุ่มอนุกรมเวลาแบบ Whole Time-Series มีการทำงาน 5 ส่วนหลักได้แก่ 1) การลดมิติหรือการแทนอนุกรมเวลา (Dimensionality Reduction or Representation Method) 2) การวัดความคล้ายคลึง/ความต่าง หรือการวัดระยะ (Similarity or Distance Measurement) 3) การกำหนดตัวแทนของกลุ่ม (Clustering Prototypes) 4) อัลกอริทึมสำหรับการจัดกลุ่ม (Clustering Algorithm) และ 5) การประเมินผลการจัดกลุ่ม (Clustering Evaluation) (Aghabozorgi et al., 2015) มีรายละเอียดแต่ละงานดังต่อไปนี้

(1) การแทนอนุกรมเวลา

การแทนอนุกรมเวลา (Dimensionality Reduction or Representation Method) ถือเป็นงานปกติสำหรับการจัดกลุ่มประเภท Whole Time-Series Clustering ซึ่งจะแทนข้อมูลอนุกรมเวลาดั้งเดิม (Raw Time-Series) ด้วยปริภูมิอื่น (Another Space) โดยการแปลงอนุกรมเวลาไปสู่มิติที่ต่ำกว่าหรืออาจใช้วิธีดึง Feature ที่มีประโยชน์ออกมาใช้ (Lin et al., 2003; Keogh, 2005; Ghysels et al., 2006; Duan et al., 2006) การลดมิติของอนุกรมเวลาให้ต่ำลงจะสามารถช่วยให้การประมวลผลมีศักยภาพ และมีประสิทธิภาพมากขึ้นได้ เนื่องจาก

1) ต้องการลดการใช้หน่วยความจำ เนื่องจากหากใช้ข้อมูล Raw Time-Series ทรัพยากรอาจไม่เพียงพอในการประมวลผลได้ (Lin et al., 2003; Keogh et al., 2000)

2) การคำนวณระยะทางระหว่าง Raw Data นั้นกินทรัพยากรเป็นอย่างมาก และนอกจากนี้การลดมิติให้ต่ำลงมีผลให้การทำงานในการจัดกลุ่มเร็วขึ้นอย่างมีนัยสำคัญ (Lin et al., 2003; Keogh et al., 2000)

3) ในขณะที่มีการวัดระยะระหว่าง Raw Time-Series สองสายอาจมีการทำงานที่จำเป็นต้องใช้ทรัพยากรสูงขึ้นอย่างไม่คาดคิดเกิดขึ้นได้ เนื่องจากการวัดระยะบางวิธีการอ่อนไหวมากกับรูปแบบที่ผิดปกติ (Ratanamahatana, 2005; Ratanamahatana et al., 2005) ดังนั้นการใช้ Raw Time-Series Data อาจพบว่าอนุกรมเวลาบางสายอาจถูกจัดให้อยู่กลุ่มเดียวกับ Noise แทนที่จะถูกจัดให้อยู่กลุ่มเดียวกับสายที่มีรูปร่างเหมือนกันได้

การแทนอนุกรมเวลาแบ่งได้เป็น 4 ประเภท (Lin et al., 2003; Ratanamahatana et al., 2005; Bagnall et al., 2006; Shieh and Keogh, 2009) ได้แก่

(1.1) วิธีแบบ Data Adaptive เป็นวิธีการแทนอนุกรมเวลาโดยการทำงานกับอนุกรมเวลาทุกตัวในชุดข้อมูลที่ให้ค่าความผิดพลาดน้อยที่สุด โดยใช้วิธีการปรับความยาวของข้อมูลที่มีความยาวไม่เท่ากัน ซึ่งหลักการนี้ประยุกต์ใช้กับหลากหลายวิธีได้แก่ Piecewise Polynomials Interpolation (PPI) (Morinaka et al., 2001), Piecewise Polynomials Regression (PPR)

(Shatkay and Zdonik, 1996), Piecewise Linear Approximation (PLA), Piecewise Constant Approximation (PCA), Adaptive Piecewise Constant Approximation (APCA) (Keogh et al., 2001), Singular Value Decomposition (SVD) (Faloutsos et al., 1994; Korn et al., 1997), Natural Language, Symbolic Natural Language (NLG) (Portet et al., 2009), Symbolic Aggregate ApproXimation (SAX) and iSAX (Lin et al., 2007) เป็นต้น

(1.2) วิธีแบบ Non-Data Adaptive เป็นวิธีแทนอนุกรมเวลาด้วยการกำหนดขนาดที่เท่ากันให้กับส่วนย่อย และทำการเปรียบเทียบการกำหนดตัวแทนของอนุกรมเวลาทั่วไปอย่างตรงไปตรงมา ซึ่งหลักการนี้ใช้ในวิธีแบบ Wavelets (Chan and Fu, 1999) โดยมีหลายเทคนิค ได้แก่ HAAR, DAUBECHIES, Coeiflets, Symlets, Discrete Wavelet Transform(DWT), Spectral Chebyshev Polynomials (Cai and Ng, 2004), SpectralDFT (Faloutsos et al., 1994), Random Mappings (Bingham, 2001), Piecewise Aggregate Approximation (PAA) (Keogh et al., 2000) และ Indexable Piecewise Linear Approximation (IPLA) (Chen et al., 2007) เป็นต้น

(1.3) วิธีแบบ Model-based เป็นวิธีแทนอนุกรมเวลาด้วยวิธีแบบสุ่ม ซึ่งเทคนิคที่อาศัยหลักการนี้ได้แก่ Markov Models และ Hidden Markov Model (HMM) (Minnen et al., 2006; Minnen et al., 2007; Panuccio et al., 2002), Statistical Models, Time-series Bitmaps (Kumar et al., 2005) และ Auto-Regressive Moving Average (ARMA) (Corduas and Piccolo, 2008; Kalpakis et al., 2001) เป็นต้น

4) วิธีแบบ Data Dictated เป็นวิธีที่ต่างจากวิธีการอื่นคือเป็นวิธีที่อาศัยสัดส่วนการบีบข้อมูล ซึ่งกำหนดโดยอัตโนมัติขึ้นกับ Raw Time-Series เทคนิคที่อาศัยหลักการนี้ได้แก่ Clipped

(2) การวัดความคล้ายคลึง/ความต่าง

การวัดความคล้ายคลึง/ความต่าง หรือวัดระยะห่าง (Similarity or Distance Measurement คือการวัดว่าอนุกรมแต่ละตัวมีความคล้ายคลึง/ความต่าง กันมากน้อยเพียงใด โดยอาศัยมาตรวัดสำหรับวัดความคล้ายคลึง/ความต่าง ซึ่งมีมากมาย ตัวอย่างเช่น Hausdorff Distance, Modified Hausdorff (MODH), HMM-based Distance, Dynamic Time Warping (DTW), Euclidean Distance, Euclidean Distance ใน PCA และ Longest Common Sub-Sequence (LCSS) (Keogh and Pazzani, 1998)

สำหรับหลักการพื้นฐานอย่างง่ายที่มักถูกใช้เป็นฐานในการวัดความคล้ายคลึงในการจัดกลุ่มข้อมูลทั่วไป หรือ การจัดกลุ่มข้อมูลอนุกรมเวลา คือ การวัดระยะแบบยุคลิด (Euclidean Distance, Euclidean Metric) มีรายละเอียดดังนี้

(2.1) ระยะทางแบบยุคลิด คือ ระยะทางปกติระหว่างจุดสองจุดในแนวเส้นตรงที่สามารถวัดได้ด้วยไม้บรรทัด หรืออุปกรณ์การวัดระยะ การวัดระยะทางแบบยุคลิดนี้จะไม่มีความโค้งและไม่สามารถวัดจากการโค้งงอ (Deza and Deza, 2009) มีรายละเอียดดังนี้

กำหนดให้ระยะทางแบบยุคลิดระหว่างจุดสองจุด p และจุด q คือความยาวของส่วนของเส้นตรง pq ถ้า $p = (p_1, p_2, \ldots, p_n)$ และ $q = (q_1, q_2, \ldots, q_n)$ ในระบบพิกัดคาร์ทีเซียน บนปริภูมิยุคลิด n มิติ เมื่อ d(p, q) คือระยะทางระหว่างจุด p กับ q คำนวณได้จาก

$$d(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \qquad (2.1)$$

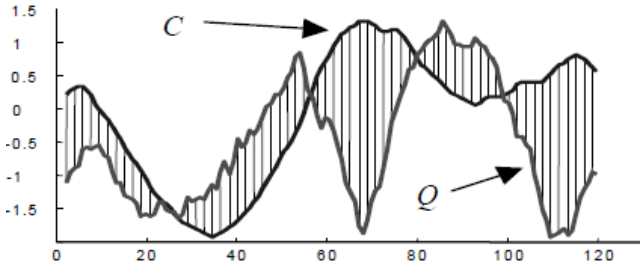
โดยที่ i คือ ตำแหน่งข้อมูล มีค่าตั้งแต่ 1 ถึง n

(2.2) การวัดระยะทางแบบยุคลิดในข้อมูลอนุกรมเวลา สำหรับการวัดระยะห่างระหว่างอนุกรมเวลา โดยอาศัยหลักการวัดระยะทางแบบยุคลิดซึ่งเป็นการวัดระยะแบบปกติทั่วไประหว่างจุดสองจุด ณ จุดเวลาเดียวกันระหว่างอนุกรมสองอนุกรม ทำได้ดังนี้ กำหนดให้ Q และ C คืออนุกรมเวลาสองอนุกรมที่มีขนาดความยาวเท่ากันคือ n ดังนั้นการหาระยะห่างระหว่างอนุกรม Q และ C ด้วยวิธีวัดแบบยุคลิด หาได้จากสมการดังต่อไปนี้ (Lin et al., 2003)

$$D(Q, C) = \sqrt{\sum_{i=1}^{n} (q_i - c_i)^2} \qquad (2.2)$$

โดยที่ i คือ จุดเวลาบนอนุกรมเวลา Q และ C ซึ่งมีค่าตั้งแต่ 1 ถึง n

แสดงแนวคิดการวัดระยะห่างระหว่างอนุกรม Q และ C ด้วยวิธีวัดแบบยุคลิดแสดงได้ดังรูปที่ 2.7



รูปที่ 2.7 การวัดระยะห่างระหว่างอนุกรมเวลา 2 อนุกรม

(ที่มา: Lin et al., 2003)

(3) การกำหนดตัวแทนของกลุ่ม

การกำหนดตัวแทนของกลุ่ม คือการกำหนดข้อมูลที่จะใช้เป็นจุดศูนย์กลาง (Centriod) กลุ่ม สำหรับอ้างอิงเพื่อวัดระยะหรือความคล้ายคลึง โดยเฉพาะในการจัดกลุ่มแบบแบ่งแยก อย่างเช่น k-Means, k-Medoids, Fuzzy C-Means ซึ่งคุณภาพของวิธีการเหล่านี้บางครั้งจะขึ้นอยู่กับคุณภาพของการกำหนดตัวแทนของกลุ่ม โดยทั่วไปมีวิธีสำหรับกำหนดตัวแทนของกลุ่ม 3 วิธี (Aghabozorgi et al., 2015) ได้แก่

(3.1) ใช้ Medoid เป็นตัวแทนกลุ่ม ในการจัดกลุ่มข้อมูลอนุกรมเวลามักใช้วิธีที่ธรรมดานี้ในการกำหนด Steiner Sequence (ระยะจากอนุกรมเวลา C กับตัวแทนของกลุ่ม R ที่มีค่าน้อยที่สุด คือ E(C, R) เรียกว่า Steiner Sequence (Gusfield, D., 1997)) ซึ่งหมายถึงจุดศูนย์กลางกลุ่มนิยามเป็นลำดับของค่าที่น้อยที่สุดของผลรวมของระยะไปยังข้อมูลอื่นภายในกลุ่ม ให้พิจารณาจากระยะของทุกคู่อนุกรมเวลาที่อยู่ภายในกลุ่ม โดยวัดระยะแบบยูคลิด หรือมาตรวัดอื่นจากนั้นให้อนุกรมเวลาหนึ่งตัวในกลุ่มที่มีค่าผลรวมความผิดพลาดต่ำที่สุด เป็นจุดศูนย์กลางของกลุ่ม (Vuori and Laaksonen, 2002)

(3.2) ใช้ค่าเฉลี่ยเป็นตัวแทนกลุ่ม สำหรับการกำหนดตัวแทนกลุ่มด้วยวิธีการนี้มีแนวทางในการทำงาน 2 กรณีคือ กรณีที่ 1 หากอนุกรมเวลาที่จะจัดกลุ่มมีขนาดเท่ากันจะใช้การเฉลี่ยแบบง่ายทั่วไปด้วยการหาค่าเฉลี่ยของอนุกรมเวลาในแต่ละจุดเวลา กรณีที่ 2 หากอนุกรมเวลามีขนาดต่างกัน (กรณีที่สนใจแค่รูปร่างของอนุกรม) การใช้วิธีจับคู่หนึ่ง-ต่อ-หนึ่ง จะกลายเป็นเรื่องยากในการที่จะเฉลี่ยรูปร่างได้ เช่น กรณีการใช้ DTW หรือ LCSS ฉะนั้นควรหลีกเลี่ยงการใช้การเฉลี่ยเป็นตัวแทนกลุ่ม (Niennattrakul and Ratanamahatana, 2007)

(4) อัลกอริทึมสำหรับการจัดกลุ่ม

อัลกอริทึมสำหรับการจัดกลุ่ม โดยทั่วไปจำแนกได้ 6 กลุ่ม (Aghabozorgi et al., 2015) ได้แก่ Partitioning, Hierarchical, Grid-based, Model-based, Density-based และ Multi-Step Clustering แต่ในที่นี้จะกล่าวถึงเพียง 5 กลุ่ม ดังนี้

(4.1) Hierarchical Clustering of Time-Series เป็นการจัดกลุ่มอนุกรมเวลาแบบลำดับชั้น เป็นการจัดกลุ่มที่มุ่งเน้นในการพิจารณาข้อมูลที่คล้ายคลึงกัน (หรือต่างกัน) มากที่สุดก่อน แล้วค่อย ๆ รวมกลุ่ม (หรือแยกตัวออก) ทีละคู่ โดยทั่วไปการจัดกลุ่มแบบลำดับชั้นมักจะมีจุดด้อยในแง่ของคุณภาพของการทำงานเนื่องจากไม่สามารถปรับแต่งกลุ่มได้อีกหลังจากที่มีการผสานกลุ่ม (หรือแยกกลุ่ม) แล้ว การจัดกลุ่มอนุกรมเวลาแบบลำดับชั้นมี 2 แบบคือ แบบล่างขึ้นบน และบนลงล่าง (Aghabozorgi et al., 2015) เป็นผลให้การจัดกลุ่มแบบนี้มักใช้งานร่วมกับอัลกอริทึมอื่น

เป็นการทำงานแบบ Hybrid Clustering Approach เช่น Chameleon (Karypis et al., 1999) CURE (Guha et al., 1998) และ BIRCH (Zhang et al., 1996) เป็นต้น

ความคล้ายคลึงในการจัดกลุ่มข้อมูลอนุกรมเวลา สร้างได้จากการจับคู่เมตริกซ์วัดระยะอนุกรมเวลา (Vlachos et al., 2003) ซึ่งการจัดกลุ่มแบบลำดับชั้นมีประโยชน์มากสำหรับจัดกลุ่มข้อมูลอนุกรมเวลาในเรื่องการแสดงผลของการจัดกลุ่ม (Keogh and Pazzani, 1998; Van Wijk and Van Selow, 1999) และลักษณะเด่นที่ต่างจากวิธีอื่น คือ ไม่จำเป็นต้องกำหนดจำนวนกลุ่มไว้ล่วงหน้า ซึ่งถือเป็นจุดแข็งของการจัดกลุ่มแบบลำดับชั้นเนื่องจากในความเป็นจริงเป็นเรื่องยากที่จะกำหนดจำนวนกลุ่มไว้ล่วงหน้าได้ นอกจากนี้การจัดกลุ่มข้อมูลอนุกรมเวลาแบบลำดับชั้นยังเป็นวิธีที่ทำงานได้ดีกับอนุกรมเวลาที่มีขนาดไม่เท่ากันอีกด้วย (Aghabozorgi et al., 2015) ซึ่งมีความเป็นไปได้ในการจัดกลุ่มข้อมูลอนุกรมเวลาที่มีขนาดไม่เท่ากันหากมีการเลือกใช้วิธีการวัดระยะแบบยืดหยุ่นด้วยวิธีที่เหมาะสมเช่น DTW (Sakoe and Chiba, 1971; Sakoe and Chiba, 1978) หรือแบบ LCSS (Vlachos et al., 2002; Banerjee and Ghosh, 2001) นอกจากนี้การจัดกลุ่มแบบลำดับชั้นยังไม่จำเป็นต้องใช้ตัวแทนกลุ่มอีกด้วย ดังนั้นจึงทำให้เป็นวิธีที่สามารถทำงานกับอนุกรมเวลาที่มีขนาดไม่เท่ากันได้ (Aghabozorgi et al., 2015)

(4.2) Partitioning Clustering เป็นการจัดกลุ่มอนุกรมเวลาแบบแบ่งแยกเป็นอีกวิธีการที่เป็นที่นิยมสำหรับการจัดกลุ่ม และอัลกอริทึมที่นิยมใช้งานทั่วไปคือ k-Means ซึ่งเป็นวิธีการที่ตัวแทนของกลุ่มคือค่าเฉลี่ยของข้อมูลในกลุ่ม ที่นิยมถัดมาคืออัลกอริทึม k-Medoids บางเครื่องมือเรียกย่อว่า PAM โดยตัวแทนของกลุ่มคือข้อมูลจริงที่มีค่าใกล้เคียงจุดศูนย์กลางของกลุ่มมากที่สุดนอกจากนี้ยังมีวิธี CLARA และ CLARANS ที่ปรับปรุงมาจาก k-Medoids โดยวิธีการเหล่านี้จำเป็นต้องมีการกำหนดค่า k (จำนวนกลุ่ม) ก่อนเริ่มการจัดกลุ่ม ซึ่งทำให้เป็นงานที่ยากสำหรับการใช้งานจริงกับข้อมูลทั่วไปที่เราไม่ทราบกลุ่มที่แท้จริงของข้อมูลมาก่อน จึงนับเป็นข้อเสียสำหรับการจัดกลุ่มแบบแบ่งแยก และยิ่งเป็นเรื่องยากและท้าทายมากขึ้นในการจัดกลุ่มข้อมูลอนุกรมเวลาเนื่องจากข้อมูลมีขนาดใหญ่และต้องวินิจฉัยสำหรับการกำหนดจำนวนกลุ่ม

(4.3) Model-based Clustering การจัดกลุ่มโดยอาศัยโมเดลเป็นฐานเป็นวิธีที่พยายามที่จะค้นคืนโมเดลดั้งเดิมของชุดข้อมูล โดยมีสมมติฐานว่าแต่ละกลุ่มจะมีโมเดลของกลุ่มและหาข้อมูลที่เข้ากันได้ที่สุดกับโมเดล ตัวอย่างอัลกอริทึมในกลุ่มนี้คือ COBWEB, Neural Network, ART หรือ Self-Organization Map (SOM) สำหรับข้อมูลอนุกรมเวลามีการใช้ SOM ในการจัดกลุ่ม

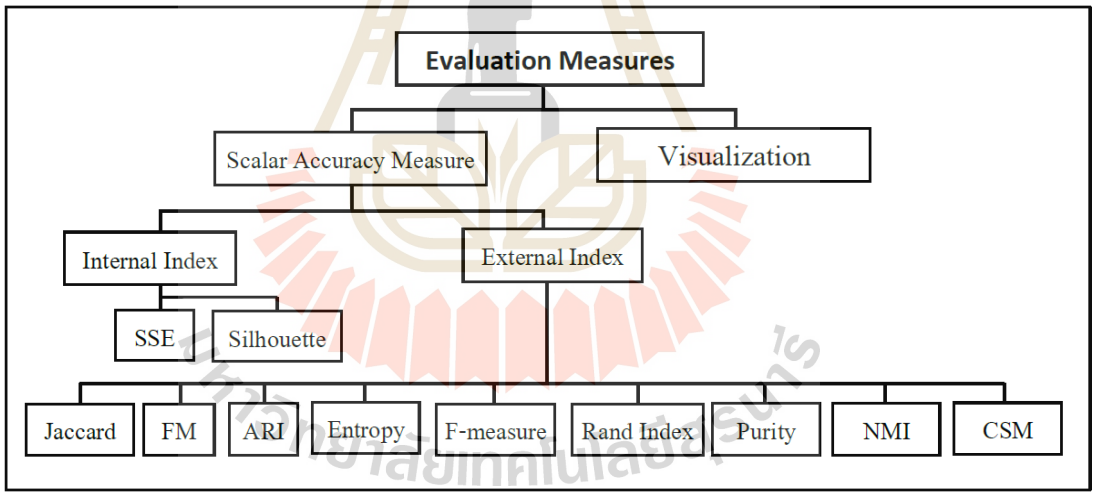
(4.4) Density-based Clustering การจัดกลุ่มโดยอาศัยความหนาแน่นเป็นฐาน จะแยกข้อมูลที่มีความหนาแน่นต่ำกว่าเกณฑ์ออกไปจากกลุ่ม ตัวอย่างอัลกอริทึมในกลุ่มนี้คือ

DBSCAN, OPTICS ซึ่งจุดสำคัญในการจัดกลุ่มด้วยวิธีการนี้คือการกำหนดค่าพารามิเตอร์ให้กับอัลกอริทึมการจัดกลุ่ม การจัดกลุ่มโดยอาศัยความหนาแน่นเป็นฐานนี้ไม่เป็นที่นิยมสำหรับการจัดกลุ่มข้อมูลอนุกรมเวลา เนื่องจากมีความซับซ้อนค่อนข้างสูง (Aghabozorgi et al., 2015)

(4.5) Grid-based Clustering การจัดกลุ่มโดยอาศัยตารางเป็นฐาน จะจัดการกับปริภูมิปริมาณให้อยู่ในรูปเซลล์ที่มีจำนวนที่แน่นอน ที่มีรูปแบบเป็น Grid แล้วจากนั้นจะดำเนินการจัดกลุ่มกับเซลล์ใน Grid โดยมีอัลกอริทึมการจัดกลุ่มข้อมูลในกลุ่มนี้คือ STING และ Wave Cluster ซึ่งโดยทั่วไปเป็นวิธีการที่อาศัยแนวคิดแบบ Grid ไม่พบว่ามีงานวิจัยใดที่อาศัยการจัดกลุ่มเทคนิคนี้สำหรับจัดกลุ่มข้อมูลอนุกรมเวลา (Aghabozorgi et al., 2015)

(5) การประเมินผลการจัดกลุ่ม

การประเมินผลการจัดกลุ่ม (Time-Series Clustering Evaluation) เป็นงานที่ทำได้ยากหากเราไม่ทราบกลุ่มที่แท้จริงของชุดข้อมูล และยังคงเป็นปัญหาที่ท้าทายเพื่อการวิจัยต่อไป โดยทั่วไปแล้วการประเมินผลจะอาศัยมาตรวัดพื้นฐานสำหรับการจัดกลุ่ม ซึ่งแบ่งเป็น 2 ประเภท คือ External Index และ Internal Index (Aghabozorgi et al., 2015) ดังรูปที่ 2.8



รูปที่ 2.8 แผนภาพแสดงกลุ่มของมาตรวัดสำหรับประเมินผลการจัดกลุ่มข้อมูลอนุกรมเวลา

2.2 อัลกอริทึมสำหรับการจัดกลุ่มข้อมูลอนุกรมเวลา

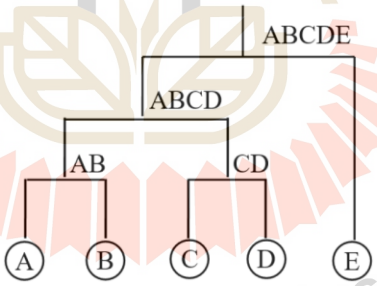
อัลกอริทึมสำหรับการจัดกลุ่มข้อมูลอนุกรมเวลามีหลายวิธี ในงานนี้จะกล่าวถึงเฉพาะเทคนิคที่เป็นที่นิยมในการจัดกลุ่มข้อมูลอนุกรมเวลา ได้แก่ การจัดกลุ่มแบบลำดับชั้นด้วยอัลกอริทึม Agglomerative และอัลกอริทึม PDC และการจัดกลุ่มแบบแบ่งแยกด้วยอัลกอริทึม k-Means มีรายละเอียดดังต่อไปนี้

2.2.1 Agglomerative Hierarchical Clustering

การจัดกลุ่มแบบลำดับชั้นด้วยอัลกอริทึม Agglomerative หรือ AGNES (Agglomerative NESting) เป็นการจัดกลุ่มที่สามารถแสดงผลแบบผังต้นไม้ แบ่งกลุ่มได้ดี มีความแม่นยำสูง โดยในการทำงานจะเริ่มต้นจากข้อมูลทีละตัว แต่ละตัวคือกลุ่มหนึ่งกลุ่มจากนั้นจะรวมกลุ่มที่มีความคล้ายคลึงกันมากที่สุดไว้ด้วยกันทีละคู่ รวมไปเรื่อย ๆ จนกระทั่งข้อมูลทุกตัวรวมกันเป็นกลุ่มเดียว (Maimon and Rokach, 2010; Mohammed and Wagner, 2014) มีหลักการที่เกี่ยวข้องในทำงาน ดังนี้

(1) เดนโดรแกรม (Dendrogram)

Dendrogram คือแผนภาพต้นไม้ที่ใช้แสดงภาพผลการจัดกลุ่มที่ผลิตจากอัลกอริทึมการจัดกลุ่มแบบลำดับชั้น (Everitt, B. 1998) ซึ่งช่วยให้สามารถเห็นลำดับชั้นของกลุ่มได้สะดวกในรูปแบบผังต้นไม้ (Mohammed and Wagner, 2014) ในผังต้นไม้ระดับที่ต่ำที่สุดคือใบ (ข้อมูลแต่ละตัว) ในทางกลับกันระดับที่สูงที่สุดคือราก (ข้อมูลทุกตัวที่รวมเป็นกลุ่มเดียว) ผู้ใช้สามารถจัดกลุ่มข้อมูลด้วยการกำหนดค่า k ซึ่งเป็นตัวระบุจำนวนกลุ่มที่ต้องการ (Mohammed and Wagner, 2014) โดยระบบจะจัดกลุ่มด้วยการตัด Dendrogram ที่ระดับความคล้ายคลึงกันที่ผลิตกลุ่มได้ k กลุ่ม (Maimon and Rokach, 2010) ซึ่งมีตัวอย่างของแผนภาพ Dendrogram ดังรูปที่ 2.9



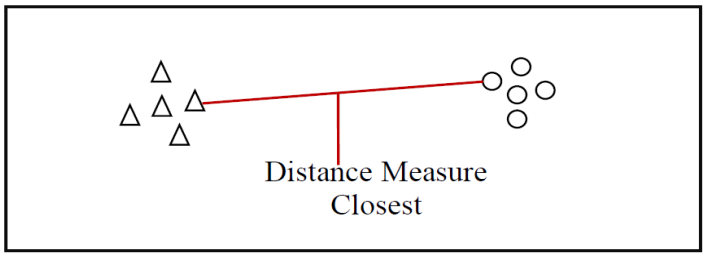
รูปที่ 2.9 ตัวอย่าง Hierarchical Clustering Dendrogram

(2) การหาระยะห่างระหว่างกลุ่ม

การหาระยะห่างระหว่างกลุ่มเป็นขั้นตอนสำคัญในการทำงานของอัลกอริทึม Agglomerative โดยอาศัยมาตรวัดความคล้ายคลึง โดยทั่วไปจะใช้การวัดระยะแบบยุคลิด วิธีวัดระยะในการจัดกลุ่มแบบลำดับชั้นในที่นี้จะกล่าวถึง 4 วิธีดังนี้

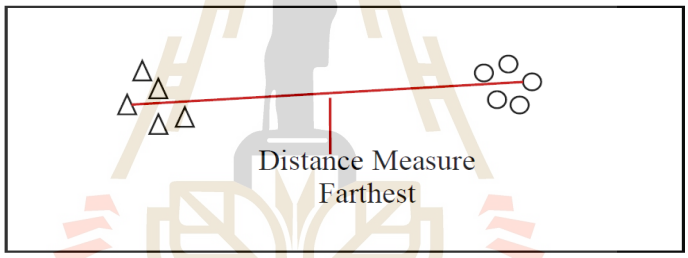
(2.1) Single Link หรือ Connectedness เป็นวิธีที่อาศัยการพิจารณากลุ่มเพื่อนบ้านที่อยู่ใกล้ชิดที่สุดหรือเพื่อนบ้านที่มีระยะใกล้ที่สุด ด้วยการวัดระยะจากสมาชิกตัวที่ใกล้ที่สุดในกลุ่มกับสมาชิกตัวที่ใกล้ที่สุดของอีกกลุ่ม (Maimon and Rokach, 2010) แสดงดังรูปที่ 2.10

ซึ่งถ้าพบว่ามีความคล้ายคลึงกัน จะจับให้เป็นกลุ่มที่มีค่าความคล้ายคลึงมากที่สุดระหว่างสมาชิกของกลุ่มไปยังสมาชิกของอีกกลุ่ม (Sneath and Sokal, 1973)



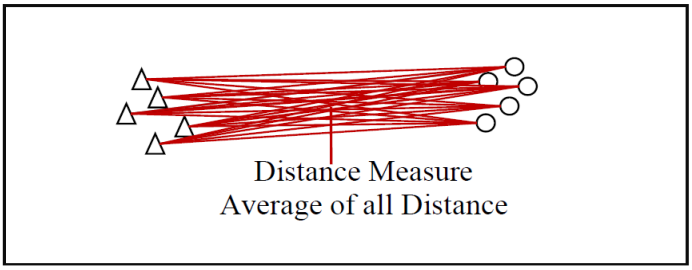
รูปที่ 2.10 ตัวอย่างการวัดระยะห่างระหว่างกลุ่มแบบ Single Link

(2.2) Complete Link หรือ Diameter เป็นวิธีที่อาศัยการพิจารณากลุ่มเพื่อนบ้านที่อยู่ไกลที่สุดหรือเพื่อนบ้านที่มีระยะห่างมากที่สุด ด้วยการหาระยะจากสมาชิกในกลุ่มตัวที่ไกลที่สุด กับสมาชิกตัวที่ไกลที่สุดของอีกกลุ่ม (King, 1967) แสดงดังรูปที่ 2.11



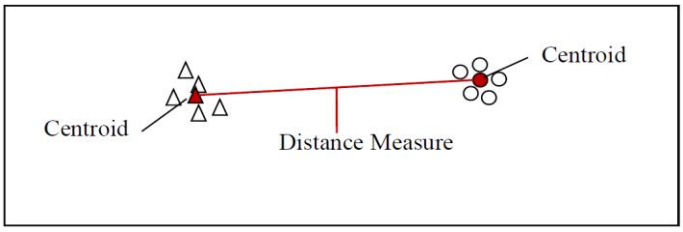
รูปที่ 2.11 ตัวอย่างการวัดระยะห่างระหว่างกลุ่มแบบ Complete Link

(2.3) Average Link เป็นวิธีที่อาศัยการพิจารณาระยะระหว่างกลุ่มด้วยค่าเฉลี่ยของระยะจากสมาชิกทุกตัวในกลุ่มกับสมาชิกทุกตัวของอีกกลุ่ม (Maimon and Rokach, 2010) แสดงได้ดังรูปที่ 2.12



รูปที่ 2.12 ตัวอย่างการวัดระยะห่างระหว่างกลุ่มแบบ Average Link

(2.4) Centroid Method หรือ Mean Distance เป็นวิธีที่อาศัยการวัดระยะห่างระหว่างจุดศูนย์กลางของกลุ่มสองกลุ่ม (Mohammed and Wagner, 2014) แสดงได้ดังรูปที่ 2.13



รูปที่ 2.13 ตัวอย่างการวัดระยะห่างระหว่างกลุ่มแบบ Centroid Method

(3) อัลกอริทึม Agglomerative

สำหรับการทำงานของอัลกอริทึม Agglomerative แสดงได้ดังรูปที่ 2.14

Algorithm: Agglomerative Hierarchical Clustering (Mohammed and Wagner, 2014)

AgglomerativeClustering (D,k)

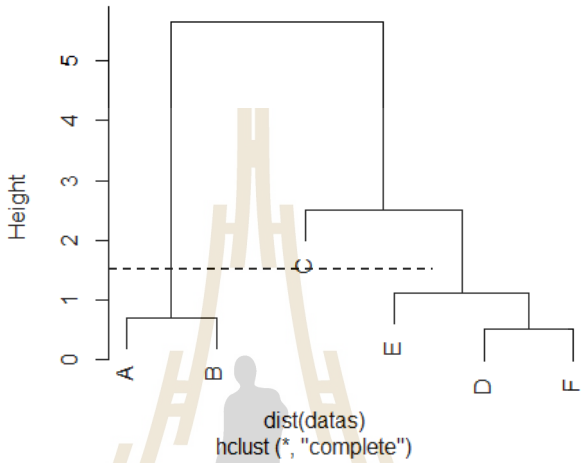
1 $C \leftarrow \{C_i = \{x_i\} | x_i \in D\}$ //Each point in separate cluster
2 $\Delta \leftarrow \{\delta(x_i, x_j) : x_i, x_j \in D\}$ //Comput distance matrix
3 **repeat**
4 Find the closest pair of clusters $C_i, C_j \in C$
5 $C_{ij} \leftarrow C_i \cup C_j$ // Merge the clusters
6 $C \leftarrow C \backslash \{\{C_i\} \cup \{C_j\}\} \cup \{C_{ij}\}$ // Update the clustering
7 Update distance matrix Δ to reflect new clustering
8 **until** $|C| = k$

รูปที่ 2.14 Agglomerative Hierarchical Clustering Algorithm

จากรูปที่ 2.14 ขั้นตอนการทำงานของอัลกอริทึม Agglomerative มี 8 ขั้นตอน อธิบายการทำงานได้ดังนี้ ขั้นที่ 1 คือการแยกข้อมูลทุกตัวออกเป็นกลุ่มแต่ละกลุ่ม (บรรทัดที่ 1) ขั้นที่ 2 เป็นการคำนวณเพื่อหาเมตริกซ์ระยะทางระหว่าง x_i และ x_j (บรรทัดที่ 2) ขั้นที่ 3-8 ดำเนินการไปตามลำดับจนกระทั่งจัดกลุ่มได้จำนวนตามต้องการ (k) (บรรทัดที่ 3-8) โดย ขั้นที่ 4 หากลุ่มที่มีความใกล้ชิดกันมากที่สุด ขั้นที่ 5 ผสานกลุ่ม C_i และC_j ที่ตรวจพบว่ามีความใกล้ชิดกันมากที่สุด ขั้นที่ 6 ทำการปรับปรุงการจัดกลุ่ม ขั้นที่ 7 ปรับปรุงเมตริกซ์ระยะทาง แสดงดังรูปที่ 2.15

(4) การตัด Dendrogram

เป็นขั้นตอนสำหรับการจัดกลุ่มข้อมูลให้มีจำนวน k กลุ่ม ซึ่งจะทำการตัด Dendrogram ในระดับที่จะแยกข้อมูลออกเป็น k กลุ่มตามต้องการ ตัวอย่างการจัดกลุ่มข้อมูลเมื่อ k=3 จะตัด Dendrogram ที่ระดับความสูงซึ่งสามารถแบ่งข้อมูลได้ 3 กลุ่ม ดังตัวอย่างตามรูปที่ 2.15 ซึ่งสามารถแบ่งข้อมูลเป็นกลุ่มได้ดังนี้ กลุ่มที่ 1 = {A, B} กลุ่มที่ 2 = {C} และกลุ่มที่ 3 = {D, E, F}



รูปที่ 2.15 ตัวอย่างการตัด Dendrogram ที่ k=3

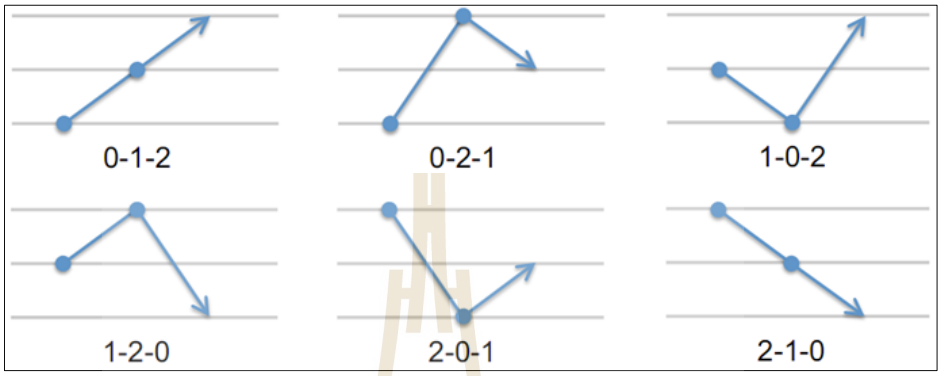
2.2.2 Permutation Distribution Clustering

อัลกอริทึม Permutation Distribution Clustering หรือ PDC เป็นการจัดกลุ่มอนุกรมเวลาแบบ Complexity-based เสนอโดย Brandmaier (2015) แนวคิดหลักคือการจัดรูปแบบความคล้ายคลึงของอนุกรมเวลาให้มีความซับซ้อนน้อยลง โดยตัวแทนของความซับซ้อนเลือกจากการกระจายของ Permutation โดยพิจารณาจาก Permutation Entropy อัลกอริทึมนี้เป็นวิธีที่ดีสำหรับการวิเคราะห์ความซับซ้อนของชุดข้อมูลจากหลากหลายแขนง หลักการทำงานที่เกี่ยวข้องมีดังนี้

(1) Permutation Distribution

Permutation Distribution (PD) ในการจัดกลุ่มแบบ PDC จะอาศัยการวัดความต่างของอนุกรมเวลาด้วยการคำนวณ PD ถูกใช้เป็นตัวแทนของอนุกรมแต่ละสาย ซึ่งจะมีการให้ค่าความน่าจะเป็นกับรูปแบบที่จะเกิดขึ้นในอนุกรมเวลา จนในที่สุดอนุกรมเวลาทั้งสายจะถูกแบ่งให้เป็นลำดับย่อยที่มีขนาดคงที่ m ซึ่งจะเรียกว่า Embedding ใน m-Space ในการคำนวณ PD ตามลำดับเวลา Embedding สามารถเป็น Time-Delayed ด้วย Delay t ดังนั้นจะมีเพียงสมาชิกทุกช่วง t-th ที่จะถูกพิจารณาเมื่อทำการแยกลำดับย่อย โดยลำดับของค่าที่ถูกคำนวณของอนุกรมทั้งสายจะเป็นผลจากการเรียงของค่าสังเกต ซึ่งค่า PD จะถูกกำหนดด้วยการนับความถี่สัมพัทธ์ของรูปแบบ

(Patterns) ที่ไม่ซ้ำกันเรียกว่า Ordinal Patterns โดยแต่ละลำดับรูปแบบที่เป็นไปได้สามารถระบุโดยการสับเปลี่ยนค่า 0 ไปเป็น (m-1) ในความน่าเชื่อถือของลำดับเป็นลักษณะเฉพาะของ PD เนื่องจากการกระจายจะถูกกำหนดโดยการสังเกตค่าที่สัมพันธ์กับค่าสัมบูรณ์ของตัวอื่นแต่ละตัว ดังแสดงในรูปที่ 2.16 ที่แสดง Ordinal Patterns สำหรับ Embedding ที่มี m=3



รูปที่ 2.16 Ordinal Patterns สำหรับ Embedding ที่มี m=3

(ที่มา: Brandmaier, 2015)

จากรูปที่ 2.16 จะเห็นว่าแต่ละส่วนของอนุกรมเวลาจะเป็นการแทนหนึ่งใน 6 รูปแบบที่ไม่ซ้ำกันสำหรับ Embedding ขนาด 3 ตัวอย่างเช่น Ordinal Pattern แรกคือ (0-1-2) ครอบคลุม m ทั้ง 3 ค่าคือ 0, 1 และ 2 โดยไม่ต้องคำนึงถึงค่าสังเกตที่แท้จริง

สำหรับการนิยาม Permutation Distribution มีหลักการดังนี้ ให้

1) อนุกรมเวลา $X = \{x(i)\}_{i=0}^{T}$, เมื่อ $x(i) \in \mathbb{R}$

2) เมื่อ Time-Delay *t* embedding ของอนุกรม X มีขนาด m มิติ จะได้ $X' = \{[x(i), x(i+t), x(i+2t), \dots, x(i+(m-1)t]\}_{i=0}^{T'}$ เมื่อ $T' = T-(m-1)t$

3) $\Pi(\mathrm{x})$ เป็น Permutation สำหรับ $x \in \mathbb{R}^m$ ที่ผ่านการจัดเรียงแล้ว และสำหรับ Permutation Distribution ของ X' นิยามได้ดังนี้

$$p_\pi = \frac{\#\{x' \in X' | \Pi(x') = \pi)\}}{T'} \tag{2.3}$$

(2) Permutation Entropy

Permutation Entropy เป็นมาตรวัดอย่างหนึ่งสำหรับใช้ในการประเมิน PD ซึ่งใน PD จะมีลำดับชั่วขณะของ Ordinal Pattern โดยจะถูกยกเลิกและจะใช้เพียงการกระจายความถี่

ของรูปแบบที่ไม่ซ้ำกันในอนุกรมเวลา ส่วน Permutation Entropy ของลำดับที่มี m >= 2 ของการกระจายความน่าจะเป็น P (Bandt and Pompe, 2002) ได้ดังนี้

$$H(P) = -\sum_{\pi \in S_m} p_\pi \log p_\pi \qquad (2.4)$$

เมื่อ S_m เป็นค่าเริ่มต้นของเซตของ m-Permutations ทั้งหมด

ความแตกต่างระหว่างสองอนุกรมเวลาสามารถจัดรูปแบบให้เป็นความแตกต่างที่แทนด้วย PD แต่สำหรับ Kullback-Leigler (KL) Divergence รู้จักกันในชื่อของ Shannon Entropy สัมพัทธ์ มักถูกนำมาใช้เป็นมาตรวัดความต่างระหว่างการกระจายของความน่าจะเป็นกับการขยายตัวตามธรรมชาติของ Entropy เหมือนเป็นดัชนีความซับซ้อนที่คำนวณ Permutation Entropy สัมพัทธ์ เป็นดัชนีความซับซ้อนสัมพัทธ์ อย่างไรก็ตาม KL Divergence ไม่ยึดหลักการของ Triangle inequality จึงไม่เป็นเมตริกซ์ ดังนั้นจึงมีการนำ The Squared Hellinger Distance มาแทน PD ในอนุกรมเวลาที่อยู่ใน Metric Space ซึ่ง Metric Space แบบนี้เป็นวิธีการที่ช่วยให้การทำงานเร็วขึ้นได้มาก ตามขนาดของ Squared Hellinger Distance ซึ่งเท่ากับค่ายูคลิดนอร์ม ของผลต่างของ Square Root Vectors ของการกระจายของความน่าจะเป็นแบบไม่ต่อเนื่อง เมื่อให้ $P= (p_1, p_2, \ldots, p_n)$ และ $Q= (q_1, q_2, \ldots, q_n)$ เป็น PD สองตัว สำหรับ Squared Hellinger Distance นิยามได้ดังนี้

$$D(P,Q) = \frac{1}{\sqrt{2}} \left\| \sqrt{P} - \sqrt{Q} \right\|_2^2 \qquad (2.5)$$

โดย Squared Hellinger Distance ได้จากค่าเฉลี่ยของการประมาณการของ KL Divergence โดยเป็นเมตริกซ์เนื่องจากเป็นไปตาม Triangle inequality ซึ่งสมมาตร ไม่ติดลบ และมีค่าระหว่าง 0-1 การวัดระยะ Squared Hellinger Distance ระหว่างชุดของ PD ใน Distance Matrix สามารถเป็นทางเลือกในการนำมาใช้ในการจัดกลุ่มสำหรับนักวิจัยได้

2.2.3 อัลกอริทึม k-Means Clustering

อัลกอริทึม k-Means เป็นเทคนิคการจัดกลุ่มที่อาศัยการวัดระยะระหว่างข้อมูลแต่ละตัวกับจุดศูนย์กลาง (Centroid) ของแต่ละกลุ่ม (Shahbaba and Beheshti, 2014) ในการจัดกลุ่มแบบ k-Means ตัวแทนของกลุ่มจะใช้ค่าเฉลี่ย (Means) ของข้อมูลที่สังกัดในกลุ่มเป็นจุดศูนย์กลาง โดยอาศัยวิธีวัดระยะแบบ Euclidean Distance สำหรับ k-Means เป็นวิธีที่มีการทำงานง่าย ประมวลผลเร็ว (Theodoridis et al., 2010) ขั้นตอนการทำงานเข้าใจง่ายไม่ซับซ้อน มีหลักการที่เกี่ยวข้องดังนี้

(1) ค่าเฉลี่ยของกลุ่ม

ดังที่กล่าวไปแล้วข้างต้นว่าการจัดกลุ่มแบบ k-Means ใช้ค่าเฉลี่ยของสมาชิกที่สังกัดในกลุ่มเป็นตัวแทนและเป็นจุดศูนย์กลางของกลุ่ม กำหนดให้ชุดข้อมูลหนึ่งมีจำนวนข้อมูลทั้งหมด n ตัว อยู่ในปริภูมิเวกเตอร์ขนาด d-มิติ แทนด้วย D= $\{x_i\}_{i=1}^{n}$ และให้ k คือจำนวนกลุ่มที่ต้องการแบ่ง กำหนดให้ $C = \{C_1, C_2, \dots, C_k\}$ สำหรับในแต่ละ Cluster ที่ i แทนด้วย C_i ดังนั้นตัวแทนของกลุ่ม C_i ที่จะถูกใช้เป็นจุดศูนย์กลางคือค่าเฉลี่ย μ_i คำนวณได้ดังนี้ (Mohammed and Wagner, 2014)

$$\mu_i = \frac{1}{n_i} \sum_{x_j \epsilon C_i} x_j \tag{2.6}$$

เมื่อ $n_i = |C_i|$ แทน จำนวนสมาชิกที่สังกัดใน Cluster C_i

x_j แทน ข้อมูลที่เป็นสมาชิกใน Cluster C_i ซึ่งจะอยู่ใกล้กับจุดศูนย์กลางของ Cluster C_i มากกว่าจุดศูนย์กลางของ Cluster อื่น

(2) การกำหนดกลุ่มให้กับข้อมูล

การกำหนดกลุ่มให้กับข้อมูล หรือวัตถุใด ๆ ในการจัดกลุ่ม ซึ่งเป็นรูปแบบการเรียนรู้แบบ Unsupervised Learning จำเป็นจะต้องมีค่าคะแนนบางอย่างเป็นตัวชี้วัดหรือประเมินว่ากลุ่มที่จัดให้กับวัตถุแต่ละตัวนั้นเหมาะสมดีแล้วหรือไม่ ซึ่งในที่นี้จะใช้ผลรวมของค่าความผิดพลาด (Sum of Squared Errors: SSE) (Mohammed and Wagner, 2014) ดังนั้นการทำงานของ k-Means จะมุ่งเน้นหา SSE ที่มีค่าต่ำสุดโดยอาศัยการทำงานแบบ Greedy ซึ่งหลักการนี้อาจเกิดปัญหาที่ทำให้ผลลัพธ์ลู่เข้าสู่ค่าที่เหมาะสมเฉพาะถิ่น (Local Optimum) มากกว่าจะลู่เข้าสู่ค่าที่เหมาะสมสาธารณะ (Global Optimum)

(3) อัลกอริทึม k-Means

สำหรับอัลกอริทึม k-Means มีขั้นตอนการทำงาน 10 ขั้นตอน อธิบายการทำงานได้ดังนี้ ขั้นที่ 1 กำหนดตัวนับจำนวนรอบ t=0 (บรรทัดที่ 1) ขั้นที่ 2 สุ่มเลือกข้อมูลในชุดข้อมูลทั้งหมดมาเท่ากับจำนวน k และกำหนดให้เป็นจุดศูนย์กลางของแต่ละกลุ่ม (บรรทัดที่ 2) ขั้นที่ 3-10 เป็นการทำงานวนซ้ำเพื่อจัดกลุ่มให้เหมาะสม ด้วยการจัดข้อมูลให้สังกัดในกลุ่มที่ใกล้ที่สุด (บรรทัดที่ 5-7) และปรับปรุงจุดศูนย์กลาง (บรรทัดที่ 8-9) โดยจะหยุดทำงานเมื่อตรวจสอบพบว่าค่า Means ที่ใช้เป็นจุดศูนย์กลางของรอบใหม่ไม่เปลี่ยนจากเดิมหรือมีค่าน้อยกว่าหรือเท่ากับค่า Threshold (ϵ) ซึ่งกำหนดให้ $\epsilon > 0$ (บรรทัดที่ 10) แสดงการทำงานดังรูปที่ 2.17

Algorithm : K-means Clustering (Mohammed and Wagner, 2014)

K-MEANS (D,k,ϵ)

1 t=0

2 Randomly initialize k centroids: $\mu_1^t, \mu_2^t, \ldots, \mu_k^t \in \mathbb{R}^d$

repeat

3 t $\longleftarrow$ t + 1

4 // Cluster Assignment Step

5 **foreach** $x_j \in D$ do

6 $j^* \leftarrow \arg\min_i \{\|x_j - \mu_i^t\|^2\}$ //Assign x_j to closest centroid

7 $C_{j*} \leftarrow C_{j*} \cup \{x_j\}$

// Centroid Update Step

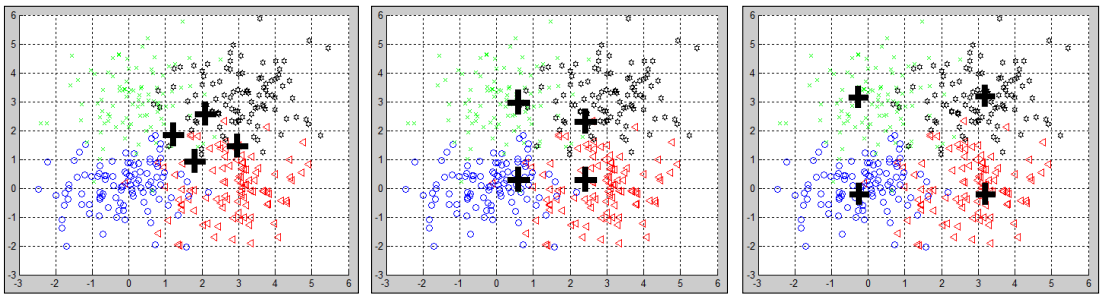
8 **foreach** $i = 1\ to\ k$ do

9 $\mu_i^t \leftarrow \frac{1}{|C_i|} \sum_{x_j \in C_i} x_j$

10 **until** $\sum_{i=1}^{k} \|\mu_i^t - \mu_2^{t-1}\| \leq \epsilon$

รูปที่ 2.17 The k-Means Clustering Algorithm

จากรหัสเทียมอัลกอริทึม k-Means สามารถแสดงภาพจำลองการทำงานการจัดกลุ่มที่มีค่า k=4 โดยข้อมูลที่ถูกจัดแทนด้วยสัญลักษณ์ที่แตกต่างกันและมีสัญลักษณ์ “+” แทนจุดศูนย์กลางของกลุ่มแต่ละกลุ่ม ได้ดังรูปที่ 2.18 (a) เมื่อเริ่มจัดกลุ่มจุดศูนย์กลางอาจอยู่ในตำแหน่งที่ไม่เหมาะสม แต่เมื่อมีการจัดกลุ่มต่อไปเรื่อย ๆ ตราบเท่าที่จุดศูนย์กลางยังเปลี่ยนแปลง ดังรูป 2.18(b) และจะหยุดการทำงานเมื่อจุดศูนย์กลางคงที่ไม่เปลี่ยนแปลง ตัวอย่างดังรูป 2.18 (c)

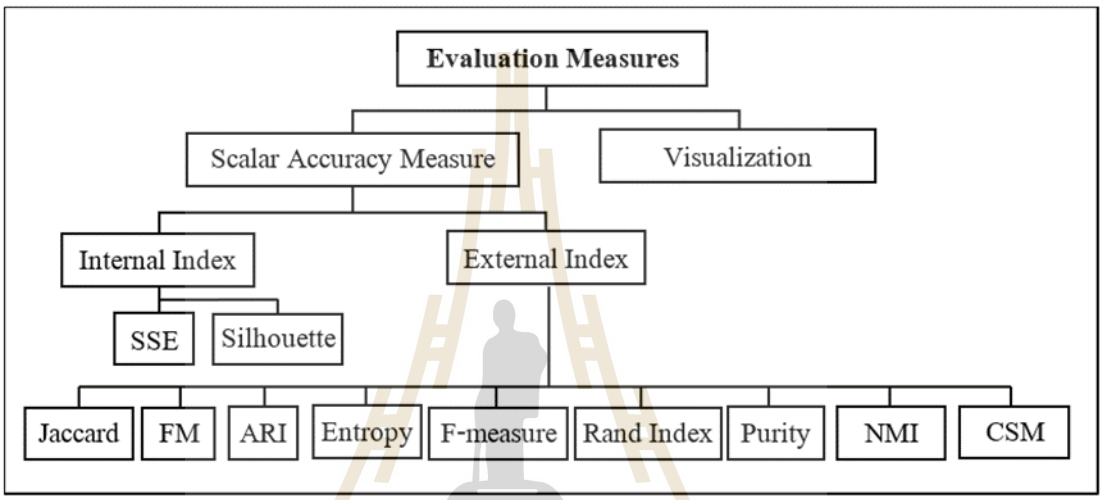


(a) เมื่อค่า Means เริ่มต้น (b) เมื่อค่า Means เปลี่ยนแปลง (c) เมื่อค่า Means คงที่

รูปที่ 2.18 ขั้นตอนการจัดกลุ่มข้อมูล ด้วยเทคนิควิธีแบบ k-Means เมื่อจัดข้อมูลเป็น 4 กลุ่ม

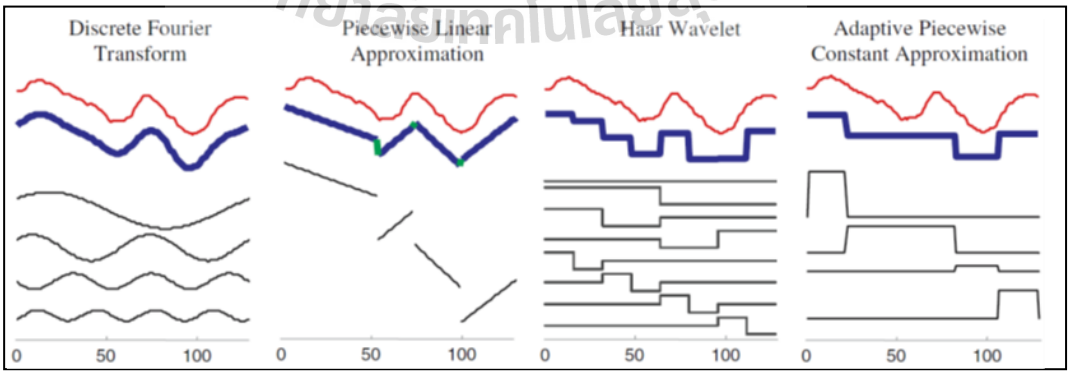
2.3 การหาตัวแทนอนุกรมเวลา

การหาตัวแทนอนุกรมเวลา ถือเป็นงานปกติสำหรับการจัดกลุ่มข้อมูลอนุกรมเวลาประเภท Whole Time-Series โดยทั่วไปวิธีการกำหนดตัวแทนอนุกรมเวลาแบ่งได้เป็น 4 ประเภท (Lin et al., 2003; Ratanamahatana et al., 2005; Bagnall et al., 2006; Shieh and Keogh, 2009) แต่ที่เป็นที่นิยมได้แก่ประเภท Data Adaptive และ Non Data Adaptive ซึ่งมีการวิจัยเทคนิคการแทนอนุกรมเวลาเพิ่มขึ้นมากมาย ดังจะเห็นได้จากแผนผังเทคนิคการแทนอนุกรมเวลา (Lin et al., 2007) ในรูปที่ 2.19



รูปที่ 2.19 แผนผังเทคนิคการกำหนดตัวแทนอนุกรมเวลา

ในบางเทคนิค เช่น Discrete Fourier Transform, Piecewise Linear Approximation, Haar Wavelet และ Adaptive Piecewise Constant Approximation ซึ่งเป็นเทคนิคที่มีการทำงานคล้ายคลึงกันและนิยมในการวิจัยต่อยอด แสดงหลักการอย่างง่ายดังรูปที่ 2.20



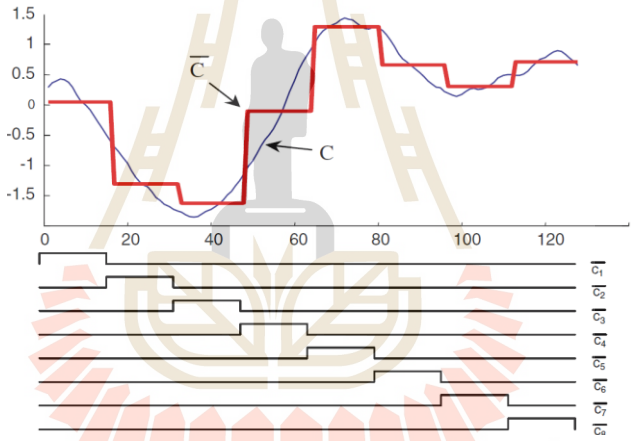
รูปที่ 2.20 ตัวอย่างการเปรียบเทียบการกำหนดตัวแทนอนุกรมเวลา

(ที่มา: Lin et al., 2007)

จากรายละเอียดในข้างต้นจะเห็นว่ามีเทคนิคการแทนอนุกรมเวลาที่น่าสนใจหลายเทคนิค ซึ่งในส่วนนี้จะกล่าวถึงสองเทคนิคที่หยิบมาใช้เปรียบเทียบในงานวิจัยนี้ ได้แก่

2.3.1 การแทนอนุกรมเวลาด้วยวิธี PAA

การแทนอนุกรมเวลาวิธี PAA (Piecewise Aggregate Approximation) เป็นวิธีที่จะทำงานโดยแบ่งอนุกรมเวลาเป็นส่วนแต่ละส่วนขนาดเท่ากัน ซึ่งตัวแทนอนุกรมเวลา คือตัวแทนจากแต่ละส่วน วิธีการแบบ PAA ถูกเสนอโดย Keogh และคณะ (2000) โดยข้อมูลอนุกรมเวลาหนึ่งสายขนาด n มิติ จะถูกลดมิติด้วยการแบ่งเป็นส่วนย่อยขนาดเท่ากันขนาด w มิติ แต่ละส่วนเรียกว่าเฟรม (Frames) จากนั้นจะคำนวณหาค่าเฉลี่ยของสมาชิกภายในเฟรมแต่ละเฟรมจะได้เป็นเวคเตอร์ของค่าเฉลี่ยซึ่งกลายเป็นตัวแทนอนุกรเวลา แผนภาพแสดงการแทนอนุกรมเวลาด้วยวิธี PAA สำหรับอนุกรมเวลาที่มีขนาด 128 มิติ และแบ่งเป็นส่วนย่อยขนาด 16 มิติ จะได้จำนวน 8 ชุดย่อย จากนั้นเปลี่ยนชุดย่อย 8 ชุดให้เป็น 8 มิติ ดังรูปที่ 2.21 ดังนี้



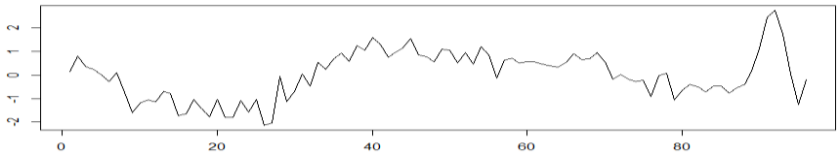
รูปที่ 2.21 การแทนอนุกรมเวลา C ด้วยวิธี PAA จากอนุกรมเวลา 128 มิติ ให้เป็น 8มิติ (ที่มา: Lin et al., 2007)

กำหนดให้ อนุกรมเวลา C มีขนาดความยาว n ซึ่งสามารถแทนให้อยู่ในขนาด w-มิติ จะได้เวคเตอร์ $\bar{C} = \bar{c_1}, \ldots, \bar{c_w}$ เมื่อ $\bar{c_i}$ คือสมาชิกตัวที่ i-th ของ $\bar{C}$ คำนวณได้ดังนี้

$$\bar{c_i} = \frac{w}{n} \sum_{j=\frac{n}{w}(i-1)+1}^{\frac{n}{w}i} c_j \qquad (2.7)$$

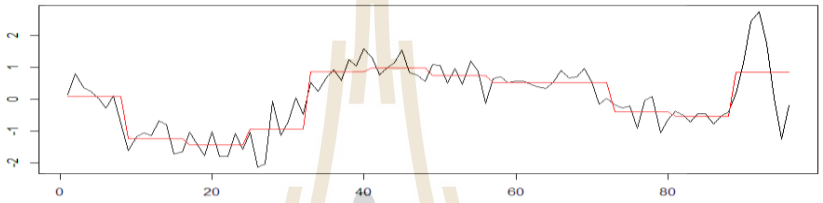
โดยที่ j แทน ตำแหน่งข้อมูลที่ถูกใช้คำนวณหา สมาชิกตัวที่ i-th ของ $\bar{C}$

ตัวอย่าง การแทนอนุกรมเวลาด้วยวิธี PAA สำหรับข้อมูลคลื่นไฟฟ้าหัวใจ 1 อนุกรม โดยคลื่นไฟฟ้าหัวใจที่นำมาใช้มีขนาดเวลายาว 96 จุดเวลา (96 มิติ) แสดงดังรูปที่ 2.22



รูปที่ 2.22 กราฟแสดงรูปร่างของข้อมูลอนุกรมเวลา ECGs อนุกรม S2

เมื่อแทนอนุกรมเวลา S2 ด้วยเทคนิค PAA โดยกำหนดให้ลดมิติจาก 96 เป็น w = 12 มิติ ดังนั้นอนุกรมจะถูกแปลงให้เป็นข้อมูลแบบไม่ต่อเนื่องสามารถพล็อตเป็นภาพแสดงลักษณะที่เปลี่ยนไปได้ดังรูปที่ 2.23 ดังนี้



รูปที่ 2.23 การลดมิติด้วยวิธี PAA ของอนุกรมเวลา S2

เมื่อพิจารณาระหว่างข้อมูลอนุกรมเวลาดั้งเดิม S2 และตัวแทนอนุกรมที่ได้จากเทคนิค PAA ซึ่งมีขนาด 60 มิติ (X1-X96) แสดงได้ดังตารางที่ 2.1

ตารางที่ 2.1 ข้อมูลดั้งเดิมของอนุกรมเวลา S2 ที่มีขนาด 96 มิติ

ข้อมูลดั้งเดิมของอนุกรม S2									
X1	0.147650	X13	-0.686040	X25	-1.036100	X37	0.573040	X49	1.084600
X2	0.804670	X14	-0.798790	X26	-2.130000	X38	1.231600	X50	1.057400
X3	0.367770	X15	-1.714300	X27	-2.038600	X39	1.043300	X51	0.503080
X4	0.243890	X16	-1.649900	X28	-0.055013	X40	1.580200	X52	0.953290
X5	0.026614	X17	-1.032700	X29	-1.140200	X41	1.290600	X53	0.454200
X6	-0.274400	X18	-1.423900	X30	-0.704280	X42	0.750520	X54	1.199000
X7	0.096731	X19	-1.782400	X31	0.048723	X43	0.977570	X55	0.849720
X8	-0.747730	X20	-1.025100	X32	-0.475850	X44	1.141600	X56	-0.137710
X9	-1.609800	X21	-1.798900	X33	0.525870	X45	1.529900	X57	0.631120
X10	-1.179600	X22	-1.793600	X34	0.244780	X46	0.847680	X58	0.694080
X11	-1.055900	X23	-1.085500	X35	0.679930	X47	0.766760	X59	0.515070
X12	-1.128800	X24	-1.574400	X36	0.928600	X48	0.565680	X60	0.559230

ตารางที่ 2.1 ข้อมูลดั้งเดิมของอนุกรมเวลา S2 ที่มีขนาด 96 มิติ (ต่อ)

ข้อมูลดั้งเดิมของอนุกรม S2									
X61	0.555710	X69	0.952600	X77	-0.037932	X85	-0.460440	X93	1.736100
X62	0.469690	X70	0.547770	X78	0.071668	X86	-0.770790	X94	0.036857
X63	0.388820	X71	-0.167690	X79	-1.048900	X87	-0.533500	X95	-1.265100
X64	0.343130	X72	0.011532	X80	-0.636220	X88	-0.400230	X96	-0.208020
X65	0.535600	X73	-0.176490	X81	-0.386640	X89	0.176080		
X66	0.900710	X74	-0.277710	X82	-0.507510	X90	1.111800		
X67	0.656850	X75	-0.224840	X83	-0.716660	X91	2.438400		
X68	0.690030	X76	-0.910720	X84	-0.463760	X92	2.734900		

ข้อมูลอนุกรมเวลา S2 เมื่อแทนอนุกรมด้วยเทคนิค PAA ได้ผลลัพธ์เป็นข้อมูลที่มีมิติต่ำลงเป็น 12 มิติ ดังตารางที่ 2.2

ตารางที่ 2.2 ข้อมูลตัวแทนอนุกรมเวลา S2 ขนาด 12 มิติ ที่ได้จากเทคนิค PAA

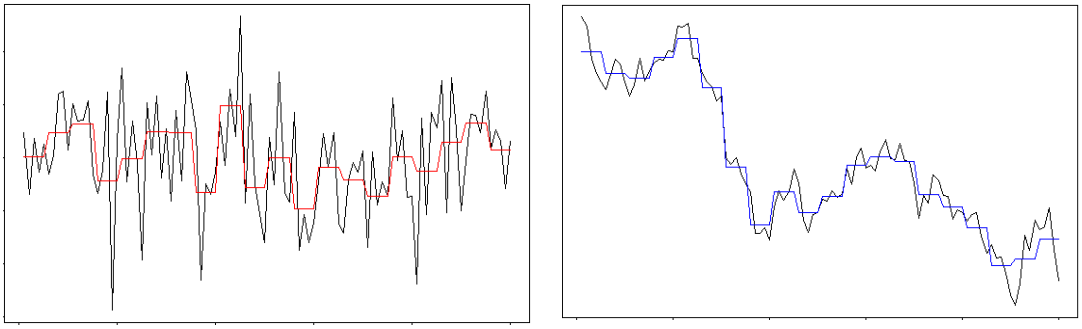
ตัวแทนอนุกรมเวลาของอนุกรม S2							
X1	0.08315	X4	-0.94140	X7	0.74545	X10	-0.40510
X2	-1.22790	X5	0.85091	X8	0.51961	X11	-0.52990
X3	-1.43960	X6	0.98379	X9	0.51592	X12	0.84513

2.3.2 การแทนอนุกรมเวลาด้วยวิธี SAX

การแทนอนุกรมเวลาด้วยวิธี SAX (Symbolic Aggregate ApproXimation) เสนอโดย Lin และคณะ (2007) เป็นวิธีที่ช่วยลดมิติข้อมูลอนุกรมเวลาและลดปริมาณการคำนวณ ด้วยการแปลงข้อมูลให้เป็นสัญลักษณ์ซึ่งมีการประยุกต์ใช้งานวิธีแบบ PAA และเทคนิค Lower Bound ซึ่งเป็นวิธีการวัดระยะที่สามารถนิยามได้ตามข้อมูลดั้งเดิม ผลงานวิจัยแสดงให้เห็นถึงคุณภาพของการนำวิธีการแทนข้อมูลนี้ไปใช้ในงานเหมืองข้อมูล หลายงาน ได้แก่ การจัดกลุ่มข้อมูล การจำแนกข้อมูล การระบุดัชนี (Indexing) การตรวจจับความผิดปกติในอนุกรมเวลา การค้นหา Motif และการแสดงผลด้วยแผนภาพ ขั้นตอนการทำงานของเทคนิค SAX ทำงานได้ดังตัวอย่างต่อไปนี้

ตัวอย่าง การหาตัวแทนอนุกรมเวลา x และ y ที่ขนาดเวลา 100 จุดเวลา (100 มิติ) โดยปรับลดมิติให้มีขนาด 20 มิติ ซึ่งมีขั้นตอนการทำงาน 2 ส่วนหลักดังนี้

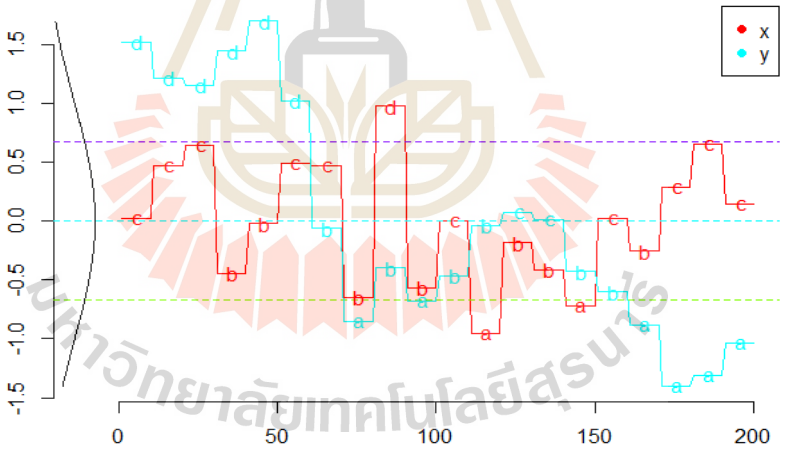
ขั้นที่ 1 การแทนข้อมูลดังเดิมด้วยวิธี PAA ซึ่งแสดงผลการแทนอนุกรมเวลา x และ y ได้ดังรูปที่ 2.24 (a) และ (b) ตามลำดับ



(a) Time-Series x (b) Time-Series y

รูปที่ 2.24 การลดมิติแบบ PAA ของอนุกรมเวลา x และ y

ขั้นที่ 2 การเปลี่ยนตัวแทนอนุกรมเวลา x และ y ที่ได้จากเทคนิค PAA ให้เป็นสัญลักษณ์ด้วยเทคนิค SAX ซึ่งสัญลักษณ์แทนระดับ (กำหนดโดยผู้ใช้) ของตัวแทนจาก PAA ตัวอย่างตัวแทนอนุกรม x และ y ที่ได้จากเทคนิค SAX แปลงเป็นสัญลักษณ์ได้เป็น “cccbbccbdbcabbacbccc” สำหรับอนุกรม x และถูกแปลงได้เป็น “ddddddbababccbbaaaa” สำหรับอนุกรม y ดังรูปที่ 2.25

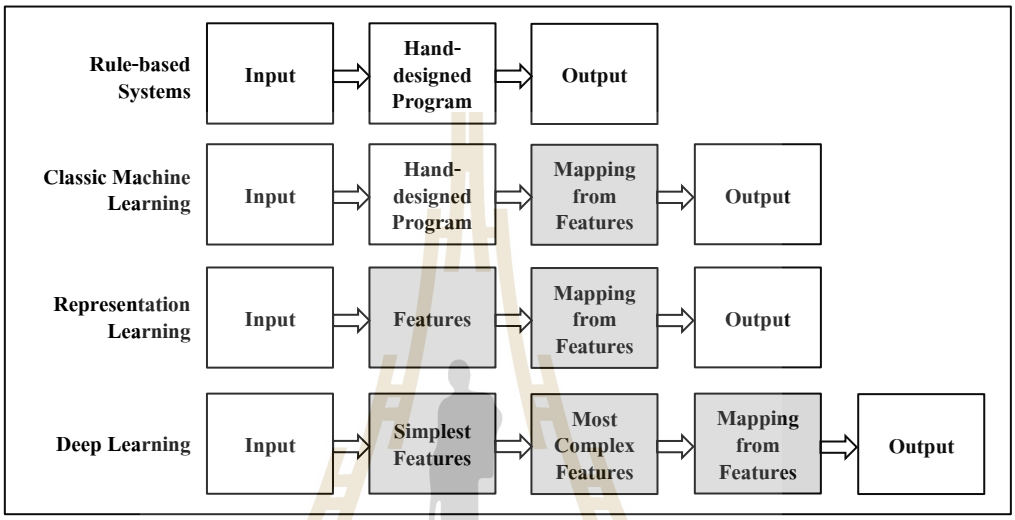


รูปที่ 2.25 ตัวแทนของอนุกรมเวลา x และ y ที่อยู่ในรูปของสัญลักษณ์

2.4 สถาปัตยกรรมโครงข่ายการเรียนรู้เชิงลึก

โครงข่ายการเรียนรู้เชิงลึก (Deep Learning Networks) เป็นเทคนิคการเรียนรู้ของเครื่องจักรที่มีประสิทธิภาพอีกเทคนิคหนึ่ง ถูกเรียกในหลายชื่อได้แก่ การเรียนรู้เชิงลึก (Deep Learning) หรือโครงสร้างการเรียนรู้เชิงลึก (Deep Structured Learning) หรือการเรียนรู้แบบลำดับชั้น (Hierarchical Learning) หรือ การเรียนรู้ของเครื่องเชิงลึก (Deep Machine Learning) โดยเป็นที่นิยมใช้สำหรับการแทนข้อมูลที่มีมิติสูง สามารถทำงานได้ทั้งแบบมีผู้สอน (Supervised) หรือไม่มีผู้สอน

(Unsupervised) (Bengio et al., 2013; Schmidhuber, 2015; Bengio et al., 2015) ซึ่งเป็นรูปแบบสถาปัตยกรรมการเรียนรู้ที่ประกอบด้วยการทำงานย่อยหลายชั้น ซึ่งแต่ละชั้นได้มาจากการแปลงและส่งผ่านข้อมูลกันระหว่างชั้นต่อชั้น (Goodfellow et al., 2016; LeCun et al., 2015; Deng and Yu, 2014; Olshausen, 1996) เพื่อพยายามสร้างแบบจำลองขึ้นมาแทนความหมายของข้อมูลในระดับสูง แนวคิดการทำงานดังแสดงในรูปที่ 2.26



รูปที่ 2.26 การเปรียบเทียบแนวคิดการเรียนรู้เชิงลึก

จากรูปที่ 2.26 แสดงให้เห็นแนวคิดการเรียนรู้เชิงลึกซึ่งเป็นรูปแบบการเรียนเพื่อกำหนดตัวแทนข้อมูลที่ประกอบด้วย Feature หลายระดับ โดยค้นหา Feature ที่เหมาะสมและรวม Feature เข้าด้วยกันในระดับอื่นเพื่อผลิตผลลัพธ์ให้ได้ตามต้องการ สำหรับหลักการของสถาปัตยกรรมเชิงที่เกี่ยวข้องในงานวิจัยอธิบายได้ดังต่อไปนี้

2.4.1 โครงข่ายความเชื่อเชิงลึก (Deep Belief Networks)

โครงข่ายความเชื่อเชิงลึก (Deep Belief Networks: DBNs) เป็นสถาปัตยกรรมเชิงลึกรูปแบบหนึ่งที่มีรูปแบบโครงข่ายการเชื่อมต่อประกอบด้วยชั้น (Layer) การทำงานหลัก 3 ชั้น ได้แก่ ชั้น Input ชั้น Hidden และชั้น Output เช่นเดียวกับโครงข่ายประสาทเทียม แต่มีหลักการภายในที่แตกต่างออกไปคือ DBNs มุ่งเน้นที่ความน่าจะเป็นในการสร้างแบบจำลองที่ประกอบด้วยชั้นหลายชั้นของการสุ่มตัวแปรแฝง ตัวแปรแฝงมักมีค่าเป็นไบนารี และมักจะเรียกว่า Hidden Units หรือ Feature Detectors โดยการเชื่อมต่อสองชั้นบนจะไม่ถูกกำกับควบคุมทิศทางการเชื่อมต่อ แต่ในชั้นที่ต่ำลงไปจะมีทิศทางการรับจากบนลงล่าง โดยชั้นล่างจะถูกควบคุมการเชื่อมต่อโดยชั้นที่อยู่สูงกว่า คุณสมบัติที่สำคัญที่สุดของโครงข่ายความเชื่อเชิงลึกได้แก่:

1) เป็นวิธีการที่มีประสิทธิภาพของการทำงานในการเรียนรู้จากบนลงล่าง ซึ่งการผลิตน้ำหนัก (Weight) ในชั้นหนึ่งขึ้นกับตัวแปรในชั้นบนที่ติดกัน แบบ ชั้น-ต่อ-ชั้น

2) หลังจากเรียนรู้ค่าของตัวแปรแฝงที่อยู่ในทุกชั้นแล้วจะสามารถสรุปเป็นหนึ่งเดียวและส่งผ่านจากล่างขึ้นบน

3) โครงข่ายความเชื่อเชิงลึกที่เวลาใดจะเรียนรู้ในชั้นหนึ่ง โดยยังรักษาค่าของตัวแปรแฝงในชั้นนั้น เมื่อมีการเริ่มสรุปข้อมูลเพื่อส่งไปใช้ในการฝึกสอนชั้นถัดไป ประสิทธิภาพนี้สามารถใช้การเรียนรู้แบบละโมบ (Greedy) ตามด้วย/หรือรวมเข้า กับขั้นตอนการเรียนรู้อื่นที่ปรับแต่ง Weight ทั้งหมดเพื่อปรับปรุงประสิทธิภาพการสร้างหรือจำแนกของทั้งโครงข่ายได้ (Hinton, 2009)

4) การปรับปรุงการจำแนกสามารถดำเนินการได้โดยการเพิ่มชั้นสุดท้ายของตัวแปรที่เป็นตัวแทนของผลที่ต้องการและแพร่ย้อนกลับ (Backpropagation) อนุพันธ์ข้อผิดพลาด โครงข่ายที่มีชั้น Hidden จำนวนมากมักใช้กับข้อมูลที่โครงสร้างมิติสูง เช่นข้อมูลภาพ สำหรับ Backpropagation ทำงานได้ดีขึ้นมากถ้าตัวตรวจจับ Feature ในชั้น Hidden เริ่มต้นเรียนรู้กับโครงสร้างของข้อมูลเข้า (Hinton and Salakhutdinov, 2006) องค์ประกอบอย่างง่ายของโครงข่ายการเรียนรู้แบบไม่มีผู้สอน DBNs ได้แก่ คือ Restricted Boltzmann Machines (RBMs) หรือ Autoencoder (Hinton, 2009)

2.4.2 สถาปัตยกรรม Restricted Boltzmann Machines

Restricted Bolzmann Machines เป็นสถาปัตยกรรมที่พัฒนาต่อยอดโดยอาศัยหลักการอื่นเป็นฐาน ซึ่งประกอบด้วยหลักการที่เกี่ยวข้องดังนี้

(1) Deep Boltzmann Machines

Boltzmann Machines (BM's) คือโครงข่ายคู่สมมาตรของการสุ่มหน่วยไบนารีซึ่งจะบรรจุเซตของหน่วยตัวแปร $\mathrm{v} \in \{0,1\}^D$ และเซตของหน่วย Hidden $\mathrm{h} \in \{0,1\}^P$ โดยพลังงานของสถานะ {v, h} (Salakhutdinov and Hinton, 2009) นิยามได้ดังนี้

$$E(\mathrm{v},\mathrm{h};\theta) = \frac{1}{2}\mathrm{v}^{\mathrm{T}}\mathrm{L}\mathrm{v} - \frac{1}{2}\mathrm{h}^{\mathrm{T}}\mathrm{J}\mathrm{v} - \mathrm{v}^{\mathrm{T}}\mathrm{W}\mathrm{h} \qquad (2.8)$$

เมื่อ $\theta = \{\mathrm{W},\mathrm{L},\mathrm{J}\}$ เป็นพารามิเตอร์สำหรับแบบจำลอง W, L, J แทน Visible-to-Hidden, Visible-to-Visible และ Hidden-to-Hidden คือเทอมของการต่อเชื่อมแบบสมมาตร โดยสมาชิกในเส้นทแยงของ L และ J กำหนดให้เป็น 0

ในการศึกษา Deep Boltzmann Machines (DBMs) โดยทั่วไปไม่สนใจการเรียนรู้ที่ซับซ้อน แต่เรามักไปสนใจการเรียนรู้ Boltzmann Machines หลายชั้นเชิงลึกแทน DBMs มีความน่าสนใจด้วยเหตุผลหลายประการ (Salakhutdinov and Hinton, 2009) ดังนี้

1) DBMs มีความเป็นไปได้ที่แฝงอยู่ของการเรียนรู้ภายในตัวแทนที่กลายเป็นความซับซ้อนที่เพิ่มขึ้น เช่นเดียวกับโครงข่ายความเชื่อเชิงลึก (Deep Belief Networks: DBNs) ซึ่งเป็นแนวทางที่มีความเป็นไปได้ที่จะสามารถแก้ปัญหาการรู้จำเสียง

2) ตัวแทนในระดับที่สูงขึ้นสามารถสร้างได้จากส่วนจัดส่งขนาดใหญ่ของข้อมูลที่ไม่มีคลาสกำกับ และข้อมูลคลาสที่มีข้อจำกัดอย่างมาก สามารถใช้เพียงแค่การปรับแต่งแบบจำลองเพียงเล็กน้อยสำหรับเฉพาะงานได้

3) ที่ต่างจาก DBNs คือในขั้นตอนการสรุปการประมาณ ในการเพิ่มค่าจะส่งผ่านจากล่างขึ้นบน สามารถรวมย้อนกลับจากบนลงล่างได้ ซึ่งช่วยทำให้ DBMs สามารถแพร่กลับแบบไม่แน่นอนได้ดีกว่า จึงทำให้มีความทนทานมากขึ้นสำหรับข้อมูลเข้าที่ไม่ค่อยชัดเจนโดย BMs 2-Layer ที่ไม่มีชั้นภายในเชื่อมต่อกัน จะพบว่าพลังงานของสถานะ {v, h1, h2} (Salakhutdinov and Hinton, 2009) นิยามได้ดังนี้

$$E(\mathrm{v}, \mathrm{h}^1, h^2;\, \theta) = -\mathrm{v}^{\mathrm{T}}\mathrm{W}^1\mathrm{h}^1 - \mathrm{h}^{1\mathrm{T}}\mathrm{W}^2\mathrm{h}^2 \tag{2.9}$$

โดยที่ $\theta = \{\mathrm{W}^1, \mathrm{W}^2\}$ คือ พารามิเตอร์สำหรับแบบจำลอง แทนด้วย Visible-to-Hidden และ Hidden-to-Hidden ซึ่งเป็นเทอมของความสัมพันธ์แบบสมมาตร

(2) Restricted Boltzmann Machines (RBMs)

RBMs เป็นแบบจำลองความน่าจะเป็นที่อาศัยพลังงานเป็นฐาน และมีการกระจายความน่าจะเป็น นิยามจาก Energy Function ดังนี้ (Ripoll et al., 2016)

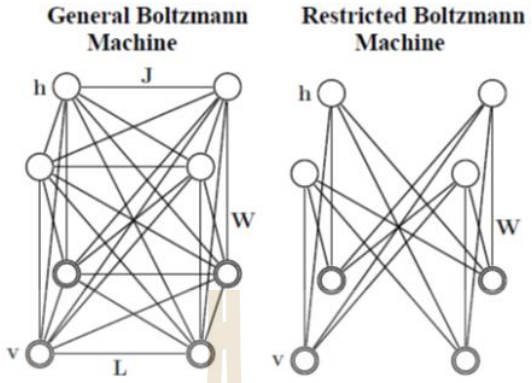
$$P(x, \mathrm{h}) = \frac{\mathrm{e}^{-Energy(x,h)}}{Z} \tag{2.10}$$

เมื่อ x คือข้อมูลเข้า h มีความเกี่ยวข้องกับ Hidden ที่ถูกแนะนำให้มีการเพิ่มการพลังงานของแบบจำลอง สำหรับปัจจัย Z เรียกว่า Partition Function ได้จาก (Ripoll et al., 2016)

$$Z = \sum_{x,h} \mathrm{e}^{-Energy(x,h)} \tag{2.11}$$

เมื่อเปรียบเทียบโครงข่ายของสถาปัตยกรรม RBMs กับ BMs จะพบว่า BMs ปกติจะเป็นโครงข่ายคู่สมมาตรที่ชั้นบนสุดเป็น Hidden และชั้นล่างสุดเป็นVisible ดังรูปที่ 2.27

(ซ้าย) แต่สำหรับ RBMs (Smolensky, 1986) จะเป็นโครงข่ายคู่สมมาตรของ Visible และ Hidden ที่ไม่มีการเชื่อมต่อภายในชั้น Visible และภายในชั้น Hidden (Hinton, 2007) ดังรูปที่ 2.27 (ขวา) ดังนี้



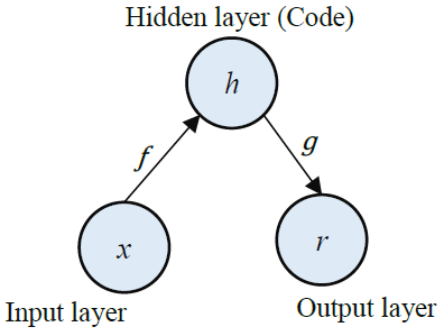
รูปที่ 2.27 แผนภาพโครงข่ายของ BMs และ RBMs
(ที่มา: Salakhutdinov and Hinton, 2009)

2.4.3 Deep Autoencoder Networks Architecture

Deep Autoencoder Networks เป็นสถาปัตยกรรมที่พัฒนาโดยอาศัยหลักการพื้นฐานของ RBMs และสถาปัตยกรรมที่เป็นที่ยอมรับเรื่องประสิทธิภาพในการแทนข้อมูลหรือลดมิติข้อมูลอย่าง Autoencoder ซึ่งอธิบายรายละเอียดได้ดังนี้

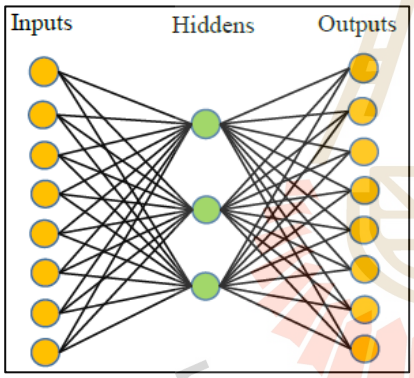
(1) Autoencoders

(1.1) Autoencoder (AE) หรือ เทคนิคเข้ารหัสอัตโนมัติจัดเป็นโครงข่ายประสาทแบบไปข้างหน้า โดยมีสถาปัตยกรรมที่มีชั้นซ่อนเร้น (Hidden Layer) มีการเชื่อมต่อกันอย่างสมบูรณ์ (Goodfellow et al., 2016; Gianniotis et al., 2016; Bengio, 2009) ซึ่งสามารถเข้ารหัสข้อมูลได้อย่างมีประสิทธิภาพสำหรับการเรียนรู้แบบไม่มีผู้สอน จึงมักถูกนำมาใช้ในการแทนข้อมูลด้วยการฝึกโครงข่ายเพื่อเปลี่ยนข้อมูลเข้า (Input) ให้อยู่ในรูปแบบอื่น โดยเฉพาะเพื่อการลดมิติของข้อมูล ซึ่งปัจจุบันแนวคิดของ Autoencoder ถูกใช้อย่างแพร่หลายสำหรับการเรียนรู้เพื่อสร้างโมเดลแทนข้อมูล (Goodfellow et al., 2016; Kingma and Welling, 2013) โดยทั่วไปโครงข่าย AE ประกอบด้วยสองส่วนหลัก ส่วนแรกคือ Encoder Function $h = f(x)$ ซึ่งส่วนที่ใช้สำหรับเข้ารหัสข้อมูลนำเข้า และส่วนที่สองคือ Decoder Function $r = g(h)$ เป็นส่วนที่ทำหน้าที่ในการเรียกคืนข้อมูลจากรหัสที่ดำเนินการโดยส่วน Encoder Function ดังนั้นโครงข่ายทั่วไปของสถาปัตยกรรม AE จะเป็นการแปลงข้อมูลนำเข้า r ให้เป็นข้อมูลนำออก (ตัวแทนข้อมูล) r (Code h) ซึ่งสามารถแสดงได้ดังรูปที่ 2.28 (Goodfellow et al., 2016)



รูปที่ 2.28 รูปแบบทั่วไปของสถาปัตยกรรม Autoencoder

ตัวอย่าง การใช้ Autoencoder แบบดั้งเดิมในการเรียนรู้เพื่อแทนข้อมูลเลขฐานสองขนาด 8 บิต โดยมีโครงข่ายดังนี้ ใน Input Layer และ Output Layer มี Neuron = 8 ใน Hidden Layer มี Neuron = 3 (8-3-8 Neuron) แสดงโครงข่ายดังรูปที่ 2.29 (ซ้ายมือ) การทำงานของโครงข่ายจะฝึกสอนโดยมีเป้าหมายให้ผลลัพธ์มีค่าเหมือนกับข้อมูลนำเข้า ดังรูปที่ 2.29 (ขวามือ)



Inputs		Hiddens				Outputs
10000000	->	.89	.04	.08	->	10000000
01000000	->	.15	.99	.99	->	01000000
00100000	->	.01	.97	.27	->	00100000
00010000	->	.99	.97	.71	->	00010000
00001000	->	.03	.05	.02	->	00001000
00000100	->	.01	.11	.88	->	00000100
00000010	->	.80	.01	.98	->	00000010
00000001	->	.60	.94	.01	->	00000001

รูปที่ 2.29 ตัวอย่าง Autoencoder Networks แบบ 8-3-8 Neuron และผลแทนข้อมูล

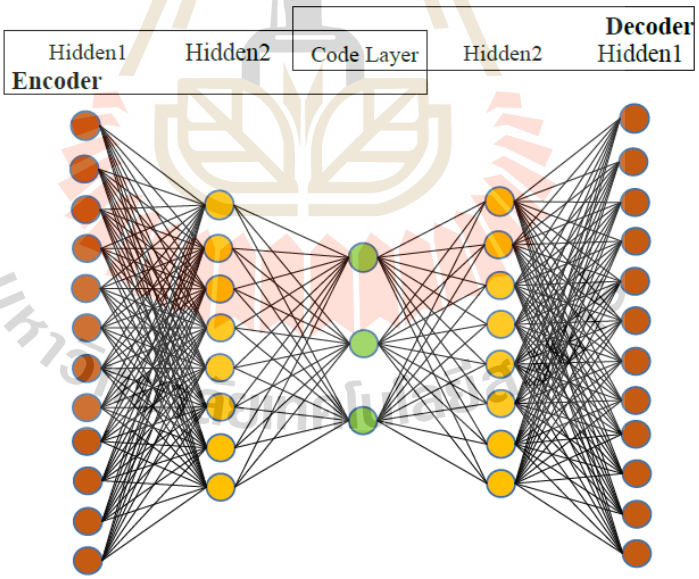
จากตัวอย่างจะแสดงถึงผลลัพธ์ที่ได้จากการเรียนรู้ของ Autoencoder คือผลลัพธ์ที่เกิดขึ้นใน Hidden Layer ซึ่งเป็นลักษณะของการสร้างตัวแทนข้อมูล

(1.2) ประเภทของ Autoencoder สามารถแบ่งได้เป็น 2 ประเภท ได้แก่ เทคนิคเข้ารหัสอัตโนมัติแบบต่ำกว่าสมบูรณ์ (Undercomplete Autoencoder) หรือเรียกว่าเป็นเทคนิคเข้ารหัสอัตโนมัติแบบดั้งเดิม และเทคนิคเข้ารหัสอัตโนมัติแบบปรับให้ปกติ (Regularized Autoencoder) คือเทคนิคการเข้ารหัสอัตโนมัติที่มุ่งเน้นในการสร้างความสามารถในการเรียนรู้ให้กับ Encoder และ Decoder ด้วยโครงข่ายที่มีความซับซ้อนขึ้น ซึ่งมักเป็นการทำงานในกรณีที่จำนวน Hidden Unit มีขนาดใหญ่กว่าจำนวน Input Unit หรือเรียกว่า เกินสมบูรณ์ (Overcomplete)

ซึ่งเทคนิคแบบ Regularized Autoencoder สามารถแบ่งเป็น เทคนิคแบบลดสัญญาณรบกวน (Denoise Autoencoder) เทคนิคแบบเบาบาง (Sparse Autoencoder) เทคนิคแบบหดตัว (Contractive Autoencoder) และเทคนิคแบบแปรผัน (Variational Autotencoder) (Goodfellow et al., 2016)

(2) Deep Autoencoder Networks

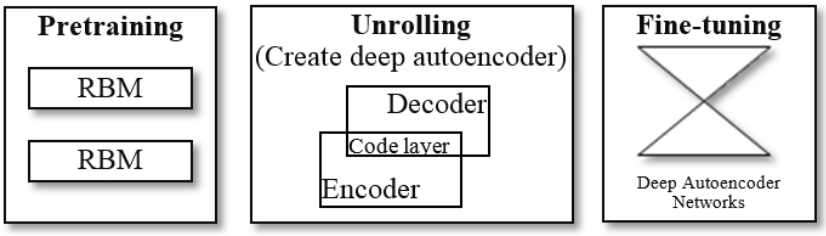
Deep Autoencoder Networks (DANs) หรือเทคนิคโครงข่ายการเข้ารหัสอัตโนมัติเชิงลึกซึ่งเป็นเทคนิคที่มีประสิทธิภาพในด้านการสร้างตัวแทนข้อมูล โดยเฉพาะความสามารถในการปรับลดข้อมูลมิติสูงให้อยู่ในรูปของรหัสข้อมูลมิติต่ำ โดย DANs เป็นเทคนิคที่อาศัยหลักการพื้นฐานของสถาปัตยกรรมเชิงลึก และโครงข่ายความเชื่อเชิงลึก โดยฝึกสอนโครงข่ายประสาทแบบหลายชั้น (Multilayer Neural Network) ด้วยประสาทชั้นกลางที่มีขนาดเล็กและอาศัยการปรับแต่ง Weights ที่มีประสิทธิภาพสูงเพื่อเรียนรู้ในรหัสข้อมูล ในกระบวนการเรียนรู้จะเข้ารหัสข้อมูลจากชั้นหนึ่งไปชั้นถัดไปแบบชั้นต่อชั้นในขั้น Encoder จนได้รหัสข้อมูลในมิติที่ต่ำลงในชั้น Code Layer และจะมีการถอดรหัสเพื่อสร้างคืนข้อมูลดั้งเดิมจากชั้นในสุดออกไปยังชั้นนอกสุดแบบชั้นต่อชั้นในขั้น Decoder แสดงแนวคิดการเชื่อมต่อสำหรับ DANs กรณี 3-Hidden Layers ได้ดังรูปที่ 2.30 ดังนี้



รูปที่ 2.30 รูปแบบโครงข่าย Encoder-Decoder ของ DANs

หัวใจการทำงานของ DANs จะอาศัยการทำงานของเทคนิคซึ่งมีความสามารถในการปรับแต่ง Weight ด้วยการฝึกสอนของโครงข่ายชั้น Hidden อย่าง RBMs ทำงานร่วมกับเทคนิคที่เป็นที่ยอมรับในด้านการแทนข้อมูลอย่าง Autoencoder สำหรับ DANs มี

ทำงานหลัก 3 ส่วนได้แก่ Pre-training, Unrolling และ Fine-tuning (Hinton and Salakhutdinov, 2006) ดังรูปที่ 2.31



รูปที่ 2.31 งานหลัก 3 ส่วนของ DANs

จากรูปที่ 2.31 อธิบายขั้นตอนการทำงานได้ดังนี้

1) ขั้น Pre-training จะประกอบด้วย การเรียนรู้ของสแตก (Stack) ของ RBMs ซึ่งแต่ละ Stack จะมีเพียง 1 ชั้นของตัวตรวจจับฟีเจอร์ (Feature Detectors) การใช้ Feature ที่ผ่านการเรียนรู้ของ RBM หนึ่งจะถูกใช้เป็นข้อมูลนำเข้าสำหรับ RBM ถัดไปใน Stack

สำหรับ Binary Vectors ทั้งหมดได้จาก RBM ในรูปแบบสุ่มตัวตรวจจับ Feature แบบไบนารีจะเชื่อมต่อด้วยการให้น้ำหนักแบบสมมาตรโดยข้อมูลเป้าหมายที่ต้องการ จะอยู่ใน Visible Units ของ RBMs (หรือ Output ของ RBMs) ส่วนตัวตรวจจับ Feature จะอยู่ใน Hidden Units ซึ่งแทนด้วย (v, h) โดยทั้งสองส่วนจะมีพลังงาน (Hinton and Salakhutdinov, 2006) นิยามได้ดังนี้

$$E(\mathrm{v},\mathrm{h}) = -\sum_{i\in pixels} b_i v_i - \sum_{j\in feature} b_j h_j - \sum_{i,j} v_i h_j w_{ij} \qquad (2.12)$$

เมื่อ v_i และ h_i เป็น Pixels (เป้าหมาย) และ Feature ในรูปแบบไบนารี ของเป้าหมาย i และ Feature j มี b_i และ b_j คือ Biases และ w_{ij} เป็น Weight ระหว่าง Visible กับ Hidden โดยในการ Pre-training นี้จะพยายามปรับแต่ง Weights และ Biases ให้เกิดพลังงานที่ต่ำ และให้เหมาะสมที่จะสามารถถอดรหัสคืนกลับเป็นข้อมูลเดิมได้ดีที่สุด

2) ขั้น Unrolling เมื่อผ่านการ Pre-training แล้ว RBM จะถูกคลี่ออก (Unrolled) เพื่อสร้างเป็น Deep Autoencoder หมายถึงการสร้างโครงข่าย Encoder และ Decoder ซึ่งทั้งสองโครงข่ายจะมีการใช้ค่า Weight เดียวกัน

3) ขั้น Fine-tuning เป็นขั้นตอนสุดท้ายซึ่งจะเป็นขั้นตอนการปรับแต่งน้ำหนักอีกครั้งเพื่อให้การเรียกคืนตัวแทนข้อมูลให้เหมือนหรือใกล้เคียงข้อมูลดั้งเดิมให้มากสุดให้โดยใช้ Backpropagation สำหรับการการตรวจสอบข้อผิดพลาดในการทำงานพิจารณาจาก Weight ที่

เปลี่ยนแปลง จากนั้นจะแทนค่าที่เกิดจากการสุ่มด้วยค่าที่มีความน่าจะเป็นค่าที่แท้จริง โดย Weight ที่เปลี่ยนแปลงสามารถนิยามได้ดังสมการต่อไปนี้

$$\Delta w_{ij} = \varepsilon(\langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{recon}) \quad (2.13)$$

โดยที่ Δw_{ij} คือ Weight ที่ปรับแต่งของ Feature ตัวที่ i และ Feature Detector ตัวที่ j

ε คือ อัตราการเรียนรู้

$\langle v_i h_j \rangle_{data}$ คือ สัดส่วนของเวลาที่ Feature ตัวที่ i และ Feature Detector ตัวที่ j

$\langle v_i h_j \rangle_{recon}$ คือ สัดส่วนเวลาที่สอดคล้องในการประมวลเพื่อสร้างคืนข้อมูล

2.4.4 การแทนอนุกรมเวลาด้วยเทคนิค DANs

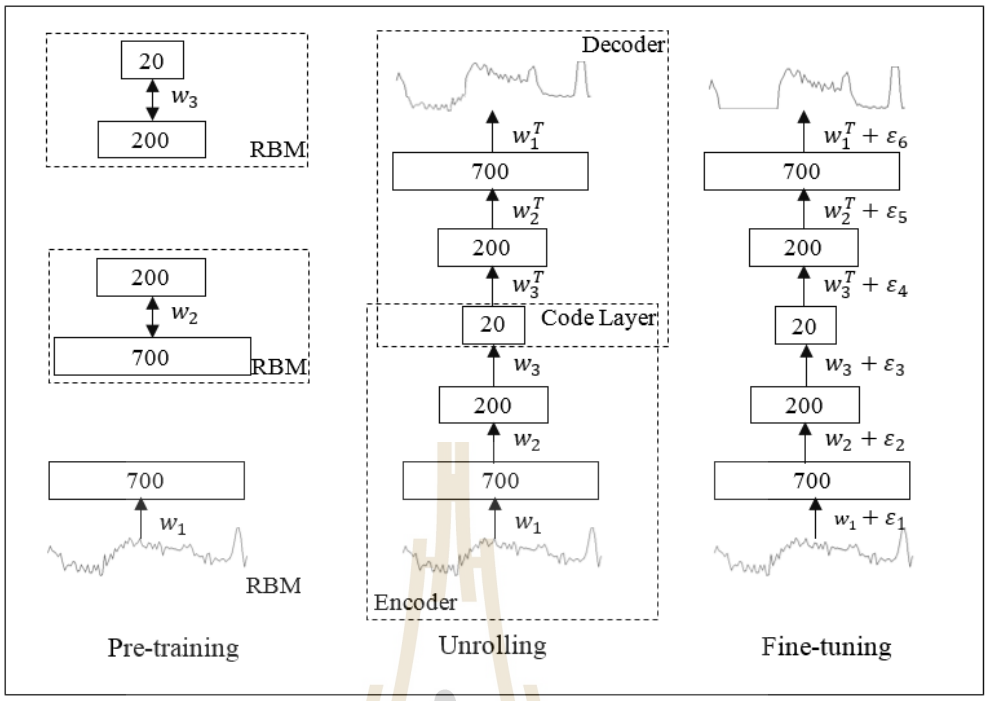
สำหรับข้อมูลที่มีขนาดมิติสูงอย่างข้อมูลอนุกรมเวลา ซึ่งธรรมชาติอาจบรรจุข้อมูลที่อาจเป็นสัญญาณรบกวนไม่จำเป็นสำหรับการนำไปวิเคราะห์ ดังนั้นแนวทางที่เป็นหัวใจสำคัญในการช่วยให้การเรียนรู้ของเครื่องจักรให้ทำงานได้มีประสิทธิภาพมากยิ่งขึ้นคือ การแทนอนุกรมเวลาด้วยตัวแทนข้อมูล โดยใช้เทคนิคบางอย่าง ซึ่ง DANs เป็นทางเลือกที่น่าสนใจ และน่าท้าทายทางเลือกหนึ่ง ดังนั้นในส่วนนี้จะกล่าวถึงการนำ DANs มาใช้เพื่อการแทนข้อมูลอนุกรมเวลา

ตัวอย่าง การหาตัวแทนอนุกรมเวลาสำหรับข้อมูล ECGs อนุกรม S1 โดยอาศัยหลักการทำงานของเทคนิค DANs มีการทำงาน ดังนี้

(1) กำหนดโมเดลของโครงข่ายการเข้ารหัส ซึ่งตัวอย่างนี้กำหนดให้โมเดลของโครงข่ายการเข้ารหัสแบบ 3 ชั้น Hidden โดยมี Encoder Networks เป็น 700-200-20 นั่นหมายความว่า ข้อมูลดั้งเดิมจะถูกแปลงให้เป็นรหัสข้อมูลที่ใกล้เคียงกับข้อมูลดั้งเดิมมากที่สุด ด้วยโครงข่าย DANs ที่ชั้น Hidden เป็น 700-200-20-200-700 (Encoder-Decoder)

(2) การ Pre-training เพื่อหา Weight เริ่มต้นที่เหมาะสมสำหรับการเชื่อมต่อของแต่ละชั้นดังต่อไปนี้ จากโครงข่ายที่กำหนดเป็น 700-200-20 ดังนั้นการเรียนรู้ของ Stack ของ RBMs จะได้ w1 คือ Weight ระหว่าง Input Layer กับ 700-Hiddens Layer, w2 คือ Weight ระหว่าง 700-Hiddens Layer กับ 200-Hidden Layer และ w3 คือ Weight ระหว่าง 200-Hiddens Layer กับ 20-Hidden Layer ดังแสดงในรูปที่ 2.31 (Pre-training)

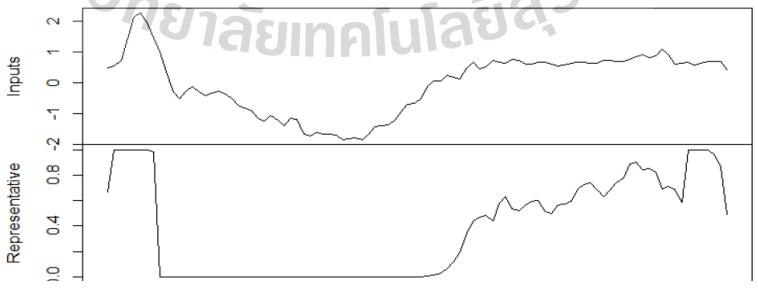
(3) การ Unrolling คือ เมื่อผ่านการ Pre-training จะได้ Weight ที่เหมาะสมใช้ตั้งต้นสำหรับการทำงานใน Encoder Networks จากนั้น RBM จะถูกคลี่ออกเพื่อสร้างเป็น Deep Autoencoder หมายถึงการสร้างโครงข่ายที่ประกอบด้วย Encoder และ Decoder ซึ่งทั้งสองโครงข่ายจะมีการใช้ค่า Weight เดียวกัน ดังรูปที่ 2.32 (Unrolling)



รูปที่ 2.32 ตัวอย่างการหาตัวแทนอนุกรมเวลาด้วยเทคนิค DANs

(4) การ Fine-tuning เป็นขั้นตอนการปรับแต่ง Weight อีกครั้งเพื่อให้การสร้างคืนข้อมูลจากตัวแทนให้เหมือนหรือใกล้เคียงข้อมูลดั้งเดิมให้มากสุด ซึ่งเมื่อทำการสร้างคืนด้วยโครงข่าย Decoder โดยอาศัย Weight ที่เปลี่ยนแปลงไปซึ่งจะขึ้นกับอัตราการเรียนรู้ (ε) แล้วจากนั้นจะแทนข้อมูลที่สร้างคืนด้วยค่าที่มีความน่าจะเป็นค่าที่แท้จริง ดังรูปที่ 2.31 (Fine-tuning)

(5) ตัวอย่างข้อมูลอนุกรมและตัวแทนของอนุกรมเวลาที่ได้จากเทคนิค DANs เมื่อแสดงกราฟเปรียบเทียบรูปร่าง Raw Data กับตัวแทน จะได้ดังรูปที่ 2.33 ดังนี้



รูปที่ 2.33 กราฟเปรียบเทียบรูปร่าง S1 ระหว่าง Raw Data กับตัวแทน S1 ที่ได้จาก DANs

เมื่อพิจารณาข้อมูล ECGs อนุกรม S1 เทียบกับตัวแทนที่ได้จาก DANs จะได้ดังข้อมูลที่แสดงในตารางที่ 2.3 ซึ่งแสดงข้อมูลเพียง 60 มิติ ดังนี้

ตารางที่ 2.3 ข้อมูลดั้งเดิมของอนุกรมเวลา S1 แสดงตัวอย่าง 60 มิติ

Features	Raw Data	Representative	Features	Raw Data	Representative
X1	0.502060	0.665691	X31	-1.649300	0.000009
X2	0.542160	0.999968	X32	-1.726600	0.000035
X3	0.722380	0.999906	X33	-1.608400	0.000014
X4	1.428900	0.999925	X34	-1.662800	0.000026
X5	2.136500	0.999969	X35	-1.650700	0.000023
X6	2.281100	0.999992	X36	-1.697300	0.000609
X7	1.936300	0.999978	X37	-1.838700	0.000239
X8	1.468900	0.981547	X38	-1.802600	0.000199
X9	1.008800	0.000009	X39	-1.780500	0.000073
X10	0.380280	0.000001	X40	-1.825200	0.000053
X11	-0.296780	0.000007	X41	-1.644800	0.000015
X12	-0.513930	0.000140	X42	-1.423800	0.000086
X13	-0.255640	0.000092	X43	-1.392200	0.000180
X14	-0.107200	0.000081	X44	-1.360400	0.000146
X15	-0.287830	0.000100	X45	-1.200200	0.000252
X16	-0.418010	0.000044	X46	-0.918630	0.000210
X17	-0.319160	0.000040	X47	-0.685920	0.000394
X18	-0.260380	0.000059	X48	-0.667940	0.000758
X19	-0.350360	0.000024	X49	-0.512720	0.001005
X20	-0.505490	0.000046	X50	-0.101690	0.001921
X21	-0.710890	0.000053	X51	0.063954	0.003904
X22	-0.823920	0.000016	X52	0.082614	0.013585
X23	-0.899700	0.000029	X53	0.237610	0.031864
X24	-1.153900	0.000038	X54	0.174790	0.065063
X25	-1.229800	0.000016	X55	0.123210	0.115009
X26	-1.044100	0.000015	X56	0.503390	0.193587
X27	-1.202000	0.000031	X57	0.683870	0.344191
X28	-1.392200	0.000006	X58	0.474990	0.436498
X29	-1.130100	0.000025	X59	0.532800	0.470723
X30	-1.179900	0.665691	X60	0.723550	0.482312

2.5 อัลกอริทึมเชิงพันธุกรรม

อัลกอริทึมเชิงพันธุกรรม (Genetic Algorithms: GAs) เป็นวิธีการค้นหาคำตอบโดยอาศัยการเลียนแบบวิวัฒนาการทางธรรมชาติของสิ่งมีชีวิตที่มีพื้นฐานแนวคิดจากทฤษฎีวิวัฒนาการทางธรรมชาติของ Charie Darwin โดยการนำเสนออัลกอริทึมเชิงพันธุกรรมของ John Holland ในปี ค.ศ. 1975 โดยประยุกต์ขั้นตอนวิวัฒนาการของสิ่งมีชีวิตในระบบชีววิทยามาใช้ในการคำนวณด้วยคอมพิวเตอร์เพื่อค้นหาคำตอบที่เหมาะสมที่สุด ซึ่งขั้นตอนการทำงานสำหรับ Genetic Algorithms (Houck et al., 1995) มีดังนี้

(1) การเข้ารหัสโครโมโซม (Chromosome Encoding)

การเข้ารหัสโครโมโซมเป็นการออกแบบให้ Chromosome แทนคำตอบของสิ่งที่ต้องการค้นหา แต่ละ Chromosome ประกอบด้วยยีน (Gene) ซึ่งเป็นพารามิเตอร์ (Parameter) ที่ต้องการหาค่าที่เหมาะสม การเข้ารหัส Chromosome มี 3 ประเภท (Holland, 1975) ได้แก่

(2.1) การเข้ารหัสแบบเลขฐานสอง (Binary Encoding) เป็นการเข้ารหัส Chromosome โดยการแปลงข้อมูลจริงเป็นเลขฐานสอง ซึ่งจะได้ Gene แต่ละตัวใน Chromosome มีรูปแบบเป็น 0 และ 1 (Holland, 1975)

(2.2) การเข้ารหัสแบบค่าจริง (Value Encoding) เป็นการเข้ารหัสโดยการแทนค่า Gene ด้วยค่าจริงที่เชื่อมโยงถึงค่าที่สามารถเป็นคำตอบที่เหมาะสมของปัญหาได้ ค่าจริงที่ใช้ในการเข้ารหัส Chromosome อาจเป็น ตัวเลขจำนวนจริง ตัวอักษร หรือคำสั่ง เป็นต้น (Wright, 1991)

(2.3) การเข้ารหัสแบบเปลี่ยนลำดับ (Permutation Encoding) เป็นการเข้ารหัสโดยการแทน Gene แต่ละตัวด้วยแหน่งของปัญหาที่ต้องการหาคำตอบ

(2) การสร้างประชากรเริ่มต้น (Initialize Population)

การสร้างประชากรเริ่มต้น หรือเรียกว่าประชากรรุ่นแรก (Initial Population) โดยทั่วไปจะใช้วิธีการสุ่มตัวเลขให้เท่ากับจำนวนประชากร (Population Size) ที่กำหนด

(3) การประเมินค่าความเหมาะสม (Fitness Value Evaluation)

การประเมินค่าความเหมาะสมของแต่ละ Chromosome เป็นขั้นตอนค้นหาคำตอบที่เหมาะสมตามเกณฑ์ที่กำหนด ด้วยการส่งประชากรเข้าสู่ฟังก์ชันวัตถุประสงค์ (Objective Function) ที่จะมีการคำนวณหรือใช้สมการที่สอดคล้องกับแต่ละปัญหาต่างกันไป

(4) การดำเนินการทางพันธุกรรม (Genetic Operations)

การดำเนินการทางพันธุกรรม เป็นงานที่ทำให้เกิดการเปลี่ยนแปลงทางพันธุกรรม เกิดประชากรรุ่นลูกหลาน (Offspring) การดำเนินการทางพันธุกรรมประกอบด้วย การคัดเลือกสายพันธุ์ (Selection) การข้ามสายพันธุ์ (Crossover) และการกลายพันธุ์ (Mutation) ดังนี้

(4.1) การคัดเลือกสายพันธุ์ (Selection) เป็นการคัดเลือก Chromosome โดยหวังผลที่จะได้สายพันธุ์ที่ดีไปเป็น Chromosome พ่อ และ Chromosome แม่ เพื่อนำไปสร้างต้นกำเนิดสายพันธุ์รุ่นถัดไป วิธีการคัดเลือกประชากรมีหลายวิธี เช่น วิธีการคัดเลือกวงล้อรูเลต (Roulette Wheel) วิธีการคัดเลือกโดยการสุ่ม (Random Selection) วิธีการคัดเลือกแบบจัดอันดับ (Ranking Selection) และวิธีการคัดเลือกแบบแข่งขัน (Tournament Selection) เป็นต้น

(4.2) การข้ามสายพันธุ์ (Crossover) เป็นการสร้าง Chromosome รุ่นลูกหลานด้วยการดำเนินการ Crossover ระหว่าง Chromosome พ่อ และ แม่ ด้วยอัตราการ Crossover ที่กำหนด ซึ่งอัตราที่เหมาะสมควรอยู่ระหว่าง 0.5 – 0.9 (Wehrens and Buydens, 1998) ซึ่งการ Crossover มี 3 วิธี ได้แก่ การข้ามสายพันธุ์แบบจุดเดียว (Single-point Crossover) การข้ามสายพันธุ์แบบหลายจุด (Multiple-point Crossover)

(4.3) การกลายพันธุ์ (Mutation) เป็นแปรผัน Gene ในบางตำแหน่ง หรือทุกตำแหน่งใน Chromosome เพื่อให้เกิดการผ่าเหล่า หรือหลุดพ้นจากคำตอบที่เหมาะสมที่สุดแบบวงแคบเฉพาะถิ่น (Local Optimum) โดยกำหนดอัตราการ Mutation ให้มีค่าต่ำซึ่งควรมีอัตราอยู่ระหว่าง 0.001-0.05 (Wehrens and Buydens, 1998)

(5) การแทนที่ (Replacement)

การแทนที่ประชากรเดิม เป็นการแทนประชากรเดิมด้วยประชากรรุ่นใหม่ที่มีความเหมาะสมที่ดีกว่า ซึ่งทำให้ได้ประชากรรุ่นใหม่ที่มี Chromosome ที่ผ่านการคัดเลือกแล้ว โดยการแทนที่ประชากรสามารถทำได้ 2 วิธีได้แก่

(5.1) การแทนที่ประชากรทั้งรุ่น (Generational Genetic Algorithm) เป็นวิธีการที่นำประชากรรุ่นลูกหลานไปแทนที่ประชากรรุ่นพ่อแม่ทั้งรุ่น ซึ่งจะส่งผลให้ Chromosome ที่ดีของพ่อแม่ถูกแทนที่ด้วยลูกหลาน ซึ่งสามารถแก้ไขปัญหานี้ได้ด้วยการเลือกเก็บ Chromosome ที่ดีที่สุดของพ่อแม่ไว้ได้บางส่วน เรียกว่า การคัดเลือกหัวกะทิ (Elitist Strategy)

(5.2) การแทนที่ประชากรบางส่วน (Partial Genetic Algorithm) เป็นวิธีการแทนที่ประชากรรุ่นเดิมโดยเลือกแทนที่ประชากรรุ่นพ่อแม่เฉพาะ Chromosome ที่ด้อยที่สุดบางตัวเท่านั้น ซึ่งหมายถึงจะมี Chromosoem รุ่นลูกหลานเพียงบางตัวที่ถูกใช้

(6) การตรวจสอบเงื่อนไขเพื่อสิ้นสุดการทำงาน (Termination Condition)

การตรวจสอบเงื่อนไขเพื่อสิ้นสุดการทำงาน เป็นขั้นตอนตรวจสอบว่าอัลกอริทึมเชิงพันธุกรรมควรสิ้นสุดการทำงานหรือยัง ซึ่งเงื่อนไขที่ใช้ เช่น สร้างจำนวนรุ่นของประชากรได้ครบตามจำนวนที่กำหนดไว้ จึงจะสิ้นสุดการทำงาน แต่หากยังไม่เป็นตามเงื่อนไข การดำเนินงานจะกลับไปวนซ้ำทำงานขั้นตอนเดิมต่อไป

ตัวอย่าง การนำเอา GAs ไปใช้ค้นหาคำตอบที่เหมาะสมที่สุด สำหรับปัญหาดังต่อไปนี้

ปัญหา ต้องการหาค่า y ที่ให้คำตอบที่ดีที่สุด 4 จำนวน สำหรับสมการ $Z = 15y - y^2$

กำหนดให้ Objective Function = $15y - y^2$, $0 \leq y \leq 16$, จำนวนประชากร = 4 การเข้ารหัส Chromosome ด้วยเลขฐานสอง อัตราการ Crossover (P_c) = 0.85 และอัตราการ Mutation (P_m) = 0.05 สามารถดำเนินตามขั้นตอนของ GAs ได้ดังนี้

ขั้นตอนที่ 1 ทำการสุ่มค่าสร้างประชากรเริ่มต้น ในที่นี้จะทำการสุ่มค่าประชากรและเข้ารหัสด้วยเลขฐานสอง (ขนาด 4 บิต) จำนวน 4 Choromosome ดังตารางที่ 2.4 ดังนี้

ตารางที่ 2.4 ข้อมูลประชากรเริ่มต้น และการเข้ารหัส Chromosome

Populations	Real Value	Chromosome Encoding			
		Gene4	Gene3	Gene2	Gene1
P1 :	9	1	0	0	1
P2 :	1	0	0	0	1
P3 :	2	0	0	1	0
P4 :	14	1	1	1	0

ขั้นตอนที่ 2 การประเมินค่าความเหมาะสมของประชากรทั้งหมด โดยการนำทุกประชากรเข้าสู่ Objective function แสดงตัวอย่างดังตารางที่ 2.5 ดังนี้

ตารางที่ 2.5 การประเมินค่าความเหมาะสมของประชากรแต่ละตัว

Populations	Real Value	Objective Function	Fitness Value
P1 :	9	$Z=15(9) - 9^2$	54
P2 :	1	$Z=15(1) - 1^2$	14
P3 :	2	$Z=15(2) - 2^2$	26
P4 :	14	$Z=15(14) - 14^2$	14

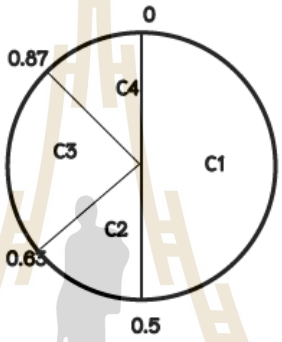
ขั้นตอนที่ 3 ทำการคัดเลือกประชากรที่จะนำไปสร้างต้นกำเนิดสายพันธ์ในรอบถัดไป เพื่อให้ได้ประชากรที่ดีที่สุด วิธีที่ใช้ในตัวอย่างนี้คือการคัดเลือกแบบ Roulette wheel ทำงานดังนี้

ใช้ Fitness Value คำนวณหาค่าพื้นที่ใน Roulette Wheel เพื่อแทนโอกาสของการสุ่มเจอประชากรที่จะใช้เป็นพ่อ และแม่ จะได้ดังตารางที่ 2.6 ดังนี้

ตารางที่ 2.6 ข้อมูลประชากรเริ่มต้น และการเข้ารหัส Chromosome

Populations	Fitness Value	Propbability	Area
P1 :	54	54/(54+14+26+14) = 0.5	C1
P2 :	14	14/(54+14+26+14) = 0.13	C2
P3 :	26	26/(54+14+26+14) = 0.24	C3
P4 :	14	14/(54+14+26+14) = 0.13	C4

และแสดงการแทนพื้นที่ใน ดังรูปที่ 2.34 ดังนี้



รูปที่ 2.34 ค่าโอกาสในการถูกคัดเลือกของ Chromosome แต่ละตัวใน Roulette Wheel

หลังจากสร้าง Roulette Wheel จะสุ่มค่าในช่วง (0,1] เพื่อเลือกพ่อและแม่ โดยสมมุติการสุ่มค่าแต่ละครั้งมีค่าดังนี้

สุ่มค่า ครั้งที่ 1 = 0.25 อยู่ในพื้นที่ C1 ดังนั้น C1 จะถูกใช้เป็น Chromosome พ่อตัวที่ 1

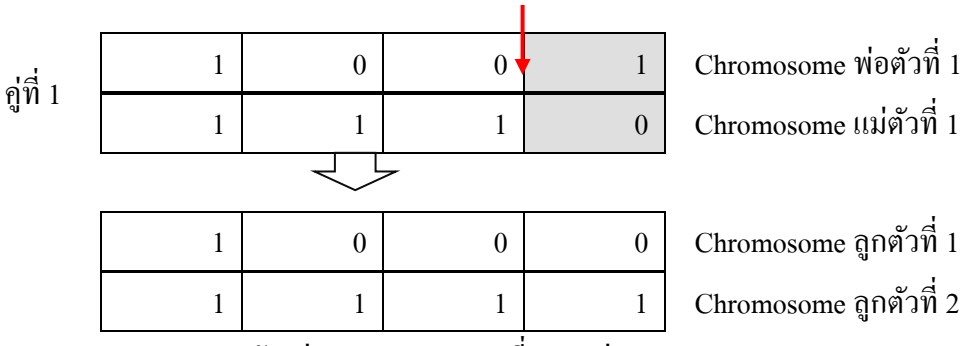
สุ่มค่าครั้งที่ 2 = 0.94 อยู่ในพื้นที่ C4 ดังนั้น C4 จะถูกใช้เป็น Chromosome แม่ตัวที่ 1

สุ่มค่าครั้งที่ 3 = 0.45 อยู่ในพื้นที่ C1 ดังนั้น C1 จะถูกใช้เป็น Chromosome พ่อตัวที่ 2

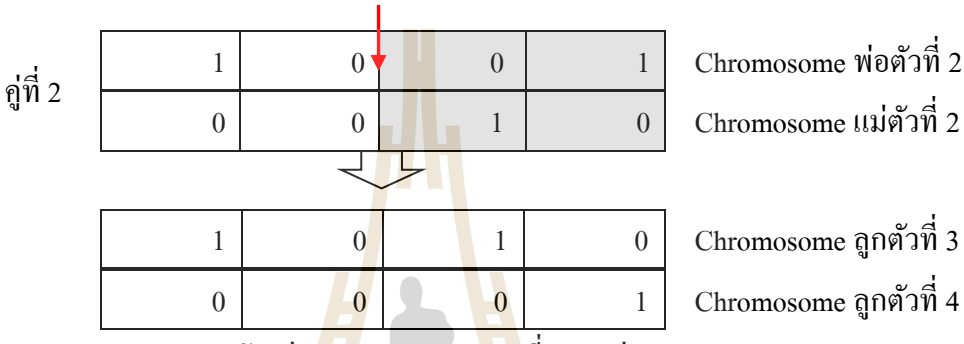
สุ่มค่าครั้งที่ 4 = 0.75 อยู่ในพื้นที่ C3 ดังนั้น C3 จะถูกใช้เป็น Chromosome แม่ตัวที่ 2

ขั้นตอนที่ 4 การดำเนินการทางพันธุกรรม (Genetic operation) การดำเนินการทางพันธุกรรมประกอบไปด้วยการทำ Crossover และ Mutation

การทำ Crossover ในตัวอย่างนี้ใช้ Single-point Crossover การสุ่มค่าในช่วง (0,1] เพื่อทำการหาค่าที่สุ่มมีค่ามากว่าหรือน้อยกว่า อัตราการ Crossover (P_c) กรณีมีค่าน้อยกว่าหรือเท่ากันจะดำเนินการทางพันธุกรรมด้วยการ Crossover กรณีมีค่ามากกว่าจะดำเนินการทางพันธุกรรมด้วยการ Mutation ซึ่งตัวอย่างแสดงการ Crossover ดังรูปที่ 2.35 ดังนี้



(a) ตัวอย่างการ Crossover ที่ตำแหน่ง Gene 1



(b) ตัวอย่างการทำ Crossover ที่ตำแหน่ง Gene 2

รูปที่ 2.35 ตัวอย่างการทำ Crossover

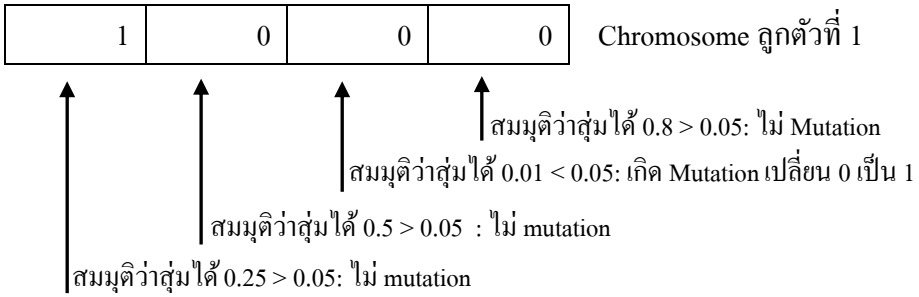
จากรูปที่ 2.35 สามารถอธิบายการสุ่ม ตำแหน่ง และอัตราการ Crossover ดังนี้

คู่ที่ 1 สมมุติให้สุ่มได้ค่า 0.6 ซึ่ง 0.6 < 0.85 (P_c ที่กำหนดไว้) ดังนั้นจะทำ การ Crossover ด้วยการสุ่มหาจุดที่จะ Crossover ซึ่งในตัวอย่างนี้ Chromosome มีขนาด 4 Gene ดั้งนั้นจะสุ่มค่าในช่วง 1-3 ซึ่งสมมุติว่าสุ่มได้จุดที่ 1 จึงดำเนินการกับคู่ที่ 1 ดังรูปที่ 2.35 (a)

คู่ที่ 2 การสุ่มได้ค่า 0.7 ต้องทำการ การ Crossover กับคู่ที่ 2 เช่นกัน และสุ่มได้จุดที่ 2 ดังนั้นจึงดำเนินการกับคู่ที่ 2 ดังรูปที่ 2.35 (b)

การทำ Mutation หลังจากผ่านกระบวนการ Crossover จะได้ประชากรรุ่นลูก 4 ประชากร ในการทำ Mutation จะนำประชากรลูกแต่ละตัวมาพิจารณาทำการ Mutation ทุก Gene แสดงตัวอย่างการทำ Muntation สำหรับ Chromosome ลูกตัวที่ 1 ได้ดังนี้

สำหรับ Gene ตัวที่ 1 (จากขวามือ) สุ่มอัตราการ Mutation ได้ 0.85 > 0.05 (P_m ที่กำหนดไว้) ดังนั้น Gene ที่ 1 ไม่เกิด Mutation และที่ Gene ตัวอื่น ๆ ก็ใช้หลักการเดียวกัน ดังรูปที่ 2.36 (a) และเมื่อดำเนินการจนครบทุก Gene จะได้ผลลัพธ์ดังรูปที่ 2.36 (b) ดังนี้



(a) ตัวอย่างการ Mutation

1	0	1	0

(b) Chromosome ลูกตัวที่ 1 เมื่อผ่านการ Mutation

รูปที่ 2.36 ตัวอย่างการทำ Mutation สำหรับ Chromosome ลูกตัวที่ 1

ดังนั้น เมื่อทำ Mutation ใน Chromosome ลูกทุกตัว จะได้ผลลัพธ์ดังตารางที่ 2.7 ดังนี้

ตารางที่ 2.7 ผลลัพธ์ที่ได้จากการดำเนินการทางพันธุกรรม: ประชากรลูกหลาน

Offspring	**Real Value**	**Chromosome Encoding**			
		Gene4	**Gene3**	**Gene2**	**Gene1**
C1 :	10	1	0	1	0
C2 :	15	1	1	1	1
C3 :	6	0	1	1	0
C4 :	4	0	1	0	0

ขั้นตอนที่ 5 ทำการประเมินค่าความเหมาะสมประชากรลูกหลานเช่นเดียวกับขั้นตอนที่ 2 ซึ่งจะได้ผลลัพธ์แสดงดังตารางที่ 2.8 ดังนี้

ตารางที่ 2.8 การประเมินค่าความเหมาะสมของประชากรรุ่นลูกหลาน

Populations	**Real Value**	**Objective Function**	**Fitness Value**
C1 :	10	$Z=15(10) - 10^2$	50
C2 :	15	$Z=15(15) - 15^2$	0
C3 :	6	$Z=15(6) - 6^2$	54
C4 :	4	$Z=15(4) - 4^2$	44

ขั้นตอนที่ 6 ทำการแทนที่ประชากร ตัวอย่างนี้กำหนดให้แทนที่ประชากรแบบแทนทั้งรุ่น โดยคัดเลือกหัวกะทิไว้ 20% จะได้ = (4*20)/100 = 0.8 คิดเป็น 1 ตัว

จากตัวอย่างจะได้ Chromosome ที่ดีที่สุดของรุ่นพ่อแม่คือ P1 มีค่า Fitness = 54 ดังนั้น ประชากรรุ่นพ่อแม่ที่เหลือจะถูกแทนที่ด้วยประชากรรุ่นลูก C1-C3 แสดงดังตารางที่ 2.9

ดังนั้นจะได้ประชากรรุ่นใหม่ที่จะถูกใช้สำหรับสืบทอดพันธุกรรมในรุ่นถัดไป คือ P1, C1, C2 และ C3 ดังตารางที่ 2.9 ดังนี้

ตารางที่ 2.9 ผลลัพธ์ที่ได้จากการแทนที่ประชากร

Offspring	Gene4	Gene3	Gene2	Gene1	Fitness Value
P1 :	***1***	***0***	***0***	***1***	***54***
C1 :	***1***	***0***	***1***	***0***	***50***
C2 :	***1***	***1***	***1***	***1***	***0***
C3 :	***0***	***1***	***1***	***0***	***54***
P2 :	0	0	0	1	14
P3 :	0	0	1	0	26
P4 :	1	1	1	0	14
C4 :	0	1	0	0	44

ขั้นตอนที่ 7 การตรวจสอบการสิ้นสุดการค้นหาผลลัพธ์ เมื่อทำการตรวจสอบหากผลลัพธ์ที่ได้ไม่ใช่คำตอบที่ดีที่สุดให้กลับไปทำใน ขั้นตอนที่ 2 ถึง ขั้นตอนที่ 7 จนกว่าจะครบเงื่อนไขที่ต้องการหรือเท่ากับจำนวนรอบที่กำหนดไว้

2.6 การตรวจสอบและประเมินผลการจัดกลุ่ม

การตรวจสอบประสิทธิภาพเป็นสิ่งจำเป็นสำหรับการทำเหมืองข้อมูลด้วยเทคนิคการจัดกลุ่ม ซึ่งมมาตรวัดสำหรับตรวจสอบประสิทธิภาพของการจัดกลุ่มที่เป็นที่นิยมแบ่งเป็น 2 กลุ่มได้แก่ 1) การประเมินการจัดกลุ่มกรณีที่รู้คลาสที่แท้จริง เรียกว่า มาตรวัดภายนอก มีมาตรวัดให้เลือกประเมินได้หลายวิธี (Mohammed and Wagner, 2014) ที่น่าสนใจมี 2 วิธีได้แก่ ค่าการประเมินกลุ่มและค่าพิวริตี 2) การประเมินที่ไม่รู้คลาสที่แท้จริง เรียกว่า มาตรวัดภายใน ซึ่งจะกล่าวถึง 2 วิธีได้แก่ ค่าซิลลูเอ็ด และค่าผลรวมความผิดพลาดโดยมีรายละเอียดดังนี้

2.6.1 ค่าการประเมินกลุ่ม (Cluster Evaluation)

ค่าการประเมินกลุ่ม หรือเรียกว่า ค่าความถูกต้อง (Accuracy) ของการจัดกลุ่ม โดยมาตรวัดนี้ใช้ในการจัดกลุ่มกรณีที่รู้ข้อมูลกลุ่มจริงก่อน ในการคำนวณจะใช้ค่าที่ตรงกันระหว่างกลุ่มจริงแทนด้วย $\mathcal{G} = \{G_1, \dots, G_k\}$ ที่รู้ก่อนแล้ว และกลุ่มที่ถูกกำหนดให้จากเทคนิคการจัดกลุ่มแทนด้วย $\mathcal{A} = \{A_1, \dots, A_k\}$ โดยประเมินภายใต้ดัชนีชี้วัดความคล้ายคลึง (Similarity Index: $Sim(\mathcal{G}, \mathcal{A})$) (Montero and Vilar, 2014) นิยามได้ดังนี้

$$Sim(\mathcal{G}, \mathcal{A}) = \frac{1}{k}\sum_{i=1}^{k} \max_{1 \le j \le k} Sim(G_i, A_j) \tag{2.14}$$

โดยที่

$$Sim(G_i, A_j) = \frac{2\left|G_i \cap A_j\right|}{|G_i| + \left|A_j\right|} \tag{2.15}$$

เมื่อ $|\cdot|$ แทนจำนวนสมาชิกในเซต

ค่าการประเมินกลุ่มที่มีค่าสูงแสดงถึงการจัดกลุ่มที่มีประสิทธิภาพมีความถูกต้องโดยรวมสูง

2.6.2 ค่าพิวริตี (Purity)

ค่าพิวริตี Purity คือค่าความบริสุทธิ์ในการเป็นสมาชิกของการจัดกลุ่มข้อมูล ซึ่งค่า Purity สำหรับสมาชิกในกลุ่ม Ci (Mohammed and Wagner, 2014) นิยามได้ดังนี้

$$purity_i = \frac{1}{n_i} \max_{j=1,\dots,k} \{n_{ij}\} \tag{2.16}$$

เมื่อกำหนดให้

i คือ ลำดับการจัดกลุ่มตั้งแต่กลุ่ม i=1 ถึง r

j คือ ลำดับคลาสเป้าหมายแต่ละคลาสตั้งแต่ คลาส j=1 ถึง k

n_i คือ จำนวนสมาชิกของกลุ่ม i

n_{ij} คือ จำนวนสมาชิกของกลุ่ม i ที่สังกัดคลาส j

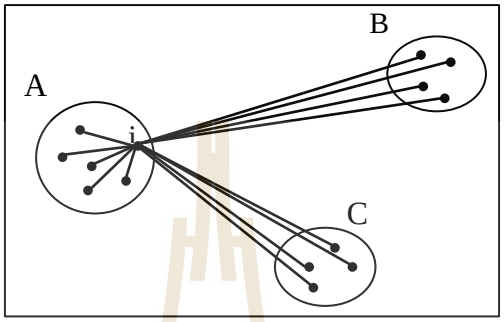
ดังนั้นค่า Purity สำหรับการจัดกลุ่มทั้งหมด C หาได้จาก Weight ทั้งหมดของการ Mapping กันระหว่าง กลุ่ม (Cluster) และ แต่ละคลาสเป้าหมาย (Partition) นิยามได้ดังนี้

$$purity = \sum_{i=1}^{r} \frac{n_i}{n} purity_i = \frac{1}{n}\sum_{i=1}^{r} \max_{j=1,\dots,k} \{n_{ij}\} \tag{2.17}$$

เมื่อค่า Purity ที่มีค่าสูง บ่งชี้ว่าการจัดข้อมูลให้อยู่ในกลุ่มที่แท้จริงได้มากหรือมีประสิทธิภาพ

2.6.3 ค่าสัมประสิทธิ์ซิลลูเอ็ต (Silhouette Coefficient: SC)

ค่าสัมประสิทธิ์ซิลลูเอ็ต เป็นมาตรวัดที่อาศัยทั้งการยึดเหนี่ยวภายในกลุ่ม และความสามารถในการแยกกันระหว่างกลุ่ม โดยใช้ค่าเฉลี่ยของระยะห่างระหว่างจุดกับกลุ่มที่อยู่ใกล้ชิด เทียบกับจุดที่อยู่ภายในกลุ่มเดียวกัน เป็นพื้นฐานในการทำงาน (Mohammed and Magner, 2014) ดังรูปที่ 2.37 แสดงการวัดระยะสำหรับวิธีของ Silhouette Coefficient



รูปที่ 2.37 การวัดระยะในมาตรวัดแบบ Silhouette

สำหรับแต่ละ x_i เราสามารถคำนวณหา Silhouette Coefficient (s_i) ได้ดังนี้

$$s_i = \frac{\mu_{out}^{\min}(\mathbf{x}_i) - \mu_{in}(\mathbf{x}_i)}{\max\left\{\mu_{out}^{\min}(\mathbf{x}_i), \mu_{in}(\mathbf{x}_i)\right\}} \tag{2.18}$$

โดยที่ $\mu_{in}(\mathbf{x}_i)$ คือค่าเฉลี่ยของระยะห่างของ x_i กับจุดอื่นภายในกลุ่ม $\hat{y}_i$ (กลุ่มเดียวกัน) ได้จาก

$$\mu_{in}(\mathbf{x}_i) = \frac{\sum_{\mathbf{x}_j \in C_{\hat{y}_i}, j \neq i} \delta(\mathbf{x}_i, \mathbf{x}_j)}{n_{\hat{y}_i} - 1} \tag{2.19}$$

และ $\mu_{out}^{\min}(\mathbf{x}_i)$ คือค่าเฉลี่ยของระยะระหว่างจุด x_i กับจุดอื่นในกลุ่มที่อยู่ใกล้ชิด หาได้จาก

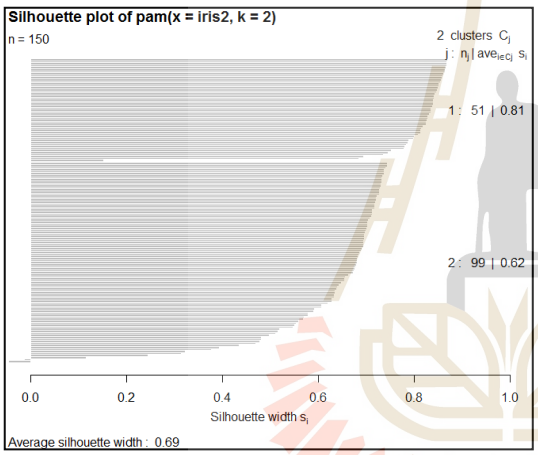
$$\mu_{out}^{\min}(\mathbf{x}_i) = \min_{j \neq \hat{y}_i} \left\{ \frac{\sum_{\mathbf{y} \in C_j} \delta(\mathbf{x}_i, \mathbf{y})}{n_j} \right\} \tag{2.20}$$

ดังนั้นค่า Silhouette Coefficient นิยามได้จากค่าเฉลี่ยของ s_i จากทุกจุดในกลุ่ม ดังนี้

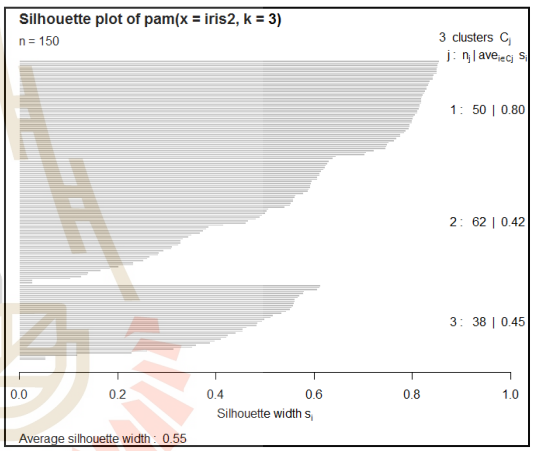
$$SC = \frac{1}{n} \sum_{i=1}^{n} s_i \tag{2.21}$$

ถ้าค่า SC มีค่ามาก ๆ เข้าใกล้ +1 หมายความว่าเป็นการจัดกลุ่มที่ดี หรือจัดกลุ่มได้เหมาะสม

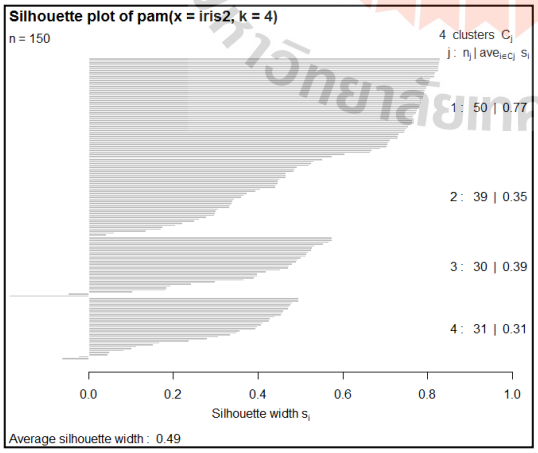
จากแนวคิดของ Rousseeuw (Rousseeuw, 1987) พบว่า Silhouette เป็นเครื่องมือที่ใช้สำหรับการจัดกลุ่มข้อมูลใช้เพื่อเลือกจำนวนกลุ่มที่เหมาะสม โดยหาค่า Silhouette จะใช้กับข้อมูลที่เป็น Ratio Scale และเหมาะกับการจัดกลุ่มข้อมูลที่ข้อมูลแต่ละกลุ่มเกาะกลุ่มกัน เมื่อมีการจัดกลุ่มจะมีการพิจารณาสมาชิกที่อยู่ใกล้เคียง (Proximities) ถึงคุณลักษณะสองประการคือ ความต่างกัน (Dissimilarities) และความคล้ายคลึงกัน (Similarities) ซึ่งมักนำมาใช้หาจำนวนกลุ่มที่เหมาะสมสำหรับการจัดกลุ่ม โดยเฉพาะกรณีที่เราไม่ทราบกลุ่มที่แท้จริงของข้อมูลมาก่อน ด้วยการนำค่า SC ที่ได้ไปสร้างเป็นแผนภาพที่แสดงความเหมาะสมของการกระจายของสมาชิก และพิจารณาค่าเฉลี่ยของ SC ร่วมด้วย ตัวอย่างดังรูปที่ 2.38 แสดงการเปรียบเทียบผลการจัดกลุ่ม เมื่อกำหนดจำนวนกลุ่ม (ค่า k) เป็น 2 ถึง 5 ซึ่งจะพบว่า ณ ค่า k=2 เป็นค่าที่เหมาะสมที่สุดสำหรับใช้จัดกลุ่ม เนื่องจากมีค่า SC ที่สูงที่สุดเมื่อเทียบกับ ค่า k=3, k=4 และ k=5



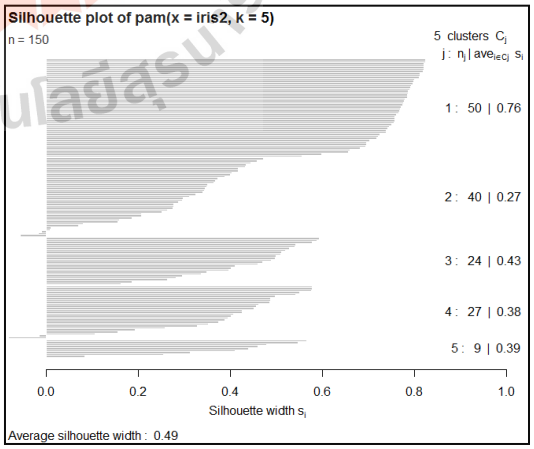
(a) k=2



(b) k=3



(c) k=4



(d) k=5

รูปที่ 2.38 Silhouette ของการจัดกลุ่มข้อมูล Iris ด้วย k-Means เมื่อกำหนดค่า k=2 ถึง k=5

2.6.4 ค่าผลรวมความผิดพลาด (Sum of Squared Error: SSE)

ในการจัดวัตถุให้อยู่ในกลุ่ม โดยเฉพาะในกรณี Unsupervised Learning หรือ กรณีที่เราไม่ทราบกลุ่มที่แท้จริงของข้อมูล จำเป็นต้องมีค่าคะแนนบางอย่างเป็นตัวชี้วัดหรือประเมินว่ากลุ่มที่จัดแบ่งให้กับวัตถุแต่ละตัวนั้นเหมาะสมดีแล้วหรือไม่ ซึ่งในที่นี้จะใช้ค่าผลรวมความผิดพลาด (Sum of Squared Errors: SSE) (Mohammed and Magner, 2014) นิยามได้ดังนี้

$$SSE(C) = \sum_{i=1}^{k} \sum_{x_j \epsilon C_i} \|x_j - \mu_i\|^2 \qquad (2.22)$$

เมื่อ x_j คือ ข้อมูลใด ๆ ตัวที่ j

μ_i คือ ค่าเฉลี่ยของกลุ่ม i

SSE เป็นเครื่องมือหนึ่งที่ใช้ในการวัดคุณภาพของการจัดกลุ่มที่ให้ผลดีไม่แพ้ Silhouette และรู้จักอย่างกว้างขวาง ในการพิจารณาค่า SSE ใช้การสังเกตจากกราฟที่สร้างจากความสัมพันธ์ระหว่าง SSE กับค่า k ที่จุดเปลี่ยนความชัน (Significant Local Change) หรือจุดที่มีลักษณะหัวเข่า “knee” (Significant “knee”) ซึ่งเป็นตำแหน่งที่บ่งชี้จำนวนกลุ่มที่เหมาะสมในการจัดกลุ่ม จากการศึกษาวิจัยของ Thinsungnoen และคณะ (2015) พบว่าควรเลือกค่า k ที่มีอัตราการเปลี่ยนแปลงสูงสุด โดยอัตราการเปลี่ยนแปลง (%Change) (Thinsungnoen et al., 2015) นิยามได้ดังนี้

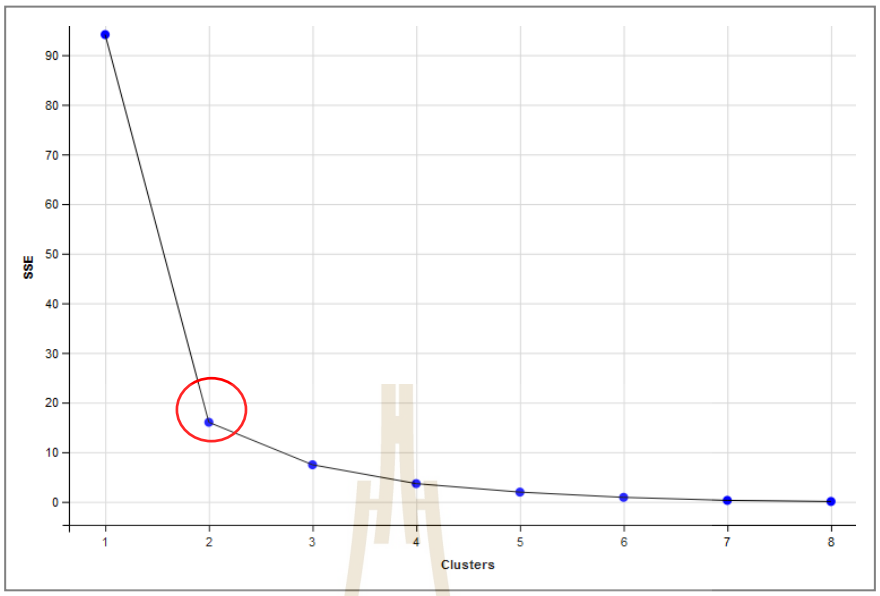
$$\%Change = \frac{(SSEofK_{i-1} - SSEofK_i) * 100}{SSEofK_{i-1}} \qquad (2.23)$$

โดยที่

$$\%Change = \begin{cases} 0 & ,if\ i = 1 \\ \%Change & ,\text{otherwise} \end{cases}$$

ตัวอย่างกราฟความสัมพันธ์ระหว่างค่า k และค่า SSE ณ จุดเปลี่ยนความชัน แสดงดังรูปที่ 2.38

ในการตรวจสอบค่า k ที่เหมาะสมสำหรับการจัดกลุ่ม โดยทดลองจัดกลุ่มกำหนดค่า k ให้มีค่าตั้งแต่ 2 ถึง 8 แล้วนำผลลัพธ์ของการจัดกลุ่มมาสร้างเป็นกราฟแสดงความสัมพันธ์ระหว่างค่า k และค่า SSE ดังรูปที่ 2.39 จากรูปสรุปได้ว่าการจัดกลุ่มนี้ควรกำหนดค่า k =2 เนื่องจากที่ตำแหน่ง k =2 เป็นตำแหน่งที่เป็นจุดเปลี่ยนความชัน (มี %Change มีค่าสูงที่สุด) ดังแสดงในรูปที่ 2.39 ดังนี้



รูปที่ 2.39 กราฟความสัมพันธ์ระหว่างค่า k และค่า SSE

2.7 งานวิจัยที่เกี่ยวข้อง

ดังที่กล่าวไปแล้วข้างต้นว่ามีงานวิจัยและวิธีการที่เกี่ยวข้องกับการค้นหาตัวแทน หรือการลดมิติของข้อมูล สำหรับข้อมูลอนุกรมเวลาจากเหล่านักวิจัยมากมายในรูปแบบที่แตกต่างกันทั้งงานวิจัยเกี่ยวกับ Subsequence Matching, Anomaly Detection และ Motif Discovery งานวิจัยเกี่ยวกับ Indexing, Clustering และ Classification งานวิจัยเกี่ยวกับ Visualization งานวิจัยเกี่ยวกับ Segmentation เป็นต้น นอกจากนี้ยังมีการใช้เทคนิคโครงข่ายการเรียนรู้เชิงลึกมาประยุกต์ในการลดมิติข้อมูลได้อีกด้วย ในงานวิจัยนี้จะสรุปการศึกษางานวิจัยที่เกี่ยวข้องได้ดังนี้

Lin และคณะ (2007) ได้นำเสนอเทคนิควิธีใหม่สำหรับกำหนดตัวแทนของข้อมูลอนุกรมเวลาในรูปแบบของการใช้สัญลักษณ์ เรื่อง Experiencing SAX: a Novel Symbolic Representation of Time-Series เป็นงานวิจัยที่ไม่ซ้ำใคร และช่วยในการลดมิติข้อมูลนำเข้า และลดปริมาณการคำนวณ โดยเทคนิคใหม่ที่นำเสนอนี้เรียกว่า SAX เป็นวิธีการแทนตัวแทนด้วยสัญลักษณ์ประยุกต์ใช้งานร่วมกับเทคนิค Lower Bound ซึ่งเป็นวิธีสำหรับการวัดระยะที่นิยามได้ตามข้อมูลดั้งเดิม ซึ่งผลงานวิจัยได้แสดงให้เห็นถึงคุณภาพของการนำวิธีการกำหนดตัวแทนข้อมูลนี้ไปใช้ในงานเหมืองข้อมูล หลายงาน ได้แก่ การจัดกลุ่มข้อมูล การจำแนกข้อมูล Indexing การตรวจจับความผิดปกติในอนุกรมเวลา การค้นหา Motif และการแสดงผลด้วยแผนภาพ

Wang และคณะ (2010) ได้เสนอแนวคิดใหม่สำหรับวิธีการกำหนดตัวแทนของข้อมูลอนุกรมเวลาและวิธีการวัดความคล้ายคลึง/ความแตกต่าง โดยเสนองานวิจัยในชื่อเรื่อง A Time-Series Analysis with Multiple Resolutions เป็นการเสนอวิธีการกำหนดตัวแทนข้อมูลแบบใหม่ที่ใช้เรียกว่า Multiresolution Vector Quantized (MVQ) Approximation ซึ่งจะใช้กับ Distance Function แบบใหม่ โดยวิธีที่นำเสนอจะมีความสามารถในการรักษาสารสนเทศในระดับเฉพาะถิ่น และระดับสาธารณะในอนุกรมเวลา และประยุกต์ใช้กับเทคนิคแบบข้อความเป็นฐาน และการค้นคืน ในการวิเคราะห์ความคล้ายคลึงของอนุกรมเวลา ซึ่งจะทำงานได้เร็วและสัมพันธ์แบบเชิงเส้นกับขนาดของข้อมูล ซึ่งต่างจากวิธีการวัดแบบเดิมอย่าง Euclidean Distance ในการทดลองได้เทียบผลกับวิธีวัดแบบ Euclidean, Dynamic Time Warping และ Piecewise Aggregate Approximation ซึ่งวิธีการที่นำเสนอได้ผลดีกว่าเกือบ 20% โดยใช้มาตรวัดเป็นค่า Precision/Recall และ Accuracy

Hinton และ Salakhutdinov (2006) ได้เสนอวิธีการเข้ารหัสข้อมูลที่มีมิติสูงให้มีมิติที่ต่ำลง โดยแนวคิดในการวิจัยคือการสอน (Training) โครงข่ายประสาทแบบหลายชั้น (Multilayer) ด้วยการใช้ชั้น Hidden ในการเข้ารหัสข้อมูลให้มีมิติต่ำและถอดรหัสคืนเป็นเวกเตอร์ข้อมูลเข้าที่มีมิติสูงใหม่ แล้วใช้ Gradient Descent แบบโครงข่าย “Autoencoder” ในการปรับแต่ง (Fine-Tuning) Weight โดยวิธีการที่นำเสนอนั้นเป็นวิธีที่มีประสิทธิอย่างมากโดยเฉพาะเมื่อเทียบกับวิธีการเลือก Feature ที่ดีที่มีอยู่อย่าง PCA จะพบว่าวิธีการที่ใช้ Deep Autoencoder Networks มีประสิทธิภาพดีกว่ามาก ซึ่งในขั้นตอนการ Training ทำงานโดยอาศัยหลักการของ Restricted Boltzmann Machines (RBMs)

Song และคณะ (2013) นำเสนองานวิจัยเรื่อง Auto-encoder Based Data Clustering เป็นงานวิจัยที่นำเสนอวิธีการจัดกลุ่มแบบใหม่โดยการจัดข้อมูลดั้งเดิมให้อยู่ในรูปแบบใหม่ที่เหมาะสำหรับการจัดกลุ่ม ด้วยเทคนิค Autoencoder Network ที่มีการเพิ่ม Objective Function ใหม่ฝังตัวเข้าไปภายใน Autoencoder Model สำหรับ Objective Function ที่สร้างขึ้นประกอบด้วย 2 ส่วนได้แก่ ข้อผิดพลาดในการสร้างใหม่ และระยะระหว่างข้อมูลดั้งเดิมกับจุดศูนย์กลางที่อยู่ในรูปแบบใหม่ โดยวิธีที่นำเสนอมีความเสถียรและจัดกลุ่มได้มีประสิทธิภาพ วัดประสิทธิภาพด้วยค่า Accuracy และค่า Normalized Mutual Information (NMI) วิธีการที่นำเสนอใช้จัดการกับข้อมูลที่มีลักษณะที่มีโครงสร้างซับซ้อน มีการกระจายไม่ปกติ หรือมีความแปรปรวนมาก

Ranzato และคณะ (2008) ได้นำเสนออัลกอริทึมใหม่ที่มีประสิทธิภาพในการเรียนรู้เพื่อหาตัวแทน Feature ที่มีลักษณะเบาบางสำหรับโครงข่ายความเชื่อเชิงลึก และเปรียบเทียบผลกับกลไกการทำงานที่คล้ายคลึงกันอย่าง RBMs โดยเสนอเกณฑ์ในการเปรียบเทียบและเลือกวิธีการเรียนรู้แบบไม่มีผู้สอนที่แตกต่างกัน โดยการชั่ง Weight ระหว่าง การปรับลดความผิดพลาดกับสาระของ

เนื้อหาของตัวแทนที่ได้ ซึ่งผลการวิจัยแสดงให้เห็นว่าการวางซ้อนกันหลายระดับของกลไกการทำงานและ Training ตามลำดับ ช่วยให้สามารถสังเกตพบคุณลักษณะที่ดีออกมาได้

Ripoll และคณะ (2016) ได้วิจัยเกี่ยวกับการวิเคราะห์ข้อมูล ECG โดยอาศัยหลักการเรียนรู้เชิงลึกด้วยการ “Pre-Training” และทำงานกับข้อมูลดั้งเดิม โดยได้นำเสนอวิธีการตรวจคัดกรองอัตโนมัติสำหรับประเมินว่าผู้ป่วยจากการดูแล Ambula-Tory หรือเหตุฉุกเฉินควรจะเรียกบริการโรคหัวใจ โดยใช้ข้อมูลที่เกิดขึ้นตามคลินิกและโรงพยาบาลในบาร์เซโลนา ระหว่างปี 2011- 2012 คัดเลือกผู้ป่วยจำนวน 1,390 คน ข้อมูลมีสองคลาสคือ Normal กับ Abnormal เปรียบเทียบประสิทธิภาพกับวิธีการจำแนกแบบอื่นที่จำแนกโดยไม่มีการ “Pre-Triaining” ผลการวิจัยพบว่าโครงข่ายประสาทที่มีชั้น Hidden จำนวน 3 ชั้นและทุกชั้นมี 700 โหนดจำแนกได้ที่ดีที่สุด ซึ่งให้ค่า Accuracy, Sensitivity และ Specificity ที่ 0.8552, 0.9176 และ 0.7827 ตามลำดับ

เพื่อสรุปการศึกษางานวิจัยที่เกี่ยวข้องกับงานวิจัยที่นำเสนอ แสดงผลได้ดังตารางที่ 2.10

ตารางที่ 2.10 สรุปเปรียบเทียบงานวิจัยที่เกี่ยวข้องกับการวิเคราะห์ข้อมูลอนุกรมเวลา

เทคนิคที่ใช้ / งานวิจัยที่เกี่ยวข้อง	**ก**	**ข**	**ค**	**ง**	**จ**	**ฉ**	**ช***
ประเภทการจัดกลุ่มข้อมูลอนุกรมเวลา (Taxonomy of Time-Series Clustering)							
Whole Time-series Clustering	✓		✓	✓	✓	✓	✓
Subsequence Time-series Clustering		✓					
การแทนข้อมูลอนุกรมเวลา (Time-Series Representation)							
Piecewise Aggregate Approximation (PAA)	✓						
Raw Time-Series Data						✓	
Symbolic Aggregate ApproXimation (SAX)	✓						
Multiresolution Vector Quantized (MVQ)		✓					
Deep Autoencoder Networks			✓		✓		✓
Deep Autoencoder Networks + Objective Function				✓			
เทคนิคนำเสนอใหม่							✓
การวัดความคล้ายคลึง/ความต่าง (Similarity or Dissimilarity Measures)							
MINDIST	✓						
Euclidean Distance	✓						✓
new Distance Function		✓	✓	✓			
Squared Hellinger Distance							✓

ตารางที่ 2.10 สรุปเปรียบเทียบงานวิจัยที่เกี่ยวข้องกับการจัดกลุ่มข้อมูลอนุกรมเวลา (ต่อ)

เทคนิคที่ใช้ / งานวิจัยที่เกี่ยวข้อง	**ก**	**ข**	**ค**	**ง**	**จ**	**ฉ**	**ช***
เทคนิคสำหรับการวิเคราะห์ (Techniques for analysis)							
k-Means Clustering	✓			✓			✓
PAM (k-Medoids) Clustering	✓	✓					
Permutation Distribution Clustering (PDC)							✓
Neural Network Classification			✓		✓	✓	
เทคนิคการประเมินการจัดกลุ่ม (Evaluation Clustering)							
Sum of Squared Error (SSE)	✓						✓
Silhouette Coefficient							✓
Cluster Evaluation (Accuracy)		✓		✓			✓
Visualization			✓	✓	✓		✓
Normalized Mutual Information				✓			
Cross-Entropy Error			✓				
Purity							✓
เวลาในการประมวลผล (Processing Time)						✓	✓
Accuracy for Classification						✓	
Precision		✓					
Recall		✓					
Sensitivity						✓	
Specificity						✓	

งานวิจัยที่เกี่ยวข้องประกอบด้วย

ก แทนงานวิจัยของ Lin และคณะ (2007)

ข แทนงานวิจัยของ Wang และคณะ (2010)

ค แทนงานวิจัยของ Hinton และ Salakhutdinov (2006)

ง แทนงานวิจัยของ Song และคณะ (2013)

จ แทนงานวิจัยของ Ranzato และคณะ (2008)

ฉ แทนงานวิจัยของ Ripoll และคณะ (2016)

ช* แทนงานของวิทยานิพนธ์ฉบับนี้

บทที่ 3
วิธีดำเนินงานวิจัย

งานวิจัยนี้เป็นการวิจัยเพื่อปรับปรุงประสิทธิภาพของการจัดกลุ่มข้อมูลอนุกรมเวลาแบบ Whole Time Series Clustering ที่จัดกลุ่มด้วยเทคนิคแบบลำดับชั้นและแบบแบ่งแยก โดยอาศัยการบูรณาการเทคนิคการเรียนรู้เชิงลึกด้วยการประยุกต์ใช้ GAs ในการหาโมเดลโครงข่ายการเรียนรู้ที่ดีที่สุดที่สามารถสร้างตัวแทนอนุกรมเวลาที่เหมาะสม สนับสนุนการจัดกลุ่มข้อมูลอนุกรมเวลาให้มีประสิทธิภาพดีขึ้น ฉะนั้นในการอธิบายวิธีดำเนินงานวิจัยจะแบ่งเนื้อหาได้ดังนี้

1. ข้อมูลสำหรับการวิจัย
2. กรอบแนวคิดงานวิจัย
3. งานวิจัยที่นำเสนอ
4. วิธีการจัดกลุ่มข้อมูลอนุกรมเวลา
5. การประเมินผลการจัดกลุ่ม

3.1 ข้อมูลสำหรับการวิจัย

ข้อมูลที่ใช้สำหรับการวิจัยนี้เป็นข้อมูลอนุกรมเวลาจำนวน 2 ชุดข้อมูล ได้แก่ ข้อมูลคลื่นไฟฟ้าหัวใจ (Electrocardiogram Signals: ECGs) และข้อมูลคลื่นไฟฟ้าสมอง (Electroencephalographic Signals: EEGs) แสดงรายละเอียดดังตารางที่ 3.1

ตารางที่ 3.1 รายละเอียดชุดข้อมูลสำหรับการวิจัย

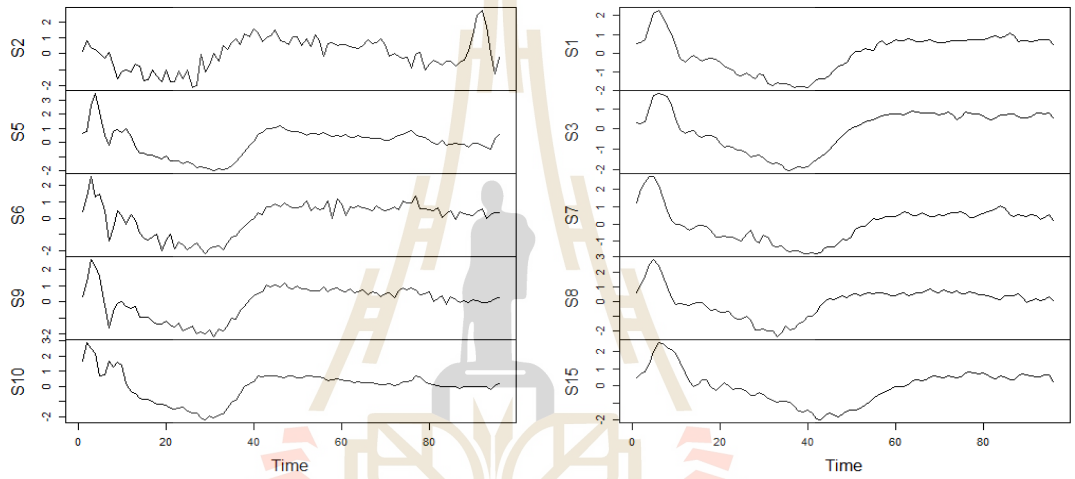
ชุดข้อมูล	ขนาดเวลา	จำนวนอนุกรม	จำนวนคลาส
ECGs	96	200	2
EEGs	4,096	100	2

โดยรายละเอียดของข้อมูลทั้ง 2 ชุดอธิบายได้ดังต่อไปนี้

(1) ข้อมูลคลื่นไฟฟ้าหัวใจ

ข้อมูลคลื่นไฟฟ้าหัวใจ (Electrocardiogram Signals: ECGs) เป็นข้อมูลจากการวินิจฉัยคลื่นไฟฟ้าหัวใจ (Electrocardiogram Diagnosis) ที่ตรวจวัดด้วยการติดขั้วไฟฟ้าไว้ตามจุดต่าง ๆ บนร่างกาย ซึ่งแต่ละอนุกรมมาจากแต่ละขั้วไฟฟ้าในช่วงหนึ่งการเต้นของหัวใจ โดยข้อมูล ECGs ที่นำ

มาใช้ในการวิจัยครั้งนี้ได้จากอุปกรณ์ตรวจบันทึกคลื่นไฟฟ้าหัวใจอย่างต่อเนื่องแบบพกพา (Holter Monitor) โดยการวัดผู้ป่วยรายเดียวในเวลาประมาณ 30 นาที ในช่วงเวลาของ RR Interval สุ่มเก็บที่ความถี่ 128 Hz ซึ่งได้คลื่นไฟฟ้าหัวใจลำดับยาวที่มี 96 Ventricular Events แบ่งออกเป็นอนุกรมที่มีความยาวเท่ากันขนาด 96 Events (คาบเวลา) ทำให้ได้ข้อมูล ECGs ที่ใช้ในการวิจัยนี้จำนวน 200 อนุกรม ที่ผ่านการวินิจฉัยจากผู้เชี่ยวชาญ โดยมี 2 คลาส คือคลาส Positive (Abnormal) ที่มีจำนวน 67 อนุกรม และคลาส Negative (Normal) จำนวน 133 อนุกรม (Olszewski, 2001) ข้อมูลนี้ดาวน์โหลดจาก UCR Time Series Classification Archive (Chen et al., 2015) แสดงตัวอย่างกราฟข้อมูล ECGs ได้ดังรูปที่ 3.1



(a) คลื่นไฟฟ้าหัวใจกรณีหัวใจเต้นปกติ (b) คลื่นไฟฟ้าหัวใจกรณีหัวใจเต้นไม่ปกติ

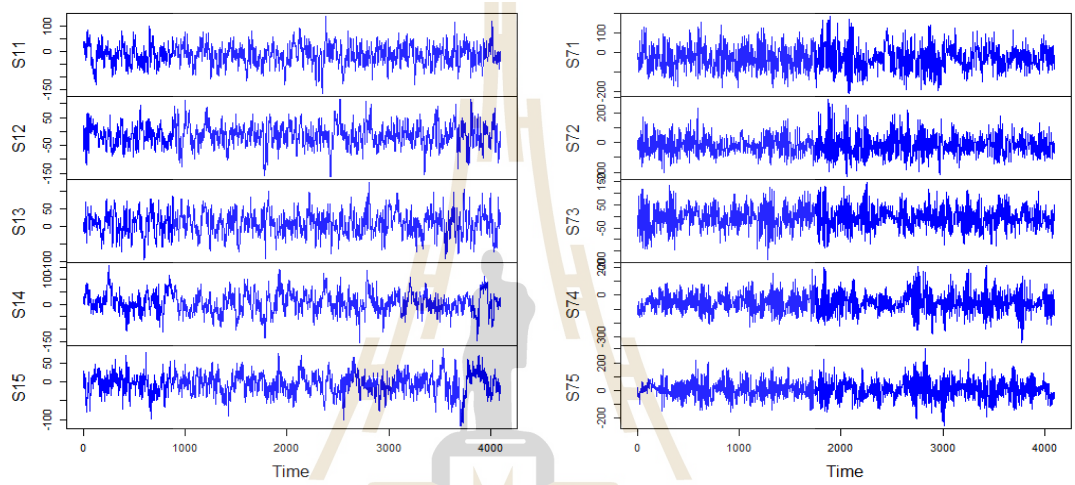
รูปที่ 3.1 ตัวอย่างกราฟข้อมูล ECGs

จากรูปที่ 3.1 แสดงตัวอย่างข้อมูล ECGs จำนวน 10 อนุกรม ได้แก่อนุกรม S1 ถึง S3, S5 ถึง S10 และ S15 โดยกลุ่ม 1 ดังรูปที่ 3.1(a) คือข้อมูลคลาส Negative เป็นการเต้นของหัวใจที่เต้นปกติได้แก่ S2, S5, S6, S9 และ S10 กลุ่ม 2 ดังรูปที่ 3.1(b) คือข้อมูลคลาส Positive เป็นการเต้นของหัวใจที่เต้นไม่ปกติ มี S1, S3, S7, S8 และ S15

(2) ข้อมูลคลื่นไฟฟ้าสมอง

ข้อมูลคลื่นไฟฟ้าสมอง (Electroencephalographic Signals: EEGs) เป็นข้อมูลการตรวจวัดคลื่นไฟฟ้าสมองแบบพิเศษทางประสาทวิทยา ที่สามารถบอกตำแหน่งและลักษณะของพยาธิสภาพของสมอง การตรวจนี้นำมาใช้ร่วมกับประวัติการตรวจร่างกายเพื่อช่วยในการวินิจฉัยโรคทางระบบประสาทที่เกี่ยวข้องกับความผิดปกติของสมอง (มณฑิรา วิทยากิตติพงษ์, 2549) ชุดข้อมูล EEGs ที่ใช้ในงานวิจัยนี้ดาวน์โหลดจาก Universitat Pompeu Fabra โดยการ

รวบรวมของ Andrzejak (2014) เป็นข้อมูลการตรวจวัดคลื่นไฟฟ้าสมองของอาสาสมัครสุขภาพดี5 คน บันทึกในช่วง 23.6 วินาทีสุ่มเก็บที่ 173.61 Hz ซึ่งส่งผลให้แต่ละอนุกรมมีขนาดความยาวเวลาเป็น 4,096 ช่วงเวลา โดยเป็นคลื่นไฟฟ้าสมองประเภท คลื่นอัลฟา (alpha) (Andrzejak et al., 2001) ซึ่งคลื่นอัลฟาเป็นคลื่นชนิดความถี่ 8-13 รอบต่อวินาที พบได้เด่นชัดที่ตำแหน่งสมองส่วนท้าย ตรวจวัดได้ในผู้ที่ปล่อยตัวตามสบาย หลับตา ไม่ได้คิดอะไร (คลาส Closed-eyes) และคลื่นอัลฟาจะหายไปเมื่อลืมตาและมีการรับรู้ของสมอง (มณฑิรา วิทยากิตติพงษ์, 2549) (คลาส Close หรือ Open-eyes) จำนวนข้อมูลที่ใช้ในการวิจัยมี 100 อนุกรม แสดงตัวอย่างกราฟข้อมูล EEGs ได้ดังรูปที่ 3.2



(a) คลื่นไฟฟ้าสมองขณะหลับตา (b) คลื่นไฟฟ้าสมองขณะลืมตา

รูปที่ 3.2 ตัวอย่างกราฟข้อมูล EEGs

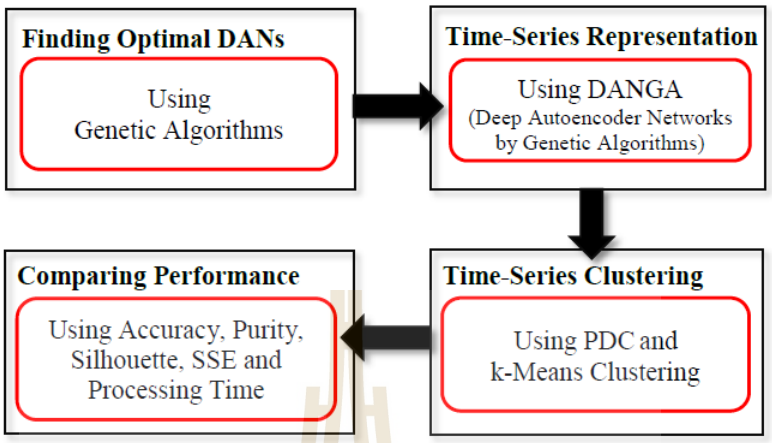
จากรูปที่ 3.2 แสดงตัวอย่างข้อมูล EEGs จำนวน 10 อนุกรม ได้แก่อนุกรม S11 ถึง S15 และ S71 ถึง S75 โดยกลุ่ม 1 คือข้อมูลคลาส Closed-eyes ได้แก่ S11 ถึง S15 ดังรูปที่ 3.2(a) กลุ่ม 2 คือข้อมูลคลาส Open-eyes ได้แก่ S71 ถึง S75 ดังรูปที่ 3.2(b)

3.2 กรอบแนวคิดงานวิจัย

จุดมุ่งหมายในการพัฒนางานวิจัยนี้คือการนำเสนออัลกอริทึมการแทนอนุกรมเวลาด้วยเทคนิคการเรียนรู้เชิงลึก (Time Series Representation with Deep Learning Technique Algorithm: TSDL Algorithm) เพื่อการจัดกลุ่มข้อมูลอนุกรมเวลาประเภท Whole Time Series Clustering ด้วยอัลกอริทึมการจัดกลุ่ม PDC และ k-Means อย่างมีประสิทธิภาพ โดยใช้ Deep Autoencoder Networks (DANs) ด้วยโมเดลที่ดีที่สุดค้นหาด้วยการประยุกต์ใช้ Genetic Algorithms (GAs) สามารถอธิบายภาพรวมของงานวิจัยได้ดังต่อไปนี้

3.2.1 ขั้นตอนหลักในการดำเนินงานวิจัย

ในการดำเนินงานวิจัยนี้ ประกอบด้วยขั้นตอนสำคัญ 4 ขั้นตอน แสดงดังรูปที่ 3.3



รูปที่ 3.3 ขั้นตอนหลักในการดำเนินงานวิจัย

แต่ละขั้นตอนชิบายรายละเอียดได้ดังต่อไปนี้

(1) การค้นหาโมเดลที่ดีที่สุดของ DANs (Finding Optimal DANs) วัตถุประสงค์หลักในการทำงานส่วนแรกคือ การค้นหาโมเดลที่ดีที่สุดของ DANs ด้วยการใช้ความสามารถของเทคนิค GAs ซึ่งผลลัพธ์ที่ได้คือ *โมเดลที่เหมาะสมสำหรับใช้ผลิตตัวแทนอนุกรมเวลา (DANGA)*

(2) การแทนอนุกรมเวลา (Time-Series Representation) การทำงานส่วนที่สองคือ การใช้โมเดล DANs ที่ดีที่สุดที่ได้จากการทำงานส่วนแรก ดังนั้นผลลัพธ์จากงานส่วนนี้คือ *ตัวแทนอนุกรมเวลาจากเทคนิค DANs (Time-Series Representative by DANGA: TSR-DANGA)*

(3) การจัดกลุ่มข้อมูลอนุกรมเวลา (Time-Series Clustering) การทำงานส่วนที่สามคือ การจัดกลุ่มข้อมูลอนุกรมเวลาด้วยอัลกอริทึมการจัดกลุ่ม PDC และ k-Means เพื่อเปรียบเทียบตัวแทนอนุกรมเวลาที่ได้จากเทคนิค PAA, SAX และ TSR-DANGA กับ Raw Data

(4) เปรียบเทียบประสิทธิภาพการจัดกลุ่ม (Comparing Ferformance) การทำงานส่วนสุดท้ายคือ ตรวจสอบประสิทธิภาพของการจัดกลุ่มตัวแทนอนุกรมเวลาที่ได้จากผลงานที่นำเสนอเปรียบเทียบกับทั้ง Raw Data และตัวแทนอนุกรมเวลาโดยใช้ 5 มาตรวัดได้แก่ Accuracy, Purity, Silhouette, SSE และ Processing Time

3.2.2 เครื่องมือที่ใช้สำหรับการวิจัย

(1) เครื่องมือที่ใช้ในการพัฒนางานวิจัยนี้ประกอบด้วย

เครื่องคอมพิวเตอร์ หน่วยประมวลผลกลาง: Intel(R) Core(TM) i7-3517U CPU @1.90GHz 2.40 GHz หน่วยความจำสำรอง: 500 GB และหน่วยความจำหลัก: 8 GB

(2) ระบบปฏิบัติการและโปรแกรมประยุกต์สำหรับการพัฒนา ประกอบด้วย

1) ระบบปฏิบัติการ : Windows 7 Home Premium (64 Bit)

2) เครื่องมือที่ใช้ในการพัฒนา : RStudio Version 1.0.143

3.3 งานวิจัยที่นำเสนอ (The Proposed Work)

งานวิจัยนี้มีเป้าหมายหลักคือนำเสนอ "*อัลกอริทึมการหาตัวแทนอนุกรมเวลาด้วยเทคนิคการเรียนรู้เชิงลึก (TSDL Algorithm)*" เพื่อให้การจัดกลุ่มข้อมูลอนุกรมเวลามีประสิทธิภาพมากขึ้น โดยใช้เทคนิค DANs ด้วยโมเดลที่ดีที่สุดจากการประยุกต์ใช้ GAs ดังนั้นการทำงานของอัลกอริทึมที่นำเสนอประกอบด้วยองค์ประกอบย่อยของการทำงานหลายส่วน อธิบายได้ดังต่อไปนี้

3.3.1 แนวคิดการกำหนดโมเดลของ DANs

ดังที่กล่าวไปแล้วว่าถึงแม้ DANs จะโดดเด่นในการเรียนรู้เพื่อแทนข้อมูลมิติสูงแต่มีข้อจำกัดเรื่องของการกำหนดโมเดลที่ดีที่สุดหรือที่เหมาะสมที่สุดของโครงข่าย DANs ซึ่งพบได้จากงานวิจัยจำนวนมากที่ใช้งาน DANs ในรูปแบบที่แตกต่างเพื่อแทนข้อมูล สามารถสรุปได้ดังนี้

1) สำหรับโครงสร้าง Encoder Networks มีการกำหนดจำนวนชั้น Hidden (Hidden Layer) จำนวนตั้งแต่ 1-5 ชั้น

2) ใน Hidden Layer เมื่อพิจารณาเฉพาะงานวิจัยที่ผู้วิจัยได้ทบทวนวรรณกรรมทั้งหมดพบว่ามีการกำหนดจำนวนชั้น ตั้งแต่ 1-6 ชั้น มีการกำหนดจำนวน Node (Unit) ในชั้น Hidden เมื่อพิจารณาในแต่ละโมเดลสรุปได้ดังนี้

กรณี 6-Hidden Layers โครงข่ายมีการกำหนดจำนวน Unit ดังนี้ 400-200-100-50-25-6 (พบเพียงงานเดียวที่มี 6-Hidden Layers)

กรณี 5-Hidden Layers จากการศึกษาไม่พบการใช้งาน

กรณี 4-Hidden Layers โครงข่ายมีการกำหนดจำนวน Unit ในแต่ละชั้น Hidden ดังนี้ Hidden1: 1000-2048, Hidden2: 250-1024, Hidden3: 50-512 และ Hidden4 (Code Layer): 10-256 Units

กรณี 3-Hidden Layers โครงข่ายกำหนดจำนวน Unit ในแต่ละชั้น Hidden ดังนี้ Hidden1: 170-700, Hidden2: 100-700 และ Hidden3 (Code Layer): 30-700 Units

กรณี 2-Hidden Layers โครงข่ายมีการกำหนดจำนวน Unit ในแต่ละชั้น Hidden ดังนี้ Hidden1: 128-2000, Hidden (Code Layer): 64-680 Units

กรณี 1-Hidden Layers กำหนดจำนวน Unit ในชั้น Hidden (Code Layer): 10 (เป็นกรณี Autoencoder แบบดั้งเดิม)

รายละเอียดดังกล่าวมาข้างต้นได้จากงานวิจัยจำนวน 9 ผลงานที่เกี่ยวข้องกับการประยุกต์ใช้ Autoencoder Networks ทั้งแบบดั้งเดิมและแบบเชิงลึกสรุปได้ดังตารางที่ 3.2

ตารางที่ 3.2 โมเดลของโครงข่ายการเข้ารหัสจากการทบทวนงานวิจัย

No	Model-of-Networks	Reference
1	Autoencoder Traditional (Input-Hidden-Output)	(Kramer, 1992)
2	Autoencoder Traditional = 1-**10**-1 (NARMA, Cauchy, Wind, Textual, Discarded Symbols)	(Gianniotis et al., 2016)
3	(28 x 28)-**400-200-100-50-25-6** (Curves) (784)-**1000-500-250-30** (MNIST) (625)-**2000-1000-500-30** (face)	(Hinton and Salakhutdinov, 2006)
4	(1024)-**200-80** (COIL.20) (1200)-**500-100-30** (Yale-B) (1024)-**2000-680** (PIE)	(Huang et al., 2014) (Manually choose)
5	(28 x 28)-**1000-250-50-10** (MNIST) (16 x 16)-**1000-250-50-10** (USPS) (1200)-**1000-250-50-10** (Yale-B)	Song et al., 2013
6	(Input)-**700-500-500** (Textual autoencoder), (Input)-**170-100-500** (Visual autoencoder)	Silberer and Lapata (2014)
7	(178)-**128-64** (Wine) (600)-**512-256** (3-NG) (1200)-**1024-512-256-128** (6-NG) (1800)-**1024-512-256-128** (9-NG) (4741)-**2048-1024-512-256** (DIP) (5964)-**2048-1024-256-128** (BioGrid)	Tian et al., 2014
8	(72 x 72)-**1000-500-250-30** (PSB, NTU) (72 x 72)-**2000-500-100-20** (ESB)	Zhu et al., 2016
9	(Input)-**700-700-700** (ECGs)	Ripoll et al., 2016

3.3.2 แนวคิดการเลือกใช้ Genetic Algorithm ค้นหาโมเดลที่ดีที่สุด

ดังที่กล่าวไปแล้วในตอนต้นว่าโครงข่าย DANs มีข้อจำกัดในเรื่องของการกำหนดโมเดลที่ดีที่สุดหรือที่เหมาะสมที่สุด เพื่อให้การดำเนินงานบรรลุตามวัตถุประสงค์ งานวิจัยนี้จำเป็นต้องอาศัยเทคนิคสำหรับการหาค่าเหมาะสมที่สุด (Optimization) ซึ่งปัจจุบันมีเทคนิคมากมายสำหรับการทำ Optimization ตัวอย่างเทคนิคที่โด่งดังเป็นที่นิยมทั่วไปดังนี้

1) วิธีหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค (Particle Sworm Optimization) โดย Kennedy และ Eberhart (1995) เป็นเทคนิคที่มีแนวคิดจากพฤติกรรมการเดินทางหรือหาอาหารของฝูงสัตว์ โดยเฉพาะฝูงนก ฝูงปลา จึงเหมาะกับผลลัพธ์ที่แทนด้วยจุดหรือเวคเตอร์ ที่ต้องการตำแหน่งที่ดีที่สุดด้วยความเร็วที่ดีที่สุด เหมาะกับงานที่มีประชากรที่หลากหลาย หากความหลากหลายของประชากรน้อยลงจะส่งผลให้การค้นหาคำตอบที่ดีที่สุดจำกัดไปด้วย

2) วิธีหาค่าเหมาะสมที่สุดด้วยระบบอาณาจักรมด (Ant Colony Optimization) โดย Marco Dorigo (1992) ที่อาศัยแนวคิดจากพฤติกรรมการหาอาหารของมดที่ค้นหาเส้นทางที่สั้นที่สุดระหว่างแหล่งอาหารและรังของมัน มักนำมาใช้แก้ปัญหาในการหาคำตอบที่อาจไม่ดีที่สุดแต่เป็นคำตอบที่ยอมรับได้ ซึ่งวิธีการนี้มีข้อจำกัดในเรื่องสมถรรนะ คือมีการใช้เวลาในการหาคำตอบที่ค่อนข้างมาก

3) วิธีหาค่าเหมาะสมที่สุดด้วยอัลกอริทึมเชิงพันธุกรรม (Genetic Algorithms) โดย John Holland (1975) ดังที่กล่าวไปแล้วในขั้นต้นว่า AGs เป็นเทคนิคที่อาศัยหลักการการวิวัฒนาการของสิ่งมีชีวิต โดยโอกาสการค้นพบคำตอบที่ดีที่สุดได้จากการดำเนินการทางพันธุกรรม คือ Selection, Crossover และ Mutation ซึ่งหลักการทำงานง่าย ไม่ซับซ้อน จัดว่าเป็นวิธีการที่ทำงานได้เร็วกว่า Ant Colony Optimization และถึงแม้ว่าวิธีแบบ GAs จะเหมาะกับงานที่มีประชากรที่หลากหลายเช่นเดียวกับวิธี Particle Sworm Optimization แต่สามารถแก้ปัญหาได้ด้วยการปรับแต่งพารามิเตอร์ที่เกี่ยวข้องกับกระบวนการในการดำเนินการทางพันธุกรรมได้

ดังนั้นเมื่อเปรียบเทียบประสิทธิภาพ และข้อจำกัดในการทำงานแล้ว ***งานวิจัยนี้จึงเลือกใช้เทคนิคการหาค่าที่เหมาะสมด้วยการใช้ GAs สำหรับค้นหาโมเดลที่ดีที่สุดสำหรับ DANs***

3.3.3 การกำหนดค่าพารามิเตอร์

งานวิจัยนี้ได้ออกแบบการดำเนินงานในส่วนของการค้นหาโมเดลที่ดีที่สุดสำหรับ DANs ด้วยเทคนิค GAs เพื่อสร้างตัวแทนอนุกรมเวลาที่เหมาะสมที่สุด โดยการกำหนดค่าพารามิเตอร์ที่เกี่ยวข้องกับกระบวนการทำงาน มี 2 ส่วนดังนี้

(1) พารามิเตอร์สำหรับ DANs ในกระบวนการทำงานของ DANs ประกอบด้วยการตั้งค่าสำหรับการประมวลผล ดังต่อไปนี้

(1.1) โครงข่ายการเรียนรู้ของ DANs: จากตารางที่ 3.2 สามารถสรุปได้ว่างานวิจัยส่วนใหญ่กำหนดจำนวนชั้น Hidden สูงสุดที่ 4 ชั้น รองลงมาคือ 3 ชั้น และเมื่อพิจารณาศักยภาพการทำงานของเครื่องมือ และอุปกรณ์ที่ใช้ในงานวิจัยพบว่าการเพิ่มจำนวนชั้นสูงขึ้น เป็น 4-6 ชั้นสำหรับข้อมูลคลื่นไฟฟ้าหัวใจ และคลื่นไฟฟ้าสมองไม่ช่วยให้ผลิตตัวแทนอนุกรมเวลาที่มีประสิทธิภาพได้สูงขึ้นมากอย่างมีนัย ทั้งยังมีการใช้ทรัพยากรเพิ่มขึ้น และเวลาในการประมวลผลสูงขึ้นมาก (จากการทดลองเพื่อสำรวจการใช้ทรัพยากร) ***ดังนั้นงานวิจัยนี้กำหนดโมเดลให้มีโครงข่าย Encoder มีจำนวน 3 ชั้น Hidden***

นอกจากนี้เมื่อพิจารณาจำนวน Unit ที่กำหนดในแต่ละชั้น Hidden พบว่ามีการกำหนดจำนวน Unit ที่หลากหลาย ขึ้นอยู่กับประเภทของข้อมูลนำเข้า ดังนี้

- ข้อมูลภาพ มี Unit สอดคล้องกับขนาดภาพเช่น 128, 512 และ 1,024 เป็นต้น
- ข้อมูลอื่นรวมถึงข้อมูลอนุกรมเวลา ส่วนใหญ่กำหนดจำนวน Unit ในชั้นบนเป็นเลขจำนวนเต็ม 10, 100 หรือ 1,000 โดยมีค่าตั้งแต่ 200-2000 และมีจำนวนลดลั่นในชั้นที่ต่ำลงไป (ซึ่งเป็นคุณลักษณะโมเดลที่มักกำหนดสำหรับ Autoencoder) เมื่อกำหนดให้ X เป็นจำนวน Unit ในชั้น Hidden ชั้นบนสุด และโครงข่ายของ Autoencoder นิยามเป็น

Hidden1 - Hidden2 - … - Code Layer

จึงสามารถจำแนกจำนวน Unit ของชั้น Hidden ได้เป็น 2 กรณีดังนี้

กรณีที่ 1 กำหนดจำนวน Unit ลดลั่นกันไปในแต่ละชั้น ดังนั้นจำนวน Unit ในชั้น Hidden แต่ละชั้นนิยามได้ดังนี้

X - {(X/4), (X/2)} - {(X/8), (X/4)} - … - {1-30}

กรณีที่ 2 กำหนดจำนวน Unit ไม่ลดลั่น ดังนั้นจำนวน Unit ในชั้น Hidden แต่ละชั้นจะนิยามได้ดังนี้

X - X - … - X

ดังนั้นเพื่อให้เหมาะสม และสอดคล้องกับศักยภาพของเครื่องมือที่ใช้ในการวิจัย ***งานวิจัยนี้จึงออกแบบให้โครงข่ายของชั้น Hidden มีลักษณะลดลั่นกันใกล้เคียงกับรูปแบบกรณีที่ 1***

เมื่อพิจารณาจากการทบทวนงานวิจัยร่วมกับความเหมาะสมของทรัพยากรที่ใช้สำหรับการพัฒนางานวิจัย ในงานวิจัยนี้จึงกำหนดโครงข่ายการเรียนรู้ของ DANs แบบ 3-Hidden Layers โดยกำหนดค่า ต่ำสุด-สูงสุด ของ Unit ในแต่ละชั้น Hidden ของโครงข่าย Encoder ดังนี้

- Hidden Layer ชั้นที่ 1 กำหนดให้มีค่าตั้งแต่ 500-1000 Units
- Hidden Layer ชั้นที่ 2 กำหนดให้มีค่าตั้งแต่ 100-500 Units
- Hidden Layer ชั้นที่ 3 (Code Layer) กำหนดให้มีค่า 1-30 Units

(1.2) จำนวน Epoch: สำหรับการเรียนรู้ใน DANs กำหนดให้มีจำนวนรอบในการเรียนรู้เพื่อสร้างตัวแทนข้อมูลทั้งชุดข้อมูล ECGS และ EEGs มีจำนวน Epoch = 50 (เนื่องจากข้อจำกัดด้านทรัพยากรในการประมวลผลทำให้ไม่สามารถกำหนดได้สูงกว่านี้)

(1.3) พารามิเตอร์อื่น ได้แก่ อัตราการเรียนรู้ = 0.1 กำหนด Unit Function คือ softplusUnit Function และ FintuneFunciton คือ Backpropagation

(2) พารามิเตอร์สำหรับ GAs เนื่องจากวัตถุประสงค์หลักในการประยุกต์ใช้ GAs คือ เพื่อหาโมเดลที่ดีที่สุดสำหรับ DANs ที่สามารถผลิตตัวแทนอนุกรมเวลาที่ดีที่สุดสำหรับการจัดกลุ่มข้อมูลอนุกรมเวลา ดังนั้นพารามิเตอร์ที่เกี่ยวข้องกับกระบวนการทำงานของ GAs มีดังนี้

(2.1) การกำหนดประชากร: กำหนดให้ประชากร *popSizes (P)* = 20

(2.2) การเข้ารหัส Chromosome: กำหนดให้เข้ารหัส Chromosome เป็นแบบค่าจริงประกอบด้วย Gene จำนวน 3 Genes ซึ่งเท่ากับจำนวนชั้น Hidden ดังนี้

Gene1 แทน Hidden1 มีค่าระหว่าง 500-1000

Gene2 แทน Hidden2 มีค่าระหว่าง 100-500

Gene3 แทน Hidden3 (Code Layer) มีค่าระหว่าง 1-30

ดังนั้นจะได้รหัส Chromosome ซึ่งใช้แทนโครงข่ายของ DANs ดังนี้

Chromosome_DANGA = (Hidden1(500-1000), Hidden2(100-500), Hidden3(1-30))

(2.3) อัตราการ Crossover และ Mutation: อัตราการ Mutation = 0.05 สำหรับอัตราการ Crossover ได้จากความน่าจะเป็นของประชากรรุ่นพ่อ-แม่ แต่ละตัวในการสร้างรุ่นลูกหลาน คำนวณได้จากการหาความสูงของการกระจายความน่าจะเป็นที่จุดสำหรับค่าเฉลี่ยและค่าเบี่ยงเบนมาตรฐานของแต่ละค่า (dnorm) ดังนี้

parentProb = dnorm(1:popSizes, mean = 0, sd = (popSizes/3))

(2.4) การประเมินค่าความเหมาะสม: ค่าที่ใช้สำหรับประเมินความเหมาะสมของแต่ละโมเดลในงานวิจัยนี้คือ ค่า Purity ซึ่งจะถูกใช้ใน Fitness Function

(2.5) การคัดเลือกประชากร: กำหนดให้เป็นวิธีแบบจัดอันดับ

(2.6) การแทนที่ประชากร: กำหนดให้เป็นวิธีแทนที่ประชากรทั้งรุ่น โดยคัดเลือกเก็บหัวกะทิไว้จำนวน 20%

(2.7) การตรวจสอบเงื่อนไขเพื่อสิ้นสุดการทำงาน: กำหนดให้การทำงานของกระบวนการ GAs สิ้นสุดเมื่อผลิตประชากรรุ่นใหม่ได้จำนวน 3 รุ่นนั่นคือ *Iteration M = 3* (เนื่องจากข้อจำกัดด้านทรัพยากรในการประมวลผล)

3.3.4 ขั้นตอนการทำงานของ TSDL Algorithm

งานวิจัยนี้มุ่งเน้นในการนำเสนอ TSDL Algorithm โดยมีองค์ประกอบสำคัญได้แก่ การค้นหาโมเดลที่ดีที่สุดสำหรับ DANs ด้วย GAs และการแทนอนุกรมเวลาเพื่อการจัดกลุ่มที่มีประสิทธิภาพ สำหรับข้อมูล ECGs และ EEGs มีขั้นตอนการทำงานดังต่อไปนี้

(1) ขั้นที่ 1 การนำเข้าข้อมูลและกำหนดค่าเริ่มต้นการทำงาน มีดังนี้

(1.1) Dataset *X* = ชุดข้อมูลอนุกรมเวลา Raw Data ซึ่งงานวิจัยนี้ใช้ข้อมูล 2 ชุดคือ ECGs และ EEGs โดยจะทำงานทีละชุดข้อมูล

(1.2) Population Size *(P)* คือจำนวนประชากรที่ต้องการมีค่า 20

(1.3) Chromosome Encoding: เข้ารหัส Chromosome แบบค่าจริง

(1.4) Initialize Population: กำหนดประชากรเริ่มต้นโดยการสุ่ม

(1.5) Iteration *M* = 3 (เงื่อนไขการสิ้นสุดการทำงานของ GAs)

(2) ขั้นที่ 2 การประเมินค่าความเหมาะสม สำหรับงานวิจัยนี้กำหนดให้การทำงานเพื่อประมวลค่า Purity ประกอบด้วยการทำงาน 3 ส่วนหลัก มีขั้นตอนการทำงานดังนี้

(2.1) การแทนอนุกรมเวลาด้วยเทคนิค DANs ที่กำหนดโครงข่ายการเรียนรู้จาก Chromosome ของประชากรแต่ละตัว ตัวอย่างเช่น

สมมุติให้รหัส Chromosome ที่ 1 คือ (500, 200, 10) ดังนั้นโครงข่ายการเรียนรู้โมเดลที่ 1 สำหรับ DANs เพื่อสร้างตัวแทนอนุกรมเวลา คือ (500-200-10-200-500) ดังนั้นเมื่อผ่านขั้นตอนนี้ จะได้ตัวแทนอนุกรมเวลาที่ผลิตจากโมเดลที่ 1 (TSR-โมเดล1)

(2.2) การจัดกลุ่มตัวแทนอนุกรมเวลาที่ได้จากเทคนิค DANs ด้วยอัลกอริทึม PDC กำหนดค่า k=2 เป็นขั้นตอนที่นำตัวแทนอนุกรมเวลาที่ผลิตได้จากประชากรแต่ละตัวมาจัดกลุ่มด้วยอัลกอริทึม PDC

(2.3) การคำนวณ Fitness Value ซึ่งงานวิจัยนี้ใช้การคำนวณค่า Purity ดังนั้นในขั้นตอนนี้จะนำผลลัพธ์จากการจัดกลุ่มตามขั้นตอนที่ (2.2) มาหาค่า Purity

(3) ขั้นที่ 3 การดำเนินงานเชิงพันธุกรรม มีการทำงานดังนี้

(3.1) การดำเนินการทางพันธุกรรม (Genetic Operations) ประกอบด้วยการ Selection, Crossover และ Mutation โดยประมวลผลตามพารามิเตอร์ที่กำหนด

(3.2) การแทนที่ (Replacement) ประชากรใหม่ ตามพารามิเตอร์ที่กำหนด

(4) ขั้นที่ 4 การตรวจสอบเงื่อนไขการสิ้นสุดการทำงาน ซึ่งงานวิจัยนี้กำหนดให้ทำงานไปจนกระทั่งผลิตประชากรรุ่นลูกหลานได้จำนวน 3 รุ่น (Iteration $M = 3$)

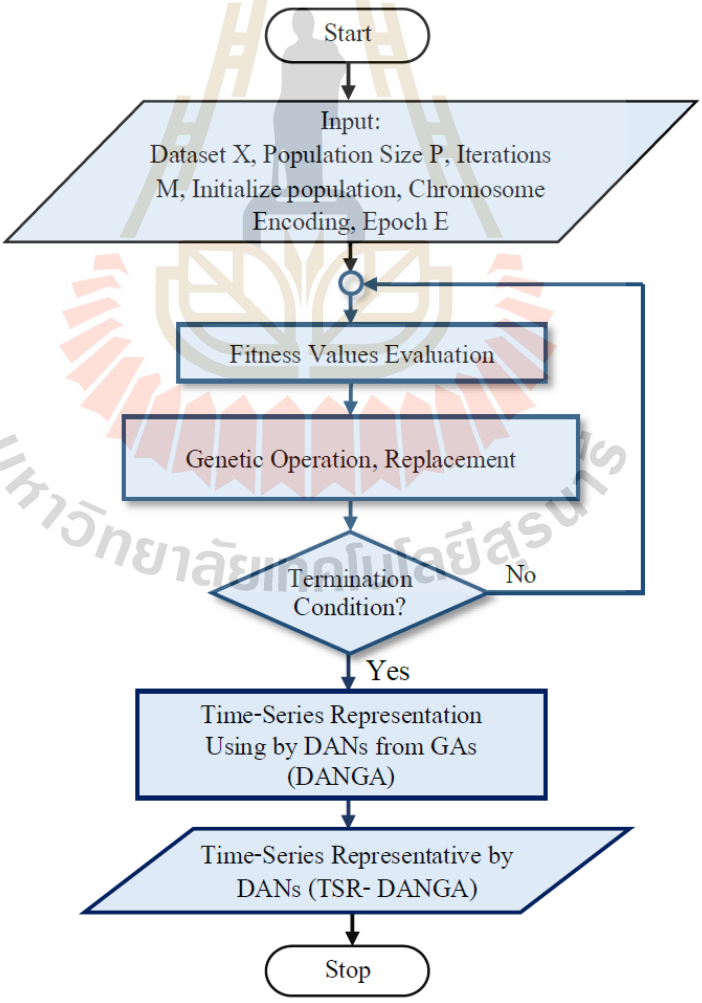
(4.1) กรณีเงื่อนไขเป็นจริง (YES) วงจรการทำงานตามขั้นตอน GAs จะสิ้นสุด และไปทำงานตามขั้นตอนของ TSDL Algorithm ในขั้นตอนถัดไป

(4.2) กรณีเงื่อนไขเป็นเท็จ (NO) ให้กลับไปทำงานในขั้นที่ 2

(5) ขั้นที่ 5 การแทนอนุกรมเวลาด้วยโมเดล DANs ที่ดีที่สุดจาก GAs ซึ่งงานวิจัยนี้เรียกโมเดลนี้ว่า ***DANGA***

(6) ขั้นที่ 6 ผลลัพธ์ที่ได้การแทนอนุกรมเวลาด้วยโมเดล DANGA ซึ่งงานวิจัยนี้ถือเป็นตัวแทนที่ดีที่สุดสำหรับใช้แทนอนุกรมเวลา ซึ่งเรียกว่า ***TSDL-DANGA***

เมื่อแสดงขั้นตอนการทำงานของ TSDL Algorithm ทั้ง 6 ขั้นตอนด้วยแผนผังขั้นตอนการทำงาน (Flowchart) จะได้ดังรูปที่ 3.4



รูปที่ 3.4 แผนผังแสดงขั้นตอนการทำงานของ TSDL Algorithm

3.3.5 รหัสเทียม TSDL Algorithm

จากการอธิบายขั้นตอนการทำงาน และแผนผังแสดงขั้นตอนการทำงานของ TSDL Algorithm สามารถนำมาแสดงให้อยู่ในรูปแบบของรหัสเทียม (Pseudo Code) จะได้รหัสเทียมของ TSDL Algorithm ดังรูปที่ 3.5 ดังนี้

Algorithm: TSDL Algorithm

```
1:  input: Dataset X; Iterations M; the number of epoch E; popSizes P; Chromosome Encoding;
2:  Randomly initialize population;
3:  set counter m to 1;
4:  while m < M do
5:      repeat
6:          Fitness values evaluation;
7:          if Crossover condition satisfied then
8:            { Select parent Chormosome;
9:              Choose crossover parameters;
10:             Perform crossover};
11:         if Mutation condition satisfied then
12:           { Choose mutation points;
13:             Perform mutation};
14:     until Sufficient offstpring created;
15:     Select new population and Replacement;
16:     m = m + 1.
17: endwhile
18: get The Best of DANs (DANGA);
19: Time-Series Representation uing DANGA (TSR-DANGA)
20: output: Time-Series Representative by DANGA (TSR-DANGA)
```

รูปที่ 3.5 รหัสเทียมของ TSDL Algorithm

3.4 วิธีการจัดกลุ่มข้อมูลอนุกรมเวลา

วัตถุประสงค์สำคัญในการวิจัยนี้คือการแทนอนุกรมเวลาที่จะช่วยให้การจัดกลุ่มมีประสิทธิภาพดีขึ้น ดังนั้นในขั้นตอนการจัดกลุ่มข้อมูลอนุกรมเวลาจะกล่าวถึงการดำเนินการจัดกลุ่ม เพื่อหาคำตอบว่าตัวแทนข้อมูลใดมีความเหมาะสม ซึ่งมีรายละเอียดในการทำงานดังนี้

3.4.1 ตัวแทนข้อมูลสำหรับการจัดกลุ่ม

เมื่อเสร็จสิ้นกระบวนการตามขั้นตอนการทำงานของ TSDL Algorithm จะได้ตัวแทนอนุกรมเวลา ซึ่งงานวิจัยนี้เรียกว่า TSR-DANGA สำหรับแทนชุดข้อมูลที่ใช้ในการวิจัยนี้จำนวน 2 ชุด นั่นคือตัวแทนสำหรับข้อมูล ECGs และ EEGs

3.4.2 เทคนิคการจัดกลุ่มข้อมูล

อัลกอริทึมการจัดกลุ่มข้อมูลในงานวิจัยนี้ใช้เทคนิคการจัดกลุ่ม 2 เทคนิค ได้แก่

(1) เทคนิคการจัดกลุ่มข้อมูลอนุกรมเวลาแบบลำดับชั้นด้วยอัลกอริทึม PDC ซึ่งจัดเป็นเทคนิคที่เป็นที่นิยมและมีความเหมาะสมสำหรับนำมาใช้เพื่อการจัดกลุ่มข้อมูลอนุกรมเวลา เนื่องจากรูปแบบการแสดงผลที่สนับสนุนกับการแสดงรูปร่างของอนุกรมเวลาหลังการจัดกลุ่ม และอัลกอริทึม PDC เป็นวิธีการที่พิจารณาจัดกลุ่มข้อมูลจากการหาความคล้ายคลึงในการเปลี่ยนแปลงรูปร่างของอนุกรมเวลาเป็นฐาน ซึ่งสอดคล้องกับคุณลักษณะเด่นของข้อมูลอนุกรมเวลา

(2) เทคนิคการจัดกลุ่มแบบแบ่งแยกด้วยอัลกอริทึม k-Means ซึ่งเป็นเทคนิคพื้นฐานที่มักถูกใช้งานในการจัดกลุ่มข้อมูลทั่วไปเนื่องจากมีหลักการทำงานที่ง่าย ไม่ซับซ้อน ประมวลผลได้เร็ว

3.5 การเปรียบเทียบประสิทธิภาพการจัดกลุ่ม

การเปรียบเทียบประสิทธิภาพของการจัดกลุ่มข้อมูลอนุกรมเวลาในงานวิจัยนี้ เป็นขั้นตอนในการนำผลการจัดกลุ่มด้วยอัลกอริทึม PDC และ k-Means จากข้อมูลทั้งหมด ได้แก่ TSDL-DANGA, Raw Data, ตัวแทนจากเทคนิค PAA และ ตัวแทนจากเทคนิค SAX ที่ได้มาเพื่อเปรียบเทียบประสิทธิภาพด้วยมาตรวัดที่กำหนดได้แก่

กรณีที่ 1 การประเมินภายนอก สำหรับอนุกรมเวลาที่ทราบกลุ่มที่แท้จริงมาก่อน

1) ประเมินด้วยค่า Accuracy
2) ประเมินด้วยค่า Purity
3) ประเมินด้วย Processing Time

กรณีที่ 2 การประเมินภายใน สำหรับอนุกรมเวลาที่ไม่ทราบกลุ่มที่แท้จริงมาก่อน

1) ประเมินด้วยค่า Silhouette เพื่อประเมินความสอดคล้องของธรรมชาติข้อมูลกับผลการจัดกลุ่ม เพื่อให้ได้ค่าจำนวนกลุ่มที่เหมาะสม (Optimal k)

2) ประเมิน Optimal k โดยอาศัยค่า Sum of Squared Error (SSE)

3) ประเมินด้วย Processing Time

บทที่ 4
ผลการศึกษา และการวิเคราะห์ผล

งานวิจัยนี้เป็นการศึกษาที่มุ่งเน้นในการพัฒนาวิธีการหาตัวแทนอนุกรมเวลาสำหรับใช้ในการจัดกลุ่มเพื่อให้การจัดกลุ่มมีประสิทธิภาพมากขึ้น ซึ่งจากการดำเนินงานตามกรอบแนวคิดของงานวิจัยที่ประกอบด้วยการทำงานหลัก 4 ส่วน ได้แก่ การค้นหาโมเดลที่ดีที่สุดสำหรับ DANs ด้วยเทคนิค GAs (DANGA) การแทนอนุกรมเวลาด้วย DANGA การจัดกลุ่มข้อมูลอนุกรมเวลา และการเปรียบเทียบประสิทธิภาพ โดยสามารถสรุปผลการศึกษา และวิเคราะห์ผลได้ดังต่อไปนี้

4.1 ผลการค้นหาโมเดลที่ดีที่สุดสำหรับ DANs

จากการประยุกต์ใช้เทคนิคที่มีประสิทธิภาพอย่างเทคนิค GAs ในการค้นหาโมเดลของ DANs ที่ดีที่สุด เพื่อใช้ในการแทนอนุกรมเวลา สำหรับข้อมูล ECGs และ EEGs ซึ่งในขั้นตอนการทำงานต้องการผลลัพธ์ของการดำเนินการทางพันธุกรรมประชากรจำนวน 20 Chromosomes (หรือโมเดล) จำนวน 3 รุ่น ซึ่งเป็นจำนวนประชากรที่ต้องการ โดยในแต่ละโมเดลประกอบด้วย Gene ซึ่งใช้แทนจำนวน Unit ในชั้น Hidden ของโครงข่าย Encoder สำหรับ DANs

ผลการดำเนินงานในขั้นตอนนี้จะอธิบายแยกออกเป็นสองส่วนได้แก่ โมเดลที่ดีที่สุดสำหรับข้อมูล ECGs และโมเดลที่ดีที่สุดสำหรับข้อมูล EEGs โดยสามารถอธิบายผลการศึกษาและวิเคราะห์ผลของแต่ละส่วนได้ดังนี้

4.1.1 โมเดลที่ดีที่สุดสำหรับข้อมูล ECGs

จากการค้นหาโมเดลที่ดีที่สุดสำหรับ DANs ด้วยเทคนิค GAs สำหรับข้อมูล ECGs สามารถแสดงผลการค้นหาโมเดลในแต่ละรุ่นจำนวน 3 รุ่น ในการแสดงผลจะเริ่มต้นที่ประชากรรุ่นที่ 1 แสดงผลการดำเนินงานได้ดังตารางที่ 4.1 ซึ่งพบว่าโมเดลที่มีค่า Fitness สูงสุด (หรือเรียกว่า หัวกะทิ) 4 อันดับแรก คือโมเดลที่มี Fitness Values สูงสุดอันดับแรกคือ 76.0 ได้แก่โมเดลที่มีจำนวน Unit ในชั้น Hidden1, Hidden2 และ Code Layer เป็น 500-500-10, 600-200-20 และ 900-200-20 อันดับสองคือ 75.5 ได้แก่โมเดล 1000-100-8 และ 1000-500-3 อันดับสามคือ 75.0 ได้แก่โมเดล 1000-100-4 และ 1000-300-10 และอันดับสี่คือ 74.5 ได้แก่โมเดล 500-400-6 และ 800-500-30 ซึ่งการแสดงผลในตารางที่ 4.1 เป็นตัวหนาสีแดง ดังนี้

ตารางที่ 4.1 ประชากรรุ่นที่ 1 ในการค้นหา DANs ด้วยเทคนิค GAs สำหรับข้อมูล ECGs

#Model	Hidden1	Hidden2	Code Layer	Fitness Values
1	**500**	**400**	**6**	**74.5**
2	**500**	**500**	**10**	**76.0**
3	600	200	8	69.0
4	**600**	**200**	**20**	**76.0**
5	600	400	20	71.0
6	700	400	10	66.5
7	800	100	20	66.5
8	800	200	1	66.5
9	800	200	6	72.0
10	800	200	10	73.5
11	800	400	20	66.5
12	**800**	**500**	**30**	**74.5**
13	**900**	**200**	**20**	**76.0**
14	900	300	8	66.5
15	900	400	10	66.5
16	**1000**	**100**	**4**	**75.0**
17	**1000**	**100**	**8**	**75.5**
18	1000	200	9	71.5
19	**1000**	**300**	**10**	**75.0**
20	**1000**	**500**	**3**	**75.5**

เมื่อประมวลผลต่อเนื่องจนสามารถผลิตประชากรรุ่นที่ 2 ผลการดำเนินงานจะพบว่าในประชากรรุ่นที่ 2 มีการเก็บหัวกะทิจากประชากรรุ่นที่ 1 ไว้จำนวน 3 อันดับ ได้แก่ Chromosome ที่มีค่า Fitness เป็นอันดับหนึ่งมีค่า 76.0 ได้แก่โมเดลที่มีจำนวน Unit ในชั้น Hidden1, Hidden2 และ Code Layer เป็น 600-200-20 และ 900-200-20 โมเดลที่มีค่า Fitness เป็นอันดับสองมีค่า 75.5 ได้แก่โมเดล 1000-500-3 และโมเดลที่มีค่า Fitness เป็นอันดับสามมีค่า 75.0 ได้แก่โมเดล 1000-300-10 โดยผลลัพธ์ดังกล่าวแสดงในตารางที่ 4.2 แสดงด้วยตัวเลขเป็นตัวหนาสีแดงดังนี้

ตารางที่ 4.2 ประชากรรุ่นที่ 2 ในการค้นหา DANs ด้วยเทคนิค GAs สำหรับข้อมูล ECGs

#Model	Hidden1	Hidden2	Code Layer	Fitness Values
1	500	200	20	66.5
2	600	100	8	66.5
3	**600**	**200**	**20**	**76.0**
4	700	400	10	66.5
5	800	200	3	67.5
6	*900*	*200*	*3*	*76.0*
7	**900**	**200**	**20**	**76.0**
8	**900**	**200**	**20**	**76.0**
9	**900**	**200**	**20**	**76.0**
10	*900*	*300*	*10*	*76.0*
11	*1000*	*100*	*30*	*78.0*
12	1000	200	20	66.5
13	1000	300	8	66.5
14	**1000**	**300**	**10**	**75.0**
15	**1000**	**300**	**10**	**75.0**
16	1000	300	30	71.0
17	**1000**	**500**	**3**	**75.5**
18	**1000**	**500**	**3**	**75.5**
19	**1000**	**500**	**3**	**75.5**
20	1000	500	10	66.5

นอกจากนี้ในตารางที่ 4.2 ยังพบว่ามีประชากรใหม่ที่ให้ค่า Fitness ดีกว่าหรือเท่ากัน เกิดขึ้นในรุ่นที่ 2 คือโมเดล 1000-100-30 ที่มีค่า Fitness เป็น 78.0 และโมเดล 900-200-3 และ 900-300-10 ที่มีค่า Fitness เป็น 76.0 แสดงด้วยตัวหนาเอียงสีฟ้า

เมื่อแสดงผลประชากรรุ่นที่ 3 ซึ่งเป็นรุ่นสุดท้ายของการประมวลผลตามงานวิจัยนี้ พบว่า มีการเก็บหัวกะทิจากประชากรรุ่นก่อนหน้าไว้ดังนี้ ประชากรรุ่นที่ 1 ยังคงเก็บหัวกะทิไว้ดังเดิม สำหรับประชากรรุ่นที่ 2 ได้แก่ Chromosome ที่มีค่า Fitness เป็น 78.0 ได้แก่โมเดล 1000-100-30 โมเดลที่มีค่า Fitness เป็น 76.0 ได้แก่โมเดล 900-200-3 และ 900-300-10 โมเดลที่มีค่า

Fitness เป็น 75.0 ได้แก่โมเดล 1000-300-10 และปรากฏ Chromosome ใหม่ในประชากรรุ่นที่ 3 ที่มีค่า Fitness สูงกว่าคือ Chromosome ที่มีค่า Fitness เป็น 82.5 ได้แก่โมเดล 700-200-20 โมเดลที่มีค่า Fitness เป็น 76.0 ได้แก่โมเดล 900-200-3 และ 900-300-10 โมเดลที่มีค่า Fitness เป็น 78.5 ได้แก่โมเดล 1000-300-20 โมเดลที่มีค่า Fitness เป็น 77.5 ได้แก่โมเดล 1000-500-30 แสดงด้วยตัวหนาสีม่วงขีดเส้นใต้ เมื่อสิ้นสุดการประมวลผลพบว่าโมเดลที่ดีที่สุดคือโมเดลที่ให้ค่า Fitness (Purity) ที่ 82.5 โดยมีโครงข่ายเป็น 700-200-20 (**) ดังแสดงในตารางที่ 4.3 ดังนี้

ตารางที่ 4.3 ประชากรรุ่นที่ 3 ในการค้นหา DANs ด้วยเทคนิค GAs สำหรับข้อมูล ECGs

#Model	Hidden1	Hidden2	Code Layer	Fitness Values
1	600	200	20	76
2	600	200	20	76
3	700	200	20	82.5**
4	900	100	30	71.5
5	900	200	3	76
6	900	200	20	76
7	900	200	20	76
8	900	200	20	76
9	900	300	10	76
10	900	500	3	76
11	1000	100	30	78
12	1000	100	30	78
13	1000	300	10	75
14	1000	300	10	75
15	1000	300	20	78.5
16	1000	500	3	75.5
17	1000	500	20	66.5
18	1000	500	30	77.5
19	1000	500	30	77.5
20	1000	500	30	77.5

4.1.2 โมเดลที่ดีที่สุดสำหรับข้อมูล EEGs

สำหรับการค้นหาโมเดลที่ดีที่สุดสำหรับ DANs ด้วยเทคนิค GAs ในข้อมูล EEGs พบว่าทุกโมเดลจะให้ค่า Fitness ไม่น่าพอใจและยังใกล้เคียงกันทุกโมเดล ดังนั้นในการศึกษาครั้งนี้จึงปรับปรุงการทำงาน โดยการเพิ่มขั้นตอนการปรับแต่งตัวแทนอนุกรมเวลา (TSR_{Adj}) โดยนำตัวแทนอนุกรมจากแต่ละโมเดล (TSR) คูณด้วยข้อมูลดั้งเดิม (Raw Data) นิยามได้ดังนี้

$$TSR_{Adj} = \text{Raw Data of EEGs} * \text{TSR of EEGs} \quad (4.1)$$

ดังนั้นค่า Fitness ที่ใช้ประเมินความเหมาะสมของข้อมูล EEGs จึงเป็นค่า Purity ที่ได้จากข้อมูล TSR_{Adj} ซึ่งผลลัพธ์ของโมเดลแต่ละรุ่น โดยเริ่มที่รุ่นที่ 1 แสดงดังตารางที่ 4.4 ดังนี้

ตารางที่ 4.4 ประชากรรุ่นที่ 1 ในการค้นหา DANs ด้วยเทคนิค GAs สำหรับข้อมูล EEGs

#Model	Hidden1	Hidden2	Code Layer	Fitness Values
1	500	100	8	66.0
2	**500**	**500**	**30**	**69.0**
3	600	400	20	64.0
4	600	500	20	55.0
5	700	100	10	54.0
6	**700**	**300**	**30**	**72.0**
7	700	500	1	63.0
8	800	400	20	65.0
9	800	400	20	65.0
10	800	500	10	52.0
11	900	200	30	51.0
12	**900**	**300**	**1**	**77.0**
13	900	300	30	52.0
14	900	400	10	68.0
15	900	400	20	56.0
16	1000	100	10	52.0
17	1000	200	30	68.0
18	1000	300	30	57.0
19	**1000**	**400**	**7**	**76.0**
20	1000	400	20	64.0

จากตารางที่ 4.4 แสดงผลประชากรรุ่นที่ 1 จากการค้นหาโมเดล DANs ด้วยเทคนิค GAs พบว่าโมเดลที่มีค่า Fitness สูงสุด 4 อันดับแรก มีดังนี้อันดับแรกมีค่า 77.0 ได้แก่โมเดล 900-300-1 อันดับสองมีค่า 76.0 ได้แก่โมเดล 1000-400-7 อันดับสามมีค่า 72.0 ได้แก่โมเดล 700-300-30 และอันดับสี่มีค่า 69.0 ได้แก่โมเดล 500-500-30 แสดงด้วยตัวหนาสีแดง

เมื่อพิจารณาประชากรรุ่นที่ 2 ในตารางที่ 4.5 พบว่ามีการเก็บหัวกะทิจากประชากรรุ่นที่ 1 ไว้ได้แก่โมเดล 900-300-1 โมเดล 1000-400-7 และโมเดล 700-300-30 ที่มีค่า Fitness เป็น 77.0, 76.0 และ 72.0 ตามลำดับ นอกจากนี้ยังพบว่ามีประชากรใหม่ที่ให้ค่า Fitness ดีกว่าเกิดขึ้นคือ โมเดล 500-400-20 มีค่า Fitness เป็น 84.0 โมเดล 600-400-10 มีค่า Fitness เป็น 82.0 และโมเดล 800-200-30 มีค่า Fitness เป็น 80.0 แสดงด้วยตัวหนาเอียงสีฟ้า ดังนี้

ตารางที่ 4.5 ประชากรรุ่นที่ 2 ในการค้นหา DANs ด้วยเทคนิค GAs สำหรับข้อมูล EEGs

#Model	Hidden1	Hidden2	Code Layer	Fitness Values
1	500	100	1	59.0
2	500	100	1	59.0
3	500	400	7	54.0
4	500	400	7	54.0
5	***500***	***400***	***20***	***84.0***
6	***600***	***400***	***10***	***82.0***
7	600	400	20	64.0
8	600	400	20	64.0
9	**700**	**300**	**30**	**72.0**
10	800	200	1	55.0
11	***800***	***200***	***30***	***80.0***
12	800	400	20	65.0
13	800	400	20	65.0
14	800	500	1	51.0
15	800	500	10	52.0
16	900	200	1	59.0
17	**900**	**300**	**1**	**77.0**
18	**1000**	**400**	**7**	**76.0**
19	1000	400	10	59.0
20	***1000***	***500***	***30***	***76.0***

ผลการดำเนินงานสำหรับประชากรรุ่นที่ 3 ซึ่งเป็นรุ่นสุดท้ายสำหรับงานวิจัยนี้ ดังตารางที่ 4.6 พบว่า ไม่พบประชากรเกิดใหม่ที่ให้ค่า Fitness สูงขึ้น และไม่พบหัวกะทิของประชากรรุ่นที่ 1 สืบทอดต่อในรุ่นที่ 3 ประชากรรุ่นที่ 3 ที่ให้ค่า Fitness สูงสุดยังคงเป็นโมเดลที่เกิดขึ้นในประชากรรุ่นที่ 2 คือโมเดลที่มีค่า Fitness เป็น 84.0 ได้แก่โมเดลที่ 500-400-20 โมเดลที่มีค่า Fitness เป็น 82.0 ได้แก่โมเดล 600-400-10 และโมเดลที่มีค่า Fitness เป็น 76.0 ได้แก่โมเดล 1000-500-30

ดังนั้นเมื่อสิ้นสุดการประมวลผลพบว่าโมเดลที่ดีที่สุดคือโมเดลที่ให้ค่า Fitness (Purity) ที่ 84.0 (**) โดยมีโครงข่ายเป็น 500-400-20 ดังแสดงในตารางที่ 4.6 ดังนี้

ตารางที่ 4.6 ประชากรรุ่นที่ 3 ในการค้นหา DANs ด้วยเทคนิค GAs สำหรับข้อมูล EEGs

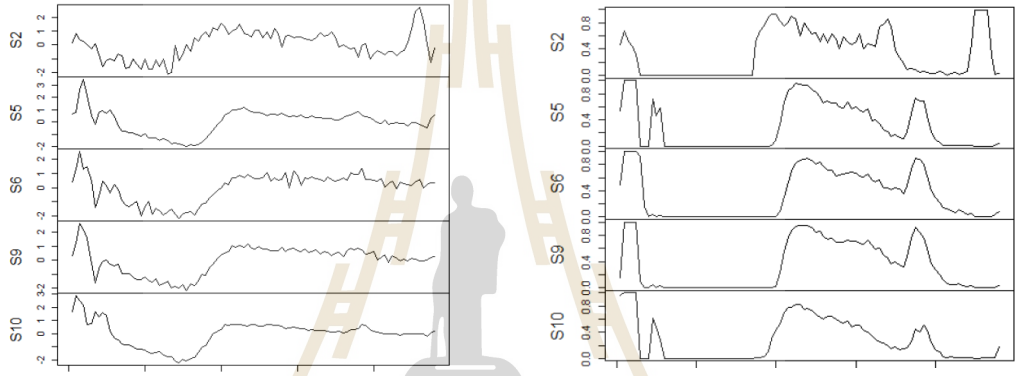
#Model	Hidden1	Hidden2	Code Layer	Fitness Values
1	500	400	1	56.0
2	***500***	***400***	***20***	***84.0****
3	***500***	***400***	***20***	***84.0***
4	***500***	***400***	***20***	***84.0***
5	***500***	***400***	***20***	***84.0***
6	500	400	30	54.0
7	600	200	30	73.0
8	***600***	***400***	***10***	***82.0***
9	***600***	***400***	***10***	***82.0***
10	600	400	20	64.0
11	800	400	10	74.0
12	800	400	20	65.0
13	800	400	20	65.0
14	800	400	20	65.0
15	800	400	30	70.0
16	900	300	30	52.0
17	1000	400	10	59.0
18	1000	400	20	64.0
19	1000	400	20	64.0
20	***1000***	***500***	***30***	***76.0***

4.2 ผลการแทนอนุกรมเวลาด้วย DANGA

การแทนอนุกรมเวลาโดยอาศัยเทคนิค DANGA ซึ่งเป็นโมเดลที่ดีที่สุดที่ได้จาก TSDL Algorithm (ผลงานวิจัยที่นำเสนอ) ซึ่งจะเป็นตัวแทนข้อมูลอนุกรมเวลาที่งานวิจัยนี้เรียกว่า *TSR-DANGA* ผลการแทนอนุกรมเวลาสำหรับข้อมูล ECGS และ EEGs อธิบายได้ดังนี้

4.2.1 TSR-DANGA สำหรับข้อมูล ECGs

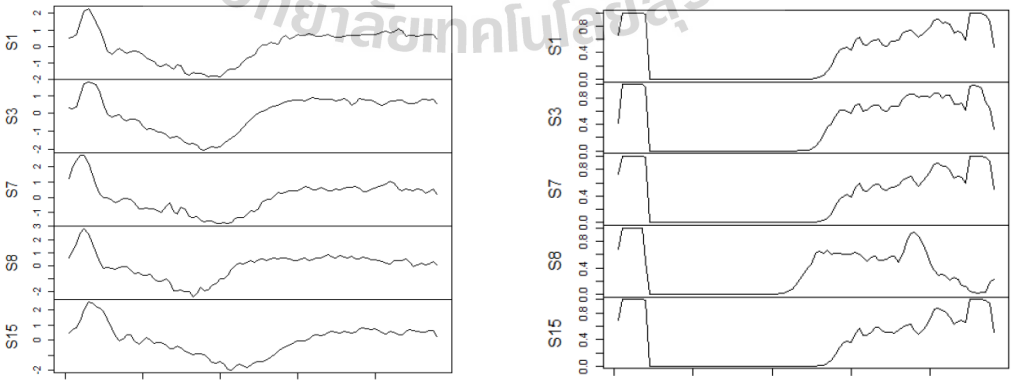
ข้อมูลคลื่นไฟฟ้าหัวใจ หรือข้อมูล ECGs เมื่อแทนอนุกรมเวลาด้วยเทคนิค DANGA จะได้ TSR-DANGA ซึ่งสามารถแสดงรูปร่างเป็นตัวอย่างบางส่วนเปรียบเทียบกับ Raw Data กรณีคลาสปกติ (Normal) ของข้อมูล ECGs ได้ดังรูปที่ 4.1 ดังนี้



(a) Raw Data (Normal) (b) TSR-DANGA (Normal)

รูปที่ 4.1 ตัวอย่าง TSR-DANGA สำหรับข้อมูล ECGs คลาส Normal

และสามารถแสดงรูปร่างของ TSR-DANGA เป็นตัวอย่างบางส่วนเปรียบเทียบกับ Raw Data กรณีคลาสไม่ปกติ (Abnormal) ของข้อมูล ECGs ได้ดังรูปที่ 4.2 ดังนี้

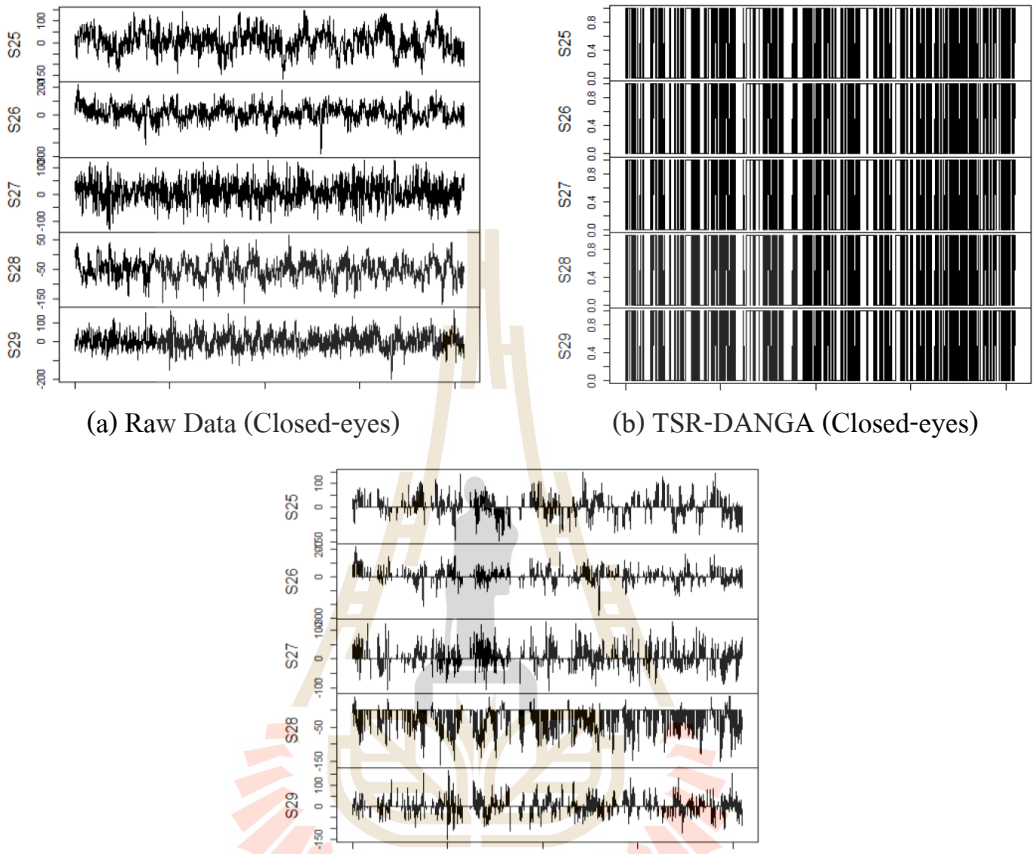


(a) Raw Data (Abnormal) (b) TSR-DANGA (Abnormal)

รูปที่ 4.2 ตัวอย่าง TSR-DANGA สำหรับข้อมูล ECGs คลาส Abnormal

4.2.2 TSR-DANGA สำหรับข้อมูล EEGs

คลื่นไฟฟ้าสมอง หรือ EEGs เมื่อหาตัวแทนด้วยเทคนิค DANGA จะได้ TSR-DANGA, TSR_{Adj}-DANGA ซึ่งแสดงตัวอย่างสำหรับ EEGs คลาส Closed-eyes ได้ดังรูปที่ 4.3



(a) Raw Data (Closed-eyes)

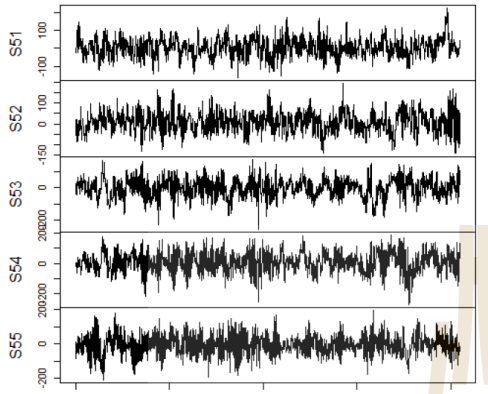
(b) TSR-DANGA (Closed-eyes)

(c) TSR_{Adj}-DANGA (Closed-eyes)

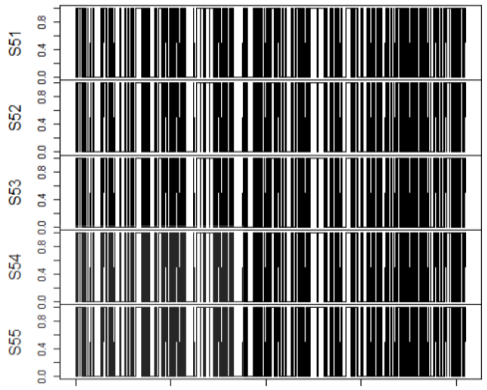
รูปที่ 4.3 ตัวอย่าง TSR-DANGA สำหรับข้อมูล EEGs คลาส Closed-eyes

จากรูปที่ 4.3 (a) แสดงรูปร่างของคลื่นไฟฟ้าหัวใจดั้งเดิมขณะตาปิด (Closed) พบว่าสัญญาณค่อนข้างหนาแน่น นอกจากนี้สัญญาณ EEGs ดั้งเดิมยังมีลักษณะเป็นรูปคลื่นที่มีความพลิ้วไหวไปตามธรรมชาติ รูปที่ 4.3 (b) แสดงรูปร่างของ TSR-DANGA ซึ่งเป็นตัวแทนข้อมูล EEGs ขณะตาปิด จะพบว่ารูปร่างไม่เป็นรูปคลื่นกลับมีลักษณะเป็นเหมือนสัญญาณดิจิทัลส่วนรูปที่ 4.3 (c) แสดงรูปร่างตัวแทนสำหรับ EEGs ขณะปิดตาที่ถูกปรับแต่ง TSR_{Adj}-DANGA พบว่ามีลักษณะที่ใกล้เคียงกับสัญญาณดั้งเดิมมากกว่า และเมื่อสังเกตจะพบว่าสัญญาณเหมือนถูกยึดไว้เป็นแนวเส้นตรงที่ค่า 0

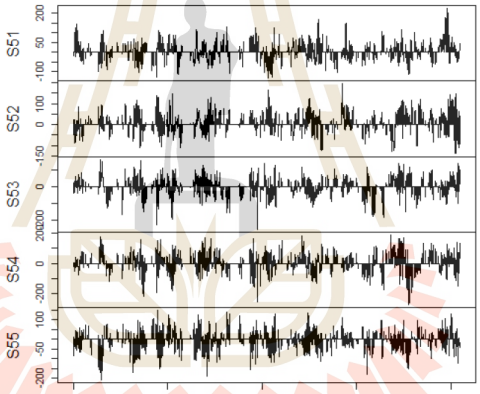
กรณีข้อมูล EEGs ขณะลืมตา (Open-eyes) สามารถแสดงรูปร่างของ Raw Data เปรียบเทียบกับ TSR-DANGA, TSR_{Adj}-DANGA เป็นตัวอย่างบางส่วนได้ดังรูปที่ 4.4 (a), (b) และ (c) ตามลำดับ ซึ่งลักษณะของรูปร่างสอดคล้องกับ ข้อมูล EEGs ขณะปิดตา แสดงดังนี้



(a) Raw Data (Open-eyes)



(b) TSR-DANGA (Open-eyes)



(c) TSR_{Adj}-DANGA (Open-eyes)

รูปที่ 4.4 ตัวอย่าง TSR-DANGA สำหรับข้อมูล EEGs คลาส Open-eyes

4.3 ผลการจัดกลุ่มข้อมูลอนุกรมเวลาและเปรียบเทียบประสิทธิภาพ

การนำเสนอนในส่วนนี้เป็นการนำเสนอผลการจัดกลุ่มข้อมูล TSR-DANGA สำหรับข้อมูล ECGs และ TSR_{Adj}-DANGA สำหรับข้อมูล EEGs ด้วยอัลกอริทึมการจัดกลุ่ม PDC และ k-Means และประเมินประสิทธิภาพของการจัดกลุ่มด้วย 5 มาตรวัด ได้แก่ Accuracy, Purity, Silhouette (พิจารณาค่า Silhouette เพื่อคัดเลือก Optimal k), Optimal k (ประเมินจำนวนกลุ่มจากค่า SSE) และ Processing Time โดยเปรียบเทียบประสิทธิภาพการจัดกลุ่มกับชุดข้อมูล Raw Data ชุดตัวแทนข้อมูลแบบ PAA และชุดตัวแทนข้อมูลแบบ SAX ได้ผลการจัดกลุ่มและเปรียบเทียบประสิทธิภาพได้ดังนี้

4.3.1 ผลการจัดกลุ่มข้อมูล ECGs

สำหรับตัวแทนของข้อมูล ECGs ที่ใช้จัดกลุ่มในขั้นตอนนี้คือ TSR-DANGA และเมื่อจัดกลุ่มด้วย อัลกอริทึม PDC และ k-Means สามารถแสดงผลลัพธ์ของการจัดกลุ่มและเปรียบเทียบประสิทธิภาพได้ดังต่อไปนี้

(1) **ผลประเมินประสิทธิภาพของการจัดกลุ่มข้อมูลด้วยอัลกอริทึม PDC** โดยใช้ 5 มาตรวัด เปรียบเทียบประสิทธิภาพกับ ชุดข้อมูล Raw Data ชุดตัวแทนข้อมูลแบบ PAA และชุดตัวแทนข้อมูลแบบ SAX แสดงผลได้ดังตารางที่ 4.7 ดังนี้

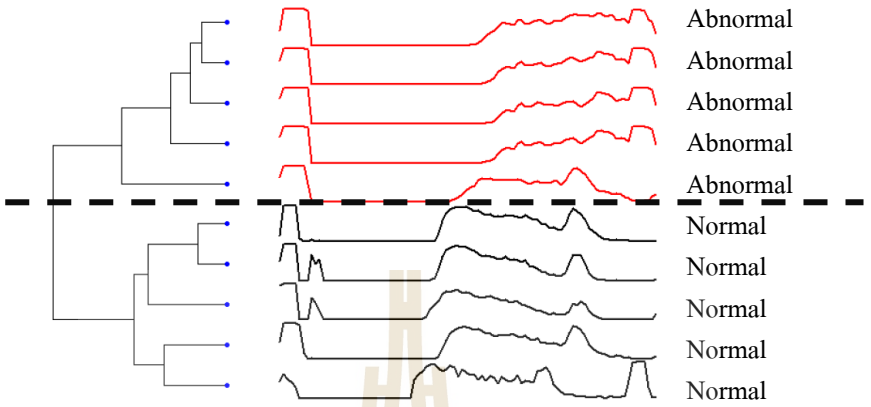
ตารางที่ 4.7 เปรียบเทียบผลการจัดกลุ่มของข้อมูล ECGs ด้วยอัลกอริทึม PDC

Dataset	Accuracy	Increased Accurary (%)	Purity	Increased Purity (%)	Silhouette (k=2)	Optimal k (SSE)	AVG Time (m)
Raw Data	62.0	-	67.0	-	0.23	2	***0.02***
PAA	67.0	8.1	74.0	10.4	***0.43***	2	***0.02***
SAX	72.3	16.6	74.0	10.4	0.16	2	0.06
TSR-DANGA	***80.6***	***30.0***	***82.5***	***23.1***	0.31	2	7.56

ผลการเปรียบเทียบการจัดกลุ่มสำหรับข้อมูล ECGs ด้วยอัลกอริทึม PDC จะพบว่าตัวแทนอนุกรมเวลา TSR-DANGA มีประสิทธิภาพจากมาตรวัด Accuracy มีค่า 80.6%, Purity มีค่า 82.5% ซึ่งประสิทธิภาพดีกว่า Raw Data โดยค่า Accuracy และ Purity เมื่อเปรียบเทียบกับ Raw Data ให้ค่าเพิ่มขึ้นถึง 30.0% และ 23.1% ตามลำดับ และยังดีกว่าเทคนิคอื่นอีกด้วย ในขณะที่ SSE และ Silhouette ให้ผลบ่งชี้ว่าการจัดกลุ่มมีความเหมาะสมตามกลุ่มจริง (k=2) ถึงแม้ว่า TSR-DANGA จะให้ค่า Accuracy และ Purity ที่ดีแต่สำหรับเวลาในการประมวลผลสูงกว่าเทคนิคอื่นมาก คือเฉลี่ยที่ประมาณเกือบ 8 นาที

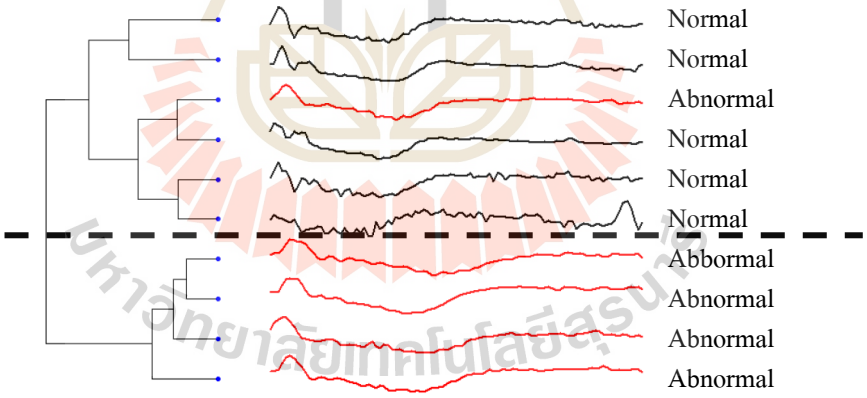
(2) **Dendrogram สำหรับข้อมูล ECGs** เป็นความสามารถในการแสดงผลสำหรับการจัดกลุ่มอนุกรมเวลาด้วยอัลกอริทึม PDC เมื่อพิจารณาภาพรวมการจัดกลุ่มสำหรับ TSR-DANGA ของข้อมูล ECGs พบว่า Dendrogram ที่แสดงมีข้อจำกัดในแสดงผลการจัดกลุ่มกับข้อมูลที่มีปริมาณมากทำให้แสดงผลได้ไม่ชัดเจน ดังนั้นเพื่อเป็นการแสดงถึงประสิทธิภาพในการจัดกลุ่มด้วยอัลกอริทึมงานวิจัยนี้ได้แสดงตัวอย่าง Raw Data และ TSR-DANGA เพียง 10 อนุกรมมา ซึ่งผลลัพธ์พบว่าข้อมูล TSR-DANGA ให้ประสิทธิภาพการจัดกลุ่มทั้งมาตรวัด Accuracy และ Purity มีค่า 100% ในขณะที่ผลการจัดกลุ่ม Raw Data ให้ค่า 89.9% และ 90.0% ตามลำดับ และ Dendrogram

ยังแสดงให้เห็นว่าข้อมูลที่ถูกจัดให้อยู่ในกลุ่มเดียวกัน (พิจารณาที่เส้นประแสดงการตัด Dendrogram) มีรูปร่างคล้ายคลึงกันมากกว่า ดังรูปที่ 4.5



รูปที่ 4.5 Dendrogram ของผลการจัดกลุ่มสำหรับข้อมูล TSR-DANGA (10 อนุกรม)

สำหรับ ข้อมูล Raw Data พบว่าจัดผิดกลุ่มไป 1 อนุกรม ทั้งที่รูปร่างของอนุกรมคล้ายคลึงกับสมาชิกภายในกลุ่ม ดังรูปที่ 4.6 (อนุกรมที่ 3 นับจากบนลงล่าง) ต่อไปนี้



รูปที่ 4.6 Dendrogram ของผลการจัดกลุ่มสำหรับข้อมูล Raw Data (10 อนุกรม)

(3) **ผลประเมินประสิทธิภาพของการจัดกลุ่มข้อมูลด้วยอัลกอริทึม k-Means** โดยใช้ 5 มาตรวัด (เวลาในการประมวลผลอ้างอิงตามอัลกอริทึม PDC) เปรียบเทียบประสิทธิภาพกับ ชุดข้อมูล Raw Data ชุดตัวแทนข้อมูลแบบ PAA และชุดตัวแทนข้อมูลแบบ SAX แสดงผลได้ดังตารางที่ 4.8

ตารางที่ 4.8 เปรียบเทียบผลการจัดกลุ่มของข้อมูล ECGs ด้วยอัลกอริทึม k-Means

Dataset	Accuracy	Increased Accurary (%)	Purity	Increased Purity (%)	Silhouette (k=2)	Optimal k (SSE)
Raw Data	69.8	-	74.5	-	0.36	2
PAA	69.0	-1.1	74.0	-0.7	0.43	2
SAX	69.1	-1.1	74.0	-0.7	0.43	2
TSR-DANGA	***72.6***	***4.0***	***76.0***	***2.0***	***0.51***	2

ผลการเปรียบเทียบการจัดกลุ่มสำหรับข้อมูล ECGs ด้วยอัลกอริทึม k-Means จะพบว่าตัวแทนอนุกรมเวลา TSR-DANGA มีประสิทธิภาพตามมาตรวัด Accuracy มีค่า 72.6%, Purity มีค่า 76.0% ซึ่งประสิทธิภาพดีกว่า Raw Data โดยค่า Accuracy และ Purity ให้ค่าเพิ่มขึ้น 4.0% และ 2.0% ตามลำดับ และยังดีกว่าเทคนิคอื่นอีกด้วย ในขณะที่ SSE และ Silhouette ให้ผลบ่งชี้ว่าการจัดกลุ่มมีความเหมาะสมตามกลุ่มจริง (k=2) และ TSR-DANGA ยังให้ค่า Silhouette เหมาะสมสูงที่สุดอีกด้วย

4.3.2 ผลการจัดกลุ่มข้อมูล EEGs

สำหรับตัวแทนของข้อมูล EEGs ที่ใช้จัดกลุ่มในขั้นตอนนี้คือ TSR_{Adj}-DANGA ของ EEGs และเมื่อจัดกลุ่มด้วย อัลกอริทึม PDC และ k-Means สามารถแสดงผลลัพธ์ของการจัดกลุ่มและเปรียบเทียบประสิทธิภาพได้ดังต่อไปนี้

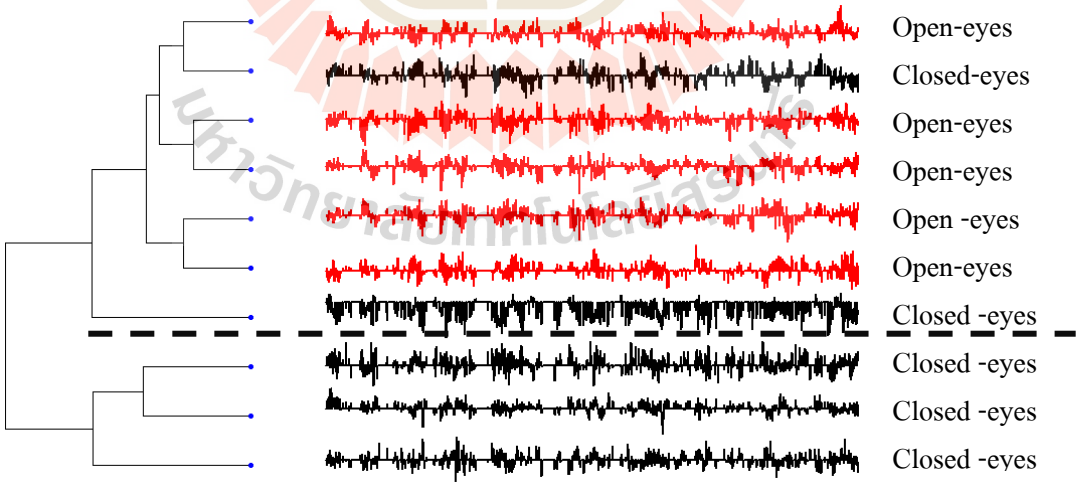
(1) **ผลประเมินประสิทธิภาพของการจัดกลุ่มข้อมูลด้วยอัลกอริทึม PDC** โดยใช้ 5 มาตรวัด เปรียบเทียบประสิทธิภาพกับ ชุดข้อมูล Raw Data ชุดตัวแทนข้อมูลแบบ PAA และชุดตัวแทนข้อมูลแบบ SAX แสดงผลได้ดังตารางที่ 4.9 ดังนี้

ตารางที่ 4.9 เปรียบเทียบผลการจัดกลุ่มของข้อมูล EEGs ด้วยอัลกอริทึม PDC

Dataset	Accuracy	Increased Accuracy (%)	Purity	Increased Purity (%)	Silhouette (k=2)	Optimal k (SSE)	AVG Time (m)
Raw Data	64.0	-	52.0	-	***0.42***	2	***0.01***
PAA	65.8	2.7	52.0	-	0.27	2	***0.01***
SAX	58.7	-8.3	55.0	5.8	0.12	2	***0.01***
TSR_{Adj}-DANGA	***83.9***	***31.1***	***84.0***	***61.5***	0.21	2	59.35

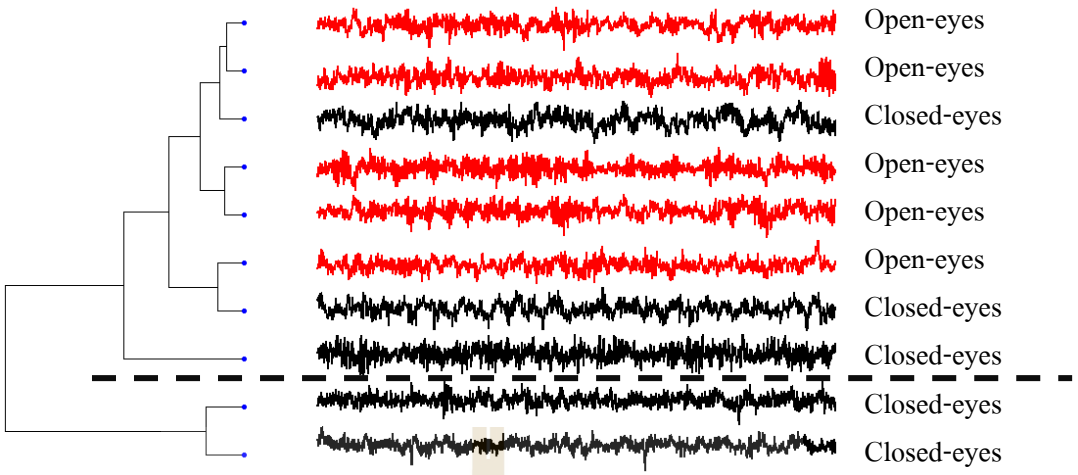
ผลการเปรียบเทียบการจัดกลุ่มสำหรับข้อมูล EEGs ด้วยอัลกอริทึม PDC จะพบว่าตัวแทนอนุกรมเวลา TSR_{Adj}-DANGA ให้ค่า Accuracy มีค่า 83.9%, Purity มีค่า 84.0% ซึ่งประสิทธิภาพดีกว่า Raw Data โดยให้ค่าเพิ่มขึ้นถึง 31.1% และ 61.5% ตามลำดับ และยังดีกว่าเทคนิคอื่นอีกด้วย ในขณะที่ SSE และ Silhouette ให้ผลบ่งชี้ว่าการจัดกลุ่มมีความเหมาะสมตามกลุ่มจริง (k=2) ถึงแม้ว่า TSR_{Adj}-DANGA จะให้ค่า Silhouette ดีเป็นอันดับสามก็ตาม สำหรับเวลาในการประมวลผลนั้น TSR_{Adj}-DANGA ใช้เวลาสูงกว่าเทคนิคอื่นมาก เฉลี่ย 59 นาที

(2) **Dendrogram สำหรับข้อมูล EEGs** ในส่วนนี้จะแสดงผลการจัดกลุ่มอนุกรมเวลาด้วยอัลกอริทึม PDC ด้วยแผนภาพ Dendrogram ซึ่งเป็นความสามารถสำหรับการแสดงผลลัพธ์ในรูปแผนผังต้นไม้เมื่อใช้เทคนิคการจัดกลุ่มแบบลำดับชั้นเมื่อพิจารณาที่ภาพรวมของการจัดกลุ่มด้วยอัลกอริทึม PDC สำหรับ TSR_{Adj}-DANGA ของข้อมูล EEGs พบว่า Dendrogram ที่แสดงมีข้อจำกัดในแสดงผลการจัดกลุ่มกับข้อมูลที่มีปริมาณมากทำให้แสดงผลได้ไม่ชัดเจน ดังนั้นเพื่อเป็นการแสดงถึงประสิทธิภาพในการจัดกลุ่มในแผนภาพ Dendrogram งานวิจัยนี้ได้เลือกแสดงข้อมูล Raw Data และ TSR_{Adj}-DANGA ที่ได้นำมาจัดกลุ่มด้วยอัลกอริทึม PDC เพียง 10 อนุกรม ซึ่งผลลัพธ์พบว่าผลการจัดกลุ่มข้อมูล TSR_{Adj}-DANGA ให้ค่า Accuracy และ Purity ให้ค่า 79.2% และ 80.0% ตามลำดับ ในขณะที่ผลการจัดกลุ่ม Raw Data ให้ค่า 67.0% และ 70.0% ตามลำดับ ซึ่ง Dendrogram ของผลการจัดกลุ่มข้อมูล TSR_{Adj}-DANGA แสดงให้เห็นชัดเจนว่าข้อมูลที่ถูกจัดให้อยู่ในกลุ่มเดียวกัน (พิจารณาที่เส้นแสดงการตัด Dendrogram) มีรูปร่างคล้ายคลึงกันมากกว่า ดังรูปที่ 4.7 ดังนี้



รูปที่ 4.7 Dendrogram ของผลการจัดกลุ่มสำหรับข้อมูล TSR_{Adj}-DANGA (10 อนุกรม)

สำหรับ ข้อมูล Raw Data พบว่าจัดผิดกลุ่มมากกว่า ดังรูปที่ 4.8 อนุกรมที่ 3, 7 และ 8 (นับจากบนลงล่าง) ดังต่อไปนี้



รูปที่ 4.8 Dendrogram ของผลการจัดกลุ่มสำหรับข้อมูล Raw Data (10 อนุกรม)

(3) **ผลประเมินประสิทธิภาพของการจัดกลุ่มข้อมูลด้วยอัลกอริทึม k-Means** โดยใช้ 5 มาตรวัด (เวลาในการประมวลผลอิงตามอัลกอริทึม PDC) เปรียบเทียบประสิทธิภาพกับข้อมูล Raw Data และตัวแทนจากเทคนิค PAA และ SAX แสดงผลได้ดังตารางที่ 4.10

ตารางที่ 4.10 เปรียบเทียบผลการจัดกลุ่มของข้อมูล EEGs ด้วยอัลกอริทึม k-Means

Dataset	Accuracy	Increased Accurary (%)	Purity	Increased Purity (%)	Silhouette (k=2)	Optimal k (SSE)
Raw Data	64.8	-	50.0	-	0.17	2
PAA	62.7	-3.2	***61.0***	***22.7***	0.14	2
SAX	63.5	-2.0	55.0	10.7	0.10	2
TSR_{Adj}-DANGA	***60.3***	-6.9	58.0	16.0	***0.20***	2

ผลการเปรียบเทียบการจัดกลุ่มสำหรับข้อมูล EEGs ด้วยอัลกอริทึม k-Means พบว่าตัวแทนอนุกรมเวลา TSR_{Adj}-DANGA มีประสิทธิภาพวัดจากค่า Accuracy มีค่า 60.3% ซึ่งให้ค่าต่ำกว่า Raw Data ถึง -6.9% ส่วนค่า Purity มีค่า 58.0% ซึ่งประสิทธิภาพดีกว่า Raw Data โดยให้ค่าเพิ่มขึ้นถึง 16.0% จัดว่าดีเป็นอันดับสองรองจาก PAA แต่สำหรับการบ่งชี้ว่าการจัดกลุ่มมีความเหมาะสมตามกลุ่มจริง (k=2) ด้วยมาตรวัด SSE และ Silhouette ให้ผลบ่งชี้ว่า TSR_{Adj}-DANGA จัดให้เหมาะสม และให้ค่า Silhouette เหมาะสมสูงที่สุดอีกด้วย

4.4 การอภิปรายผล

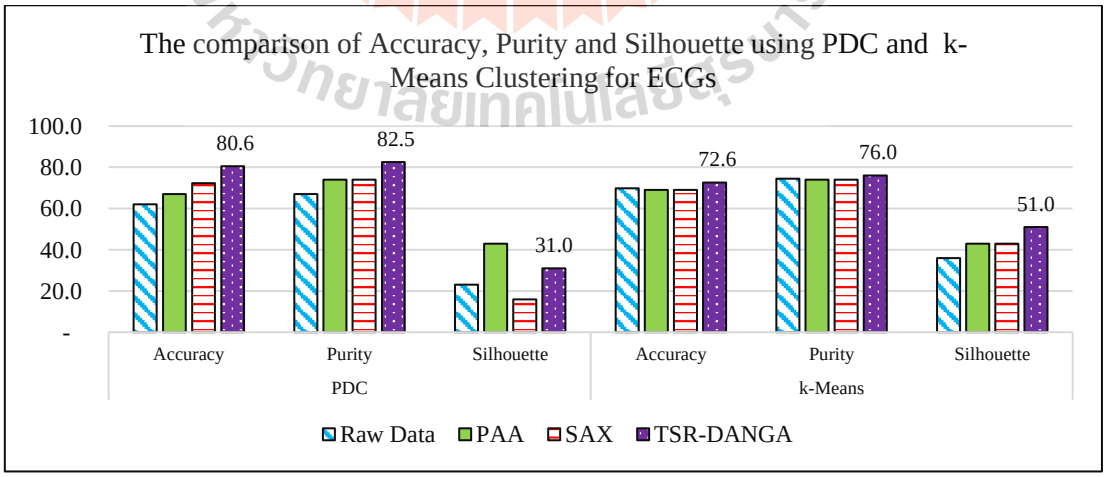
ผลลัพธ์สำหรับการวิจัยนี้คือตัวแทนอนุกรมเวลา TSR-DANGA สำหรับ ECGs และ TSR_{Adj}-DANGA สำหรับ EEGs ซึ่งได้จากโมเดลที่ดีที่สุดของ DANs ที่ค้นหาด้วยเทคนิค GAs (DANGA) ซึ่งผลลัพธ์จากขั้นตอนต่าง ๆ ในงานวิจัยสามารถอภิปรายสรุปเป็นประเด็นได้ดังนี้

(1) โมเดลที่ดีที่สุดของ DANGA ผลการวิจัยพบว่าโมเดลที่ดีที่สุดสำหรับข้อมูล ECGs คือโมเดลที่มีโครงข่าย Encoder = 700-200-20 และโมเดลที่ดีที่สุดสำหรับข้อมูล EEGs คือโมเดลที่มีโครงข่าย Encoder = 500-400-20 ซึ่งจากผลลัพธ์ดังกล่าวมีจุดที่น่าสนใจคือทั้งข้อมูล ECGs และ EEGs ได้โมเดลที่ดีที่สุดที่มีจำนวน Unit ในชั้น Code Layer = 20 Units เท่ากัน

(2) ตัวแทนอนุกรมที่ได้จาก DANGA ผลการวิจัยพบว่า ตัวแทนอนุกรมสำหรับข้อมูล ECGs มีรูปร่างที่ชัดเจนแตกต่างกันทั้งสองคลาสและคงลักษณะเด่นของข้อมูลดั้งเดิม สำหรับข้อมูล EEGs ตัวแทนอนุกรมเวลาที่ได้มีความคล้ายคลึงกับรูปร่างของข้อมูลเดิมเช่นกัน แต่ลดความหนาแน่นของสัญญาณรบกวนลงมีแนวโน้มจะช่วยให้พบรูปแบบของสัญญาณได้ง่ายขึ้น

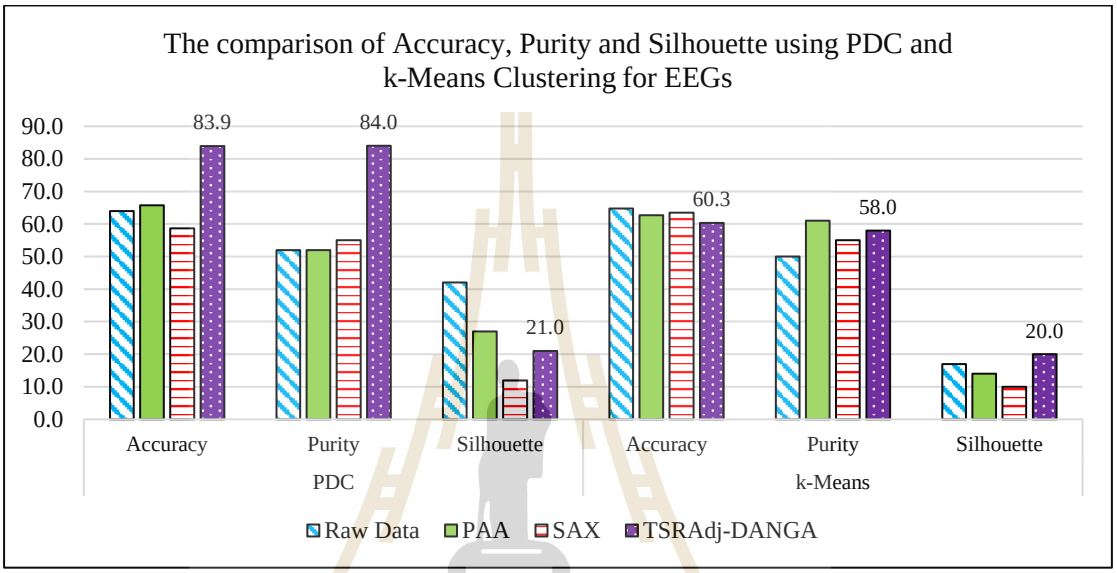
(3) ผลการจัดกลุ่มข้อมูลอนุกรมเวลาและเปรียบเทียบประสิทธิภาพ จากผลการจัดกลุ่มสำหรับข้อมูล ECGs และ EEGs อภิปรายผลการวิจัยแยกเป็นประเด็นได้ดังนี้

(3.1) พิจารณาที่ข้อมูล ECGs ทั้งการจัดกลุ่มด้วย PDC และ k-Means เมื่อให้ความสนใจที่การจัดกลุ่มสำหรับข้อมูล ECGs เป็นหลัก โดยพิจารณาที่มาตรวัด Accuracy, Purity และ Silhouette จะพบว่าเกือบทุกมาตรวัดทั้งการจัดกลุ่มด้วย PDC และ k-Means ตัวแทนอนุกรมที่ได้จากงานวิจัยนี้ ให้ประสิทธิภาพดีที่สุดยกเว้นค่า Silhouette จากการจัดกลุ่มด้วย PDC มีประสิทธิภาพเป็นอันดับสอง แสดงดังรูปที่ 4.9

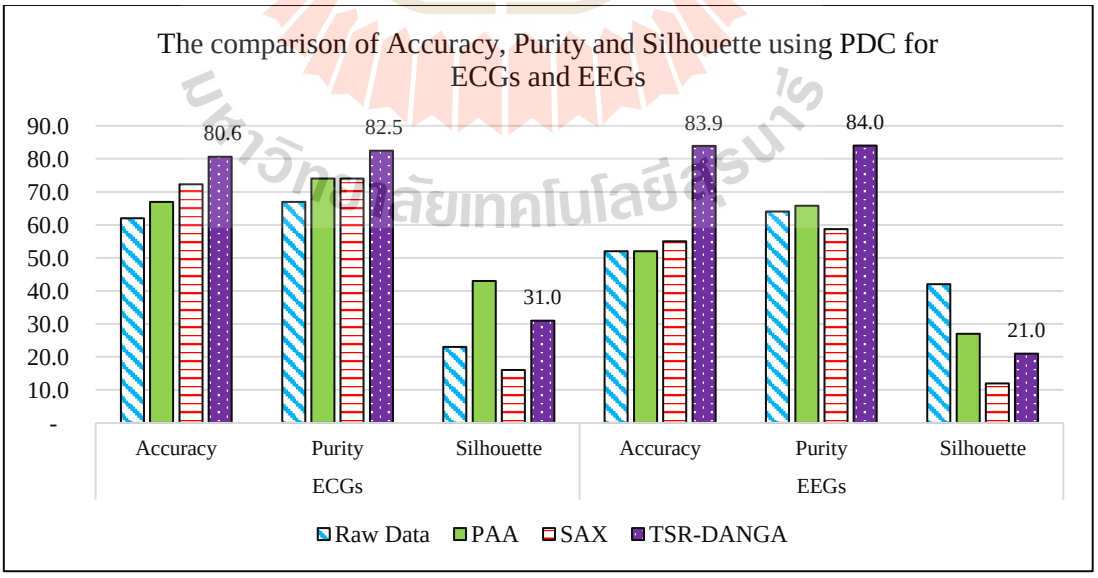


รูปที่ 4.9 เปรียบเทียบประสิทธิภาพผลการจัดกลุ่ม สำหรับข้อมูล ECGs

(3.2) พิจารณาที่ข้อมูล EEGs ทั้งการจัดกลุ่มด้วย PDC และ k-Means โดยพิจารณาที่มาตรวัด Accuracy, Purity และ Silhouette จะพบว่าที่การจัดกลุ่มด้วย PDC ประสิทธิภาพดีกว่าเทคนิคอื่นอย่างมาก แต่สำหรับอัลกอริทึม k-Means ให้ค่า Accuracy ต่ำกว่าเทคนิคอื่นเพียงเล็กน้อย แต่พบว่าเมื่อพิจารณาที่ค่าความเหมาะที่สอดคล้องกับธรรมชาติของข้อมูล เทคนิคนี้ให้ความเหมาะสมสูงกว่าเทคนิคอื่นดังแสดงในรูปที่ 4.10



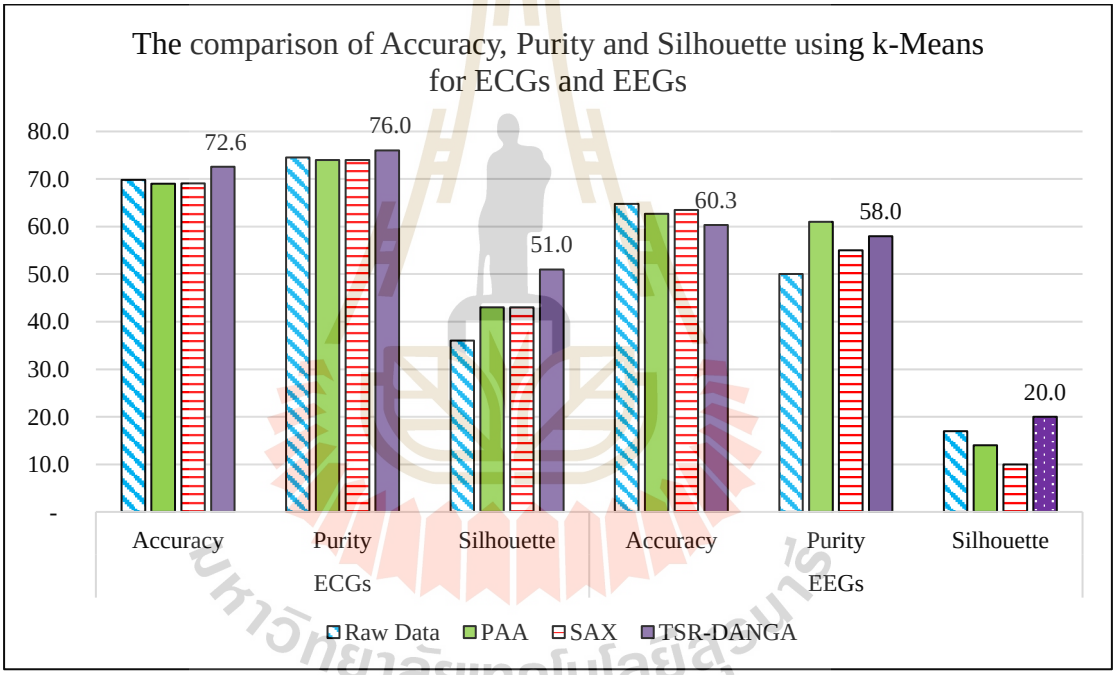
รูปที่ 4.10 เปรียบเทียบประสิทธิภาพผลการจัดกลุ่ม สำหรับข้อมูล EEGs



รูปที่ 4.11 เปรียบเทียบประสิทธิภาพผลการจัดกลุ่มด้วยอัลกอริทึม PDC

(3.3) พิจารณาอัลกอริทึม PDC พบว่าตัวแทนอนุกรมเวลาที่ได้จาก DANGA ทั้งข้อมูล ECGs และ EEGs มีประสิทธิภาพดีกว่า Raw Data และตัวแทนอนุกรมเวลาที่ได้จากเทคนิคอื่นอย่างมาก ถึงแม้ว่า Silhouette ไม่ใช่ค่าที่สูงที่สุดแต่ทั้งสองก็แสดงให้เห็นว่าสามารถจัดกลุ่มได้เหมาะสมกับธรรมชาติข้อมูล ดังแสดงในรูปที่ 4.11

(3.4) เมื่อพิจารณาอัลกอริทึม k-Means ทั้งข้อมูล ECGs และ EEGs จะพบว่า ตัวแทนอนุกรมเวลาที่ได้จาก DANGA ของข้อมูล ECGs จะให้ประสิทธิภาพดีกว่าทุกมาตรวัด ในขณะที่ของข้อมูล EEGs พบค่า Purity ให้ประสิทธิภาพดีเป็นอันดับสองรองจากเทคนิค PAA ซึ่งภาพรวมของประสิทธิภาพของการจัดกลุ่มให้ตรงกับกลุ่มจริงทำได้ดีกว่าเทคนิคอื่นเพียงเล็กน้อยเท่านั้น แต่สำหรับค่า Accuracy นั้นประสิทธิภาพด้อยกว่า Raw Data และเทคนิคอื่น ดังแสดงในรูปที่ 4.12



รูปที่ 4.12 เปรียบเทียบประสิทธิภาพผลการจัดกลุ่มด้วยอัลกอริทึม k-Means

บทที่ 5
บทสรุป

งานวิจัยนี้พัฒนาขึ้นโดยมีวัตถุประสงค์ในการนำเสนออัลกอริทึมการหาตัวแทนอนุกรมเวลาด้วยเทคนิคการเรียนรู้เชิงลึก (TSDL Algorithm) โดยในการเรียนรู้เชิงลึกได้ใช้วิธีค้นหาโมเดลที่มีโครงสร้างโครงข่ายเหมาะสมสำหรับ DANs ด้วยGAs การแทนอนุกรมเวลามีจุดมุ่งหมายเพื่อปรับปรุงประสิทธิภาพการจัดกลุ่มข้อมูลอนุกรมเวลาให้มีประสิทธิภาพมากยิ่งขึ้นกว่าการจัดกลุ่มข้อมูลดั้งเดิม (Raw Data) ที่ไม่มีการเลือกตัวแทน จากผลการดำเนินงานสรุปผลการวิจัยได้ดังนี้

5.1 สรุปผลการวิจัย

จากผลการศึกษาและการวิเคราะห์การดำเนินงานวิจัยเพื่อการจัดกลุ่มที่มีประสิทธิภาพสามารถสรุปผลการวิจัยอธิบายเป็น 3 ส่วนได้ดังนี้

5.1.1 สรุปผลการพัฒนา TSDL Algorithm

จากผลการหาตัวแทนอนุกรมเวลาตามขั้นตอนการทำงานของ TSDL Algorithm สามารถสรุปผลการวิจัยได้ดังต่อไปนี้

(1) งานวิจัยนี้สามารถพัฒนาอัลกอริทึมสำหรับหาตัวแทนอนุกรมเวลาด้วยเทคนิคการเรียนรู้เชิงลึก (TSDL Algorithm) ที่ผสานการทำงานร่วมกับเทคนิค GAs เพื่อค้นหาโมเดลที่ดีที่สุดที่สามารถผลิต TSR-DANGA ที่ช่วยให้การจัดกลุ่มข้อมูลอนุกรมเวลาทั้งการจัดกลุ่มแบบลำดับชั้นด้วยอัลกอริทึม PDC และการจัดกลุ่มแบบแบ่งแยกด้วยอัลกอริทึม k-Means มีประสิทธิภาพมากยิ่งขึ้นได้

(2) สรุปผลการพัฒนา TSDL Algorithm ที่นำเสนอในงานวิจัยนี้ มีองค์ประกอบการทำงานที่สำคัญ 4 ส่วน ได้แก่

(2.1) ส่วนนำเข้าข้อมูลและกำหนดค่าเริ่มต้นการทำงาน

(2.2) ส่วนงานการประเมินความเหมาะสม ประกอบด้วยส่วนงาน การแทนอนุกรมเวลาด้วยเทคนิค DANs ส่วนงานการจัดกลุ่มตัวแทนอนุกรมเวลาที่ได้จากเทคนิค DANs ด้วยอัลกอริทึม PDC กำหนดค่า k=2 และส่วนงานการคำนวณ Fitness Value

(2.3) ส่วนงานการดำเนินงานตามขั้นตอนทางพันธุกรรม

(2.4) ส่วนงานการแทนอนุกรมเวลาด้วย DANGA

5.1.2 สรุปผลการแทนอนุกรมเวลาด้วย DANGA

จากผลการวิจัยสรุปได้ว่าการแทนอนุกรมเวลาด้วย DANGA สามารถใช้เป็นเครื่องมือที่ผลิตตัวแทนอนุกรมเวลาที่ช่วยให้การจัดกลุ่มข้อมูลมีประสิทธิภาพดีขึ้นได้ทั้งในข้อมูล ECGs และ ข้อมูล EEGs สรุปผลการวิจัยได้ดังนี้

(1) เทคนิค DANGA เป็นเทคนิคที่สามารถผลิตตัวแทนอนุกรมเวลา (TRS-DANGA) ที่ช่วยเพิ่มประสิทธิภาพการจัดกลุ่มให้ดีขึ้นได้ทั้งในอัลกอริทึม PDC และ k-Means อีกทั้งยังเป็นตัวแทนอนุกรมเวลาที่ให้ประสิทธิภาพดีกว่าข้อมูล Raw Data และดีกว่าการใช้ตัวแทนอนุกรมเวลาแบบอื่น (แบบ PAA และ SAX) อีกด้วย โดยโมเดลที่ดีที่สุดที่งานวิจัยนี้ค้นพบ ได้แก่

(1.1) โมเดลที่ดีที่สุดสำหรับข้อมูล ECGs คือโมเดลที่มีโครงข่ายของ DANGA ซึ่งประกอบด้วยทั้ง Encoder-Decoder = 700-200-20-200-700

(1.2) โมเดลที่ดีที่สุดสำหรับข้อมูล EEGs คือโมเดลที่มีโครงข่ายของ DANGA ประกอบด้วย Encoder-Decoder = 500-400-20-400-500

(1.3) จุดที่น่าสนใจคือทั้งข้อมูล ECGs และ EEGs ได้โมเดลที่ดีที่สุดที่มีจำนวน Unit ในชั้น Code Layer = 20 Units เท่ากัน

(2) ในการจัดกลุ่มสำหรับข้อมูล ECGs

(2.1) ผลการจัดกลุ่มสำหรับข้อมูล ECGs ด้วยอัลกอริทึม PDC จะพบว่าตัวแทนอนุกรมเวลา TSR-DANGA มีประสิทธิภาพจากมาตรวัด Accuracy มีค่า 80.6%, Purity มีค่า 82.5% ซึ่งประสิทธิภาพดีกว่า Raw Data โดยค่า Accuracy และ Purity ให้ค่าเพิ่มขึ้นถึง 30.0% และ 23.1% ตามลำดับ และยังดีกว่าเทคนิคอื่นอีกด้วย ในขณะที่ SSE และ Silhouette ให้ผลบ่งชี้ว่าการจัดกลุ่มมีความเหมาะสมตามกลุ่มจริง (k=2) ทั้งนี้ TSR-DANGA ให้ค่า Silhouette ดีเป็นอันดับสอง ในขณะที่ตัวแทนอนุกรมเวลาแบบ PAA ให้ค่าดีเป็นอันดับหนึ่ง สำหรับเวลาในการประมวลผลเทคนิค TSR-DANGA ใช้เวลาสูงกว่าเทคนิคอื่นคือเฉลี่ยที่ 5 นาที

(2.2) ผลการจัดกลุ่มสำหรับข้อมูล ECGs ด้วยอัลกอริทึม k-Means จะพบว่าตัวแทนอนุกรมเวลา TSR-DANGA มีประสิทธิภาพจากมาตรวัด Accuracy มีค่า 72.6%, Purity มีค่า 76.0% ซึ่งประสิทธิภาพดีกว่า Raw Data โดยค่า Accuracy และ Purity ให้ค่าเพิ่มขึ้น 4.0% และ 2.0% ตามลำดับ และยังดีกว่าเทคนิคอื่นอีกด้วย ในขณะที่ SSE และ Silhouette ให้ผลบ่งชี้ว่าการจัดกลุ่มมีความเหมาะสมตามกลุ่มจริง (k=2) และ TSR-DANGA ยังให้ค่า Silhouette เหมาะสมสูงที่สุดอีกด้วย

(3) ในการจัดกลุ่มสำหรับข้อมูล EEGs

(3.1) ผลการจัดกลุ่มสำหรับข้อมูล EEGs ด้วยอัลกอริทึม PDC จะพบว่าตัวแทนอนุกรมเวลา TSR_{Adj}-DANGA มีประสิทธิภาพจากมาตรวัด Accuracy มีค่า 83.9%, Purity มีค่า 84.0% ซึ่งประสิทธิภาพดีกว่า Raw Data โดยค่า Accuracy และ Purity ให้ค่าเพิ่มขึ้นถึง 31.1% และ 61.5% ตามลำดับ และยังดีกว่าเทคนิคอื่นอีกด้วย ในขณะที่ SSE และ Silhouette ให้ผลบ่งชี้ว่าการจัดกลุ่มมีความเหมาะสมตามกลุ่มจริง (k=2) ถึงแม้ว่า TSR_{Adj}-DANGA จะให้ค่า Silhouette ดีเป็นอันดับสาม สำหรับเวลาในการประมวลผลเมื่อเปรียบเทียบกับ Raw Data และการแทนอนุกรมเวลาแบบอื่น สรุปได้ว่า TSR_{Adj}-DANGA ให้ผลด้อยกว่า เนื่องจากใช้เวลาการประมวลผลสูงกว่าเทคนิคอื่นเฉลี่ยที่ 59 นาที

(3.2) ผลการเปรียบเทียบการจัดกลุ่มสำหรับข้อมูล EEGs ด้วยอัลกอริทึม k-Means จะพบว่าตัวแทนอนุกรมเวลา TSR_{Adj}-DANGA มีประสิทธิภาพจากมาตรวัด Accuracy มีค่า 60.3% ซึ่งให้ค่าต่ำกว่า Raw Data ถึง -6.9% และต่ำกว่าเทคนิคอื่น ส่วนค่า Purity มีค่า 58.0% ซึ่งประสิทธิภาพดีกว่า Raw Data โดยให้ค่าเพิ่มขึ้นถึง 16.0% จัดว่าดีเป็นอันดับสองรองจาก PAA แต่สำหรับการบ่งชี้ว่าการจัดกลุ่มมีความเหมาะสมตามกลุ่มจริง (k=2) ด้วยมาตรวัด SSE และ Silhouette ให้ผลบ่งชี้ว่า TSR_{Adj}-DANGA จัดให้เหมาะสม และให้ค่า Silhouette เหมาะสมสูงที่สุดอีกด้วย

5.2 สรุปผลงานของการวิจัย

จากผลการศึกษาและการดำเนินงานวิจัยจนสำเร็จลุล่วง ผลงานของการวิจัยที่ได้พัฒนาและค้นพบในงานวิจัยนี้สามารถสรุปได้ดังต่อไปนี้

(1) เทคนิคสำหรับการหาตัวแทนอนุกรมเวลาที่สามารถเพิ่มประสิทธิภาพในการจัดกลุ่มข้อมูลอนุกรมเวลาได้ดีขึ้นทั้งการจัดกลุ่มแบบลำดับชั้นด้วยอัลกอริทึม PDC และการจัดกลุ่มแบบแบ่งแยกด้วยอัลกอริทึม k-Means

(2) ค้นพบโมเดลของโครงข่ายการเรียนรู้เชิงลึกที่สามารถผลิตตัวแทนอนุกรมเวลาที่เหมาะสมสามารถเพิ่มประสิทธิภาพในการจัดกลุ่มข้อมูลอนุกรมเวลาให้ดีขึ้น สำหรับชุดข้อมูล ECGs และชุดข้อมูล EEGs ได้ทั้งการจัดกลุ่มด้วยอัลกอริทึม PDC และ k-Means

(3) ค้นพบการใช้เทคนิค GAs สามารถช่วยแก้ปัญหาในการค้นหาโมเดลที่มีสถาปัตยกรรมโครงข่ายดีที่สุดของโครงข่ายการเรียนรู้เชิงลึกเพื่อใช้แทนข้อมูลอนุกรมเวลาได้ทั้งข้อมูล ECGs และชุดข้อมูล EEGs

5.3 ข้อเสนอแนะ

จากผลการดำเนินงานวิจัยนี้ได้พบข้อจำกัดในการดำเนินงาน การนำไปใช้ และประเด็นการวิจัยที่น่าสนใจบางประการ จึงสรุปเป็นข้อเสนอแนะได้ดังต่อไปนี้

5.3.1 ข้อแนะนำสำหรับการใช้ผลงานวิจัย

งานวิจัยนี้มีข้อแนะนำสำหรับการนำผลงานวิจัยไปใช้ดังนี้

(1) งานวิจัยนี้เป็นการศึกษาและพัฒนาวิธีการเพื่อหาตัวแทนอนุกรมเวลาที่ช่วยให้การจัดกลุ่มข้อมูลอนุกรมเวลาประเภท Whole Time Series มีประสิทธิภาพเพิ่มขึ้น ดังนั้นจึงเหมาะกับการนำไปใช้หาตัวแทนข้อมูลสำหรับข้อมูลที่มีขนาดใหญ่ มีลำดับต่อเนื่อง ให้สามารถทำงานได้อย่างมีประสิทธิภาพ และรวดเร็ว

(2) วิธีการที่นำเสนอ หากมีการกำหนดโครงข่ายการเรียนรู้ที่มีจำนวนชั้น Hidden มาก ควรตรวจสอบทรัพยากรที่จะนำมาใช้ในการประมวลผลให้เหมาะสมทั้งความจุของหน่วยความจำหลักและความเร็วของหน่วยประมวลผลกลาง

(3) จำนวนชั้น Hidden สำหรับโครงข่ายการเรียนรู้ ซึ่งงานวิจัยนี้กำหนดไว้ที่ 3 ชั้น หากต้องการใช้งานกับโครงข่ายที่มีจำนวนชั้นแตกต่างออกไปควรออกแบบพารามิเตอร์เพิ่ม

5.3.2 ข้อแนะนำการต่อยอดงานวิจัย

เพื่อให้ประสิทธิภาพของตัวแทนอนุกรมเวลามีประสิทธิภาพดียิ่งขึ้นจากการศึกษางานวิจัยนี้พบว่ามีประเด็นที่น่าสนใจในการศึกษาวิจัยเพิ่มเติม ดังนี้

(1) พัฒนาเทคนิคที่นำเสนอนี้เพื่อใช้สำหรับงาน การจัดกลุ่มข้อมูลอนุกรมเวลาประเภท Subsequence Time Series Clustering เพื่อประโยชน์ในการประเมินผลเพื่อตรวจสอบความเหมาะสมของการจัดกลุ่มด้วยวิธีแบบอาศัยรูปร่างเป็นฐาน

(2) ศึกษาและวิจัยเทคนิคการนำเสนอสำหรับชุดข้อมูลอื่นเพิ่มเติมเพื่อให้ได้เทคนิควิธีที่มีประสิทธิภาพและมีเสถียรภาพมากยิ่งขึ้น

(3) ศึกษาและทดสอบตัวแทนอนุกรมเวลาด้วยเทคนิคการจัดกลุ่มแบบอื่นเพิ่มเติม เพื่อศึกษาความเหมาะสมของธรรมชาติตัวแทนข้อมูลกับวิธีการจัดกลุ่มที่เลือกใช้

(4) ศึกษาและพัฒนาเทคนิคการวัดความคล้ายคลึง/ความแตกต่าง ที่อาศัยรูปร่างของอนุกรมเป็นฐาน เพื่อปรับปรุงประสิทธิภาพของการจัดกลุ่มเมื่อใช้กับตัวแทนอนุกรมเวลาที่ผลิตจากเทคนิค DANGA

รายการอ้างอิง

กฤษณะ ไวยมัย, ชิดชนก ส่งศิริ, ธนาวินท์ รักธรรมานนท์. (2544). "**เทคนิคการทำเหมืองข้อมูล**".

บุญเสริม กิจศิริกุล. (2548). "**ปัญญาประดิษฐ์**". เอกสารคำสอนวิชา 2110654, ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย.

มณฑิรา วิทยากิตติพงษ์. (2549). การตรวจคลื่นไฟฟ้าสมองในผู้ใหญ่. **สงขลานครินทร์เวชสาร**. ปีที่ 24 ฉบับที่ 5, หน้า 445-452.

สำนักนโยบายการออมและการลงทุน สำนักงานเศรษฐกิจการคลัง กระทรวงการคลัง. (2548). เศรษฐศาสตร์น่ารู้. ค้นเมื่อ 2 สิงหาคม 2559, จาก http://www.fpo.go.th/S-I/Source/ECO/eCO24.htm.

Aghabozorgi, S., Shirkorshidi, S. A. and Wan, Y. T. (2015). Time-series clustering-A decade review. **Information Systems**. Vol. 53, pp.16-38.

Agrawal, R., Faloutsos, C. and Swami, A. (1993). Efficient similarity search in sequence databases, **Found. Data Organ. Algorithms**. Vol. 46, pp. 69–84.

Andrzejak, R. (2014). **Data Files**. [Online].http://ntsa.upf.edu/downloads/andrzejak-rg-et-al-2001-indications-nonlinear-deterministic-and-finite-dimensional.

Andrzejak, R. G., Lehnertz, K., Mormann, F., Rieke, C., David, P., and Elger, C. E. (2001). Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state, **Physical Review E**, Vol. 64, no. 6.

Bandt, C., and Pompe, B. (2002). Permutation Entropy: A Natural Complexity Measure for Time Series. **Physical Review Letters**, Vol. 88, no. 17, pp. 174102–1–174102–4. doi:10.1103/physrevlett.

Bagnall, A.A.J., Ratanamahatana, C. "Ann", Keogh, E., Lonardi, S. and Janacek, G. (2006). A bit level representation for time series data mining with shape based similarity, **Data Min. Knowl. Discov**. Vol. 13, no. 1, pp. 11–40.

Banerjee, A. and Ghosh, J. (2001). Clickstream clustering using weighted longest common subsequences, in: **Proceedings of the Workshop on Web Mining, SIAM Conference on Data Mining**, pp. 33-40.

Bengio, Y. (2009). Learning deep architectures for AI. **Foundations and trends® in Machine Learning**, Vol. 2(1), pp. 1-127.

Bengio, Y., Courville, A. and Vincent, P. (2013). Representation Learning: A Review and New Perspectives. **IEEE Transactions on Pattern Analysis and Machine Intelligenc**e. Vol. 35, no. 8, pp. 1798–1828.

Bengio, Y., LeCun, Y. and Hinton, G. (2015). Deep Learning. **Nature**. Vol. 521, pp. 436–444.

Bingham, E. (2001). Random projection in dimensionality reduction: applications to image and text data, in: **Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**, pp. 245–250.

Brandmaier, M. A. (2015). pdc: An R Package for Complexity-Based Clustering of Time Series. **Journal of Statistical Software**. October 2015, Vol. 67, Issue 5, pp. 1-23.

Brockwell, P. J. and Davis, R. A. (2016). Introduction to time series and forecasting. **Springer**.

Burrus, C. S., Gopinath, R. A. and Guo, H. (1997). Introduction to wavelets and wavelet transforms: A Primer. **Prentice Hall**.

Busseti, E., Osband, I. and Wong, S. (2012). Deep learning for time series modeling. **Technical report, Stanford University**.

Cai, Y. and Ng, R. (2004). Indexing spatio-temporal trajectories with Chebyshev polynomials, in: **Procedings of 2004 ACM SIGMOD International**, pp. 599.

Chan, K., Fu, A. W. (1999). Efficient time series matching by wavelets. In: **Proceedings of the 15th IEEE International conference on data engineering**, Sydney, Australia, March 23–26, pp 126–133.

Chen, Q., Chen, L., Lian, X. and Liu,Y. (2007). Indexable PLA for efficient similarity search, in: **Proceedings of the 33rd International Conference on Very large Data Bases**, pp. 435–446.

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

Chuentawat, R., Kerdprasop, N. and Kerdprasop, K. (2017). The Forecast of PM10 Pollutant by Using a Hybrid Model. **International Journal of Future Computer and Communication**, Vol. 6(3), pp. 128.

Ciresan, D., Meier, U. and Schmidhuber, J. (2012). Multi-column deep neural networks for image classification. **2012 IEEE Conference on Computer Vision and Pattern Recognition**: pp. 3642–3649.

Corduas, M. and Piccolo, D. (2008). Time series clustering and classification by the autoregressive metric, **Comput. Stat. Data Anal**. Vol. 52, no. 4, pp. 1860–1872.

Deng, L. and Yu D. (2014). Deep Learning: Methods and Applications. **Foundations and Trends in Signal Processing**. Vol. 7, Issues. 3-4, pp.197-387. ISSN: 1932-8346.

Demongeot, M. J., Mazoyer, M. J., Peretto, M. P. and Whitley, M. D. (1994). NEURAL NETWORK SYNTHESIS USING CELLULAR ENCODING AND THE GENETIC ALGORITHM.

Deza, M. M., Deza, E., (2009). Encyclopedia of Distances. **Springer-Verag Berlin Heidelberg**. ISBN 978-3-642-00233-5.

Dorigo, M. (1992). Optimization, Learning and Natural Algorithms, **PhD thesis, Politecnico di Milano, Italy**.

Duan, G., Suzuki, Y. and Kawagoe, K. (2006). Grid representation of time series data for similarity search, in: **The institute of Electronic, Information, and Communication Engineer**.

Everitt, Brian. (1998). Dictionary of Statistics. Cambridge, UK**: Cambridge University Press**. pp. 96. ISBN 0-521-59346-8.

Faloutsos, C., Ranganathan, M. and Manolopoulos, Y. (1994). Fast subsequence matching in time-series databases, **ACM SIGMOD Rec**. Vol. 23, no. 2, pp. 419–429.

Gianniotis, N., Kügler, S. D., Tiňo, P., and Polsterer, K. L. (2016). Model-coupled autoencoder for time series visualisation. **Neurocomputing**, Vol. 192, pp. 139-146.

Ghysels, E., Santa-Clara, P. and Valkanov, R. (2006). Predicting volatility: getting the most out of return data sampled at different frequencies, **J. Econom**. Vol. 131, no. 1–2, pp. 59–95.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. **MIT Press**.

Grabusts, P. and Borisov, A. (2009). Clustering Methodology for Time Series Mining. **Scientific Journal of RIGA Technical University**. Computer Science (1407-7493). Vol. 40, pp. 81-86.

Graps, A. (1995). An introduction to wavelets. **IEEE computational science and engineering**, Vol. 2, no. 2, pp. 50-61.

Guha, S., Rastogi, R. and Shim, K. (1998). CURE: An efficient clustering algorithm for large databases. **ACM SIGMOD Rec**. Vol. 27, no. 2, pp. 73–84.

Gusfield, D. (1997). Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology, **Cambridge University Press**.

Han, J. and Kamber, M. (2001). Data Mining: Concepts and Techniques. **Morgan Kaufmann Publishers**.

Hinton, G. E. (2007). "Boltzmann Machine". **Scholarpedia**, 2(5):1668. [Online]. Avialable:http://www.scholarpedia.org/article/Boltzmann_machine#Restricted_Boltzmann_machines.

Hinton, G. E. (2009). Deep belief networks. **Scholarpedia**, Vol. 4, no. 5, pp. 5947. [Online].Available:http://www.scholarpedia.org/article/Deep_belief_networks.

Hinton, G. E., and Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. **Science**, Vol.313, Issues.5786, pp.504-507.

Hinton, G. E., Osindero, S. and Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. **Neural computation**, Vol. 18, Issues.7, pp.1527-1554.

Holland, J. H. (1975). “Adaptation in Natural and Artificial System”. **Michigan: University of Michigan Press**. 1975.

Huang, P., Huang, Y., Wang, W. and Wang, L. (2014). Deep embedding network for clustering. **In Pattern Recognition (ICPR), 2014 22nd International Conference on**, pp. 1532-1537. IEEE.

Houck, C. R., Joines, J. A., and Kay, M. G. (1995). A genetic algorithm for function optimization: a Matlab Implementation. **Ncsu-ie tr**, 95(09).

Jain, A., Murty, M. N., Flynn, P. J. (1999). Data clustering: a review. **ACM Comput. Surv**. Vol.31, no.3, pp.264–323.

Kalpakis, K., Gada, D. and Puttagunta, V. (2001). Distance measures for effective clustering of ARIMA time-series, in: **Proceedings 2001 IEEE International Conference on Data Mining**, pp. 273-280.

Karypis, G., Han, E.H. and Kumar, V. (1999). Chameleon: hierarchical clustering using dynamic modeling, **Comput. (Long. Beach. Calif)**. Vol. 32, no. 8, pp. 68–75.

Kaufman, L. and Rousseeuw, P. J. (1990). Finding Groups in Data: An Introduction to Cluster Analysis. **A Wiley-Science Publication John Wiley & Sons**.

Kennedy, J., Eberhart, R. (1995). "Particle Swarm Optimization". **Proceedings of IEEE International Conference on Neural Networks**. IV. pp. 1942–1948.

Keogh, E. (2005). Hot sax: efficiently finding the most unusual time series subsequence, in: **Proceedings of Fifth IEEE International Conference on Data Mining ICDM05**, pp. 226–233.

Keogh, E. and Lin, J. (2005). Clustering of Time-Series Subsequence is Meaningless: Implications for Previous and Future Research. **Knowledge and Information Systems**. Vol. 8, pp. 154-177.

Keogh, E., Chakrabarti, and K., Pazzani, M. (2001). Locally adaptive dimensionality reduction for indexing large time series databases. In: **Proceedings of ACM SIGMOD conference on management of data, Santa Barbara**, May 21–24, pp 151–162.

Keogh, E., Pazzani, M. (1998). An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback, in: **Proceedings of the 4th International Conference of Knowledge Discovery and Data Mining**, pp. 239–241.

Keogh, E., Pazzani, M., Chakrabarti, K. and Mehrotra, S. (2000). A simple dimensionality reduction technique for fast similarity search in large time series databases, **Knowl. Inf. Syst**. Vol. 1805. no. 1, pp. 122-133.

King, B. (1967). Step-wise Clustering Procedures, **J. Am. Stat. Assoc**. Vol. 69, pp. 86-101.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. **arXiv preprint arXiv:**1312.6114.

Korn, F., Jagadish, H.V. and Faloutsos, C. (1997). Efficiently supporting ad hoc queries in large datasets of time sequences, **ACM SIGMOD Record 26**, pp. 289–300.

Kramer, M. A. (1992). Autoassociative neural networks. **Computers & chemical engineering**, 16(4), 313-328.

Kumar, N., Lolla, N., Keogh, E., Lonardi, S. (2005). Time-series bitmaps: a practical visualization tool for working with large time series databases, **SIAM2005DataMin**, pp. 531–535.

Kwedlo, W. (2011). A clustering method combining differential evolution with the K-means algorithm. **Pattern Recognition Letters.** Vol. 32, pp. 1613–1621.

Längkvist, M., Karlsson, L. and Loutfi, A. (2014). A review of unsupervised feature learning and deep learning for time-series modeling. **Pattern Recognition Letters**, Vol. 42, pp. 11-24.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. **Nature**, Vol. 521(7553), pp. 436-444.

Li, G., Braysy, O., Jiang, L., Wu, Z. and Wang, Y. (2013). Finding time series discord based on bit representation clustering. **Knowledge-Based Systems 54**, pp. 243–254.

Lin, J., Keogh, E., Lonardi, S. and Chiu, B. (2003). A symbolic representation of time series, with implications for streaming algorithms, in: **Proceedings of 8th ACM SIGMOD Workshop on Research Issues Data Mining and Knowledge Discovery – DMKD '03**, pp. 2.

Lin, J., Keogh, E., Wei, L., Lonardi, S. (2007). Experiencing SAX: a novel symbolic representation of time series, **Data Min. Knowl. Discov**. Vol. 15, Issue 2, pp. 107–144.

Maimon, O., Rokach, L. (2010). Data Mining and Knowledge Discovery Handbook: Second Edition. **Springer, New York Dordrecht Heidelberg London**, pp. 270.

Minnen, D., Isbell, C.L., Essa, I. and Starner, T. (2007). Discovering multivariate motifs using subsequence density estimation and greedy mixture learning, **Proc. Natl. Conf. Artif. Intell.** Vol. 22, no. 1, pp. 615.

Minnen, D., Starner, T., Essa, M. and Isbell, C. (2006). Discovering characteristic actions from on-body sensor data, in: **Proceedings of 10th IEEE International Symposium on Wearable Computers**, pp. 11–18.

Mohammed J.Z. and Wagner M.J. (2014). Data Mining and Analysis: Fundamental Concepts and Algorithms. **Cambridge University Press**, USA.

Montero, P. and Vilar, A.J. (2014). TSclust: An R Package for Time Series Clustering. **Journal of Statistical Software**. November 2014, Vol. 62, Issue 1, pp. 1-43.

Morinaka, Y., Yoshikawa, M., Amagasa, T. and Uemura, S. (2001). The L-index: an indexing structure for efficient subsequence matching in time sequence databases, in: **Proceedings of 5th PacificAisa Conference on Knowledge Discovery and Data Mining**, pp. 51–60.

Niennattrakul, V. and Ratanamahatana, C. (2007). On clustering multimedia time series data using k-means and dynamic time warping, in: **Proceedings of the International Conference on Multimedia and Ubiquitous Engineering, MUE '07**, pp. 733-738.

_____. (2007). Inaccuracies of shape averaging method using dynamic time warping for time series data, **Comput. Sci. 2007**. pp. 513-520.

Olshausen, B. A. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. **Nature**. Vol. 381 (6583), pp. 607-609.

Olszewski, R. T. (2001). Generalized feature extraction for structural pattern recognition in time-series data (No. CMU-CS-01-108). **CARNEGIE-MELLON UNIV PITTSBURGH PA SCHOOL OF COMPUTER SCIENCE**.

Panuccio, A., Bicego, M. and Murino, V. (2002). A Hidden Markov Model-based approach to sequential data clustering, in **Structural, Syntactic, and Statistical Pattern Recognition**, T. Caelli, A. Amin, R. Duin, R. De, and M. Kamel, Eds.

Portet, F., Reiter, E., Gatt, A., Hunter, J., Sripada, S., Freer, Y. and Sykes, C. (2009). Automatic generation of textual summaries from neonatal intensive care data, **Artif. Intell**. Vol. 173, no. 7, pp. 789–816.

Rai, P. and Singh, S. (2010). A survey of clustering techniques, **Int. J. Comput. Appl**. Vol. 7, no. 12, pp. 1–5.

Ranzato, M. A., Boureau, Y. L., & Cun, Y. L. (2008). Sparse feature learning for deep belief networks. **In Advances in neural information processing systems**, pp. 1185-1192.

Ratanamahatana, C. (2005). Multimedia retrieval using time series representation and relevance feedback, in: **Proceedings of 8th International Conference on Asian Digital Libraries (ICADL 2005)**, pp. 400-405.

Ratanamahatana, C., Keogh, E., Bagnall, A.J. and Lonardi, S. (2005). A novel bit level time series representation with implications for similarity search and clustering, in: **Proceedings of 9th Pacific-Asian International Conference on Knowledge Discovery and Data Mining (PAKDD'05)**, pp. 771–777.

Ripoll, V. J. R., Wojdel, A., Romero, E., Ramos, P., and Brugada, J. (2016). ECG assessment based on neural networks with pretraining. **Applied Soft Computing**, Vol. 49, pp. 399-406.

Roiger, R. J. and Geatz, M. W. (2003). Data Mining A Tutorial – Based Primer. **Pearson Education, Inc. Addison Wesley**. pp. 11-12.

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. **Journal of Computational and Applied Mathematics**. Vol. 20, pp. 53-65.

Sakoe, H. and Chiba, S. (1971). A dynamic programming approach to continuous speech recognition, **Proceedings of the Seventh International Congress on Acoustics**. Vol. 3, pp. 65-69.

_____. (1978). Dynamic programming algorithm optimization for spoken word recognition, **IEEE Trans. Acoust. Speech Signal Process**. Vol. 26, no. 1, pp. 43-49.

Salakhutdinov, R. and Hinton, G. E. (2009). Deep Boltzmann Machines. In **AISTATS**, Vol. 1, pp. 3.

Schmidhuber, J. (2015). Deep Learning in Neural Networks: An Overview. **Neural Networks**. Vol. 61, pp. 85–117. PMID 25462637.

_____. (2015). Deep Learning. **Scholarpedia**, Vol. 10(11):32832. [Online]. http://www.scholarpedia.org/article/Deep_Learning.

Shahbaba, M., Beheshti, S. (2014). MACE-means clustering. **Singnal Processing**. Vol. 105, pp. 216-225.

Shatkay, H. and Zdonik, S.B. (1996). Approximate queries and representations for large data sequences, in: **Proceedings of the Twelfth International Conference on Data Engineering**, pp. 536–545.

Shen, C., Wang, L. and Li, Q. (2007). Optimization of injection molding process parameters using combination of artificial neural network and genetic algorithm method. **Journal of Materials Processing Technology**, 183(2), 412-418.

Shieh, J. and Keogh, E. (2009). iSAX: disk-aware mining and indexing of massive time series datasets, **Data Min. Knowl. Discov**. Vol. 19, no. 1, pp. 24–57.

Silberer, C. and Lapata, M. (2014). Learning Grounded Meaning Representations with Autoencoders. **In ACL** Vol. 1, pp. 721-732.

Smolensky, P. (1986). Information processing in dynamical systems: Foundations of harmony theory. In Rumelhart, D. E. and McClelland, J. L., editors, Parallel Distributed Processing: Vol. 1: Foundations, pp.194-281. **MIT Press**, Cambridge, MA.

Sneath, P., and Sokal, R. (1973). Numerical Taxonomy. **W.H. Freeman Co.**, San Francisco, CA.

Song, C., Liu, F., Huang, Y., Wang, L., & Tan, T. (2013). Auto-encoder based data clustering. In Iberoamerican Congress on Pattern Recognition, pp. 117-124. **Springer Berlin Heidelberg**.

Suksut, K., Kerdprasop, K., and Kerdprasop, N. (2017). Support Vector Machine with Restarting Genetic Algorithm for Classifying Imbalanced Data. **International Journal of Future Computer and Communication**, Vol. 6(3), pp. 92.

Tian, F., Gao, B., Cui, Q., Chen, E., & Liu, T. Y. (2014, July). Learning Deep Representations for Graph Clustering**. In AAAI**, pp. 1293-1299.

Theodoridis, S., Pikrakis, A., Koutroumbas, K., Cavouras, D. (2010). An Introduction to Pattern Reccognition: A MATLAB Approach. **Academic Press**, USA.

Thinsungnoen, T., Kaoungku, N., Durongdumronchai, P., Kerdprasop, K., and Kerdprasop, N. (2015). The Clustering Validity with Silhouette and Sum of Squared Errors. **Proceedings of the 3rd International Conference on Industrial Application Engineering 2015**, pp. 44-51.

Thinsungnoen, T., Kerdprasop, K. and Kerdprasop, N. (2017). A Deep Learning of Time Series for Efficient Analysis. **International Journal of Future Computer and Communication**, Vol. 6(3), pp. 123.

Van Wijk, J.J. and Van Selow, E.R. (1999). Cluster and calendar based visualization of time series data, in: **Proceedings of 1999 IEEE Symposium on Information Vision**, pp. 4–9.

Vlachos, M., Kollios, G. and Gunopulos, D. (2002). Discovering similar multi-dimensional trajectories, in: **Proceedings of 18th International Conference on Data Engineering**, pp. 673-684.

Vlachos, M., Lin, J. and Keogh, E. (2003). A wavelet-based anytime algorithm for k-means clustering of time series, **Proc. Work. Clust**, pp. 23–30.

Vuori, V. and Laaksonen, J. (2002). A comparison of techniques for automatic clustering of handwritten characters, **Pattern Recognit**., Vol. 3, pp. 330168.

Wang, Q., Megalooikonomou, V. and Faloutsos, C. (2010). Time series analysis with multiple resolutions. **Information Systems**. Vol. 35, pp. 56-74.

Wang, X., Mueen, A., Ding, H., Trajcevski, G., Scheuermann, P. and Keogh, E. (2013). Experimental comparison of representation methods and distance measures for time series data, **Data Min. Knowl. Discov**. Vol. 26, pp. 275-309.

Wehrens, R. and Buydens, L. M. C. (1998). Evolutionary optimization: a tutorial, Trends in **Analytical Chemistry**. Vol. 17(4), pp. 193-203.

Wenke Lee, Salvatore J. Stolfo, Philip K. Chan, WeiFan. (2001). Real Time Data Minin-based Intrusion Detecion. **Columbia University**, NY 10027.

Wright, A. H. (1991). Genetic algorithms for real parameter optimization. **Foundations of genetic algorithms**, Vol. 1, pp. 205-218.

Zhang, T., Ramakrishnan, R. and Livny, M. (1996). BIRCH: an efficient data clustering method for very large databases, **ACM SIGMOD Rec.** Vol. 25, no. 2, pp. 103–114.

ภาคผนวก

ภาคผนวก ก

รหัสต้นฉบับ ภาษา R การแทนอนุกรมเวลาด้วย TSDL Algorithm

รหัสต้นฉบับ ภาษา R การแทนอนุกรมเวลาด้วย TSDL Algorithm

```
###========== 1.EEGs data ==============================
# lenght 4097 times, It's has true_cluster(closed->col 1:50, close->col 51:100)
files.A <- paste("./Data/setA.zip")
files.B <- paste("./Data/setB.zip")
instances.each <- 50
EEGot <- matrix(NA, nrow = 4097, ncol = instances.each * 2)

for (i in 1:instances.each) {
 z <- unz(files.A, paste("Z", formatC(i, width=3, flag="0"), ".txt", sep=""))
 EEGot[,i] <- as.numeric(readLines(z))
 close(z)
 z <- unz(files.B, paste("O", formatC(i, width=3, flag="0"), ".txt", sep=""))
 EEGot[, instances.each + i] <- as.numeric(readLines(z))
 close(z)
}

#---For Deep Learning Networks is not time series (not Transpost)
trueClust1 <- rep(c(1, 2), each = instances.each)
EEGot <- EEGot[-4097,]
EEGo <- data.frame(t(EEGot))
EEGolabel <- data.frame(trueClust1)
colnames(EEGolabel) <- "trueClust"
trueClust <- trueClust1
EEGs <- data.frame(EEGo, EEGolabel)

###========== 2.ECGs data ==============================
ECG200t <- read.csv("./Data/ECG200.csv")
trueClust3 <- c(2,1,2,2,1,1,2,2,1,1,1,1,1,2,1,1,1,2,2,1,1,1,2,1,1,1,2,1,1,1,1,1,1,1,2,1,1,1,1,
          1,1,1,2,2,1,2,2,1,2,1,1,1,2,1,1,2,2,1,1,1,2,1,1,1,2,1,1,1,1,1,1,2,1,1,1,1,1,1,
          1,1,1,2,2,2,2,1,2,1,2,1,1,2,1,1,2,2,1,1,1,1,1,1,2,1,1,1,1,2,2,1,2,2,2,2,1,1,1,2,1,
          2,1,1,1,1,1,2,1,2,1,1,1,2,2,1,1,1,1,2,2,1,1,2,1,1,1,1,1,1,1,1,2,2,2,2,2,1,2,2,2,
          2,1,1,2,2,2,1,1,1,1,1,1,1,1,2,1,1,1,1,1,2,1,1,2,2,2,2,1,1,1,2,1,1,1,1,1,2,2,1)
ECG200 <- data.frame(t(ECG200t))
ECG200label <- data.frame(trueClust3)
trueClust <- trueClust3
colnames(ECG200label) <- "trueClust"
ECGs <- data.frame(ECG200, ECG200label)
```

```
#======== Purity Measures ========================
ClusterPurity <- function(clusters, classes) {
  sum(apply(table(classes, clusters), 2, max)) / length(clusters)
}
###====== Finding Appropriate number by Silhouette, SSE ====
library(cluster)  #---for silhouette;   library(GMD) #-- for css, cssobj;   library(fpc);
findNum <- function(datas, nloop, CMethod, dists ) {
  library(cluster)  #---for silhouette;   library(GMD) #-- for css, cssobj;   library(fpc);
  wss <- 0;  si_i <- 0;  si <-  0;  sse <- 0;  maxsi <- 0;  maxs <- 0;  apk <- 0;  ksse <- 2

  for (i in 1:nloop) {
    if (CMethod=="kmeans") {  set.seed(5555); (clusts <- kmeans(datas,i)$cluster) }
    else if (CMethod=="hclust") { clusts <- cutree(datas,k=i) }
    else if (CMethod=="pdc") { clusts <- cutree(datas,k=i) }
    else if (CMethod=="pam") {set.seed(5555); clusts <- pam(dists, k=i)$clustering }

    #-- for Silhouette
    if (i>1) {
      si_i <- silhouette(clusts, dists)    #plot(si_i) # silhouette plot
      si[i] <-  ave(si_i[,c("sil_width")])[1] #or ave(si_i[,3])
    }
    if (((si[i]) > maxsi) || (i<3))  {
      maxsi <- si[i];       apk <- i;
    }

    #--- for SSE
    cssobj <- css(dists,clusts)
    wss[i] <- (cssobj$totwss/i)  #average of tot.withinSSE
  }
    for (j in 2:nloop){
    if (wss[j] > 0.00009){
      sse[j] <- (abs(wss[j-1] - wss[j]) * 100 )/ wss[j]
      if (sse[j] > maxs) {
        maxs <- sse[j];       ksse=j;
      }
    }else
      break
  }
  out <- list(clusts=clusts, si=si, wss=wss, ksse=ksse, apk=apk)
  return(out)
}
```

```
#======== Plot SSE Vs. Number of Clusters =========
plotSSE <- function(wss,n){
  library(ggvis);
  wsss = data.frame(c(1:n), c(wss));
  names(wsss)[1] = 'Clusters';
  names(wsss)[2] = 'SSE'
  wsss %>%
    ggvis(~Clusters, ~SSE) %>%
    layer_points(fill := 'blue') %>%
    layer_lines() %>%
    set_options(height = 300, width = 400)
}
#======== Darch Package on autoencoder ========
library(darch)
Autoencoder <- function(datas, enLayer, epch, flname) {
  set.seed(999)
  darch <- darch(datas, datas, enLayer, darch.isClass = F,
                 preProc.params = list(method = c("center", "scale")), preProc.targets = T,
                 darch.numEpochs = epch, darch.batchSize = 3,
                 darch.unitFunction = softplusUnit, bp.learnRate = 0.1,
                 darch.fineTuneFunction = backpropagation)
  predictions <- predict(darch, newdata = datas)
  write.csv(predictions, flname, row.names = TRUE)

  out <- list(darchobj=darch, predicts=predictions, mse=mse)
  return(out)
}
#======== Genetic Optimization ==============
library(genalg)
library(ggplot2)
library(TSclust)
library(cluster)
# ---- 1.Monitor function for Evaluate function ----
monitor <- function(obj) {
  xlim = c(obj$stringMin[1], obj$stringMax[1]);
  ylim = c(obj$stringMin[2], obj$stringMax[2]);
  zlim = c(obj$stringMin[3], obj$stringMax[3]);

  flname <- paste0("Data/ENC/Popiter_",obj$iter,".csv")
  write.csv(obj$population, flname, row.names = TRUE)
}
```

```
evaluate <- function(string=c()) {
   returnVal = NA;
   if (length(string) == 3) {
   #--- 1. Chromosome Encoding --> Randomize number of Nuerons of Hidden unit
   #--- 2. Population Initialization c(Hidden1, Hidden2, CodesLayer)
   Hdd1 <- round(string[1]/1000,1)*1000; print(paste0(string[1],"-",Hdd1))
   Hdd2 <- round(string[2]/100,0)*100; print(paste0(string[2],"-",Hdd2))
   ifelse(string[3] < 10, Cds <- round(string[3],0),
          Cds <-  round(string[3]/10,0)*10) ;print(paste0(string[3],"-",Cds))
   #--- 3. Fitness Function
   #------ 3.1 TSDL: Time Series Representation with Deep Learninging Technique
   enCodeLayer <- c(NCOL(TsData),Hdd1,Hdd2,Cds,Hdd2,Hdd1,NCOL(TsData))
   fln <- paste0("Data/ENC/",datan,Hdd1,"_",Hdd2,"_",Cds,".csv")
   features <- Autoencoder(TsData, enCodeLayer, epch, fln)
   #----- After autoencoder then Retrive TS Representative
   predictions <- read.csv(fln);        predictions$X <- NULL
   # predictions <- TsData * predictions #---for EEGs Dataset
   #------ 3.2 Finding the Purity value
   hc <- pdclust(t(predictions))
   memb <- cutree(hc, k=2)
   Prt <- ClusterPurity(memb, trueClust)*100; print(Prt)
   #----- Write every neurons to file
   write(Hdd1, paste0("Data/ENC/",datan,"_rbga_Hdd1"), append = TRUE)
   write(Hdd2, paste0("Data/ENC/",datan,"_rbga_Hdd2"), append = TRUE)
   write(Cds, paste0("Data/ENC/",datan,"_rbga_Cds"), append = TRUE)
   write(Prt, paste0("Data/ENC/",datan,"_rbga_Prt"), append = TRUE)
   returnVal = (100 - Prt) #The optimal is a minimal data
  } else {    stop("Expecting a chromosome of length 3!");  }
  returnVal
}
#===== Processing of Genetic Algorithm ===========
TsData <- EEGo; trueClust <- trueClust1; datan <- "eegs"    # for EEGs
# TsData <- ECG200; trueClust <- trueClust3;datan <- "ecgs"    # for  ECGs
epch <- 50; iter <- 3; popSizes <- 20
(ptm <- proc.time())
GAmodel = rbga(c(500, 100, 1), c(1000, 500, 30), popSize=popSizes, iters=iter,
               monitorFunc=monitor, evalFunc=evaluate, verbose=TRUE,
               mutationChance=0.05,elitism = T)
proc.time() - ptm
cat(summary(GAmodel))
```

ภาคผนวก ข

รายการบทความวิจัยตีพิมพ์

รายการบทความวิจัยตีพิมพ์

Thinsungnoen, T., Kaoungku, N., Durongdumronchai, P., Kerdprasop, K., and Kerdprasop, N. (2015). The Clustering Validity with Silhouette and Sum of Squared Errors. In: **Proceedings of the 3rd International Conference on Industrial Application Engineering 2015**, pp. 44-51.

Thinsungnoen, T., Kerdprasop, K., Kerdprasop, N. (2558). ON IMPROVING K-MEANS CLUSTERING EFFICIENCY WITH IMPORTANCE AND CORRELATION ANALYSES. **9th SOUTH EAST ASIAN TECHNICAL UNIVERSITY CONSORTIUM (SEATUC) SYMPOSIUM**, pp. 56

Kaoungku, N., Thinsungnoen, T., Durongdumronchai, P., Kerdprasop, K., Kerdprasop, N. (2558). Discretization Based on Chi2 Algorithm and Visualize Technique for Association Rule Mining. In: **Proceedings of the 3rd International Conference on Industrial Application Engineering 2015**, pp. 254-260.

Thinsungnoen, T., Kerdprasop, K. and Kerdprasop, N. (2017). A Deep Learning of Time Series for Efficient Analysis. **International Journal of Future Computer and Communication**, Vol. 6(3), pp. 123.

The Clustering Validity with Silhouette and Sum of Squared Errors

Tippaya Thinsungnoen[a*], Nuntawut Kaoungku[b], Pongsakorn Durongdumronchai[b],
Kittisak Kerdprasop[b], Nittaya Kerdprasop[b]
[a]Informatics Program, Faculty of Science and Technology, Nakhon Ratchasima Rajabhat University, Thailand
[b]Data Engineering Research Unit, School of Computer Engineering, Institute of Engineering,
Suranaree University of Technology, Thailand
*Corresponding Author: tippayasot@hotmail.com

Abstract

The data clustering with automatic program such as k-means has been a popular technique widely used in many general applications. Two interesting sub-activity of clustering process are studied in this paper, selection the number of clusters and analysis the result of data clustering. This research aims at studying the clustering validation to find appropriate number of clusters for k-means method. The characteristics of experimental data have 3 shapes and each shape have 4 datasets (100 items), which diffusion is achieved by applying a Gaussian distributed (normal distribution). This research used two techniques for clustering validation: Silhouette and Sum of Squared Errors (SSE). The research shows comparative results on data clustering configuration k from 2 to 10. The results of both Silhouette and SSE are consistent in the sense that Silhouette and SSE present appropriate number of clusters at the same k-value (Silhouette value: maximum average, SSE-value: knee point).

Keywords: Clustering Validity, Silhouette Measure, Sum of Squared Errors, k-means Algorithm.

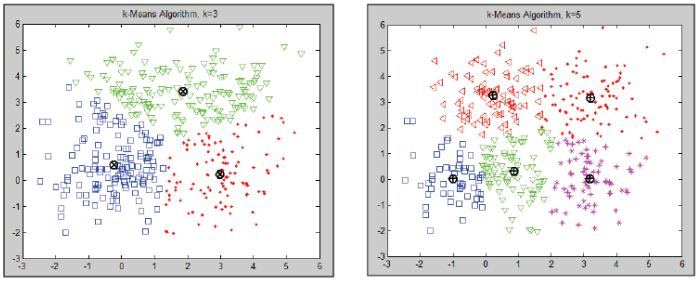
1. Introduction

A clustering is to group data. Although the clustering is similar to the data classification in terms of data input, the clustering is learning without target class. The clustering algorithm forms groups based on object similarities[1]. The clustering was applied to many fields such as bioinformatics, genetics, image processing, speech recognition, market research, document classification, and weather classification[2]. In addition, the clustering was applied to document data analysis that was one of big data learning[3-7].

There are various algorithms for the data clustering. But the most popular one is k-means algorithm. The k-means algorithm is very simple in operation and suitable for unraveling compact clusters and a fast iterative algorithm[8]. The principle of k-means algorithm has divide n objects from dataset for k clusters that used center-based clustering methods[2]. In addition, each cluster has represented by the means of objects[8]. Although k-means is a popular technique, k-means is not known the correct number of clusters a priori. Consequently, the main challenge for these clustering methods is in determining the number of clusters[2]. In general, the number of clusters has been set by users or archives from knowledge of research[1, 9-11].

Fig. 1 shows the distribution of each cluster when k=3, and k=4. The researcher found that the determination of suitable k value is not clear as shown in fig. 1a and fig. 1b. As mentioned above about problem of clustering, there are various research for selecting an appropriate number of clusters[10-13]. Each of the proposed technique is suitable for each of data distribution such as Gaussianity and non-Gaussianity[14]. Therefore, finding the correct k-value for clustering is still a fundamental problem of clustering methods[15-16].

In this research, we study the clustering validity techniques to quantify the appropriate number of clusters for k-means algorithm. These techniques are Silhouette and Sum of Squared Errors. The rest of this paper is organized as follows. Section 2 discusses related research. Section 3 contains a description of methodology. Section 4 presents the results of experiments. The last section contains conclusions.



(a) k=3 (b) k=5

Fig. 1. Data clustering when k=3 and k=5

2. Related Research

Rousseeuw[12] have proposed the concept of the monitoring cluster. In the research proposed for Silhouette technique, which is based on the comparison of objects tightness and separation. The silhouette can reflect the data is grouped that objects are organized into groups that match it. This is a tool to assess the validity of the clustering to be used for selecting the optimal k in the cluster.

Kwedlo[17] have proposed the concept of problem solving in order to know the number of cluster using the Sum of Squared Errors (SSE). The research for developed a new method, called DE-KM (Differential Evolution Algorithm: DE) technique is the combination of algorithms, k-means clustering by tuning in DE and sort data to see the evolution. Experimental results show that the highest k values appropriate to the clustering, and DE-KM SSE values than the other methods they tested.

Shahbaba and Beheshti[2] have proposed the concept of dealing with the problem of determining the correct number of cluster to be grouped. The method used to estimate the probability of error point average (ACE), which is the difference between the actual point and estimate point. The idea is to explore how to use the k-means clustering with ACE k-means low (MACE-Means) is used with the UCI data and the other synthetic. They found that a correct number of cluster that can be spent on items that are sturdy little overlap and time less.

Jun et al.[9] have proposed the concept of clustering the documents based on the concept of reducing the dimension of the data, combined with the clustering k-means based on clustering with support vector and silhouette measure. They have experimented with the patent, documents from UCI to analyze separately each group of documents to clustering for technology forecasting.

3. Proposed Methodology

3.1 A Framework of Data Clustering and Validation Approach

The clustering is a data mining at an unsupervised learning technique[2, 17-20]. The principle of data clustering that objects in the same cluster will have to look very similar, while objects in other similar less[17]. There are various algorithms of clustering technique, for example, Basic Sequential Algorithms Scheme (BSAS), Partitioning Around Medoids Algorithm (PAM), Fuzzy c-Means Algorithm (FCM), k-means Algorithm[8] and so on.

The main steps in the work of the clustering has 5 steps[21]. There are (a) set a number of cluster for clustering (k) and cluster feature, (b) set a function for objects similarity measurement, (c) run clustering algorithm, (d) set Visualization to display cluster and (e) clustering validity analysis, as fig. 2.

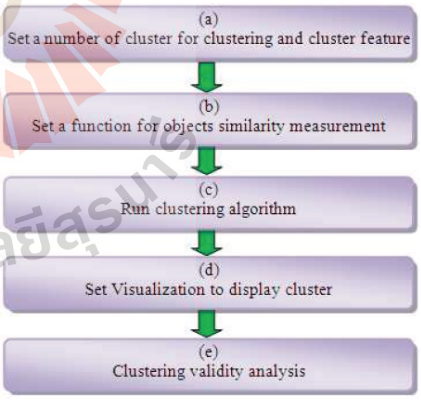


Fig. 2. Show 5 steps in clustering process.

3.2 k-Means Clustering

The k-means clustering is a technique that relies on the center of cluster. This is often represented by the average (Means) of cluster. The clustering measure the similarity of the group by iterating the measurement distance between each object and the center of each cluster[2] using Euclidean distance measuring.

The k-means algorithm is an iterative algorithm which can be described by the following steps.

Algorithm: k-means Clustering[17].

(a) Choose initial centroids $\{m_1,\ldots,m_k\}$ of the clusters $\{C_1,\ldots,C_k\}$.

(b) Calculate new cluster membership. A feature vector x_j is assigned to the cluster C_i if and only if

$$i = \arg\min_{k=1,\ldots,k} ||x_j - m_k||^2 \quad (1)$$

(c) Recalculate centroids for the cluster according.

$$m_i = \frac{1}{|C_i|}\sum_{x_j \in C_i} x_j. \quad (2)$$

(d) If none of the cluster centroids have changed, finish the algorithm. Otherwise go to Step (b).

3.3 Clustering Validity Methods

3.3.1 Silhouette Measure

The concept of Rousseeuw[12] is described as follows: the Silhouette is a tool used to assess the validity of clustering. The silhouette constructed to select the optimal number of cluster with a ratio scale data (as in the case of Euclidean distances) that suitable for clearly separated cluster. The clustering are considered average proximities as the two are dissimilarities and similarities, which work best in a situation with roughly spherical clusters.

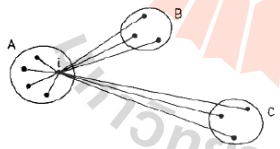


Fig. 3. Show computation *s(i)* for each object, where object *i* belong to cluster A[12].

Case #1 considered dissimilarities[12].

From fig. 3 described for take the object *i* in the data set, and assigned to cluster *A*, then define as follows:

s(i) = in case of dissimilarities.

i = object *i* belong to cluster *A*.

a(i) = average dissimilarity of *i* to all other objects of *A*.

d(i, C) = average dissimilarity of i to all objects of *C*.

b(i) = minimum *d(i, C)*, where $C \neq A$.

B = the cluster *B* for which minimum is attained the neighbor of object *i*

The cluster *B* is like the second-best choice for object *i*: if it could not be accommodated into cluster *A*, which cluster B would be the closest competitor In Fig. 3. The number *s(i)* write this in formula:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}. \quad (3)$$

The number *s(i)* is obtained by combining *a(i)* and *b(i)* as follows:

$$s(i) = \begin{cases} 1 - a(i)/b(i) & if\ a(i) < b(i), \\ 0 & if\ a(i) = b(i), \\ b(i)/a(i) - 1 & if\ a(i) > b(i), \end{cases}$$

s(i) can will be $-1 \leq s(i) \leq 1$

Case #2 considered similarities[12].

In this case consideration similarities and define *a'(i), d'(i, C)*, and put *b'(i)* = maximum *d'(i, C)*, where $C \neq A$. The numbers *s(i)* is obtained by

$$s(i) = \begin{cases} 1 - b'(i)/a'(i) & if\ a'(i) > b'(i), \\ 0 & if\ a'(i) = b'(i), \\ a'(i)/b'(i) - 1 & if\ a'(i) < b'(i), \end{cases}$$

For Example, fig. 4 shows the results silhouette of clustering, when fig. 4 (a) present clustering on k = 2 and fig. 4 (b) clustering on k = 3. The Figure shows the comparison of result: density and separation, Neighbors, the average Silhouette of each cluster. Which silhouette is used to support the evaluation clustering with the maximum of silhouette.

3.3.2 Sum of Squared Errors

The k-means clustering techniques defines the target object (x_i) to each group (C_i), which relies on the Euclidean distance measurement (m_i) is the reference point to check the quality of clustering. The Sum of Squared Errors: SSE is another technique for clustering validity. SSE is defined as follows[17].

$$SSE(X, \Pi) = \sum_{i=1}^{K}\sum_{x_j \in C_i} ||x_j - m_i||^2 \quad (4)$$

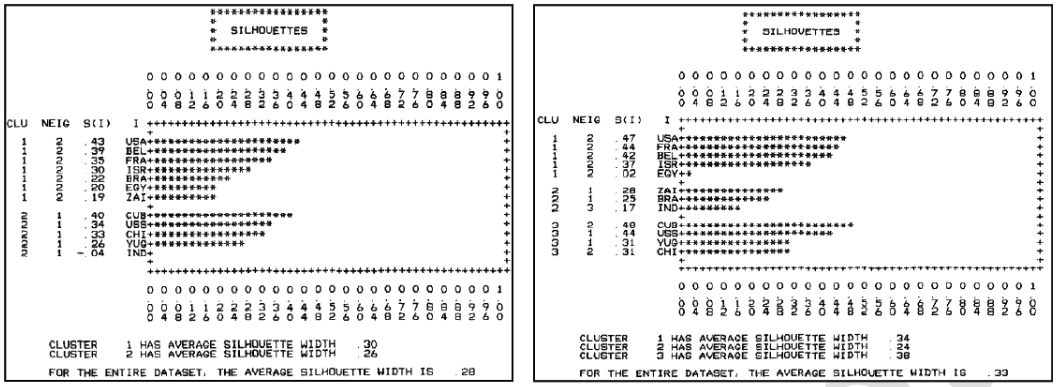
where

N = Feature vectors

$X = \{x_1, \ldots, x_i, \ldots x_N\}$, $x_i \in \Re^M$

$\Pi = \{C_1, C_2, \ldots C_K\}$, $\forall_{i \neq j} C_i \cap C_j = \emptyset, \bigcup_{i=1}^{K} C_i = X, \forall_i C_i \neq \emptyset.$

$||\cdot||$ = Euclidean distance and m_i is centroid of cluster C_i which can computed as Eq.(2).



(a) k=2. (b) k=3.

Fig. 4. Shown Silhouette was present clustering when k=2 and k=3[12].

Conditions of applied the SSE for clustering, is to determine k ≥ 2[22]. When the SSE is applied in graph that generated from the relationship between the SSE and k value at knee point (Significant "knee"), which is positioned to indicate the appropriate number of cluster in the k-means clustering [8] as shown in fig. 5.

3.4 Selection an Appropriate Number of Cluster

The principle of the monitoring tool for clustering, can support the selection of correct k values for the k-means clustering, consider the following.

Fig. 4, Silhouette is used to assist in cluster monitoring. This analysis is compared between Fig. 4 (a) and (b) it is found that the average silhouette of clustering when k = 3, the value 33 will be greater than k = 2, the value 28.

Fig. 5 the SSE is used in the inspection cluster. This analysis was shows the appropriate number at the knee clearly was 5(a) k = 3, 5(b) k = 4 and 5(c) k = 5, which the appropriate number of cluster.

4. Experimentation and Results

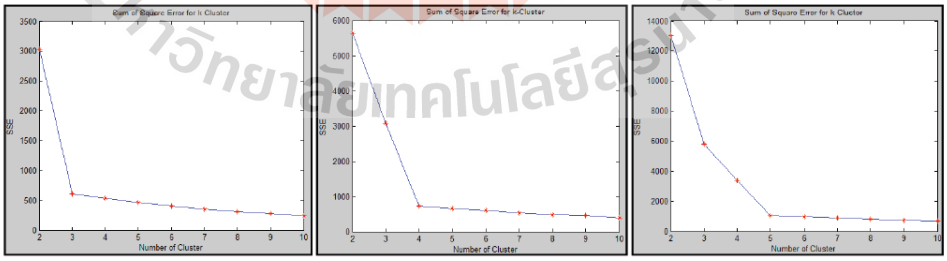
4.1 Experimental Data

The research uses data synthesized with 3 shapes and each shape have 4 datasets (100 items), which is applying a Gaussian distribution (normal distribution). Fig. 6 is the distribution of a spherical around the center of dataset, fig. 7 the distribution is non-spherical lying on the x-axis, and fig. 8 the distribution is spherical, but each group will have some overlap.

4.2 Results of k-Means Clustering Method

In experiments, the researchers repeated the k-means clustering algorithm with datasets by changing the value of k, set k = 2 to k = 10, which illustrate the specific clustering when k = 2, 4, and 6 shown in fig. 6, fig. 7 and fig. 8.

The next step is to investigate the cluster. This relies on the analysis of both Silhouette and SSE of above mentioned, are as follows.



(a) k=3 (b) k=4 (c) k=5

Fig. 5. Number of cluster consideration from the relationship between SSE and the k value.

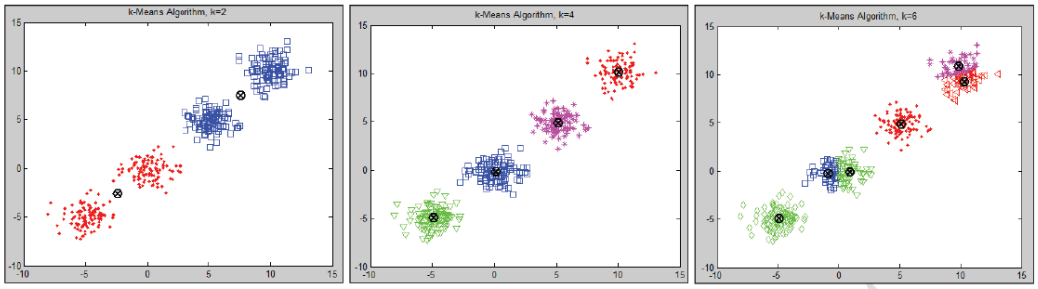


Fig. 6. Clustering with spherical data shape where k=2 (left), k=4 (middle), and k=6 (right)

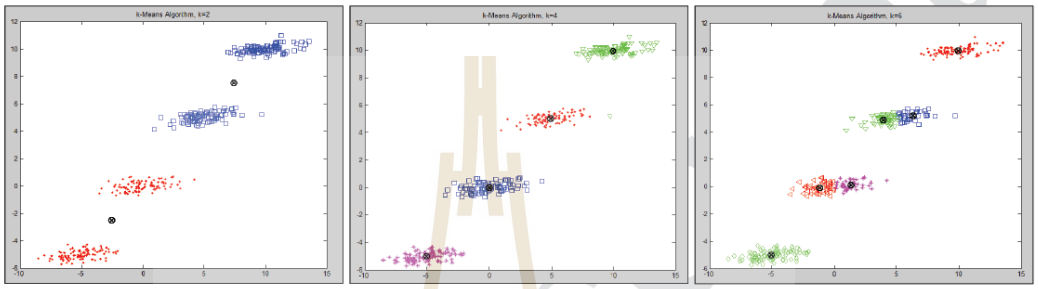


Fig. 7. Clustering with non-spherical data shape where k=2 (left), k=4 (middle), and k=6 (right)

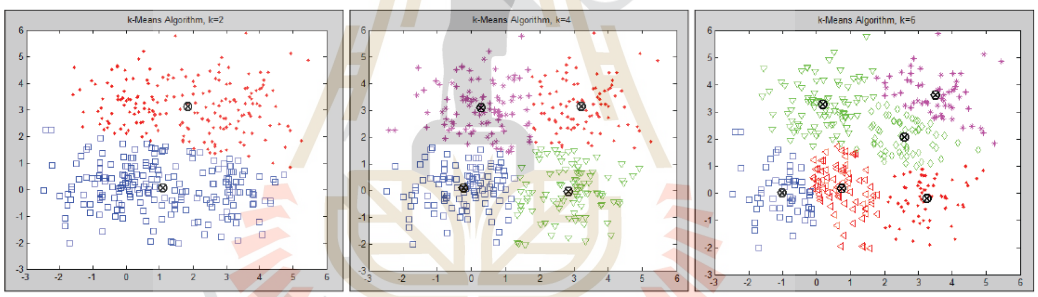


Fig. 8. Clustering with overlap data where k=2 (left), k=4 (middle), and k=6 (right)

4.3 Clustering Validity with Silhouette Measure

Consider Fig. 9 is an illustration Silhouette of a clustering technique to the k-means repeating the grouping by changing the value of k from 2 to 10, which shows a comparison of the density and separation of each cluster. Which found that the density of the k values of k = 2 and k = 4 show the density and separation is optimal.

Using the silhouette to assess the quality of clustering not silhouette diagrams only in addition need to consider the average of silhouette. It was found that the average of all silhouette values when k=4 the highest shown in table 1.

4.4 Clustering Validity with SSE

The Result of SSE for inspection the cluster is shown in Table 2. The table 2 shows the SSE value and rate of change of the SSE when k = 2 to 10, found that when k = 4 SSE is the maximum rate of change. The rate of change (%Change) defined as follows.

$$\%Change = \frac{(SSEofK_{i-1} - SSEofK_i) * 100}{SSEofK_i}. \qquad (5)$$

i can will be $i \geq 2$

where

$SSEofK_{i-1}$ = SSE values of k_{i-1}

$SSEofK_i$ = SSE values of k_i

Table 1. Show comparison of the average of the Silhouette of a k-means clustering when k = 2 to 10.

Number of Cluster	Average of Silhouette		
	Spherical	Non-Spherical	Spherical & Overlap
K=2	0.8305	0.8318	0.5262
K=3	0.7715	0.7553	0.5603
K=4	**0.9117**	**0.9018**	**0.6150**
K=5	0.8182	0.8558	0.5720
K=6	0.7109	0.7982	0.5407
K=7	0.5639	0.7466	0.5177
K=8	0.6198	0.7333	0.5104
K=9	0.5072	0.6945	0.5217
K=10	0.5162	0.6850	0.5177

As the Silhouette to assess the quality of clustering not the data in table 2 that should set correct k value only. In order to investigate the effect is therefore necessary to consider a graph showing the relationship between k and the SSE values at the knee point are shown in fig. 10.

Table 2. Show SSE values and %change from k-means algorithm when k=2 to 10.

Number of Cluster	Sum of Squared Errors					
	Spherical		Non-Spherical		Spherical & Overlap	
	SSE	%Change	SSE	%Change	SSE	%Change
K=2	5,972.97	-	6,042.24	-	1505.40	-
K=3	3,179.66	87.85	3,265.32	85.04	958.94	56.99
K=4	**771.76**	**312.00**	**834.01**	**291.52**	**608.08**	**57.70**
K=5	682.02	13.16	679.50	22.74	518.62	17.25
K=6	612.04	11.43	552.51	22.98	441.48	17.47
K=7	544.41	12.42	430.75	28.27	387.16	14.03
K=8	512.90	6.14	403.10	6.86	337.27	14.79
K=9	436.02	17.63	364.19	10.68	302.21	11.60
K=10	403.84	7.97	245.55	48.32	276.06	9.47

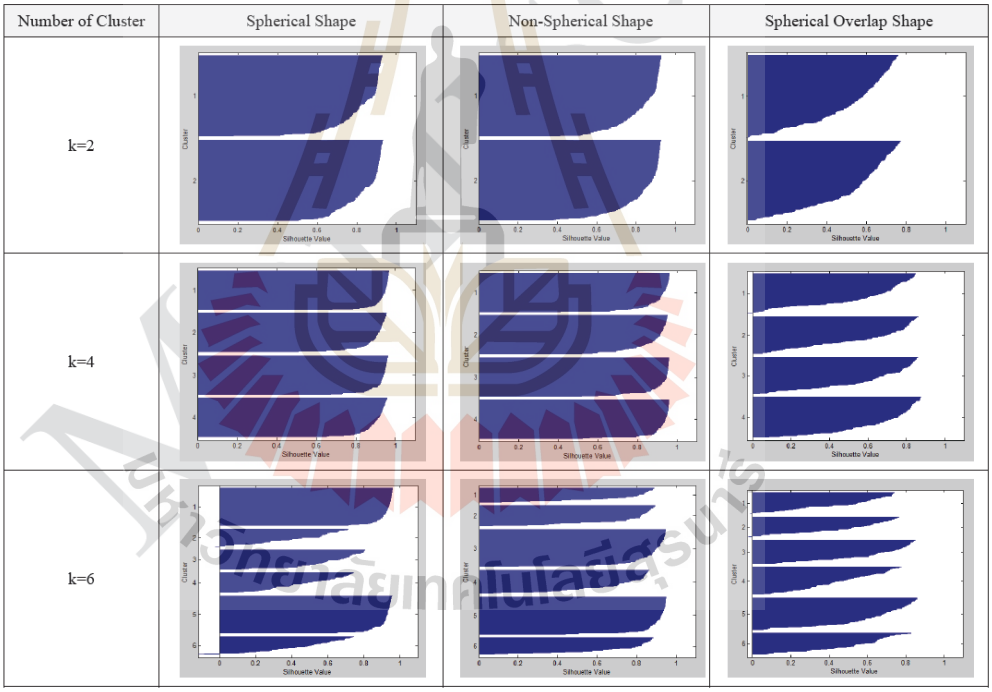
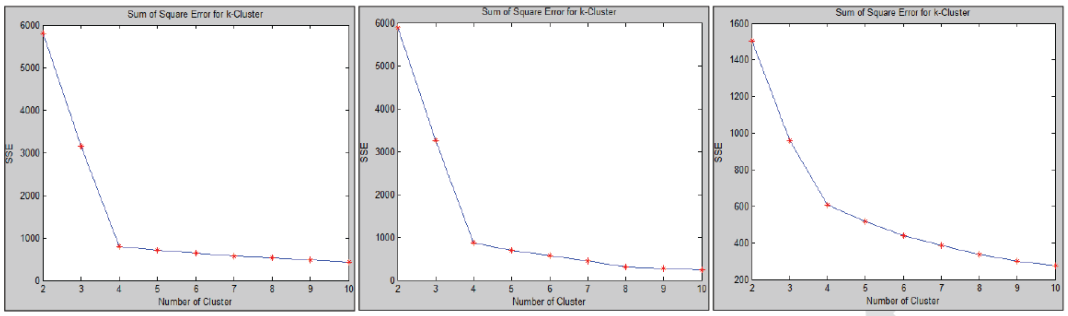


Fig. 9. Silhouette of a clustering technique when k=2, k=4, and k=6



(a) Spherical (b) Non-Spherical (c) Spherical & Overlap

Fig. 10. The graph showing the relationship between k and the SSE values of different data shapes

5. Conclusions

The results of research above when examining the Silhouette clustering analysis is to determine the k = 4 was the highest average Silhouette with all the data sets. When examining the clustering of the graph that shows the relationship between the SSE and k value with k = 4 the result was the knee point. That means the examination of both Silhouette and SSE are result inconsistent. Is that the number of cluster as the same number that k = 4.

However, a comparison of SSE and Silhouette have to attention is if the data does not overlap. Assessment the number of cluster is appropriate both SSE and Silhouette. However, when the data begin to overlap SSE will provide an assessment that is more close to the true value.

Acknowledgment

The first author has been supported by grant from the Informatics Program, Faculty of Science and Technology, Rajabaht Nakhon Ratchasrima University (NRRU). Data Engineering Research Unit has been funded by Suranaree University of Technology.

References

(1) Han, J., and Kamber, M. (2006). Data mining concepts and techniques (2nd ed.). United States of America: Morgan Kaufman Publishers.

(2) Shahbaba, M., Beheshti, S. (2014). MACE-means clustering. Singnal Processing. Vol.105, pp.216-225.

(3) Aliguliyev, R. M. (2009). Clustering of document collection–A weighting approach. Expert Systems with Applications, Vol.36, pp. 7904-7916.

(4) Isa, D., Kallimani, V. P., and Lee, L. H. (2009). Using the Self Organizing map for Clustering of Text Documents. Expert Systems with Applications, Vol.36, pp.9584–9591.

(5) Maziere, P. A. D., and Hulle, M. M. V. (2011). A clustering study of a 7000 EU document inventory using MDS and SOM. Expert Systems with Applications, Vol.38, pp. 8835–8849.

(6) Saracoglu, R., Tutuncu, K., and Allahverdi, N. (2007). A fuzzy clustering approach for finding similar documents using a novel similarity measure. Expert Systems with Applications, Vol.33, pp.600–605.

(7) Tseng, Y. H. (2010). Generic title labeling for clustered documents. Expert Systems with Applications, Vol.37, pp.2247–2254.

(8) Theodoridis, S., Pikrakis, A., Koutroumbas, K., Cavouras, D. (2010). An Introduction to Pattern Reccognition : A MATLAB Approach. Academic Press, USA.

(9) Jun, S., Park, S., and Jang, D. (2014). Document clustering method using dimension reduction and support vector clustering to overcome sparseness. Expert Systems with Applications. Vol.41, pp.3204–3212.

(10) Everitt, B. S., Landau, S., and Leese, M. (2001). Cluster analysis (4th ed.). Apnold.

(11) Jun, S., and Uhm, D. (2010). Patent and statistics, What's the connection? Communications of the Korean Statistical Society, Vol.17(2), pp.205–222.

(12) Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics. Vol.20, pp.53-65.

(13) Wang, L., Leckie, C., Ramamohanarao, K., and Bezdek, J. (2009). Automatically determining the number of clusters in unlabeled data sets. IEEE Transactions on Knowledge and Data Engineering, Vol.21(3), pp.335–350.

(14) McNicholas, P. D., Subedi, S. (2012). Clustering gene expression time course data using mixtures of multivariate t-distributions, J. Stat. Plan. Inference 142 (May (5)). p.1114–1127.

(15) Jain, A. K. (2010). Data clustering: 50 years beyond k-means, Pattern Recogn. Lett. Vol.(8) 31, pp.651–666, http://dx.doi.org//10.1016j.patrec..2009.09.011

(16) Aggarwal, C. C., Reddy, C. K. (2013). Data Clustering: Algorithms and Applications, Vol.31, CRC Press, Hoboken, New Jersey, p.648. (Chapman & Hall/CRC Data Mining and Knowledge Discovery Series, ISBN: 1466558210).

(17) Kwedlo, W. (2011). A clustering method combining differential evolution with the k-means algorithm. Pattern Recognition Letters. Vol.32, pp.1613–1621.

(18) Roiger, R. J., Geatz, M. W. (2003). Data Mining A Tutorial – Based Primer. Pearson Education, Inc. Addison Wesley. pp. 11-12.

(19) Jain, A., Murty, M. N., Flynn, P. J. (1999). Data clustering: a review. ACM Comput. Surv. Vol.31(3), pp.264–323.

(20) Kaufman, L., Rousseeuw, P. J. (1990). Finding Groups in Data: An Introduction to Cluster Analysis. Wiley.

(21) Kerdprasop, K. (2006). Density Biased Sampling for Incremental Data Clustering. Research Final Report. School of Computer Engineering, Suranaree University of Technology.

(22) Aloise, D., Deshpande, A., Hansen, P., Popat, P. (2009). NP-hardness of Euclidean sum-of-squares clustering. Mach. Learn Vol.75 (2), pp.245–248.

ON IMPROVING K-MEANS CLUSTERING EFFICIENCY WITH IMPORTANCE AND CORRELATION ANALYSES

Tippaya Thinsungnoen*, Kittisak Kerdprasop, Nittaya Kerdprasop
Data engineering research unit, School of computer engineering
Suranaree university of technology, Thailand

ABSTRACT

The k-means clustering has been a popular technique widely used and used with data distribution both gaussianity and non-gaussianity, which if data distribution is non-gaussianity making the efficiency of data clustering will be low. This research aims at study of improve the efficiency for k-means clustering with feature selection. The experimental comparing feature selection techniques between random forests and correlation, using real data various 4 data sets from UCI machine learning repository. Evaluate the effectiveness of the clustering used by the purity value and f-measure value. The results showed that both feature selection technique provide higher efficiency for k-means clustering in all data sets.

Keyword: k-Means clustering, Random forest, Correlation, Purity, F-measure.

1. INTRODUCTION

A data clustering is process of knowledge discovery in databases. There are various algorithms for data clustering, but one popular algorithm is k-means algorithm is due to a simple operation and a fast processing (Theodoridis et al., 2010). The principle of k-means algorithm is separate data into k cluster and then calculating the means to represent each cluster, which the means represented centroid for measuring the similarity and dissimilarity of the same cluster (Shahbaba and Beheshti, 2014).

Despite such advantages as previous mention, however, k-means clustering has also a limitation, it works well with data that is hyper-spherical cluster shapes and performs poorly when the true underlying clusters are arbitrarily shaped (Ryali et al., 2015). In addition, the data used for clustering may contain features that are not suitable for processing such as has the number of attributes more, each attribute irrelevant, the scale of the data is very different (high or low) and so on. The limitations mentioned above, we must find solution to improve the efficiency of clustering method.

Lee et al. research on the automatic detection of pulmonary edema can help show abnormalities in the lung by CT image using by random forest technique, and presented structure for the hybrid-random forest base on according to the classification of pulmonary nodule from the clustering. The experiments conducted by using scans of the lung patients, including 32 patients and 5,721 images, which are marked by projecting an expert radiologist. The result shows sensitivity best is 98.33%, and the specificity best is 97.11%.

Saha et al. has proposed clustering algorithm named Incremental learning based multiobjective fuzzy clustering for categorical data. The research offered the adjustment multiobjective base on the principles of fuzzy clustering algorithms then integrated with random forest. The experiment finding from a comparison with other method to synthesize 6 data sets and real 4 data sets, the results shows that better than other methods.

Ai-li et al. have proposed clustering and correlation Analysis of the Industry Networks so considered the structure of the network and the correlation weight to determine the correlation analysis. The study has found that it can optimize the industrial structure.

Tosi et al have proposed clustering traffic information based on the correlation analysis for select a variable that is outstanding for the group. The study evaluated the experiment with multiple data sets. The synthetic data and real data The results showed that the accuracy and durability than conventional clustering methods are available.

Yu et al. have presented the concept of clustering that can be grouped to data from multiple different sources and different structure of the data. By strategies called selection features important information for the group. This new concept in the evaluation to selection features by combining the use of the correlation included in their selection techniques. The experimental data and information from the UCI cancer gene operating results showed that the method presented better results with all data sets and more effective strategy than all other options are compared.

Therefore, this paper, has studied performance improvements for k-means clustering technique chosen by comparative feature selection such as random forest and correlation. By selecting a specific importance value of feature assign to k-means clustering. In experiments using real 4 data sets from the UCI, using purify measure and f-measure for evaluating the performance of the clustering.

2. METHODOLOGY

2.1 k-Means clustering method

The k-means clustering technique has been the most widely used because it is easy to understand the process is not complicated. The Means was the center of the cluster. The k-Means algorithm can be described by the following steps.

Algorithm: k-means clustering (Kwedlo, W., 2011)

(a) Choose initial centroids $\{m_1,\ldots,m_k\}$ of the clusters $\{C_1,\ldots,C_k\}$.

(b) Calculate new cluster membership. A feature vector x_j is assigned to the cluster C_i if and only if

$$= \arg\min_{k=1,\ldots,k} ||x_j - m_k||^2 \quad (1)$$

(c) Recalculate centroids for the cluster according.

$$m_i = \frac{1}{|C_i|} \sum_{x_j \in C_i} x_j. \quad (2)$$

(d) If none of the cluster centroids have changed, finish the algorithm. Otherwise go to Step (b).

2.2 Feature selection with random forest technique

Random forest is a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and the same distribution of trees. An application randomly features selection for each node. Internal estimation to determine the strength of the error and relationships and is used to show an increasing number of features used in isolation. The estimation also used to measure the importance of feature (Breiman, L., 2001).

The random forest technique is the classification accuracy than other methods due to (i) reduce the average of variance of feature (ii) reduce of correlation between specific feature and other feature (Lee, SLA, 2010). The selected appropriate feature led to classify successfully. An exploration for importance of feature can be observed from the plot of the means of accuracy show in fig. 1. The figure shows that relationship between means of accuracy and feature for example, if the appropriate feature greater than and equal to 2 we have to be select including V2, V3, V7, V8, V9, V10, V12 and V13.

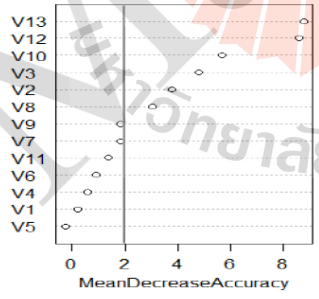


Fig. 1 Measure of variable importance for random forest.

2.3 Feature selection with correlation technique

Correlation coefficient or another word that Pearson's product moment is shows the linear relationship. It has ranged from -1.0 to +1.0 (Correlation coefficient. 2015).

An exploration for correlation coefficient can be observed from the plot of importance value show in fig. 2. if an appropriate is importance >= 0.3 we have to be select including V3, V8, V9, V10, V12 and V13.

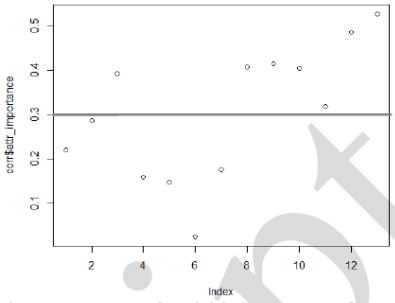


Fig. 2 Measure of variable importance for correlation analysis.

3. EXPERIMENTAL AND RESULT

3.1 Experiments on real data sets

This research was, experiment on real 4 data sets that are characteristic of different data, different number of features and different scale of value inside each feature. The experimental setting data are training data and testing data such as 70:30 percent respectively. Details of the trial are shown in Table 1.

Table 1 Statistics of real data sets.

Data set	Instance	Attribute	Class
Thyroid-disease	7,200	21	3
Heart disease	270	13	2
Pima Indians diabetes	768	8	2
Breast cancer Wisconsin	699	10	2

The experiment we used k-means clustering techniques to clustering with 4 data sets of comparative data in three formats. i.e, original data and data from using feature selection techniques such random forest and correlation analysis. The result shows that the clustering method with data sets was through feature selection can be separate very clear. As shown, the plot results grouped in Table 2.

3.2 Evaluate the clustering efficiency

In evaluating the performance of the clustering has various measures (Mohammed et al., 2014). This paper, we has two measures for clustering evaluate that are purity value and f-measure value, principle as follows.

a. The purity value is the measurement that member are clustering situation, the Purity for cluster C_i defined below.

$$purity_i = \frac{1}{n_i} \max_{j=1}^{k} \{n_{ij}\} \quad (3)$$

Therefore, the purity of clustering C is defined as the weighted sum of the cluster-wise purity values.

$$purity = \sum_{i=1}^{r} \frac{n_i}{n} purity_i = \frac{1}{n} \sum_{i=1}^{r} \max_{j=1}^{k} \{n_{ij}\} \quad (4)$$

b. The f-measure consisting of the two necessary values as precision value and recall value. The precision value of the members in the cluster C_i defined as follows.

$$prec_i = \frac{1}{n_i} \max_{j=1}^{k} \{n_{ij}\} = \frac{n_{ij_i}}{n_i} \quad (5)$$

The recall value of the cluster C_i defined as follows.

$$recall_i = \frac{n_{ij_i}}{|T_{j_i}|} = \frac{n_{ij_i}}{m_{j_i}} \quad (6)$$

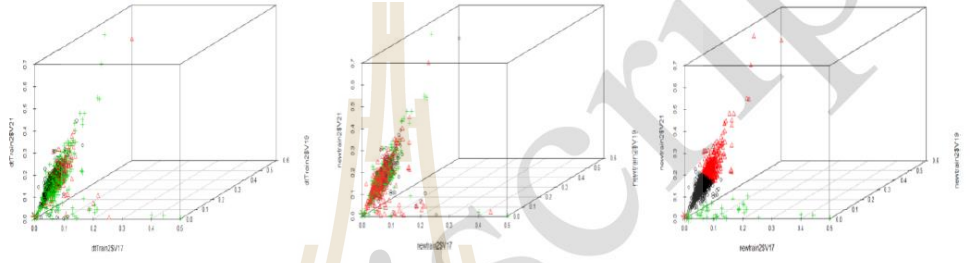
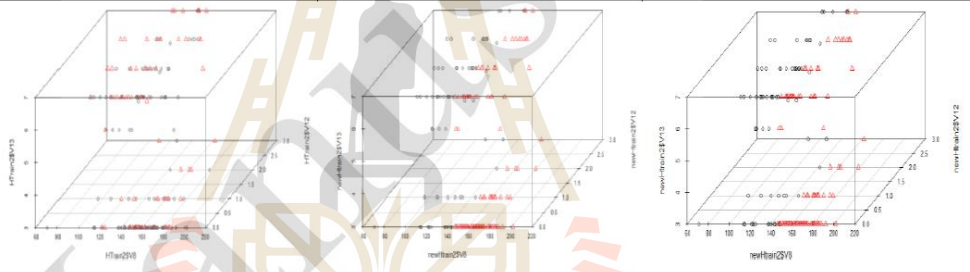
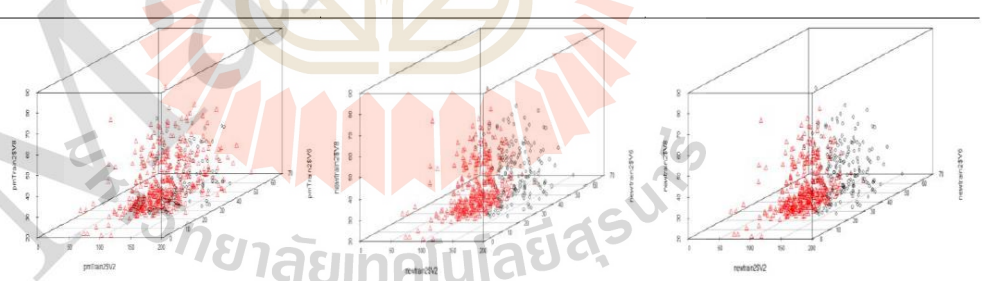
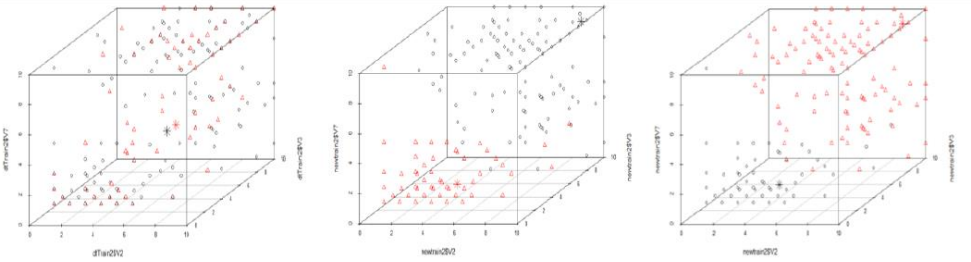
where $m_{ji} = |T_{ji}|$. It measures the fraction of point in partition T_{ji} shared in common with cluster C_i. Therefore, the f-measure for cluster C_i is given as.

$$F_i = \frac{2}{\frac{1}{prec_i} + \frac{1}{recall_i}} = \frac{2 \cdot prec_i \cdot recall_i}{prec_i + recall_i} = \frac{2\, n_{ij_i}}{n_i + m_{j_i}} \quad (7)$$

The f-measure for the clustering C is the mean of cluster-wise f-measure values.

$$F = \frac{1}{r} \sum_{i=1}^{r} F_i \quad (8)$$

Table 2 Visualization of k-means clustering in comparison with original data and feature selection (random forest, correlation).

Data set	k-means clustering		
	Original data	Random forest	Correlation
Thyroid disease			
Heart disease			
Pima Indians diabetes			
Breast cancer Wisconsin			

The experiment we evaluate the performance of the data clustering by compare results showed that the clustering via the feature selection present the purify value and f-measure value are better than original data. As shown in Table 3.

Table 3 Performance evaluation of k-means clustering in comparison with original data and feature selection (random forest, correlation)

Data set	k-Means clustering		
Evaluate	Original data	Random forest :% increment	Correlation :% increment
Thyroid-disease			
#feature	21	7	4
Purity	0.929	0.929 : 0.0%	0.929 : 0.0%
F-measure	**0.464**	**0.474 : 2.3%**	0.449 : -3.2%
Heart disease			
#feature	13	8	9
Purity	0.581	**0.716 : 23.3%**	**0.716 : 23.3%**
F-measure	0.579	**0.712 : 22.9%**	**0.712 : 22.9%**
Pima Indians diabetes			
#feature	8	5	4
Purity	0.628	**0.756 : 20.41%**	**0.756 : 20.41%**
F-measure	0.562	**0.740 : 31.66%**	**0.740 : 31.66%**
Breast cancer Wisconsin			
#feature	10	8	8
Purity	0.659	**0.967 : 46.8%**	**0.967 : 46.8%**
F-measure	0.539	**0.963 : 78.7%**	**0.963 : 78.7%**

CONCLUSIONS

The data clustering with not suitable data such as distribution is non-gaussianity, each feature are irrelevant, or the scale of each value inside feature is very different etc, led to low efficiency clustering. This study, we propose to improve the efficiency of the data clustering via a principle of feature selection technique as random forest and correlation analysis. The experiments with real data results show that the performance of k-means clustering given improving purity value and f-measure for all data sets. In addition, also found that if you want to use clustering method with imbalance data then the result of data clustering will performance increase only slightly.

REFERENCES

Ai-li, F., Qi-sheng, G., Si-ying, Z. (2009). Clustering and correlation analysis of the industry networks. System engineering–theory & practice. Vol.29(6). Online english edition of the Chinese language journal.

Breiman, L. (2001). Random forests. Machine learning. Vol.45(1), pp.5–32.

Correlation coefficient. (2015). Wikipedia the free encyclopedia, Retrieved May 8, 2015, from http://en.wikipedia.org/wiki/Correlation_coefficient.

Kwedlo, W. (2011). A clustering method combining differential evolution with the k-means algorithm. Pattern recognition letters. Vol.32, pp.1613–1621.

Lee, S.L.A., Kouzani, A.Z., Hu, E.J. (2010). Random forest based lung nodule classification aided by clustering. Computerized medical imaging and graphics. Vol.34, pp.535–542l.

Mohammed J. Zaki, Wagner Meira Jr. (2014). Data Mining and Analysis: Fundamental Concepts and Algorithms. Cambridge university press, USA.

Ryali, S., Chen, T., Padmanabhan, A., Cai, W. and Menon, V. (2015). Development and validation of consensus clustering-based framework for brain segmentation using resting fMRI. Journal of neuroscience methods. Vol.240, pp.128–140.

Shahbaba, M., Beheshti, S. (2014). MACE-means clustering. Singnal processing. Vol.105, pp.216-225.

Saha, I., Maulik, U. (2014). Incremental learning based multiobjective fuzzy clustering for categorical data. Information sciences. Vol.267, pp.35–57.

Theodoridis, S., Pikrakis, A., Koutroumbas, K., Cavouras, D. (2010). An introduction to pattern reccognition: A MATLAB approach. academic press, USA.

Tosi, S., Casolari, S., Colajanni, M. (2013). Data clustering based on correlation analysis applied to highly variable domains. Computer networks. Vol.57, pp.3025–3038.

Yu, Z., Li, L., Gao, Y., You, J., Liu, J., Wong, H., Han, G. (2014). Hybrid clustering solution selection strategy. Pattern recognition. Vol.47, pp.3362–3375.

Tippaya Thinsungnoen received the B.E. (1999) degrees in computer science from Nakhon Ratchasima Rajabhat Institute, M.E. (2007) degrees in computer engineering from Suranaree University of Technology. She is a Lecturer, Informatics program, Faculty of science and technology, Nakhon Ratchasima Rajabhat University. Her research of interest includes Knowledge Discovery in Databases, k-means clustering.

Nittaya Kerdprasop is an associate professor at the School of Computer Engineering, Suranaree University of Technology, Thailand.

She received her bachelor degree in Radiation Techniques from Mahidol University, Thailand, in 1985, master degree in Computer Science from the Prince of Songkla University, Thailand, in 1991 and doctoral degree in Computer Science from Nova Southeastern University, U.S.A, in 1999. She is a member of ACM and IEEE Computer Society. Her research of interest includes Knowledge Discovery in Databases, Artificial Intelligence, Logic Programming, and Intelligent Databases.

Kittisak Kerdprasop is an associate professor and chair of the School of Computer Engineering, Suranaree University of Technology, Thailand.

He received his bachelor degree in Mathematics from Srinakarinwirot University, Thailand, in 1986, master degree in Computer Science from the Prince of Songkla University, Thailand,in 1991 and doctoral degree in Computer Science from Nova Southeastern University, U.S.A., in 1999. His current research includes Data mining, Artificial Intelligence, Functional and Logic Programming Languages, Computational Statistics.

Discretization Based on Chi2 Algorithm and Visualize Technique for Association Rule Mining

Nuntawut Kaoungku*, Tippaya Thinsungnoen, Pongsakorn Durongdumrongchai,
Kittisak Kerdprasop, Nittaya Kerdprasop
School of Computer Engineering, Institute of Engineering, Suranaree University of Technology, Thailand.

*Corresponding Author: b5111299@gmail.com

Abstract

This research aims at studying the discretization based on Chi2 algorithm and visualize technique for association rule mining. Numeric attributes with large distinct values normally do not appear in the association rules. We thus study the discretization method for numeric attributes with the integrated Chi2 algorithm and visualize technique to handle numeric attributes prior to the association analysis phase. We comparatively experiment with our proposed method against existing techniques. The comparative metrics are accuracy and number of rules.

Keywords: Discretization, Association Rule Mining, Chi2, Visualize.

1. Introduction

Currently, many organizations and merchants do not use paper to record data, but they use computers for recording. When the data are in the digital form, we can use computer to analyze these data in order to obtain the model for future use such as to predict patient's disease, to understand customer purchase behavior, or to assess learning behavior of student. The automatic induction of model from electronic data is known as data mining [(1)].

Association rule mining is one well-known technique in data mining. It is the induction of relationships of events or objects and generate these relationship as association rules to understand the current data or to predict the occurrence of an event or object in the future. There are many researches about the increase in efficiency in association rule mining, such as increase the speed or the accuracy of the association rule mining.

The data currently available are in a variety of types, such as numeric, text or character. But, the result of relationship induction from the numerical data in association rule mining was not good enough because the numerical data have a wide range of values. Thus, the solution of numerical data handling for association rule mining is discretization technique. There exist have many techniques for discretizing numerical data[(2)], such as Chi2 algorithm and Extend-Chi2 algorithm[(3)].

This research aims at proposing the efficient discretization technique based on Chi2 algorithm and visualized cut-point analysis for association rule mining to handle numerical data. The problem caused by the numerical data in association rule mining is that they make association rules disappear due to the sparseness of each numeric value. We, therefore, propose an efficient algorithm to solve this problem.

2. Related Work

Related researches can be divided into several types: research for proposing the new idea, research to improve the original algorithm, research to discretize numerical data for classification, and research to discretize numerical data for association rule mining. Thus, we study related work along these research themes with the details as follows:

Gyenesei[(4)] has proposed an algorithm to discretize numerical data by using fuzzy sets technique to reduce runtime and to increase number of effective rules in the association rule mining. The experimental result has been compared between discretization by non-normal distributed data and normal distributed data. The result of the discretization by normal distributed data gives more effective association rules than non-normal distributed data,

which is measured by the minimum support, minimum confidence and runtime in association rule mining.

Tong et al.[5] has proposed a method for association rule mining with numerical data using k-means clustering algorithm and Euclidean-based distance calculation. They used synthetic data to test the performance of the algorithm. Their algorithm results in less number of association rules, but higher in the values of support and confidence compared to association rule mining without any handling method for numerical data.

Ke et al.[6] has proposed a method of the association rule mining with numerical data using mutual information and clique (MIC). This technique has divided the work into 3 parts. First, applying discretization to numerical data. Second, using the data obtained from the first step to create the MI graph. And finally, the result of the second step was used in the finding of frequent itemsets. The experimental result used 6 datasets and compared the runtime, number of rules and other measures.

Wei[7] has proposed discretization technique to handle numerical data for association rule mining with clustering and genetic algorithm. His proposed technique is multivariate discretization based on density-based clustering and genetic algorithm (MVD-CG), which is an algorithm that improves the multivariate discretization algorithm (MVD). The experiment used real data and compared between MVD-CG algorithm and MVD algorithm. Measurement metrics are number of rules and effective of rules. The result is that MVD-CG algorithm has higher confident than MVD algorithm.

Sug[8] has proposed multi-dimensional association rule mining, which is different from original association rule mining in term of the difference of column. This method can reduce the data size and runtime in association rule mining. The experiment used real data from UCI. The result is that the algorithm can create small multi-dimensional table and reduce the number of rules.

From the related work, it can be seen that there exist many techniques for discretization. However, most researches do not take into consideration the distribution of the data in each rang of the divided value. We notice that it may be possible that some of the data that was in the range with very small amount might be unnecessary to be used in the association rule mining. We thus propose the visualize technique to detect ranges of values with limited amount of data in order to consider a cut point during discretization process.

3. Background

This research aims at studying the discretization and proposing a new method based on Chi2 algorithm and visualize technique for association rule mining. The related theories are divided into 4 parts, that is, association rule mining, discretization algorithms, the cut points in Chi2 algorithm, and visualization by cut points.

3.1 Association Rule Mining

Association rule mining is a popular analysis technique to automatically find the relationships between the data. There are many methods for association rule mining. In this papers we use Apriori[9] algorithm for association rule mining. In the table 1 is a list of customer purchases that will be used as an example to explain the association rule mining process. The data are counted to find the frequent customer purchases on each itemset, and then take the frequent itemsets to generate the association rules, which is a rule in the form of "If condition Then result". The measurement metrics used in the selection of frequent itemsets and rules are the following:

- Support is the frequency of the occurring event. Give the items A and B, the computation for support of A and B to be purchased is as follows:

$$Support(A \rightarrow B) = P(A \wedge B) \quad (1)$$

As an example, support (Coca cola $\rightarrow$ Bread) = 2/5 = 0.4 or 40%.

- Confidence is the frequency of the incident with other events occurring together. The computation for confident is as follows:

$$Confidence(A \rightarrow B) = \frac{Support(A \rightarrow B)}{Support(A)} \quad (2)$$

For the same example as above, confidence (Coca cola $\rightarrow$ Bread) = 0.4/0.4 = 1.0 or 100%.

Table 1. Purchase transactions of customers.

Order	Coca cola	Bread	Candy	Milk
1	1	1	0	0
2	0	1	1	0
3	0	0	0	1
4	1	1	1	0
5	0	1	0	0

Table 2. Contingency table for cut point computing demonstration.

	Class 1	Class 2	Sum
Interval 1	A_{11}	A_{12}	R_1
Interval 2	A_{21}	A_{22}	R_2
Sum	C_1	C_2	N

Sample	K=1	K=2	
2	0	1	1
3	1	0	1
total	1	1	2

$\mathbf{E_{11}} = (1/2)*1 = .05$
$\mathbf{E_{12}} = (1/2)*1 = .05$
$\mathbf{E_{21}} = (1/2)*1 = .05$
$\mathbf{E_{22}} = (1/2)*1 = .05$

$\mathbf{X^2} = (0\text{-}.5)^2/.5 + (1\text{-}.5)^2/.5 + (1\text{-}.5)^2/.5 + (0\text{-}.5)^2/.5 = \mathbf{2}$

Sample	K=1	K=2	
3	1	0	1
4	1	0	1
total	2	0	2

$\mathbf{E_{11}} = (1/2)*2 = 1$
$\mathbf{E_{12}} = (0/2)*2 = 0$
$\mathbf{E_{21}} = (1/2)*2 = 1$
$\mathbf{E_{22}} = (0/2)*2 = 0$

$\mathbf{X^2} = (1\text{-}1)^2/1 + (0\text{-}0)^2/0 + (1\text{-}1)^2/1 + (0\text{-}0)^2/0 = \mathbf{0}$

Fig. 1. Example of calculation the Chi2 cut point between intervals.

3.2 Discretization algorithms

(a) Chi2 algorithm

Chi2 algorithm[10] that is based on the X^2 statistics was used to perform discretization over the numerical data. The computation for x^2 is as follows:

$$X^2 = \sum_{i=1}^{x}\sum_{j=1}^{k}\frac{\left(A_{ij} - E_{ij}\right)^2}{E_{ij}} \quad (3)$$

where:

k = number of classes,

A_{ij} = number of patterns in the ith interval, jth class,

Eij = expected frequency of $A_{ij} = R_i * C_j / N$,

R_i = number of patterns in the ith interval $= \sum_{j=1}^{k} A_{ij}$,

C_j = number of patterns in the jth class $= \sum_{i=1}^{2} A_{ij}$,

N = total number of patterns $= \sum_{i=1}^{2} R_i$

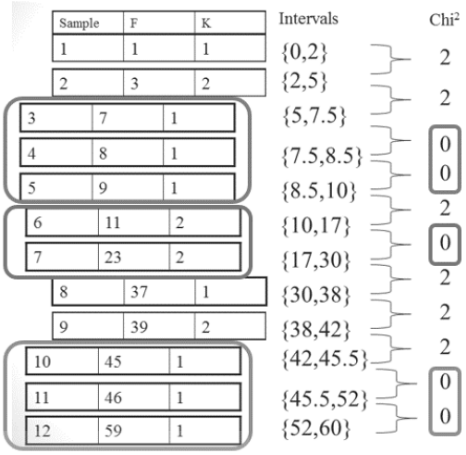


Fig. 2. Example of interval integration by Chi2 considering value.

(b) CAIM algorithm

Class-Attribute Interdependence Maximization (CAIM)[11] is a discretization algorithm by supervised learning. Main idea of the CAIM algorithm is to use class-attributed interdependence of the class and the numeric column to create minimal interval. The computation for CIAM is as follows:

$$CAIM(C, D \mid F) = \frac{\sum_{r=1}^{n}\frac{\max_r^2}{M_{+r}}}{n} \quad (4)$$

where:

C = class,
D = discretization,
F = columns,
n = number of intervals,
$\max_r$ = maximum of the q_{ir},
M_{+r} = number of all continuous columns

(c) CACC algorithm

Class-Attribute Contingency Coefficient (CACC)[12] is a discretization algorithm developed from CAIM algorithm for solving the overfitting problem. The computation for CACC is as follows:

$$CACC = \sqrt{\frac{y'}{y'+M'}} \quad (5)$$

where:

M' = number of sampling data

$$y' = M\left[\left(\sum_{i=1}^{S}\sum_{r=1}^{n}\frac{q_{ir}^2}{M_{i+}M_{+r}}\right)-1\right]/\log(n)$$

when

M = number of sampling data,

n = number of data,

q_{ir} = number of class i by sampling data (i=1,2,...,S and r=1,2,...,n) in the interval $(d_{r-1},d_r]$,

M_{i+} = number of class i by sampling data

(d) AMEVA algorithm

AMEVA algorithm[13] is discretization algorithm by supervised learning. This algorithm increase performance in terms of correlations between parameter and decrease number of intervals of the Chi2 algorithm. The computation for AMEVA is as follows:

$$Ameva(k) = \frac{x^2(k)}{k(l-1)} \quad (6)$$

where:

k = number of intervals,

$$x^2(k) = N\left(-1+\sum_{i=1}^{l}\sum_{j=1}^{k}\frac{n_{ij}^2}{n_i n_j}\right)$$

3.3 Cut point in Chi2 Algorithm

Discretization by Chi2 algorithm is to find the cut points to divide the numerical data to interval data. The algorithm is based on the bottom-up division of numerical data in each row into intervals, and then gradually merging each interval based on independence, which can be computed by Chi2 value. We can be demonstrate the cut point calculation from contingency table in table 2 and equation (3). Figure 1 shows an example of calculating the Chi2 in each interval, such as intervals 2, 3 and 4 has Chi2 0. Figure 2 shows an example of integration the intervals by considering minimal Chi2 value, because the minimal Chi2 value means less independent between intervals. If the intervals are less independent, they should be in the same interval. For instance, the intervals 3, 4 and 5 have the Chi2 values 0. They should be in the same interval. Repeat the interval merging until Chi2 values of all intervals are greater than threshold that the user has defined.

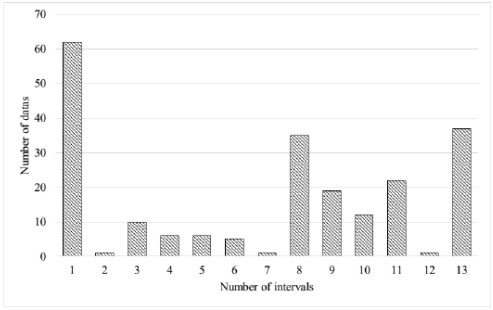


Fig. 3. Example of comparing the number of rules in each interval.

3.4 Visualization Cut Point Consideration Through

Discretization numerical data by various algorithms has to consider the cut points, which can be used for grouping the numerical data to intervals. But in some situation the data in the discretized interval has too small amount of data. This can lead to an inefficient association rule mining. Thus, we propose to use visualization technique[14] in the post-processing of the discretization steps to see the distribution of data in each interval, and then adjust the cut points to fit the data distribution in each interval. Figure 3 shows an example chart comparing the number of data in each interval. It can be seen that the data in the intervals 1, 7 and 12 are minimal comparing to other intervals. The cut points should be adjusted to re-distribute data in the discretized intervals. Figure 4 example of interval integration by visualize technique.

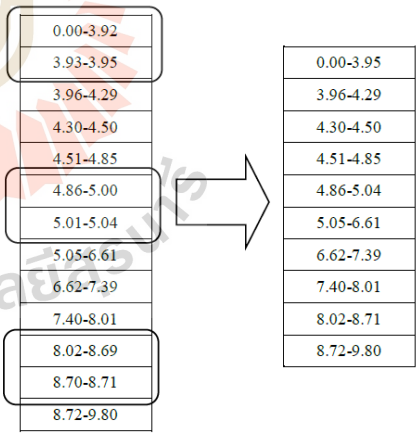


Fig. 4. Example of interval integration by visualize technique

Table 3. Comparative results of accuracy and number of rules by five discretization algorithms.

Datasets / Algorithms	Simulated Data		ECOLI		APD		CLE	
	#Rules	Acc.	#Rules	Acc.	#Rules	Acc.	#Rules	Acc.
AMEVA	81	88.19	107	97.57	987	95.29	111	45.31
CAIM	79	88.39	52	97.47	971	94.87	43	83.13
CACC	81	88.19	107	97.57	987	95.21	75	65.99
Chi2	41	88.17	47	97.67	40	81.85	239	81.37
Chi2+Visualize	37	95.41	47	97.68	36	87.08	220	81.58

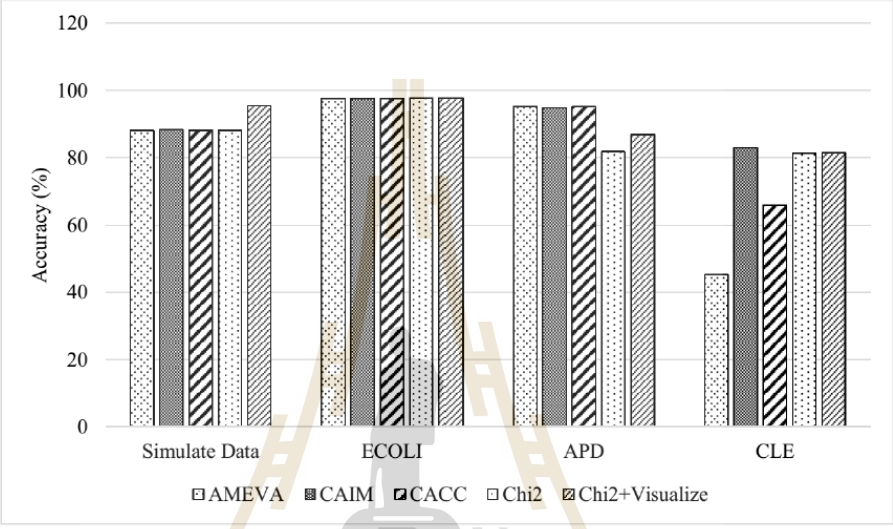


Fig. 5. The accuracy comparison of the five discretization algorithms.

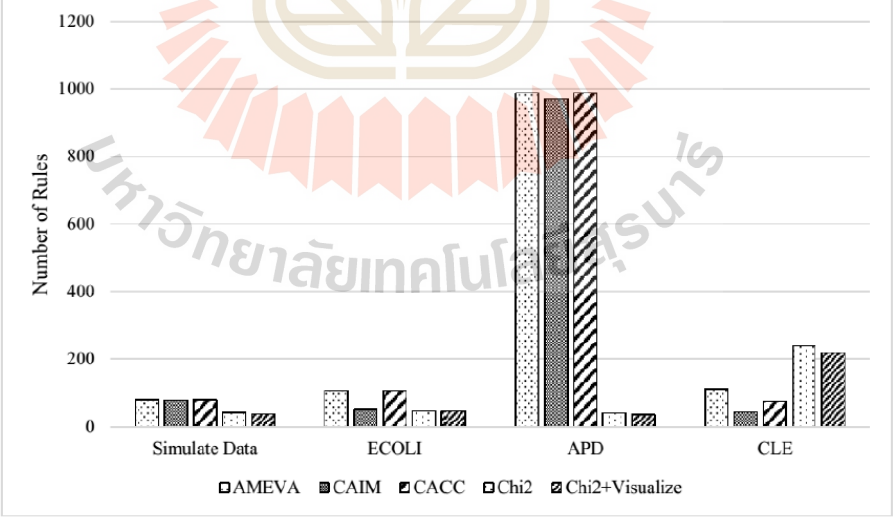


Fig. 6. The number of rules comparison of the five discretization algorithms.

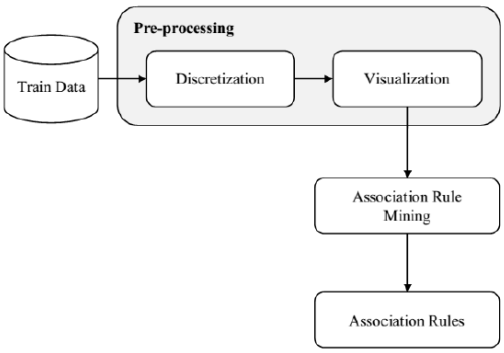


Fig. 7. Conceptual framework of the research.

4. Proposed Discretization Method

This research has proposed a methodology to perform discretization based on the Chi2 algorithm and the visualize technique for association rule mining. Figure 7 sketches the proposed method to discretize numerical attributed for association rule mining. The pre-processing module can be divided into two parts: discretization step and visualization by cut point consideration step. Discretization in our method is based on the Chi2 algorithm because it is easy to understand. After the discretization step to visualization has been applied to see distribution of data in each interval. If amount of data in any interval is too small, that interval will be merged with the previous interval or the next interval, which is considered by the distance to the nearest interval. Finally, the result is data discretized by new cut points. These data will be further used in the association rule mining

5. Experimental Setting and Results

The proposed discretization method has been experimented with both synthetic data and real data from the UCI Machine Learning Repository. The UCI data are Appendicitis data (APD) with 106 records and 7 attributes. Ecoli data with 336 records and 8 attributes. Cleveland data (CLE) with 303 records and 13 attributes. Each of these datasets hasbeen divided into training dataset (70%) and test dataset (30%). The discretization algorithm is encoded with the R language[15]. The algorithm proposed in this research is called Chi2+Visualize discretization algorithm. Its performance has been compared with the AMEVA, CAIM, CACC, and Chi2 algorithms. The performance metrics are number of rules and accuracy of the algorithms.

Table 3 shows the number of rules and accuracy of algorithms with different datasets. It can be seen that the Chi2+Visualize algorithm with the synthetic data and Ecoli data shows higher accuracy with less number of rules when compared to other algorithms. But the performance of our algorithm on the CLE and APD data shows lower accuracy than some algorithms.

Figure 5 shows a chart comparing the accuracy of algorithms with different datasets. It can be seen that the Chi2+Visualize with synthetic data and Ecoli data show higher accuracy when compared to other algorithms. Figure 6 shows a chart comparing the number of rules with different datasets. It can be seen that the Chi2+Visualize with synthetic, Ecoli and APD data has less number of rules when compared with other algorithms.

6. Conclusions

This research aims at studying the discretization method based on Chi2 algorithm and visualize technique for association rule mining. The problem of association rule mining with numerical data is that there will be large number of rules and the obtained association rules are not effective enough to predict the future data. Thus, we propose to use the cut point from discretization by Chi2 algorithm to see the distributed data in each interval, and then adjust the cut points to fit the distribution in each interval. The experimental results reveal that the proposed algorithm can reduce the number of rules and increase accuracy in predicting the future data. However, the application of our method over some data show low accuracy, but can be traded-off by small number of rules. But in some data our method shows low accuracy and large number of rules. We hypothesize that this dataset may be non-normal distribution and this kind of distribution has strong effect to our method. However, this hypothesis needs theoretical and experimental proofs further.

References

(1) Berry, Michael J., and Gordon Linoff. : “Data mining techniques: for marketing, sales, and customer support.”, John Wiley & Sons, Inc., 1997.

(2) Liu, Huan, et al. : “Discretization: An enabling technique.”, Data mining and knowledge discovery, vol. 6, No. 4, pp. 393-423, 2002

(3) Su, Chao-Ton, and Jyh-Hwa Hsu. : "An extended chi2 algorithm for discretization of real value attributes.", Knowledge and Data Engineering, IEEE Transactions on, vol. 17, No. 3, pp. 437-441, 2005

(4) Gyenesei, Attila : "A Fuzzy Approach for Mining Quantitative Association Rules.", Acta Cybern, Vol. 15, No. 2, pp. 305-320, 2001

(5) Tong, Qiang, et al. : "A method for mining quantitative association rules.", Jisuanji Gongcheng/ Computer Engineering, vol. 33, No. 10, pp. 34-35, 2007

(6) Ke, Yiping, James Cheng, and Wilfred Ng. : "MIC framework: an information-theoretic approach to quantitative association rule mining.", Data Engineering, 2006. ICDE'06. Proceedings of the 22nd International Conference on. IEEE, pp. 112-112, 2006.

(7) Wei, Hantian. : "A novel multivariate discretization method for mining association rules.", Information Processing, 2009. APCIP 2009. Asia-Pacific Conference on, Vol. 1, pp. 378-381, 2009

(8) Sug, Hyontai. : "Discovery of multidimensional association rules focusing on instances in specific class.", International Journal of mathematics and Computers in Simulation, vol. 5, No. 3, pp. 250-257, 2011

(9) Agrawal, Rakesh, Tomasz Imieliński, and Arun Swami : "Mining association rules between sets of items in large databases.", ACM SIGMOD Record, Vol. 22, No. 2, ACM, 1993.

(10) Liu, Huan, and Rudy Setiono. : "Chi2: Feature selection and discretization of numeric attributes.", 2012 IEEE 24th International Conference on Tools with Artificial Intelligence. IEEE Computer Society, pp. 388-388, 1995

(11) Kurgan, Lukasz A., and Krzysztof J. Cios. : "CAIM discretization algorithm.", Knowledge and Data Engineering, IEEE Transactions on, vol. 16, No. 2, pp. 145-153, 2004

(12) Tsai, Cheng-Jung, Chien-I. Lee, and Wei-Pang Yang. : "A discretization algorithm based on class-attribute contingency coefficient.", Information Sciences, vol. 178, No. 3, pp. 714-731, 2008

(13) Gonzalez-Abril, L., et al. : "Ameva: An autonomous discretization algorithm.", Expert Systems with Applications, Vol. 36, No. 3, pp. 5327-5332, 2009

(14) Keim, Daniel A. : "Information visualization and visual data mining.", Visualization and Computer Graphics, IEEE Transactions on, vol. 8, No. 1, pp. 1-8, 2002

(15) Ihaka, Ross, and Robert Gentleman : "R: a language for data analysis and graphics.", Journal of computational and graphical statistics, Vol. 5, No. 3, pp. 299-314, 1996

A Deep Learning of Time Series for Efficient Analysis

Tippaya Thinsungnoen, Kittisak Kerdprasop, and Nittaya Kerdprasop

***Abstract*—The main problem for analyzing time series data with machine learning techniques such as classification and clustering is that a high-dimensional nature of this kind of data can cause computational difficulty in finding optimal solution. Currently, advanced learning strategy such as deep learning has been used extensively and effectively to improve learning performance. In this research, we propose a method to optimize time series analysis by adding a pre-training and fine-tuning process of deep learning based on Deep Belief Networks and Restricted Boltzmann Machines. On evaluating performance of the proposed method, we use electroencephalographic, electrocardiogram, and synthetic time series data to analyze with classification task. The induced classification models are assessed with the four several metrics including cluster evaluation, purity, mean squared error, and processing time. We comparatively compare the three learning schemes: traditional neural networks, deep learning networks, and deep learning networks with added a pre-training and fine-tuning process. The results showed that all three schemes show the same performance on predicting time series data when assessed with mean squared error. For the processing time comparison, neural networks technique is slightly faster than others. But when assessed with cluster formation and purity metrics, we found that deep learning based on the concept of Deep Belief Networks and Restricted Boltzmann Machines that adds a pre-training and fine-tuning process outperforms other learning techniques.**

***Index Terms*—Deep belief networks, deep learning, restricted Boltzmann machines, time series analysis.**

I. Introduction

Time series data are a set of values that had been stored consecutively in terms of occurring time and kept for a long period. Data in many applications are being stored as time series such as sales records, stock prices, weather data, and biomedical measurements [1]. An efficient analysis of time series data is a challenging problem in machine learning. The difficulty is due to the fact that time-series data is a type of temporal data, which is naturally high dimensional and large in data size. Researchers have tried to improve efficiency by reducing dimensionality to low-dimensions or introducing new representation method for time-series. The proposed time series representation appeared in the literature includes a symbolic [2], [3] and a grid [4] representation of time series for similarity search.

Besides representation, efficient learning techniques that are appropriate for time series analysis are also in the main focus of my research teams. One potential learning technique is deep learning, which is a complex learning architectural style composing of sub-layers, where each layer contains multiple linear and non-linear transformations attempting to model high level abstractions in data [5]. Researchers offer several learning architectures based on principles of a deep learning such as deep neural networks and deep belief networks. These learning styles have been successfully applied in computer vision, speech recognition, natural language processing, bioinformatics, and others. In the research of [6], they proposed a fast learning algorithm for deep belief nets. In the work of [7], they proposed to reduce the dimensionality of data with neural networks based on the deep autoencoder method. In the same year, the research team [8] proposed a novel and efficient algorithm to sparse feature learning for deep belief networks to capture high-order dependencies between the input observed variables. Furthermore, in the work of [9], they proposed learning algorithm called Deep Boltzmann Machines (DBM's) such that efficiency can be achieved through the pre-training process. Lately, in research work of [10], they also applied pre-training to the ECG assessment based on Restricted Boltzmann Machines (RBM's).

The works mentioned above are a small review of deep learning applied for efficient data analysis, with a particularly interest in high-dimensional data forming themselves as sequences and important knowledge is hidden in these series. This paper thus proposes a study of deep learning of time series for efficient analysis by using a Deep Architecture and Restricted Boltzmann Machines. The data domains include electroencephalographic (EEG), electrocardiogram (ECG), and synthetic time series data. Learning efficiency is evaluated by comparing the results of classification between the three solutions of learning. We use four evaluation measures, that are, cluster evaluation to assess group formation of data, purity in each data group, mean squared error for series prediction, and time used for processing.

II. Background

A. Artificial Neural Network

Artificial neural networks (ANN) are a computational approach, based on a large collection of neural units for information processing with connectionist theory. This machine learning approach simulates the pattern recognition function of neural networks in human brain. The bioelectric network in the human brain consists of neurons and synapses, and the interoperation to connect neurons [11]–[13]. Basically, the neural network consists of three layers including input layers, hidden layer and output layer.

Manuscript received February 15, 2017; revised April 7, 2017. This work was supported in part by grants from Suranaree University of Technology and Nakhon Ratchasima Rajabhat University, Thailand.

The authors are with the School of Computer Engineering, Suranaree University of Technology (SUT), 111 University Avenue, Muang, Nakhon Ratchasima 30000, Thailand (corresponding author: T. Thinsungnoen; Tel.: +66819671907; e-mail: tippayasot@hotmail.com, kerdpras@sut.ac.th, nittaya@sut.ac.th).

B. Deep Learning

Deep learning (or deep structured learning, hierarchical learning, deep machine learning) is an advanced learning technique relying on a set of algorithms that attempt to model high-level abstractions in data. Deep learning is part of a broader family of machine learning methods based on learning representations of data. The concepts include the processing among multiple layers such that each layer is derived from both the linear and nonlinear conversions [5], [13], [14].

C. Deep Belief Networks

Deep Belief Networks (DBNs) is a kind of networks that use probability in a modeling process including multiple layers of stochastic and latent variables. The latent variables typically have binary values called hidden units or feature detectors. The top two layers are undirected graph between itself and memory. The layer below have been directed by the above layer, as shown on the left network of Fig. 1. This network concept has been used as a simple element of unsupervised learning such as RBMs and autoencoder [15].

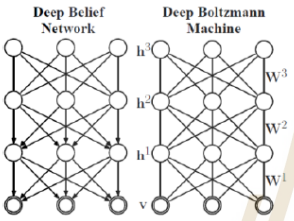


Fig. 1. Model of DBNs and DBMs [9].

D. Deep Boltzmann Machines

A *Boltzmann Machine* is a network of symmetrically coupled stochastic binary units [14]. It consists of a set of visible units $v \in \{0, 1\}$ and a set of hidden units $h \in \{0, 1\}$, as shown on the right of Fig. 1 [9]. The energy of the state $\{v, h\}$ can be defined as:

$$E(v, h; \theta) = -\frac{1}{2} v^T L v - \frac{1}{2} h^T J h - v^T W h, \quad (1)$$

where $\theta = \{W, L, J\}$ is a set of the model parameters: W, L and J represent visible-hidden, visible-visible, and hidden-hidden, respectively. The diagonal elements of L and J are set to 0.

Deep *Boltzmann Machines* (DBMs) are generally used for learning a deep multilayer Boltzmann Machine [9]. Consider a two-layer Boltzmann Machine with no within-layer connections, the energy of the state $\{v, h^1, h^2\}$ is defined as [9]:

$$E(v, h^1, h^2; \theta) = -v^T W^1 h^1 - h^{1T} W^2 h^2, \quad (2)$$

where $\theta = \{W^1, W^2\}$ is a pair of model parameters representing visible-hidden and hidden-hidden symmetric interaction terms, respectively.

E. Restricted Boltzmann Machines

A general Boltzmann Machines consist of the top layer to represent a hidden connection and the bottom layer to represent a visible connection, as shown in Fig. 2 (left). But RBMs is a network of symmetrically coupled stochastic binary units, with no hidden-to-hidden and no visible-to-visible connections [9], [16]-[18], as shown in Fig. 2 (right).

RBMs are energy-based probabilistic model. In these models a probability distribution is defined from energy function as [10]:

$$P(x, h) = \frac{e^{-Energy(x,h)}}{Z}, \quad (3)$$

where x is a set of input variables, and h corresponds to the hidden variables introduced to increase the expressive power of the model. The normalization factor Z is called the partition function defined as [10]:

$$Z = \sum_{x,h} e^{-Energy(x,h)}, \quad (4)$$

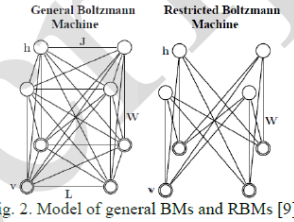


Fig. 2. Model of general BMs and RBMs [9].

F. Performance Evaluation

There are various measures for evaluating performance if we know previous label of data [18]. This research, we use the three measures: Cluster Evaluation, Purity, and Mean Squared Error (MSE).

Cluster Evaluation is often used for clustering evaluation criteria based on the known label. It is a calculation of an index that measures the amount of agreement between the true cluster partition $G = \{G_1, \ldots, G_k\}$ (the "ground-truth"), and the experimental cluster solution $A = \{A_1, \ldots, A_k\}$ obtained by a machine learning method. The similarity index Sim (G, A) is defined by [19]:

$$Sim(g, A) = \frac{1}{k} \sum_{i=1}^{k} \max_{1 \le j \le k} (G_i, A_j) \quad (5)$$

$$Sim(g, A) = \frac{|G_i \cap A_j|}{|G_i| + |A_j|} \quad (6)$$

where $|\bullet|$ denote the member of the set.

Purity is an assessment of how members are organized into class or cluster with the innocence of a member of the cluster. The Purity for members of cluster Ci is defined as follows [18]:

$$purity_i = \frac{1}{n_i} \max_{j=i}^{k} \{n_{ij}\} \quad (7)$$

where i is an order of clusters from i to r,

j is an order of targets from j to k,

n_i is a member of cluster i,

n_{ij} is a member of cluster i that belongs to class j.

Thus, purity of clustering or classification can be defined as:

$$purity = \sum_{i=1}^{r} \frac{n_i}{n} purity_i = \frac{1}{n_i} \max_{j=i}^{k} \{n_{ij}\} \quad (8)$$

Mean *Squared Error* (MSE) is the accuracy index for neural networks model and can be defined as follows [19]:

$$MSE = \sum_{t=1}^{N} \frac{(E_t)^2}{N} \quad (9)$$

where N is total number of data for prediction, E_t is difference (or error) between actual and predicted values of object t.

III. Materials and Methodology

A. Dataset Description

The datasets used in this study consist of EEG signals, ECG signals, and Time Series Synthetics. Details of datasets are summarized in Table I.

TABLE I: The Description of Dataset

Dataset	Time Series Length	#Series	#Class
EEGs	4,096	20	2
ECGs	96	200	2
Synthetic	200	18	6

Electroencephalographic (EEG) data, taken from [20], [21], consist of five sets: A to E. All EEG signals were recorded with 128-channel amplifier system, using an average common reference. Sets A and B are data sensed from volunteers in a relaxing and awake state with eyes-open (A) and eyes-closed (B), Sets C, D, and E are data originated from EEG archive of presurgical diagnosis. In this research, we use only sets A and B of EEGs data comprising time series of 4,096 samples each, and 10 instances of each class.

Electrocardiogram (ECG) are data taken from [22], [23]. This dataset contains measurements of cardiac electrical activity as recorded from electrodes at various locations on the body; each data set in the ECG recorded by one electrode during one heartbeat. The dataset in each database were analyzed by domain experts; from 200 data records, 133 records were identified as normal and 67 were identified as abnormal.

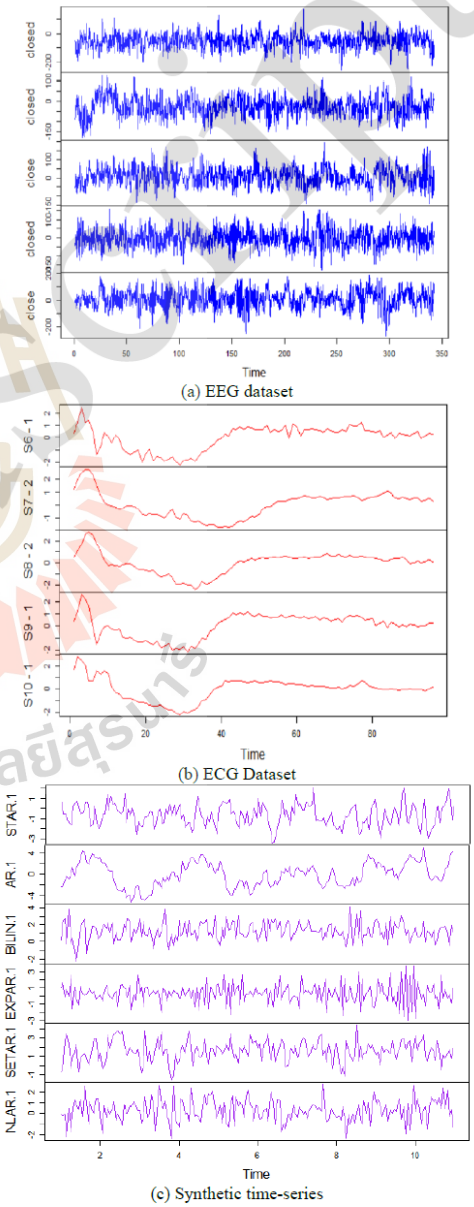
The Synthetic of Time series (Synthetic.tseries) follows the work of [24] by synthesizing data from six different models; three models were set with both linear and non-linear distributions. Each profile model is shown in Table II.

Preview time series of EEG and ECG data, as well as the synthetic data, are shown in Fig. 3 (a), (b), (c), respectively.

In Fig. 3 (a), the preview of EEG signals shows two classes of the whole signal that were very similar. Fig. 3(b) shows previews ECG signals with two classes, that are, classes normal as the S6, S9 and S10 series, and class abnormal as S7 and S8. It can be noticed at time points 40-45 that the two groups have difference changes on the inverse. Fi 3(c) shows previews of synthesis of time series complicated shapes derived from different distributi models, thus, resulting in the six different groups that can clearly differentiated.

TABLE II: The Description of Six Synthetics [24]

Name	Model
AR	$X_t = 0.6X_{t-1} + \varepsilon_t$
Bilinear	$X_t = (0.3 - 0.2\varepsilon_{t-1})X_{t-1} + 1.0 + \varepsilon_t$
EXPAR	$X_t = (0.9\exp(-X_{t-1}^2) - 0.6)X_{t-1} + 1.0 + \varepsilon_t$
SETAR	$X_t = (0.3X_{t-1} + 1)/(X_{t-1} \geq 0.2) - (0.3X_{t-1} + 1)/(X_{t-1} < 0.2) + \varepsilon_t$
NLAR	$X_t = 0.7\lvert X_{t-1}\rvert(2 + \lceil X_{t-1}\rceil)^{-1} + \varepsilon_t$
STAR	$X_t = 0.8X_{t-1} - 0.8X_{t-1}(1 + \exp(-10X_{t-1}))^{-1} + \varepsilon_t$



(a) EEG dataset

(b) ECG Dataset

(c) Synthetic time-series

Fig. 3. Preview time series dataset for experimental.

B. Methodology

This research aims at studying a deep learning performance when applied to time series analysis. We employ Deep Architecture and RBMs to find out solution for efficient analysis. The framework of our research shows in figure 4. The four main steps can be explained as follows:

Input Raw Time series: This research use the whole raw time series as input to the machines.

Experimental Design: At this step, we divide our experimental method as two sections. Section 1 is the use of Artificial Neural Nets to learn patterns in time series. Section 2 is the application of Deep Architecture and RBMs. The parameter setting of both sections are explained as follows.

Networks consist of three layers. The input layer and a number of neurons equal to the length of time series. Hidden layer contains 10 neurons (which is the best setting from our experiment), and output layer contain one neuron (figure 5).

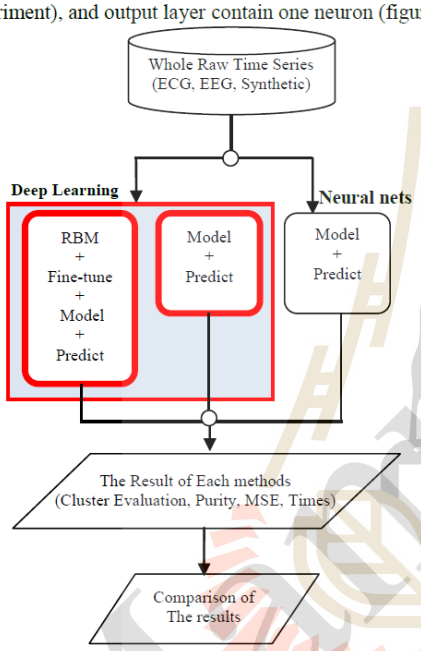


Fig. 4. The framework for this research.

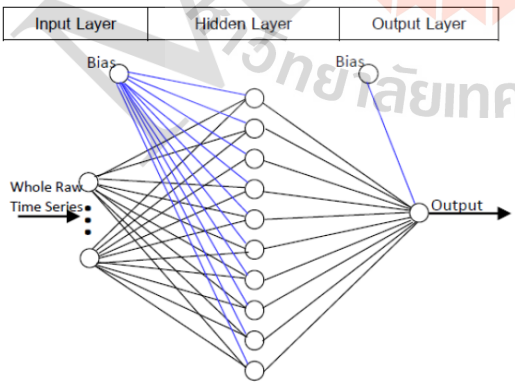


Fig. 5. Model of Neural Networks for this research.

Solutions are based on the setting of 500 epochs for running and stop before the end if small error exists compared to the previous iteration. We design three solutions for analysis. Solution#1 is the analysis with the application of neural nets as a training process for modeling using back propagation. Solution#2 is the use of deep architecture. Solution#3 is a deep learning with pre-training as a startup, then fine-tune with back propagation and sigmoid function, and end the process with prediction step.

Collection the results: after complete all solutions.

Comparison: comparing of cluster evaluation, purity, MSE, and running time between EEG, ECG and Synthetic datasets.

IV. Experimental Results

This research concerns the study of deep learning based on Deep Architecture and RBMs for effectiveness of time series analysis. The summary of the results is shown in table III, and the details of the experimental results are follows.

An Evaluated by MSE, the lower is the better. The results show that in Solution#3, Solution#2 and Solution#1, MSE values for EEG, ECG, and synthetic data from all solutions are almost the same at 0.5, 0.335 and 9.17, respectively.

An Evaluated by process time, the lower is the better. The results show that Solution#1 uses the least processing time, but only slightly less than the others. The running time of EEG data for Solution#3, Solution#2 and Solution#1 are 4.78, 3.61, and 2.80 minutes, respectively. The running time of ECG data for Solution#3, Solution#2 and Solution#1 are 4.95, 3.01, and 1.60 minutes, respectively. For synthetic data, the running time from the three solutions are 2.48, 0.95, and 0.76 minutes, respectively.

An Evaluated by cluster evaluation and purity, the higher is the better. These assessment metrics raise an interesting result. In Solution#3, which is the learning based on DBNs and RBMs with the additional pre-training and fine-tuning procedure, the results show the best performance in almost all datasets. The results of cluster evaluation and purity for EEG dataset are 0.949 and 0.95, and for ECG dataset are 0.65 and 0.7, respectively. For the synthetic data, we found that Solution#1 performs slightly better than others with the cluster evaluation and purity showing the same value at 0.39.

TABLE III: The Comparing of the Results

Dataset	Measures	Solution#1	Solution#2	Solution#3
EEGs	Time (m)	*2.8*	3.611	4.783
	Purity	0.7	0.5	*0.95*
	MSE	*0.5005*	*0.5*	*0.5*
	Cluster Evaluation	0.7	0.67	*0.949*
	Incorrect (%)	6(30%)	10(50%)	*1(5%)*
ECGs	Time (m)	*1.6*	3.01	4.9508
	Purity	*0.7*	*0.7*	*0.7*
	MSE	*0.335*	*0.335*	*0.335*
	Cluster Evaluation	0.55	*0.65*	*0.65*
	Incorrect (%)	88(44%)	*67(33.5%)*	*67(33.5%)*
Synthetic	Time (m)	*0.76*	0.951	2.481
	Purity	*0.39*	0.2	0.2
	MSE	*9.17*	*9.17*	*9.17*
	Cluster Evaluation	*0.39*	0.29	0.29
	Incorrect (%)	*14(77.78%)*	15(83.33%)	15(83.33)

method using Deep Architecture and RBMs for analyzing the time series data. The findings for this research are as follows.

Some of Deep architecture such as DBNs and RBMs can be an appropriate method for analyzing the time series data with high-dimensions and a clear sequence pattern. These methods can solve the problem with satisfied accuracy, effectiveness and speed.

An adaptation, such as pre-training and fine-tuning in RBMs, has been experimentally proven to be efficient supporting steps for deep learning more efficiently with a slightly increase in processing time. The study can also conclude that both ANN and RBMs can be used for time series analysis when MSE is the sole concern. But if cluster quality and purity are also additional criteria, RBMs with pre-training and fine-tuning is a better choice for time series analysis.

B. *Future Work*

Based on the result of synthesis data, created from six different distribution models, we found that ANN performs the best. Deep learning with pre-training and fine-tuning shows remarkably high predictive performance on the EEG dataset. Thus, we plan to extend our research for more studies on the correlation between time series characteristic and deep architectures.

ACKNOWLEDGMENT

The first author has been supported by Nakhon Ratchasima Rajabhat University (NRRU) for Scholarship of Doctoral degree. The second and third authors from Knowledge Engineering Research Units have been funded by Suranaree University of Technology.

REFERENCES

[1] S. Aghabozorgi, A. S. Shirkhorshidi, and T. Y. Wah, "Time-series clustering — A decade review," *Information Systems*, vol. 53, pp. 16-38, April 2015.

[2] J. Lin, E. Keogh, S. Lonardi, and B. Chiu, "A symbolic representation of time series, with implications for streaming algorithms," in *Proc. 8th ACM SIGMOD Workshop on Research Issues Data Mining and Knowledge Discovery – DMKD '03*, 2003, p. 2.

[3] E. Keogh, "Hot sax: Efficiently finding the most unusual time series subsequence," in *Proc. Fifth IEEE International Conference on Data Mining ICDM05*, 2005, pp. 226–233.

[4] G. Duan, Y. Suzuki, and K. Kawagoe, "Grid representation of time series data for similarity search," *The Institute of Electronic, Information, and Communication Engineer*, 2006.

[5] L. Deng and D. Yu, "Deep learning: Methods and applications. foundations and trends," *Signal Processing*, vol. 7, issues 3-4, pp. 197-387, ISSN: 1932-8346, 2014.

[6] G. E. Hinton, S. Osindero, and Y. W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, issues 7, pp. 1527-1554, 2006.

[7] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, issues 5786, pp. 504-507, 2006.

[8] M. A. Ranzato, Y. L. Boureau, and Y. L. Cun, "Sparse feature learning for deep belief networks," *Advances in Neural Information Processing Systems*, pp. 1185-1192, 2008.

[9] R. Salakhutdinov and G. E. Hinton, "Deep Boltzmann machines," *AISTATS*, vol. 1, p. 3, 2009.

[10] V. J. R. Ripoll, A. Wojdel, E. Romero, P. Ramos, and J. Brugada, "ECG assessment based on neural networks with pretraining," *Applied Soft Computing*, vol. 49, pp. 399-406, 2016.

[11] K. Gurney, "An introduction to neural networks," CRC Press, 1997.

[12] H. M. Yao, H. B. Vuthaluru, M. O. Tade, and D. Djukanovic, "Artificial neural network based prediction of hydrogen content of coal in power station boilers," *Fuel*, vol. 84, 2005, pp. 1535–1542.

[13] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436-444, 2015.

[14] L. Goodfellow, Y. Bengio, and A. Courville, "Deep learning," MIT Press, 2016.

[15] G. E. Hinton. (2009). Deep belief networks. *Scholarpedia*. [Online]. *4(5)*, p. 5947. Available: http: //www.scholarpedia.org/article/Deep_belief_networks

[16] P. Smolensky, "Information processing in dynamical systems: Foundations of harmony theory," *Parallel Distributed Processing*, vol. 1, pp. 194-281, 1986.

[17] G. E. Hinton, "Boltzmann machine," *Scholarpedia*, vol. 2, no. 5, 1668, 2007.

[18] J. Z. Mohammed and M. Wagner, "Data mining and analysis: fundamental concepts and algorithms," Cambridge University Press, 2014, USA.

[19] T. T. Nguyen, "Neural network optimal-power-flow," *APSCOM-97*, Hong Kong, November 1997, pp. 266-271.

[20] R. Andrzejak. (2014). Data files. [Online]. Available: http://ntsa.upf.edu/downloads/andrzejak-rg-et-al-2001-indications-nonlinear-deterministic-and-finite-dimensional

[21] R. G. Andrzejak, K. Lehnertz, F. Mormann, C. Rieke, P. David, and C. E. Elger, "Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state," *Physical Review E*, vol. 64, no. 6, p. 061907, 2001.

[22] Y. Chen, E. Keogh, B. Hu, N. Begum, A. Bagnall, A. Mueen, and G. Batista. (2015). The UCR time series classification archive. [Online]. Available: www.cs.ucr.edu/~eamonn/time_series_data/

[23] R. T. Olszewski, "Generalized feature extraction for structural pattern recognition in time-series data," School of Computer Science, Carnegie-Mellon University, Pittsburgh, U.S.A., 2001.

[24] P. Montero and A. J. Vilar, "TSclust: An r package for time series clustering," *Journal of Statistical Software*, vol. 62, issue. 1, pp. 1-43, November 2014.

T. Thinsungnoen is currently a doctoral student with the School of Computer Engineering, Suranaree University of Technology (SUT), Thailand. She received her bachelor degree in computer science from Nakhon Ratchasima Rajabhat Institute, Thailand, in 1999, the master degree in computer engineering from SUT in 2007. Her current research of interest includes data mining, deep Boltzmann machines, deep learning of time series data.

K. Kerdprasop is an associate professor and chair of the School of Computer Engineering, SUT. He received his bachelor degree in mathematics from Srinakarinwirot University, Thailand, in 1986, the master degree in computer science from the Prince of Songkla University, Thailand, in 1991 and doctoral degree in Computer Science from Nova Southeastern University, U.S.A., in 1999. His current research includes data mining, artificial intelligence, computational statistics.

N. Kerdprasop is an associate professor with Computer Engineering School, SUT. She received her bachelor degree in radiation techniques from Mahidol University, Thailand, in 1985, the master degree in computer science from the Prince of Songkla University, Thailand, in 1991 and the doctoral degree in computer science from Nova Southeastern University, U.S.A, in 1999. Her research of interest includes data mining, artificial intelligence, and intelligent databases.

ประวัติผู้เขียน

นางทิพยา ถินสูงเนิน เกิดเมื่อวันที่ 2 พฤษภาคม พ.ศ. 2519 เริ่มศึกษาชั้นประถมที่โรงเรียนบ้านขาม ชั้นมัธยมศึกษาตอนต้นที่โรงเรียนสตรีศึกษาร้อยเอ็ด ชั้นมัธยมศึกษาตอนปลายที่โรงเรียนร้อยเอ็ดวิทยาลัย จังหวัดร้อยเอ็ด และสำเร็จการศึกษาระดับปริญญาตรี สาขาวิชาวิทยาการคอมพิวเตอร์ เกียรตินิยมอันดับสอง สถาบันราชภัฏนครราชสีมา จังหวัดนครราชสีมา เมื่อปี พ.ศ. 2542 ระหว่างศึกษาระดับปริญญาตรี ได้เข้าทำงานเป็นพนักงานชั่วคราว ตำแหน่งโปรแกรมเมอร์ ที่บริษัทสยามมัลติซอฟต์ หลังจากสำเร็จการศึกษาได้เข้าทำงานตำแหน่งนักวิเคราะห์และออกแบบระบบ ที่บริษัทอัลฟาออฟฟิซออโตเมชั่น ระหว่าง เมษายน - พฤศจิกายน พ.ศ. 2542 และตำแหน่งนักวิเคราะห์และออกแบบระบบ ที่บริษัท เทอร์โบซอฟต์ ระหว่าง ธันวาคม พ.ศ. 2542 – พฤษภาคม พ.ศ. 2544 และได้รับบรรจุเข้าทำงานตำแหน่งอาจารย์พิเศษตามสัญญา โปรแกรมวิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏนครราชสีมา เมื่อพฤษภาคม พ.ศ. 2544 เป็นต้นมา ในปี พ.ศ. 2547 ได้เข้าศึกษาต่อในระดับปริญญาโท สาขาวิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเทคโนโลยีสุรนารี โดยขณะศึกษาได้รับทุนผู้ช่วยสอนและวิจัยสาขาวิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเทคโนโลยีสุรนารี ขณะเป็นผู้ช่วยวิจัยได้พัฒนาบทความวิจัยเรื่อง Efficient Progressive Sampling for Data Mining ในขณะศึกษาระดับปริญญาโทได้จัดทำวิทยานิพนธ์เรื่อง การพัฒนาเครื่องมือสร้างกราฟควบคุมกระแสและข้อมูลทดสอบสำหรับภาษาซี และสำเร็จการศึกษาระดับปริญญาโท เมื่อปีการศึกษา 2549

ปี พ.ศ. 2557 ได้เข้าศึกษาต่อในระดับปริญญาเอก สาขาวิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเทคโนโลยีสุรนารี ในระหว่างศึกษาระดับปริญญาเอกได้พัฒนาบทความวิจัยดังแสดงในรายการบทความวิจัยตีพิมพ์ในภาคผนวก ข และได้จัดทำวิทยานิพนธ์เรื่อง การแทนข้อมูลอนุกรมเวลาเพื่อการจัดกลุ่มที่มีประสิทธิภาพ

ผลงานวิจัย : ได้รับทุนอุดหนุนการดำเนินงานวิจัยจากสำนักงานคณะกรรมการวิจัยแห่งชาติ (วช.) ในปี พ.ศ. 2554 เรื่อง การพัฒนาแบบจำลองข้อมูลการสำเร็จการศึกษาและไม่สำเร็จการศึกษาสำหรับนักศึกษาคณะวิทยาศาสตร์และเทคโนโลยีโดยเปรียบเทียบเทคนิคดาต้าไมน์นิ่งแบบ C4.5 และเบย์ ในปี พ.ศ. 2557 ชุดโครงการวิจัย เรื่อง การบูรณาการข้อมูลคุณภาพน้ำเพื่อการบริหารจัดการมลพิษทางน้ำและการใช้ประโยชน์ทรัพยากรน้ำอย่างยั่งยืน กรณีศึกษา : ลุ่มน้ำลำตะคอง และได้รับทุนจากมหาวิทยาลัยราชภัฏนครราชสีมา ในปี พ.ศ. 2555 เรื่อง การพัฒนาระบบสนับสนุนการตัดสินใจผลิตลายผ้าไทยในเขตจังหวัดนครราชสีมาด้วยเทคนิคดาต้าไมน์นิงแบบดีซีชันทรี