การสร้างกฎที่มีประสิทธิภาพสำหรับการจำแนกเชิงความสัมพันธ์

วิทยานิพนธ์

ของ

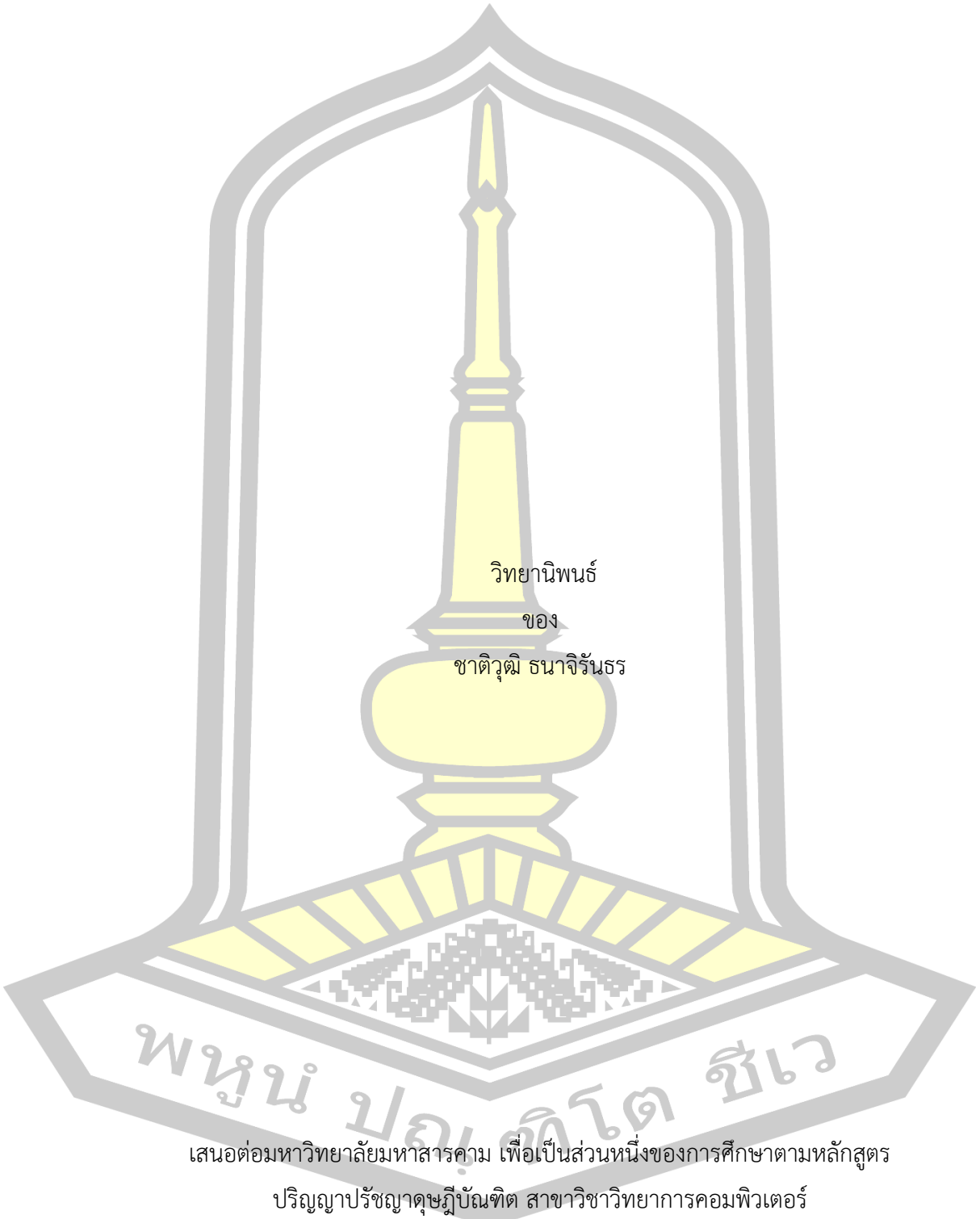
ชาติวุฒิ ธนาจิรันธร

เสนอต่อมหาวิทยาลัยมหาสารคาม เพื่อเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาปรัชญาดุษฎีบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์

ธันวาคม 2563

การสร้างกฎที่มีประสิทธิภาพสำหรับการจำแนกเชิงความสัมพันธ์



วิทยานิพนธ์

ของ

ชาติวุฒิ ธนาจิรันธร

เสนอต่อมหาวิทยาลัยมหาสารคาม เพื่อเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาปรัชญาดุษฎีบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์

ธันวาคม 2563

Generating Efficient Rules for Associative Classification

Chartwut Thanajiranthorn

A Thesis Submitted in Partial Fulfillment of Requirements

for Doctor of Philosophy (Computer Science)

December 2020

คณะกรรมการสอบวิทยานิพนธ์ ได้พิจารณาวิทยานิพนธ์ของนายชาติวุฒิ ธนาจิรันธร แล้วเห็นสมควรรับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาปรัชญาดุษฎีบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ ของมหาวิทยาลัยมหาสารคาม

คณะกรรมการสอบวิทยานิพนธ์

.....ประธานกรรมการ
(ผศ. ดร. วรารัตน์ สงฆ์แป้น)

.....อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(ผศ. ดร. พนิดา ทรงรัมย์)

.....กรรมการ
(ผศ. ดร. พัฒนพงษ์ ชมภูวิเศษ)

.....กรรมการ
(ผศ. ดร. ฉัตรเกล้า เจริญผล)

.....กรรมการ
(ผศ. ดร. มนัสวี แก่นอำพรพันธ์)

มหาวิทยาลัยอนุมัติให้รับวิทยานิพนธ์ฉบับนี้ เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญา ปรัชญาดุษฎีบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ ของมหาวิทยาลัยมหาสารคาม

.....
(ผศ. ศศิธร แก้วมั่น)
คณบดีคณะวิทยาการสารสนเทศ

.....
(รศ. ดร. กริสน์ ชัยมูล)
คณบดีบัณฑิตวิทยาลัย

ชื่อเรื่อง การสร้างกฎที่มีประสิทธิภาพสำหรับการจำแนกเชิงความสัมพันธ์
ผู้วิจัย ชาติวุฒิ ธนาจิรันธร
อาจารย์ที่ปรึกษา ผู้ช่วยศาสตราจารย์ ดร. พนิดา ทรงรัมย์
ปริญญา ปรัชญาดุษฎีบัณฑิต **สาขาวิชา** วิทยาการคอมพิวเตอร์
มหาวิทยาลัย มหาวิทยาลัยมหาสารคาม **ปีที่พิมพ์** 2563

บทคัดย่อ

การจำแนกเชิงความสัมพันธ์เป็นเทคนิคการจำแนกชุดข้อมูลที่รวมการจำแนกและกฎความสัมพันธ์เข้าด้วยกัน จากการวิจัยที่ผ่านมาพบว่าการจำแนกเชิงความสัมพันธ์สามารถจำแนกข้อมูลได้ถูกต้องมากกว่าเทคนิคจำแนกแบบดั้งเดิมและให้แบบจำลองที่ง่ายต่อการแปลความหมาย เพราะอยู่ในรูปบบของกฎความสัมพันธ์ อย่างไรก็ตามการจำแนกเชิงความสัมพันธ์เผชิญกับปัญหาการสร้างกฎรายการจำนวนมากเมื่อค่าสนับสนุนขั้นต่ำถูกกำหนดให้มีค่าน้อย ซึ่งบางกฎไม่ได้ถูกนำมาใช้ในการจำแนก มีความซ้ำซ้อนและต้องถูกกำจัดในภายหลัง ทำให้เวลาในการประมวลผลเพิ่มขึ้นและการใช้หน่วยความจำปริมาณมากสำหรับการสร้างแบบจำลอง ปัญหาเหล่านี้แปรผันตรงตามจำนวนชุดข้อมูลที่เพิ่มมากขึ้น งานวิจัยจึงนำเสนอขั้นตอนวิธีใหม่สำหรับการจำแนกเชิงความสัมพันธ์เพื่อกำจัดกฎรายการที่ไม่จำเป็น โดยมุ่งค้นหาเฉพาะกฎที่มีประสิทธิภาพสำหรับการจำแนก การแทนค่าข้อมูลแนวตั้งได้ถูกนำเข้ามาใช้เพื่อหลีกเลี่ยงกฎรายการที่ไม่จำเป็นและลดเวลาในกระบวนการขุดค้นข้อมูล ผลการทดลองแสดงว่าขั้นตอนวิธีที่นำเสนอมีประสิทธิภาพทางด้านความถูกต้องในการจำแนกข้อมูล เวลาและหน่วยความจำในการประมวลผลเมื่อเปรียบเทียบกับขั้นตอนวิธี CBA CMAR และ FACA

คำสำคัญ : การจำแนกเชิงความสัมพันธ์, กฎความสัมพันธ์ระบุคลาส, การแสดงข้อมูลแนวตั้ง, การจำแนกข้อมูล

TITLE Generating Efficient Rules for Associative Classification

AUTHOR Chartwut Thanajiranthorn

ADVISORS Assistant Professor Panida Songram , Ph.D.

DEGREE Doctor of Philosophy **MAJOR** Computer Science

UNIVERSITY Mahasarakham University **YEAR** 2020

ABSTRACT

Associative classification is a classification technique that combines classification and association rule mining for classifying unseen data. In the literature, associative classification technique has been found to be more accurate than traditional classification techniques and gives classifier that is easy to interpret by utilizing association rules. However, if a low minimum support threshold is given, a large number of frequent ruleitems will be generated. Some of the ruleitems are not used for classification and needed to be pruned. Moreover, computation time and memory are massively consumed. These problems are highly intensive especially when an input dataset has a large number of dimensions. In this paper, a new associative classification algorithm is proposed to eliminate unnecessary ruleitems. It directly discovers efficient rules for classification. A vertical data representation technique is implemented to avoid unnecessary ruleitems and speeds up mining processes. The experimental results show that the proposed algorithm archives in terms of accuracy, a number of generated ruleitems, classifier building time, and memory consumption, when comparing to the well-know algorithms, CBA, CMAR, and FACA.

Keyword : Associative Classification, Class Association Rule, Vertical Data Representation, Classification

กิตติกรรมประกาศ

วิทยานิพนธ์นี้ได้รับทุนอุดหนุนการวิจัยสำหรับนิสิตระดับบัณฑิตศึกษา (ปริญญาเอก) ประจำปีงบประมาณ พ.ศ. 2564 จากมหาวิทยาลัยมหาสารคาม จนกระทั่งได้รับการตีพิมพ์เผยแพร่ผลงานวิจัยในวารสารวิชาการระดับสากล

การดำเนินการพัฒนาวิทยานิพนธ์เล่มนี้มิอาจสำเร็จไปได้หากปราศจาก การชี้แนะและช่วยเหลืออย่างดียิ่งจาก ผู้ช่วยศาสตราจารย์ ดร.พนิดา ทรงรัมย์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ให้คำแนะนำและสละเวลาตรวจแก้ไขข้อบกพร่องอย่างละเอียด เพื่อให้วิทยานิพนธ์ออกมาสมบูรณ์มากที่สุด ตลอดจนมอบความเชื่อมั่นในการสร้างสรรค์งานวิจัยให้แก่ผู้เขียนพร้อมผลักดันให้การดำเนินงานจนได้รับความสำเร็จ ผู้เขียนขอกราบขอบพระคุณด้วยความเคารพอย่างสูงไว้ ณ โอกาสนี้

ผู้เขียนขอกราบขอบพระคุณของคณาจารย์ในสาขาวิทยาการคอมพิวเตอร์ ที่ให้คำแนะนำ และมอบความรู้ที่เกี่ยวข้องกับวิทยานิพนธ์ อีกทั้งเพื่อน ๆ พี่ ๆ น้อง ๆ สาขาวิทยาการคอมพิวเตอร์ซึ่งให้คำปรึกษาและแลกเปลี่ยนความรู้ จนสามารถพัฒนาวิทยานิพนธ์จนสำเร็จ

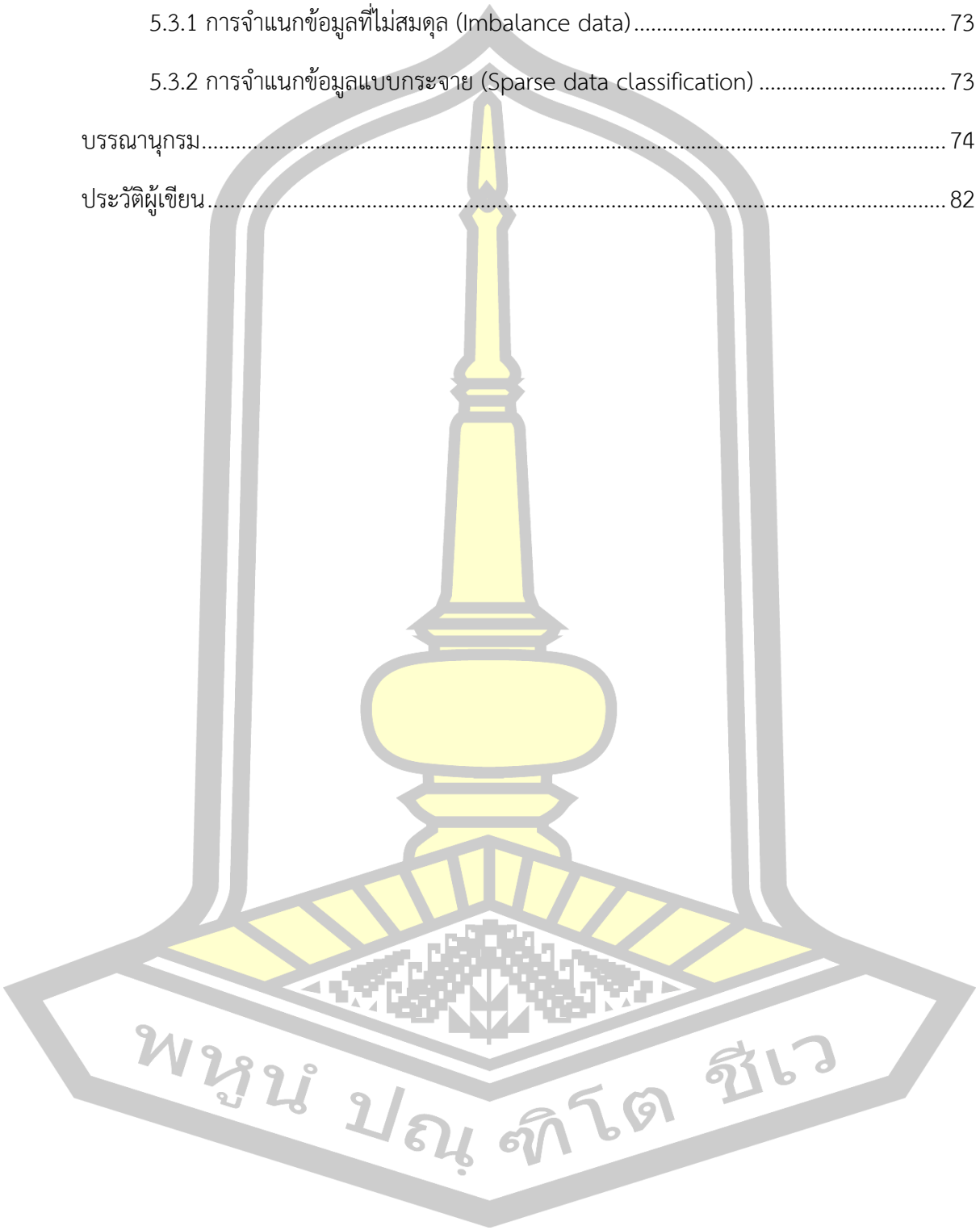
ผู้เขียนขอกราบขอบพระคุณ ดร.ทิพวัลย์ แสนคำ ที่ช่วยเหลือสนับสนุนทั้งด้านการวางแผนการเรียนและความช่วยเหลือด้านกำลังใจและกำลังทรัพย์ด้วยดีตลอดมา อีกทั้งขอขอบคุณเพื่อนร่วมงานในสาขาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยราชภัฏบุรีรัมย์ทุกท่านที่เป็นกำลังใจ

สุดท้ายผู้วิจัยขอขอบคุณครอบครัวธนาจิรันธร ครอบครัวเพียงไธสง ครอบครัวโยโพธิ์ ที่ให้กำลังใจ และให้การสนับสนุนตามกำลังความสามารถ อีกทั้งแบ่งเบาภาระในชีวิตประจำวันเพื่อให้ผู้เขียนสามารถทุ่มเทในการเขียนวิทยานิพนธ์ได้อย่างเต็มที่ จนงานวิจัยสำเร็จลุล่วงในที่สุด นอกจากนี้ยังมีผู้ที่ให้ความช่วยเหลืออีกหลายท่าน ซึ่งผู้เขียนไม่สามารถกล่าวนามในที่นี้ได้หมด จึงขอขอบคุณทุกท่านเหล่านั้นไว้ ณ โอกาสนี้ด้วย คุณค่าและประโยชน์อันพึงมีจากการศึกษาวิจัยนี้ ผู้วิจัยขอน้อมบูชาพระคุณบิดามารดาและบูรพาจารย์ทุกท่าน ที่ได้อบรมสั่งสอนวิชาความรู้ และให้ความเมตตาแก่ผู้วิจัยมาโดยตลอด และเป็นกำลังใจสำคัญ ที่ทำให้การศึกษาวิจัยฉบับนี้สำเร็จลุล่วงได้ด้วยดี

ชาติวุฒิ ธนาจิรันธร

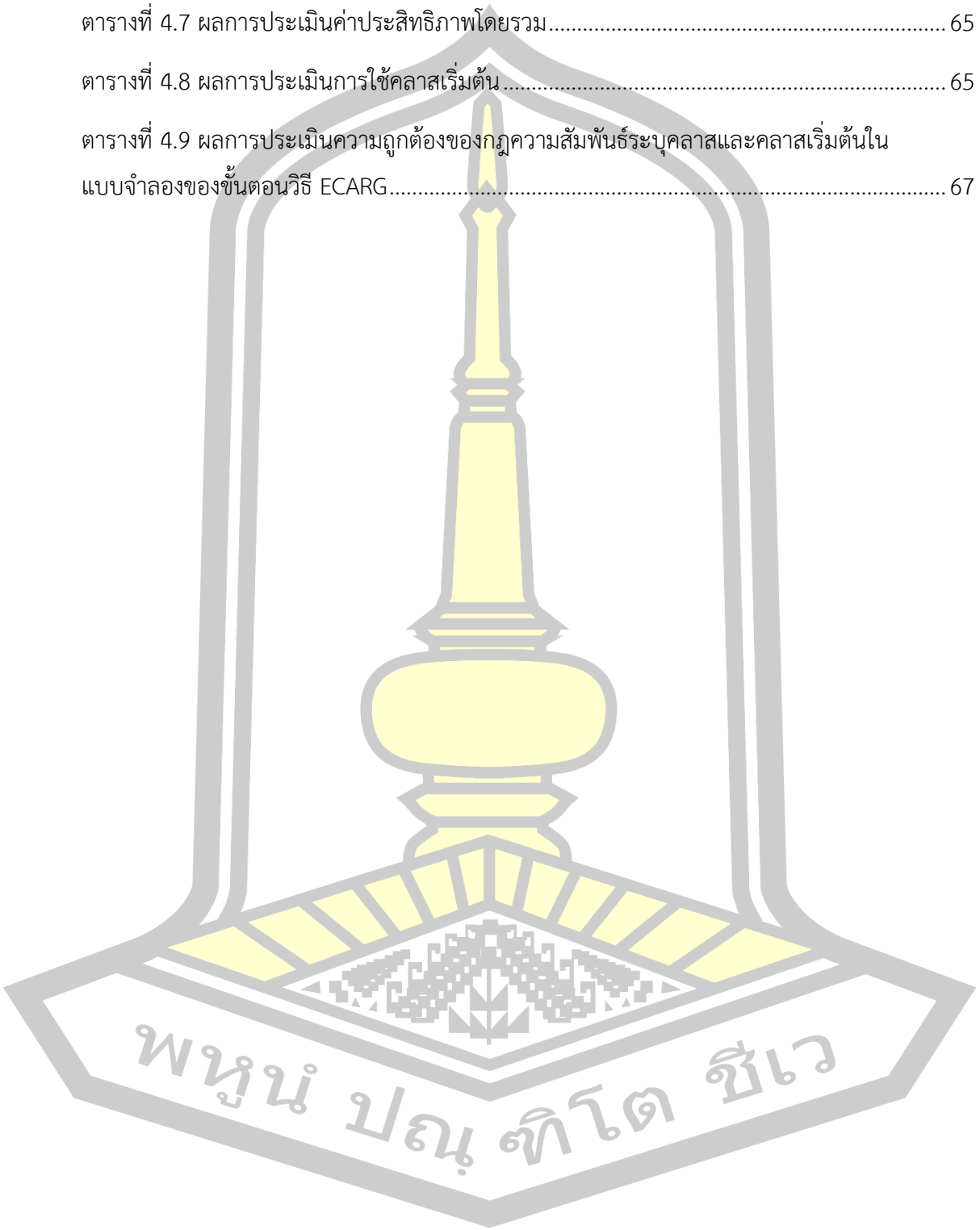
สารบัญ

หน้า



สารบัญตาราง

หน้า



สารบัญรูป

หน้า

บทที่ 1

บทนำ

1.1 หลักการและเหตุผล

การจำแนกข้อมูลเชิงความสัมพันธ์ (Associative Classification) คือ การจำแนกข้อมูลด้วยวิธีการเรียนรู้แบบมีผลเฉลย (Supervised Learning) โดยใช้กฎความสัมพันธ์ การจำแนกเชิงความสัมพันธ์ถูกคิดค้นครั้งแรกโดย Lui และคณะ [1] เป็นการรวม 2 เทคนิคเข้าด้วยกัน ได้แก่ กฎความสัมพันธ์ (Association Rule) และการจำแนกข้อมูล (Classification) โดยกฎความสัมพันธ์ใช้ในการหาความสัมพันธ์ระหว่างรายการข้อมูล และการจำแนกข้อมูลใช้ในการทำนายคลาส การจำแนกข้อมูลเชิงความสัมพันธ์ใช้กฎความสัมพันธ์ระบุคลาส (Class Association Rules) หรือ CARs ในการจำแนกข้อมูล โดย CARs อยู่ในรูปแบบ $itemsets \rightarrow c$ โดย itemsets คือ เซตรายการที่ประกอบด้วยแอททริบิวต์และค่าที่จัดเก็บ เขียนแทนด้วย $\langle (A_i, a_j) \rangle$ และ c คือ คลาสที่เป็นไปได้ในชุดข้อมูล การหากฎความสัมพันธ์ระบุคลาสประกอบไปด้วย 2 ขั้นตอนหลัก คือ 1) การหากฎที่มีค่าสนับสนุนมากกว่าหรือเท่ากับค่าสนับสนุนขั้นต่ำ กฎดังกล่าวเรียกว่า กฎรายการความถี่ (Frequent Ruleitem) 2) จากนั้นคัดเลือกกฎรายการที่มีค่าความเชื่อมั่นมากกว่าหรือเท่ากับค่าความเชื่อมั่นขั้นต่ำและทำการเรียงลำดับกฎ เพื่อสร้างเป็นแบบจำลองในการทำนายข้อมูล การสร้างกฎเพื่อใช้ในการทำนายข้อมูลมีความแม่นยำในการทำนายสูง [2, 3] นอกจากนั้นกฎที่ใช้ในการทำนายถูกสร้างให้มีลักษณะเป็น if-then ทำให้ผู้ใช้งานสามารถอ่านและเข้าใจได้ง่าย ส่งผลให้การจำแนกเชิงความสัมพันธ์ถูกประยุกต์ใช้ในงานที่หลากหลาย เช่น การตรวจจับเว็บไซต์ล่อลวงเหยื่อ [4, 5] การทำนายอาการโรคหัวใจ [6, 7] การจำแนกข้อมูลแหล่งน้ำ [8] การตรวจจับข้อมูลคุณภาพต่ำในโซเชียลมีเดีย [9] เป็นต้น

เนื่องจากการสร้างกฎความสัมพันธ์ระบุคลาสมีพื้นฐานมาจากการสร้างกฎความสัมพันธ์ซึ่งกฎรายการถูกสร้างขึ้นเป็นจำนวนมากแต่บางกฎไม่ได้ถูกนำมาใช้และมีความซ้ำซ้อน เช่น ขั้นตอนวิธี CBA (Classification Based on Association Rules) [1] สร้างกฎความสัมพันธ์ระบุคลาสด้วยพื้นฐานเทคนิค Apriori ซึ่งสแกนฐานข้อมูลหลายรอบและได้กฎจำนวนมาก ขั้นตอนวิธี FACA (Fast Associative Classification Algorithm) [5] ใช้เทคนิคเซตผลต่าง (Different Sets หรือ Diffset) เพื่อคำนวณค่าสนับสนุนและค่าความเชื่อมั่นของกฎเพื่อลดการสแกนฐานข้อมูลหลายรอบแต่ FACA ต้องสร้างกฎที่มี 1 เซตรายการจำนวนมาก ขั้นตอนวิธี PCAR (Predictability-Based Collective Class Association Rules) [10] นำเสนอวิธีสร้างค่าความสามารถในการทำนาย (Predictability-value) ขึ้นมาเพื่อใช้ในการเรียงกฎ การเพิ่มค่าความสามารถในการทำนายส่งผลให้ขั้นตอนวิธีนี้ใช้เวลาประมวลผลมากขึ้นและยังไม่สามารถหลีกเลี่ยงการสร้างกฎคู่แข่งจำนวนมากได้ ขั้นตอนวิธี CAR-Miner [11] และ CAR-Miner-Diff [12] ใช้โครงสร้างเซตของตัวระบุวัตถุซึ่งบรรจุเซตรายการ (Obidset) ในการสกัดกฎความสัมพันธ์ระบุคลาส เพื่อให้สามารถสร้างกฎได้อย่างรวดเร็ว แต่กฎคู่แข่งจำนวนมากถูกสร้างขึ้นมาก่อนแล้วคัดเลือกกฎที่มีประสิทธิภาพในการจำแนกภายหลัง

เพื่อแก้ไขปัญหาการสร้างกฎคู่แข่งโดยไม่จำเป็น ขั้นตอนวิธี CMAR (Classification Based On Multiple Class-Association Rules) [13] ใช้เทคนิค FP-Growth ร่วมกับโครงสร้างต้นไม้ FP-Tree (Frequent Pattern Tree) ทำให้มีความเร็วในการประมวลผลสูงและหลีกเลี่ยงการสร้างกฎคู่แข่ง แต่อย่างไรก็ตามการทำงานกับฐานข้อมูลขนาดใหญ่ที่มีจำนวนข้อมูลมากจะทำให้โครงสร้าง FP-Tree มีขนาดใหญ่เกินกว่าที่จะจัดเก็บลงในหน่วยความจำได้ นอกจากนี้ ขั้นตอนวิธี CCAR (Constraint Class Association Rule) [14] และ LD-CARM-IC [15] นำเสนอแนวทางการสร้างกฎรายการตามเงื่อนไขของผู้ใช้เพื่อลดการสร้างกฎคู่แข่ง ผลการทดลองแสดงให้เห็นว่า LD-CARM-IC ซึ่งใช้โครงสร้างตาข่าย (Lattice Structure) ร่วมกับเซตผลต่างสามารถสร้างกฎรายการได้เร็วกว่าและประหยัดหน่วยความจำมากกว่า อย่างไรก็ตามวิธีนี้จำเป็นต้องกำหนดแอททริบิวต์และระบุค่าเพื่อใช้คัดกรองข้อมูล ขั้นตอนวิธี Top-k CARs [16] ค้นหากฎความสัมพันธ์ระบุคลาสลำดับสูงสุด k ลำดับแรกด้วยวิธีการเรียงข้อมูลแบบเร็ว (Quick Sort) โดยมีแนวคิดจาก TopKRules [17] ซึ่งเห็นว่าค่าสนับสนุนขั้นต่ำยากต่อความเข้าใจของผู้ใช้ จึงสร้างตัวแปร k เพื่อให้ผู้ใช้กำหนดจำนวนกฎที่ต้องการให้เกิดขึ้น อุปสรรคของวิธีการนี้ คือ หากผู้ใช้เลือกค่า k น้อยเกินไป จำนวนกฎที่ถูกสร้างอาจไม่เหมาะสมทำให้ค่าความถูกต้องของแบบจำลองลดลง RAJAB นำเสนอวิธีการประเมินและการตัดกฎแบบใหม่ในขั้นตอนวิธี APR (Active Pruning Rules) [18] ขั้นตอนวิธี ARP ค้นหากฎรายการความถี่ทั้งหมด แล้วเรียงลำดับและประเมินกฎแบบเดียวกับขั้นตอนวิธี CBA แตกต่างกันที่เมื่อกฎถูกแทรกลงในตัวจำแนกทุกครั้ง ข้อมูลชุดสอนทั้งหมดที่เกี่ยวข้องกับกฎดังกล่าวจะถูกลบออกทันที กฎที่ยังไม่ถูกประเมินแต่ละรายการที่เกี่ยวข้องกับข้อมูลชุดสอนที่ถูกลบจะได้รับการปรับปรุงค่าสนับสนุนและค่าความมั่นใจแล้วจัดเรียงใหม่ อย่างไรก็ตาม ขั้นตอนวิธี ARP ใช้การค้นหาแบบค้นหาทั้งหมด (Exhaustive search) เพื่อสร้างกฎคู่แข่งทั้งหมดก่อนการตัดกฎเช่นเดียวกับขั้นตอนวิธี CBA WCBA และ FACA

ขั้นตอนวิธีที่กล่าวมีลักษณะการหากฎความสัมพันธ์ระบุคลาสโดยสร้างกฎคู่แข่งทั้งหมดก่อนแล้วจึงหากฎที่มีผ่านเกณฑ์ค่าสนับสนุนขั้นต่ำและค่าความเชื่อมั่นขั้นต่ำ ทำให้กฎที่ไม่จำเป็นและซ้ำซ้อนจำนวนมากถูกสร้างขึ้น ส่งผลให้เวลาในการประมวลผลสูงตลอดจนการใช้หน่วยความจำยังสูงอีกด้วย จึงได้มีการประยุกต์วิธีการเหนี่ยวนำกฎ (Rule Induction) หรือ RI เพื่อสร้างกฎความสัมพันธ์ระบุคลาสที่มีค่าความเชื่อมั่นสูงที่สุด (100%) ก่อน แล้วจึงหากฎที่มีความเชื่อมั่นต่ำกว่าถัดไป เช่น ขั้นตอนวิธี PRISM [19].ใช้วิธีการค้นหากฎที่มีค่าความเชื่อมั่น 100% เพื่อให้ได้ผลการทำงานที่สูงที่สุด อย่างไรก็ตาม PRISM ไม่ได้ให้ความสำคัญกับค่าสนับสนุนของกฎ แม้ว่ากฎจะมีค่าสนับสนุนต่ำกว่า 1% แต่มีค่าความเชื่อมั่น 100% กฎนั้นยังถูกนำไปสร้างแบบจำลอง ขั้นตอนวิธี eDRI (Enhance Dynamic Rules Induction) [20] แก้ปัญหาของ PRISM โดยตั้งเกณฑ์ค่าความถี่ขั้นต่ำ เพื่อคัดกรองกฎที่มีค่าสนับสนุนต่ำออก และใช้เทคนิคการสร้างแบบจำลองจากกฎที่มีค่าความเชื่อมั่น 100% เมื่อกฎถูกสร้าง eDRI จะลดพื้นที่การค้นหาข้อมูลด้วยการลบข้อมูลที่เกี่ยวข้องกับกฎนั้น ผลการทดลองพบว่าขั้นตอนวิธี eDRI สร้างกฎได้จำนวนน้อยและมีความแม่นยำสูงกว่า PRISM อย่างไรก็ตามปัญหาที่สำคัญของขั้นตอนวิธี eDRI คือ การคำนวณค่าสนับสนุนและค่าความเชื่อมั่นทุกครั้งหลังจากลบข้อมูลที่เกี่ยวข้องกับกฎที่สร้างแล้วออกไป ขั้นตอนวิธี ACPRISM (Associative Classification Based on PRISM) นำเสนอการรวมเทคนิคของการจำแนกข้อมูลเชิงสัมพันธ์และการเหนี่ยวนำกฎเข้า

ด้วยกัน โดยสร้างกฎความสัมพันธ์จากแอททริบิวต์ที่มีความสำคัญในการจำแนกสูง ซึ่งคำนวณจากค่า Information Gain ทำให้สามารถตัดกฎที่ซ้ำซ้อนและไม่เกี่ยวข้องออกไปได้อย่างรวดเร็ว แต่อย่างไรก็ ACPRISM เริ่มสร้างกฎจากค่าภายในแอททริบิวต์ที่มีค่า Information gain สูงสุดเท่านั้น อาจทำให้ค่าภายในแอททริบิวต์อื่นที่มีความสามารถในการจำแนกข้อมูลสูงไม่ถูกนำไปสร้างเป็นกฎที่มีประสิทธิภาพ

งานวิจัยที่ผ่านมาพยายามสร้างกฎที่ให้ประสิทธิภาพในการจำแนกสูง แต่อย่างไรก็ตามมีการสร้างกฎที่ไม่ถูกนำไปใช้ในการจำแนก ดังนั้นงานวิจัยนี้จึงนำเสนอแนวทางใหม่ในการสร้างกฎ ซึ่งรวมการค้นหากฎและการประเมินคุณภาพของกฎเข้าด้วยกัน เพื่อค้นหากฎที่มีประสิทธิภาพในการทำนายคลาสข้อมูลและหลีกเลี่ยงการสร้างกฎที่ไม่จำเป็นในการจำแนก ด้วยวิธีการดังกล่าวนอกจากจะสามารถสร้างแบบจำลองที่ประกอบด้วยกฎจำนวนน้อยที่ให้ความถูกต้องสูง แบบจำลองยังง่ายต่อการวิเคราะห์

1.2 วัตถุประสงค์ของการวิจัย

เพื่อพัฒนาวิธีการค้นหากฎที่มีประสิทธิภาพสำหรับการจำแนกเชิงความสัมพันธ์

1.3 ความสำคัญของการวิจัย

1. งานวิจัยนำเสนอขั้นตอนวิธีสร้างกฎที่มีประสิทธิภาพโดยตรงที่ให้ค่าความถูกต้องในการจำแนกสูง
2. งานวิจัยนำทฤษฎีเซตอย่างง่าย ได้แก่ อินเทอร์เซกชันและเซตผลต่าง มาใช้เพื่อลดเวลาและหน่วยความจำในการขุดค้นกฎที่มีประสิทธิภาพ
3. งานวิจัยนำเสนอโครงสร้างข้อมูลอย่างง่าย ได้แก่ กฎรายการและรายการหมายเลขแทรนเซกชัน ถูกนำมาใช้เพื่อลดการใช้หน่วยความจำ
4. วิธีการที่นำเสนอในงานวิจัยมีความโดดเด่นในด้านความถูกต้อง เวลาในการประมวลผลและการใช้งานหน่วยความจำ เมื่อเปรียบเทียบกับขั้นตอนวิธี CBA CMAR และ FACA

1.4 ขอบเขตของการวิจัย

1. งานวิจัยนี้เป็นการจำแนกข้อมูลเชิงความสัมพันธ์
2. ชุดข้อมูลที่ใช้ในงานวิจัยเป็นชุดข้อมูลมาตรฐานที่นำมาจาก University of California, Irvine (UCI) Machine learning Repository จำนวน 14 ชุด
3. งานวิจัยนี้วัดประสิทธิภาพของขั้นตอนวิธีที่นำเสนอ ประสิทธิภาพในการจำแนกของกฎซึ่งพิจารณาจากค่าความถูกต้อง (Accuracy) ค่าความแม่นยำ (Precision) ค่าความระลึก (Recall) ค่าเฉลี่ยประสิทธิภาพโดยรวม (F-Measure) ด้วยจำนวนกฎที่สร้างได้ เวลาที่ใช้ในการสร้างแบบจำลอง และปริมาณหน่วยความจำที่ใช้ในการสร้างแบบจำลอง
4. เปรียบเทียบประสิทธิภาพของขั้นตอนวิธีที่นำเสนอกับขั้นตอนวิธี CBA CMAR และ FACA

1.5 นิยามศัพท์เฉพาะ

1. เซตรายการ (Itemset) คือ กลุ่มข้อมูลที่ประกอบด้วยแอททริบิวต์และรายการในแอท-ทริบิวต์

2. กฎรายการ (Ruleitem) คือ กฎที่ประกอบด้วยเซตรายการที่สามารถจำแนกคลาสได้

3. ค่าสนับสนุน (Support) คือ อัตราส่วนจำนวนแทรนแซกชันที่มีกฎรายการเทียบกับจำนวนแทรนแซกชันในชุดข้อมูลทั้งหมด

4. กฎรายการความถี่ (Frequent Ruleitem) คือ กฎรายการที่ค่าสนับสนุนมากกว่าหรือเท่ากับค่าสนับสนุนขั้นต่ำ

5. ค่าความเชื่อมั่น (Confidence) คือ อัตราส่วนจำนวนแทรนแซกชันที่มีกฎรายการในแต่ละคลาสเทียบกับจำนวนแทรนแซกชันที่มีเซตรายการ

6. กฎรายการคู่แข่ง (Candidate Ruleitem) คือ กฎรายการที่สร้างขึ้นแต่มีค่าความเชื่อมั่นน้อยกว่าค่าความเชื่อมั่นขั้นต่ำจึงไม่ถูกใช้งานและตัดออกภายหลัง

7. กฎความสัมพันธ์ระบุคลาส (CARs) คือ กฎรายการที่มีค่าความเชื่อมั่นมากกว่าหรือเท่ากับความเชื่อมั่นขั้นต่ำและค่าสนับสนุนมากกว่าหรือเท่ากับค่าสนับสนุนขั้นต่ำ

8. ประสิทธิภาพ คือ ประสิทธิภาพในการจำแนกของกฎที่สร้างขึ้น โดยวัดจากค่าความถูกต้อง จำนวนกฎสามารถค้นพบ เวลาในการค้นหากฎ การใช้หน่วยความจำหลักในการสร้างแบบจำลองเพื่อการจำแนกข้อมูล รวมถึงค่าความแม่นยำ ค่าความระลึก และค่าประสิทธิภาพโดยรวม

บทที่ 2

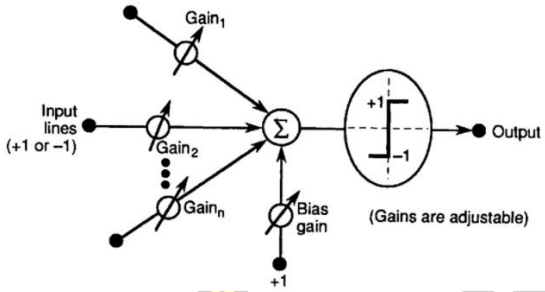
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

งานวิจัยนี้ได้ศึกษาเอกสาร แนวคิด ทฤษฎีและงานวิจัยต่าง ๆ ที่เกี่ยวข้องกันการจำแนกข้อมูลเชิงความสัมพันธ์ (Associative Classification) เพื่อเป็นแนวทางในการพัฒนาขั้นตอนวิธีที่มีประสิทธิภาพในการลดจำนวนกฎสำหรับการจำแนกข้อมูลเชิงความสัมพันธ์ โดยในบทนี้จะกล่าวถึง การจำแนกข้อมูล กฎความสัมพันธ์ การจำแนกข้อมูลเชิงความสัมพันธ์ การแทนข้อมูล การเรียงกฎ การตัดกฎ การวัดประสิทธิภาพในการจำแนก และงานวิจัยที่เกี่ยวข้อง ซึ่งรายละเอียดจะกล่าวดังหัวข้อต่อไปนี้

2.1 การจำแนกข้อมูล (Classification)

การจำแนกข้อมูลเป็นการจัดกลุ่มข้อมูลด้วยวิธีการเรียนรู้แบบมีผลเฉลย (Supervised Learning) ใช้ในการทำนายข้อมูลใหม่ซึ่งยังไม่ได้ระบุคลาส เช่น การจำแนกอีเมล์ขยะจากอีเมล์ทั่วไป โดยใช้คำที่ปรากฏในอีเมล์เป็นต้น ขั้นตอนการจำแนกเริ่มจากการแบ่งชุดข้อมูล (Dataset) เพื่อสร้างเป็นข้อมูลชุดสอน (Train Set) และข้อมูลชุดทดสอบ (Test Set) แล้วค้นหารูปแบบของข้อมูลที่สามารถสร้างกฎเพื่อจำแนกข้อมูลที่ต้องการได้อย่างถูกต้อง สุดท้ายสร้างแบบจำลองจากกฎแล้ววัดประสิทธิภาพแบบจำลองด้วยข้อมูลชุดทดสอบ เทคนิคที่ใช้จำแนกข้อมูลมีหลากหลาย เช่น วิธีการต้นไม้ตัดสินใจ (Decision Tree) วิธีการค้นหาเพื่อนบ้านใกล้สุด K ตัว (K-Nearest Neighbors; KNN) เป็นต้น

วิธีการต้นไม้ตัดสินใจประกอบไปด้วย โหนดตัดสินใจ (Decision node) เพื่อเก็บเงื่อนไขการตัดสินใจ และ โหนดใบ (Terminal Node) ใช้เก็บคลาส ข้อมูลจะถูกเปรียบเทียบกับโหนดตัดสินใจไปจนสามารถระบุคลาสได้จากโหนดใบ [21] วิธีการโครงข่ายประสาท (Neural Network) คือ ระบบโครงข่ายประสาทเทียมจำลองเลียนแบบการทำงานสมองมนุษย์ ถูกประยุกต์ใช้ในงานหลายด้าน เช่น การสร้างรูปภาพธรรมชาติที่สมจริงมาก [22] การตรวจจับภาพบุคคลจากกล้องวงจรปิดเพื่อระบุตัวตน [23] วิธีการโครงข่ายประสาทจะแบ่งการทำงานแยกเป็นหลายชั้น (Layer) โดยชั้นแรกสุดเรียกว่าชั้นนำเข้า (Input Layer) ต่อจากนั้นเป็นชั้นซ่อน (Hidden Layer) และชั้นผลลัพธ์ (Output Layer) โดยข้อมูลที่ต้องการเรียนรู้จะเข้าสู่ชั้นนำเข้า ซึ่งภายในบรรจุโหนดนำเข้า (Input Node) โดยมีจำนวนเท่ากับแอททริบิวต์ของข้อมูล แล้วปรับค่าน้ำหนัก (Weight) ให้เหมาะสมโดยที่ค่านี้เกิดจากการสุ่มระหว่างเลข 0-1 หลังจากนั้นข้อมูลที่ถูกปรับด้วยค่าน้ำหนักจะถูกส่งเข้าไปยังโหนดซ่อน (Hidden Node) ดังรูปที่ 2.1 เพื่อรวมผลและปรับค่าอคติ (Bias) ด้วยแอคทีฟฟังก์ชัน ซึ่งโดยทั่วไปเป็นฟังก์ชัน Sigmoid ในท้ายที่สุดจึงได้เป็นผลลัพธ์ส่งไปยังชั้นผลลัพธ์ ซึ่งอาจมีข้อผิดพลาดเหลืออยู่ ระบบจะนำข้อผิดพลาดจากผลลัพธ์กลับไปปรับค่าน้ำหนักซ้ำอีกหลายครั้ง จนกว่าผลลัพธ์จะเกิดข้อผิดพลาดน้อยที่สุด ขั้นตอนนี้จัดเป็นขั้นตอนสำคัญของระบบโครงข่ายประสาท



รูปที่ 2.1 แผนผังจำลองการทำงานของประสาทเทียม

ที่มา [24]

วิธีการค้นหาเพื่อนบ้านใกล้สุด K ตัว (K-Nearest Neighbors; KNN) เป็นการหาข้อมูลที่ใกล้เคียงที่สุด K ลำดับ วิธีนี้ได้รับการประยุกต์ใช้ในหลายด้าน อาทิ การตรวจจับใบหน้า การเข้ารหัสข้อมูลที่จัดเก็บในพื้นที่จัดเก็บข้อมูลหมู่เมฆ [25] การจัดหมวดหมู่พฤติกรรมสัตว์ตามสายพันธุ์โดยอัตโนมัติ [26] วิธีการ KNN ประกอบด้วยการทำงาน 2 ขั้นตอน คือ การสอนและการจำแนก ในขั้นตอนการสอนข้อมูลตัวอย่างถูกกำหนดตำแหน่งพร้อมระบุคลาสลงในพื้นที่คุณลักษะหลายมิติ [27] โดยส่วนใหญ่จำลองอยู่ในรูปแบบของกราฟ ข้อมูลตัวอย่างที่มีคุณลักษณะใกล้เคียงกันจะอยู่ในตำแหน่งที่ใกล้กันภายในกราฟ ในขั้นตอนการจำแนก วัดระยะห่างระหว่างข้อมูลตัวอย่างภายในกราฟกับข้อมูลทดสอบ หากข้อมูลตัวอย่างคลาสใดอยู่ใกล้ข้อมูลทดสอบที่สุดเป็นจำนวน K จัดได้ว่าข้อมูลทดสอบจัดอยู่ในกลุ่มนั้น โดยการวัดระยะห่าง นิยมใช้วิธีการวัดระยะทางแบบ Euclidean Distance คือ การหาค่ารากที่สองของผลต่างระหว่างข้อมูลใหม่ที่ตำแหน่ง x_i แล้วข้อมูลที่ตำแหน่ง x_j แต่ละตัวยกกำลังสองดังสมการที่ 2.1

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^{n} (a_r(x_i) - a_r(x_j))^2} \qquad 2.1$$

2.2 กฎความสัมพันธ์ (Association Rule)

กฎความสัมพันธ์เป็นวิธีการหนึ่งของการขุดค้นเหมืองข้อมูล เพื่อค้นหาความสัมพันธ์ระหว่างข้อมูลจากข้อมูลที่ปรากฏขึ้นบ่อยในชุดข้อมูลขนาดใหญ่ ผลลัพธ์ที่ได้จะอยู่ในรูปของกฎที่แสดงความสัมพันธ์ระหว่างข้อมูลในรูปแบบเหตุและผล โดยรูปแบบของกฎแสดงอยู่ในรูป A⟶B โดยที่ A คือ เหตุ (Antecedent) และ Y เป็นผลที่ตามมา (Consequence) กฎความสัมพันธ์สามารถประยุกต์ใช้ในงานที่หลากหลาย เช่น การวิเคราะห์ความสัมพันธ์ของสินค้าจากพฤติกรรมการซื้อของลูกค้า [28] การวิเคราะห์ความสัมพันธ์ของเวชภัณฑ์ในการรักษาผู้ป่วย [28, 29] เป็นต้น วิธีการกฎความสัมพันธ์นิยมใช้เพื่อค้นหาความสัมพันธ์ของสินค้าจากพฤติกรรมการซื้อของลูกค้าในห้างสรรพสินค้า รวมถึงสั่งสินค้าทางอีเมล์และซื้อสินค้าออนไลน์ [30] โดยเจาะจงไปยังกลุ่มของสินค้าซึ่งมีความถี่ในการถูกซื้อร่วมกันบ่อยครั้ง การสร้างกฎความสัมพันธ์ประกอบไปด้วย 2 ขั้นตอนหลัก คือ การสร้างเซตรายการความถี่ และการสร้างกฎความสัมพันธ์ ซึ่งรายละเอียดแสดงได้ดังหัวข้อต่อไปนี้

2.2.1 การทำเหมืองเซตรายการความถี่ (Frequent Itemset Mining)

การทำเหมืองเซตรายการความถี่ คือ การค้นหาข้อมูลหรือความสัมพันธ์ระหว่างข้อมูลที่มีความถี่มากกว่าหรือเท่ากับค่าสนับสนุนขั้นต่ำที่ผู้ใช้กำหนด วิธีการทั่วไปที่นิยมใช้ในการหาเซตรายการความถี่ คือ Apriori [31] และ FP-Growth [32] การทำเหมืองเซตรายการความถี่เป็นขั้นตอนแรกของกระบวนการกฎความสัมพันธ์โดยมีนิยามที่เกี่ยวข้องดังนี้

กำหนด $I=\{i_1,i_2,...,i_3\}$ เป็นเซตของรายการทั้งหมดในฐานข้อมูล $T=\{t_1,t_2,...,t_3\}$ คือ เซตของแทรนแซกชันทั้งหมด (Transaction) ซึ่งทุกแทรนแซกชันประกอบด้วยเซตย่อยของ I และกำหนดให้ x คือ เซตรายการ (Itemset) โดยที่ $x \subseteq I$ และ $g(x)$ เท่ากับแทรนแซกชันที่มีเซตรายการ x โดยที่จำนวนแทรนแซกชันที่มีเซตรายการ x แทนด้วย $|g(x)|$

นิยามที่ 2.1 ความยาวของเซตรายการ *x* คือ จำนวนรายการที่ปรากฏอยู่ในเซตรายการ *x*

ตัวอย่างที่ 2.1 กำหนดชุดข้อมูลดังตารางที่ 2.1 เซตรายการ (CD) มีความยาวเท่ากับ 2 เนื่องจากเซตรายการ (CD) ประกอบไปด้วยรายการ C และ D

ตารางที่ 2.1 ตัวอย่างข้อมูล 1

แทรนแซกชัน	เซตรายการ
1	A C D F
2	A B C
3	A C D F
4	B C D E F

นิยามที่ 2.2 ค่าสนับสนุนสัมบูรณ์ (Absolute Support) ของเซตรายการ x คือ ความถี่ของการเกิด x หรือจำนวนแทรนแซกชันที่พบ x แทนด้วย $|g(x)|$ ดังสมการที่ 2.2

$$sup(x)=g(x) \tag{2.2}$$

นิยามที่ 2.3 ค่าสนับสนุนสัมพันธ์ (Relative Support) คือ เปอร์เซ็นที่พบเซตรายการ *x* เมื่อเทียบกับจำนวนแทรนแซกชันทั้งหมด สามารถคำนวณได้ ดังสมการที่ 2.3

$$sup(x)=\frac{|g(x)|}{|g(T)|}\times 100 \tag{2.3}$$

ตัวอย่างที่ 2.2 จากตารางที่ 2.1 เซตรายการ (CD) ปรากฏอยู่ในแทรนแซกชันที่ 1 3 และ 4 ดังนั้นค่าสนับสนุนสัมบูรณ์มีค่าเท่ากับ 3 ส่วนค่าสนับสนุนสัมพันธ์มีค่าเท่ากับ 3/4 * 100 = 75% ซึ่งแสดงให้เห็นว่าเซตรายการ (CD) พบถึง 75% จากข้อมูลทั้งหมด

นิยามที่ 2.4 ค่าสนับสนุนขั้นต่ำ (Minimum Support) เงื่อนไขค่าสนับสนุนที่กำหนดโดยผู้ใช้ เขียนแทนด้วย *minsup*

นิยามที่ 2.5 เซตรายการความถี่ (Frequent Itemset) คือ เซตรายการที่มีค่าสนับสนุนมากกว่าหรือเท่ากับค่าสนับสนุนขั้นต่ำ

ตัวอย่างที่ 2.3 สมมติกำหนดค่าสนับสนุนขั้นต่ำแบบค่าสนับสนุนสัมบูรณ์และให้มีค่าเท่ากับ 2 เซตรายการ (CD) เป็นเซตรายการความถี่เนื่องมาจากค่าสนับสนุนของเซตรายการ (CD) มีค่าเท่ากับ 3 ซึ่งมีค่ามากกว่าค่าสนับสนุนขั้นต่ำ

ปัจจุบันมีงานวิจัยที่เกี่ยวข้องกับการทำเหมืองเซตรายการความถี่ เช่น Zaki และคณะ [33] ใช้ Diffsets เพื่อสร้างเซตรายการความถี่ ต่อมา Deng [34] ได้ประยุกต์เป็นการสร้างเซตรายการด้วย DiffNodesets งานวิจัยของ Uy และคณะ [35] ใช้สถาปัตยกรรมคำนวณอุปกรณ์รวม (Compute Unified Device Architecture หรือ CUDA) ในการสร้างเซตรายการความถี่ความยาว 1 รายการ (1-Itemset)

2.2.2 กฎความสัมพันธ์ (Association Rule)

กฎความสัมพันธ์เป็นการค้นหาความสัมพันธ์ที่น่าสนใจระหว่างรายการภายในฐานข้อมูล โดยไม่สนใจลำดับการเกิดของรายการ เช่น เมื่อมีรายการ x แล้วมีโอกาสเกิดรายการ y จำนวน z แทรนแซกชัน การสร้างกฎความสัมพันธ์โดยมีนิยามที่เกี่ยวข้องดังต่อไปนี้

นิยามที่ 2.6 กฎความสัมพันธ์ (Association Rule) ของ *r* เขียนแทนด้วย $r{=}x \longrightarrow y$ หมายถึง เซตข้อมูลที่พบรายการ x แล้วจะพบรายการ y เมื่อ $x \subset I, y \subset I$ และ $x \bigcap y = \varnothing$

นิยามที่ 2.7 ค่าความเชื่อมั่น (Confidence) ของกฎ $x \rightarrow y$ คือ ค่าที่แสดงให้เห็นถึงโอกาสการเกิด x แล้วเกิด y ร่วมกัน ค่าความเชื่อมั่นสามารถคำนวณได้จากสมการ 2.4

$$conf(r) = \frac{sup(x \cup y)}{sup(x)} \times 100 \qquad 2.4$$

ตัวอย่างที่ 2.4 จากตารางที่ 2.1 ค่าความเชื่อมั่นของกฎความสัมพันธ์ C⟶D สามารถคำนวณได้จาก $sup(C \cup D) / sup(C) \times 100$ = 3/4x100 = 75% ซึ่งแสดงให้เห็นว่าเมื่อเกิด C จะมีโอกาสเกิด D ถึง 75% ส่วนค่าความเชื่อมั่นของกฎความสัมพันธ์ D⟶C สามารถคำนวณได้จาก $sup(D \cup C) / sup(D) \times 100$ = 3/3x100 = 100% ซึ่งแสดงให้เห็นว่าเมื่อเกิด D มีโอกาสเกิด C ถึง 100%

นิยามที่ 2.8 กฎความสัมพันธ์ *r* เป็นกฎที่สามารถยอมรับได้ก็ต่อเมื่อค่าความเชื่อมั่นของกฎมีค่ามากกว่าหรือเท่ากับค่าความเชื่อมั่นขั้นต่ำ (Minimum Confidence) เขียนแทนด้วย *minconf*

ตัวอย่างที่ 2.5 สมมติกำหนดค่าความเชื่อมั่นขั้นต่ำเท่ากับ 80% กฎความสัมพันธ์ C⟶D ถือว่าเป็นกฎที่ยอมรับไม่ได้ จะถูกตัดทิ้ง เนื่องจากค่าความเชื่อมั่นของกฎความสัมพันธ์ C⟶D มีค่า

น้อยกว่าค่าความเชื่อมั่นขั้นต่ำ ในขณะที่กฎความสัมพันธ์ D⟶C ถือว่าเป็นกฎที่ยอมรับได้ เนื่องมาจากค่าความเชื่อมั่นของกฎ D⟶C มีค่าเท่ากับ 100% ซึ่งมีค่ามากกว่าค่าความเชื่อมั่นขั้นต่ำ

2.3 การจำแนกข้อมูลเชิงความสัมพันธ์ (Associative Classification)

การจำแนกข้อมูลเชิงความสัมพันธ์ (Associative Classification) คือ การจำแนกข้อมูลด้วยการเรียนรู้แบบมีผลเฉลย เพื่อทำนายตัวอย่างข้อมูลที่ยังไม่เคยถูกระบุคลาส ถูกคิดค้นโดย Lui และคณะ [1] เป็นการนำวิธีการขุดค้นข้อมูลที่สำคัญ 2 วิธีการเข้าร่วมกันได้แก่ กฎความสัมพันธ์ (Association Rule) ที่ใช้ในการหาความสัมพันธ์ระหว่างรายการข้อมูล และการจำแนกข้อมูล (Classification) โดยเริ่มจากการใช้กฎความสัมพันธ์สร้างกฎรายการความถี่ทั้งหมดที่ผ่านเงื่อนไขค่าสนับสนุนขั้นต่ำ แล้วคัดเลือกกฎที่ผ่านเงื่อนไขค่าความเชื่อมั่นขั้นต่ำเพื่อสร้าง CARs หลังจากนั้นจึงสร้างตัวจำแนกจาก CARs โดยนิยามที่เกี่ยวข้องดังนี้

กำหนด $A=\{a_1,a_2,..,a_n\}$ เป็นเซตของแอททริบิวต์ทั้งหมดในฐานข้อมูล $I=\{i_1,i_2,..,i_n\}$ เป็นเซตของรายการทั้งหมด $C=\{c_1,c_2,..,c_n\}$ คือ เซตของคลาสทั้งหมด $T=\{t_1,t_2,..,t_n\}$ คือ เซตของแทรนแซกชันทั้งหมด (Transaction) ซึ่งทุกแทรนแซกชันประกอบด้วยเซตย่อยของ I กำหนดให้ x คือ เซตรายการ (Itemset) โดยที่ $x\subseteq I$ และ $g(x)$ เท่ากับแทรนแซกชันที่มีเซตรายการ x โดยที่จำนวนแทรนแซกชันที่มีเซตรายการ x แทนด้วย $|g(x)|$

นิยามที่ 2.9 เซตรายการ (Itemset) หมายถึง รายการที่อธิบายแอททริบิวต์ a_i และค่า i_j ที่จัดเก็บในแอททริบิวต์ a_i เขียนแทนด้วย $\langle a_i,i_j\rangle$

นิยามที่ 2.10 กฎรายการ (Ruleitem) หมายถึง เซตรายการ $\langle a_i,i_j\rangle$ ใน d_j สัมพันธ์กับคลาส c_k เขียนแทนด้วย $\langle a_i,i_j\rangle\rightarrow c_k$

ตัวอย่างที่ 2.6 จากตารางที่ 2.2 ในแทรนแซกชันหมายเลข 4 เซตรายการ $(\langle A,a1\rangle)$ สัมพันธ์กับ C เขียนแทนด้วย $(\langle A,a1\rangle)\rightarrow C$

ตารางที่ 2.2 ตัวอย่างข้อมูล 2

แทรนแซกชัน	A	B	C	D	Class
1	a1	b1	c1	d1	A
2	a1	b2	c1	d2	B
3	a2	b3	c2	d3	A
4	a1	b2	c3	d3	C
5	a1	b2	c1	d3	C

นิยามที่ 2.11 ความยาวของกฎรายการ คือ จำนวนรายการที่ปรากฏอยู่ในเซตรายการ *x*

ตัวอย่างที่ 2.7 กฎรายการ $\langle(A,a1),(B,b2),(D,d3)\rangle \rightarrow C$ ประกอบด้วย $(\langle A,a1\rangle)$ $(\langle B,b2\rangle)$ และ $(\langle D,d3\rangle)$ ดังนั้นกฎรายการดังกล่าวมีความยาวของกฎรายการเท่ากับ 3

นิยามที่ 2.12 ค่าสนับสนุนสัมบูรณ์ของกฎรายการ r คือ ความถี่ของการเกิดกฎ r หรือจำนวนแทรนแซกชันที่พบ r แทนด้วย $|g(r)|$ ดังสมการ 2.5

$$sup(r) = g(r) \tag{2.5}$$

นิยามที่ 2.13 ค่าสนับสนุนสัมพันธ์ของกฎ คือ เปอร์เซ็นที่พบกฎ r เมื่อเทียบกับจำนวนแทรนแซกชันทั้งหมด สามารถคำนวณได้ ดังสมการ 2.6

$$sup(r) = \frac{|g(r)|}{|g(T)|} \times 100 \tag{2.6}$$

ตัวอย่างที่ 2.8 กฎรายการ $\langle(A,a1),(B,b2),(D,d3)\rangle \rightarrow C$ ปรากฏอยู่ในแทรนแซกชันที่ 4 และ 5 ดังนั้นค่าสนับสนุนสัมบูรณ์มีค่าเท่ากับ 2 ส่วนค่าสนับสนุนสัมพันธ์มีค่าเท่ากับ 2/5 * 100 = 40% ซึ่งแสดงให้เห็นว่าเซตรายการ $\langle(A,a1),(B,b2),(D,d3)\rangle$ ซึ่งสัมพันธ์กับคลาส C พบถึง 40% จากข้อมูลทั้งหมด

นิยามที่ 2.14 r เป็นกฎรายการความถี่ (Frequent Ruleitem) เมื่อค่าสนับสนุนของ r มากกว่าหรือเท่ากับค่าสนับสนุนขั้นต่ำ

ตัวอย่างที่ 2.9 สมมติกำหนดค่าสนับสนุนขั้นต่ำแบบค่าสนับสนุนสัมบูรณ์และให้มีค่าเท่ากับ 2 กฎรายการ $\langle(A,a1),(B,b2),(D,d3)\rangle \rightarrow C$ เป็นกฎรายการความถี่เนื่องมาจากค่าสนับสนุนของเซตรายการกฎ มีค่าเท่ากับ 2 ซึ่งมีค่าเท่ากับค่าสนับสนุนขั้นต่ำ

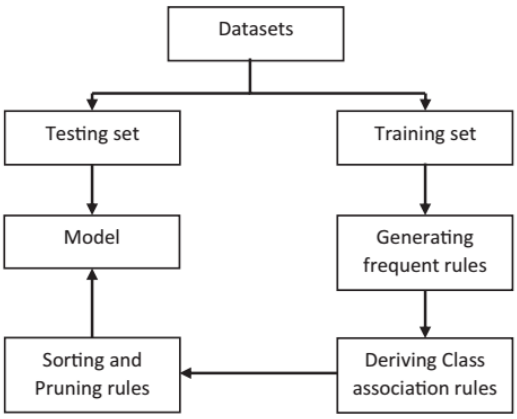
นิยามที่ 2.15 ค่าความเชื่อมั่น (Confidence) ของกฎ $x \rightarrow c$ คือ ค่าที่แสดงให้เห็นถึงโอกาสการเกิด x แล้วเกิด c ร่วมกัน ดังสมการที่ 2.7

$$conf(r) = \frac{sup(x \cup c)}{sup(x)} \times 100 \tag{2.7}$$

ตัวอย่างที่ 2.10 จากตารางที่ 2.1 ค่าความเชื่อมั่นของกฎความสัมพันธ์ *R1* = $\langle(A,a1),(B,b2)\rangle \rightarrow C$ สามารถคำนวณได้จาก $sup(R1) / sup(\langle(A,a1),(B,b2)\rangle) \times 100$ = 2/3 x 100 = 66.67% ซึ่งแสดงให้เห็นว่าเมื่อเกิด $\langle(A,a1),(B,b2)\rangle$ จะมีโอกาสสัมพันธ์กับคลาส C ถึง 66.67%

นิยามที่ 2.16 กฎความสัมพันธ์ระบุคลาส (Class Association Rules) $r: x \rightarrow c$ คือ กฎรายการ r ที่มีค่าความเชื่อมั่นของกฎมากกว่าหรือเท่ากับค่าความเชื่อมั่นขั้นต่ำซึ่งกำหนดโดยผู้ใช้

วิธีการทำงานทั่วไปของการจำแนกข้อมูลเชิงความสัมพันธ์สามารถอธิบายได้ดังรูปที่ 2.2



รูปที่ 2.2 การทำงานทั่วไปของการจำแนกข้อมูลเชิงความสัมพันธ์
ที่มา [8]

2.4 การเรียงกฎ (Rule Sorting)

การเรียงกฎเป็นการจัดลำดับความสำคัญของกฎ โดยกฎที่มีลำดับความสำคัญสูง ซึ่งสามารถครอบคลุมข้อมูลได้มากจะถูกพิจารณาเป็นลำดับแรกในการคัดเลือกกฎ ข้อมูลใดที่กฎครอบคลุมแล้วจะถูกตัดออกเพื่อลดเวลาในการพิจารณากฎอื่น ๆ ดังนั้นหากสามารถครอบคลุมข้อมูลทั้งหมดได้เร็วจะส่งผลต่อประสิทธิภาพโดยรวมด้วย ขั้นตอนวิธี CBA ใช้วิธีการเรียงกฎที่มีค่าความเชื่อมั่นสูงก่อน หากมีค่าความเชื่อมั่นเท่ากันเรียงกฎที่มีค่าสนับสนุนสูงก่อน หากค่าสนับสนุนยังเท่ากันกฎที่ถูกสร้างก่อนจะถูกเรียงลำดับก่อนดัง รูปที่ 2.3 ขั้นตอนวิธี CMAR [13] มีวิธีการเรียงลำดับกฎแตกต่างจาก -ขั้นตอนวิธี CBA เล็กน้อยในขั้นตอนที่สาม โดยหากค่าสนับสนุนและค่าความเชื่อมั่นเท่ากัน จะเลือกเรียงกฎที่มีจำนวนแอททริบิวต์น้อยกว่าขึ้นมาก่อนดังรูปที่ 2.4 ขั้นตอนวิธี MMAC (Multiple-class Multiple-label Associative Classification) [36] ใช้วิธีการเรียงข้อมูลเช่นเดียวกับขั้นตอนวิธี CBA ขั้นตอนวิธี FACA ใช้การเรียงกฎตามจำนวนเซตรายการจากจำนวนน้อยไปมาก หากมีจำนวนเซตรายการเท่ากันจึงเรียงตามค่าความเชื่อมั่นจากมากไปน้อย หากค่าความเชื่อมั่นเท่ากันจึงเรียงตามค่าสนับสนุนจากมากไปน้อย ท้ายที่สุดจึงเรียงตามลำดับการปรากฏของดังรูปที่ 2.5 ขั้นตอนวิธี PCAR (Predictability-Based Class Collative Class Association Rules) แตกต่างจากขั้นตอนวิธีอื่นเป็นอย่างมากเนื่องจากเรียงกฎโดยค่าความสามารถในการทำนายก่อนค่าความเชื่อมั่นและค่าสนับสนุน ดังรูปที่ 2.6 ขั้นตอนวิธี ACPRISM (Associative Classification Based on PRISM) มีวิธีการเรียงลำดับแตกต่างจากวิธีอื่น โดยเรียงลำดับการสร้างกฎเท่านั้น ขั้นตอนวิธี WCBA (Weighted Classification Based on Association Rules) [37] เรียงลำดับด้วยค่าฮาร์-โมนิก ของกฎที่มากที่สุดก่อน หลังจากนั้นจึงใช้การเรียงแบบ CBA คือ ค่าความเชื่อมั่น ค่าสนับสนุน และกฎที่เกิดก่อน ในขณะที่ขั้นตอนวิธี MRMCAR (Map-Reduce Multi-Class Classification Based on Association Rule) [38] อนุญาตให้มีการเรียงกฎตามที่ผู้ใช้ต้องการโดยใช้ค่าสนับสนุน ค่าความเชื่อมั่น และจำนวนกฎ ซึ่งปรับเปลี่ยนได้ถึง 5 รูปแบบ

If Confidence(*R1*) > Confidence (*R2*) then

→ → *R1* have higher rank

Else If Confidence(*R1*) = Confidence (*R2*) and Support(*R1*) > Support(*R2*) then

→ → *R1* have higher rank

Else If Confidence(*R1*) = Confidence (*R2*) and Support(*R1*) = Support(*R2*)

and R1 has generated before R2 then *R1* have higher rank

Else *R2* have higher rank

รูปที่ 2.3 การเรียงลำดับกฎของขั้นตอนวิธี CBA และ MMAC

If Confidence(*R1*) > Confidence (*R2*) then

→ → *R1* have higher rank

Else If Confidence(*R1*) = Confidence (*R2*) and Support(*R1*) > Support(*R2*) then

→ → *R1* have higher rank

Else If Confidence(*R1*) = Confidence (*R2*) and Support(*R1*) = Support(*R2*)

And Size(*R1*) < Size(*R2*) then *R1* have higher rank

Else *R2* have higher rank

รูปที่ 2.4 การเรียงลำดับกฎของขั้นตอนวิธี CMAR และ MCAC

If Size (*R1*) < Size (*R2*) then

→ → *R1* have higher rank

Else If Size (*R1*) = Size (*R2*) and Confidence (*R1*) > Confidence (*R2*) then

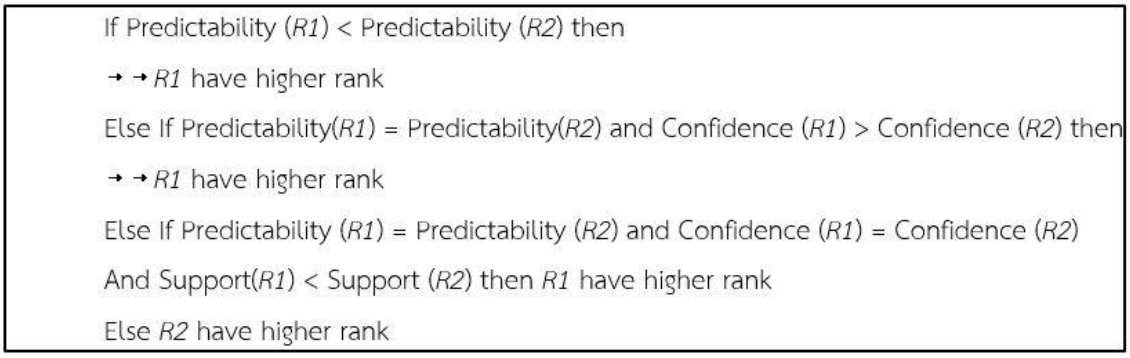
→ → *R1* have higher rank

Else If Size (*R1*) = Size (*R2*) and Confidence (*R1*) = Confidence (*R2*)

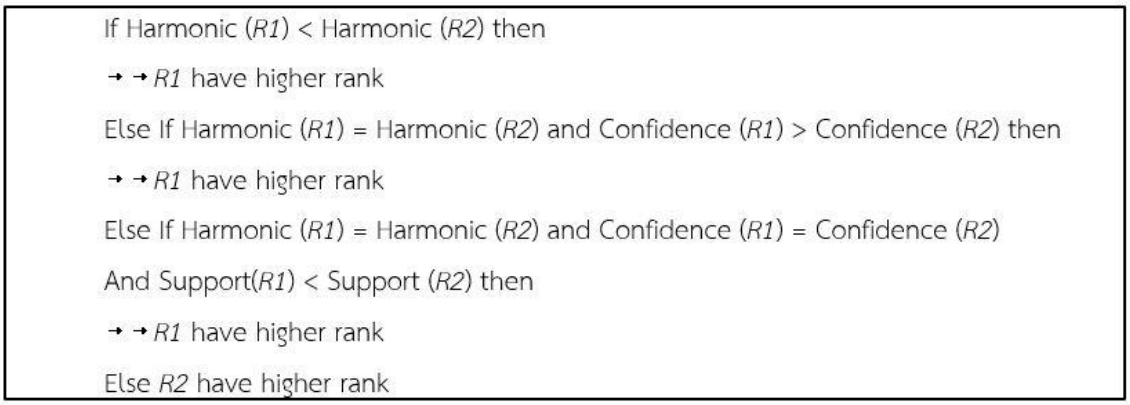
And Support(*R1*) < Support (*R2*) then *R1* have higher rank

Else *R2* have higher rank

รูปที่ 2.5 การเรียงลำดับกฎของขั้นตอนวิธี FACA



รูปที่ 2.6 การเรียงลำดับกฎของขั้นตอนวิธี PCAR



รูปที่ 2.7 การเรียงลำดับกฎของขั้นตอนวิธี WCBA

2.5 การแทนค่าข้อมูล (Data Representation)

การแสดงผลข้อมูลมีผลต่อประสิทธิภาพโดยรวมของการจำแนกข้อมูลเชิงความสัมพันธ์ ขั้นตอนวิธี CBA พัฒนาการจำแนกข้อมูลเชิงความสัมพันธ์โดยใช้การแทนค่าข้อมูลแนวนอน ซึ่งเป็นการแสดงผลข้อมูลที่ได้รับการนิยม หลังจากนั้นขั้นตอนวิธี MMAC [36] ได้นำเสนอการแทนค่าข้อมูลแนวตั้ง จากผลการทดลองแสดงให้เห็นว่าการแทนค่าข้อมูลแนวตั้งใช้หน่วยความจำน้อยกว่าและส่งผลให้เวลาในการประมวลผลขั้นตอนวิธีเร็วขึ้นมากกว่าขั้นตอนวิธีที่ใช้การแทนข้อมูลแนวนอน โดยการแสดงผลข้อมูลมีวิธีการดังนี้

2.5.1 การแทนค่าข้อมูลแนวนอน (Horizontal Data Representation)

การแทนค่าข้อมูลแนวนอน เป็นการจัดรูปแบบข้อมูลชุดสอนก่อนการประมวลผล โดยแต่ละแถวข้อมูลประกอบด้วย หมายเลขแทรนแซกชัน (TID) และรายการข้อมูลที่บรรจุในแทรนแซกชันนั้น ๆ ตารางที่ 2.3 แสดงรูปแบบการแทนค่าข้อมูลหลังจากจัดการข้อมูลตามแนวนอน

ตารางที่ 2.3 การจัดข้อมูลตามแนวนอน

หมายเลขแทรนแซกชัน	รายการ
1	Sugar, Cola
2	Beer
3	Sugar, Beer
4	Cola
5	Sugar

2.5.2 การแทนค่าข้อมูลแนวตั้ง (Vertical Data Representation)

การแทนค่าข้อมูลแนวตั้ง ได้รับการนำเสนอครั้งแรกโดย Zaki และคณะ[33] โดยแต่ละคอลัมน์หมายถึงรายการข้อมูลในชุดข้อมูล ข้อมูลภายในแต่ละคอลัมน์หมายถึงหมายเลขแทรนแซกชันที่ปรากฏรายการข้อมูลนั้น ตารางที่ 2.4 แสดงรูปแบบการแทนค่าข้อมูลหลังจากจัดการข้อมูลตามแนวตั้ง

ตารางที่ 2.4 การจัดข้อมูลตามแนวตั้ง

Sugar	Beer	Cola
1	2	1
3	3	4
5		

การแทนค่าข้อมูลแนวตั้งสามารถลดการอ่านข้อมูลหลายครั้งซึ่งเกิดขึ้นในการแทนค่าข้อมูลแนวนอน โดยใช้การอ่านชุดข้อมูลเพียงครั้งเดียวแล้วจัดข้อมูลให้อยู่ในรูปแบบเซตหมายเลขแทรนแซกชันซึ่งมีการใช้โครงสร้างข้อมูลที่ไม่ซับซ้อน ทำให้สามารถค้นหาเซตรายการความถี่ได้ง่ายด้วยใช้การดำเนินการอินเทอร์เซกชัน [39]

2.5.3 การดำเนินการเซตผลต่าง (Different Sets)

Zaki และคณะ [33] นำเสนอวิธีการหาเซตรายการความถี่ด้วยเซตผลต่าง (Diffsets) ซึ่งเป็นการแทนค่าข้อมูลแนวตั้งเป็นครั้งแรก วิธีการนี้ให้ความสนใจกับหมายเลขแทรนแซกชันที่ไม่บรรจุเซตรายการนั้น โดยแสดงให้เห็นว่าเทคนิคนี้สามารถลดขนาดหน่วยความจำที่ใช้จัดเก็บผลลัพธ์ลงได้อย่างมาก

นิยามที่ 2.17 เซตผลต่างของ p เขียนแทนด้วย $d(p)$ หมายถึง รายการหมายเลขแทรน-แซกชันที่ไม่บรรจุเซตรายการ p

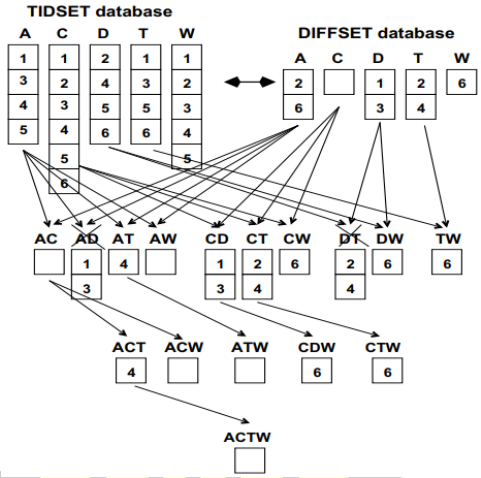
นิยามที่ 2.18 ค่าสนับสนุนโดยเซตผลต่างสำหรับเซต 1 รายการ คำนวณโดยลบจำนวนแทรนแซกชันทั้งหมดในชุดข้อมูลด้วยเซตผลต่างของเซตรายการนั้นของ p ดังสมการที่ 2.8

$$sup(p) = |D| - |d(p)| \quad 2.8$$

นิยามที่ 2.19 ค่าสนับสนุนโดยเซตผลต่างสำหรับเซต *k* รายการ คำนวณโดยลบจำนวนแทรนแซกชันในเซตผลต่างของเซต *k-1* รายการด้วยจำนวนจำนวนแทรนแซกชันในเซตผลต่างของ *k* รายการ ดังสมการที่ 2.9

$$sup(p\ q) = sup(p) - d(p\ q) \quad 2.9$$

ตัวอย่างการหาเซตรายการความถี่ด้วยเซตผลต่าง แสดงดังรูปที่ 2.8 สมมุติข้อมูลตัวอย่างมี 6 แทรนแซกชันและค่าสนับสนุนขั้นต่ำเท่ากับ 2 รายการ A ปรากฏในแทรนแซกชันที่ 1 3 4 และ 5 ดังนั้นเซตผลต่างของ A คือ 2 ค่าสนับสนุนของรายการ A คำนวณจากจำนวนรายการทั้งหมดลบด้วยจำนวนสมาชิกในเซตผลต่างของ A ดั้งนั้น ค่าสนับสนุนของ A คือ 6-2 = 4 รายการ A ผ่านค่าสนับสนุนขั้นต่ำ เซตผลต่างระหว่าง A และ D เกิดจากผลต่างของสมาชิกเซตผลต่างทั้งสองรายการ คือ 4 ค่าสนับสนุนของเซตรายการ AD คำนวณจากค่าสนับสนุนของ A ลบด้วยจำนวนสมาชิกเซตผลต่างของ AD ดังนั้นค่าสนับสนุนของ AD คือ 2 – 1 = 1 รายการ AD ไม่ผ่านค่าสนับสนุนขั้นต่ำ



รูปที่ 2.8 การหาเซตรายการความถี่ด้วยเซตผลต่าง
ที่มา [33]

วิธีการเซตผลต่างถูกประยุกต์ใช้กับการขุดค้นข้อมูลที่หลากหลาย การจำแนกข้อมูลด้วยกฎความสัมพันธ์ ได้แก่ ขั้นตอนวิธี FACA (Fast Associative Classification Algorithm) [5] ใช้เซตผลต่างคำนวณค่าสนับสนุนและค่าความเชื่อมั่นเพื่อสร้างกฎคู่แข่ง การขุดค้นรายการความถี่ (Frequent Itemset Mining) ได้แก่ ขั้นตอนวิธี dFin (Diffnodeset Frequent Itemset Using Nodesets) [34] ใช้เทคนิคการขุดค้นข้อมูลในแนวตั้งเพื่อสร้างเซตรายการความถี่อย่างรวดเร็ว ขั้นตอนวิธี CAR-Miner-Diff [12] LD-CARM-IC [15] ซึ่งจัดเป็นการขุดค้นกฎความสัมพันธ์ระบุคลาสประยุกต์เซตผลต่างเข้ากับขั้นตอนวิธีเดิมของตนเองผลการทดลองพบว่ามีความเร็วในการสร้างกฎมากขึ้น

2.6 การวัดประสิทธิภาพการจำแนก (Evaluation)

การวัดประสิทธิภาพการจำแนกอธิบายได้โดยใช้เมทริกซ์ความสับสน (Confusion Matrix) ซึ่งเป็นตารางที่จำนวนแถวเท่ากับจำนวนคอลัมน์ โดยจำนวนของคอลัมน์และแถวขึ้นกับจำนวนคลาสที่พิจารณา เช่น ในชุดข้อมูลมีจำนวน 2 คลาส ทำให้เมทริกซ์ความสับสน มีขนาด 2x2 โดยที่ข้อมูลด้านคอลัมน์ เป็นคลาสที่อยู่ในข้อมูลชุดสอน (Actual) และข้อมูลด้านแถวเป็นคลาสที่ทำนายได้จากแบบจำลอง (Predict) สมมุติให้ข้อมูลชุดสอนมีจำนวน 2 คลาส คือ X และ Y เมทริกซ์ความสับสนมีลักษณะดังตารางที่ 2.5

ตารางที่ 2.5 เมทริกซ์ความสับสน (Confusion Matrix)

Actual / Predict	Class X	Class Y
Class X	*a*	*b*
Class Y	*c*	*d*

โดยที่ a (True Positive) คือ จำนวนข้อมูลที่ทำนายถูกว่าเป็นคลาส X

b (False Positive) คือ จำนวนข้อมูลที่ทำนายว่าเป็นคลาส X แต่คำตอบคือ Y

c (False Negative) คือ จำนวนข้อมูลที่ทำนายว่าเป็นคลาส Y แต่คำตอบคือ X

d (True Negative) คือ จำนวนข้อมูลที่ทำนายถูกว่าเป็นคลาส Y

2.6.1 ค่าความถูกต้อง (Accuracy)

การวัดค่าความถูกต้องของการจำแนกเป็นการวัดประสิทธิภาพโดยรวมในการจำแนก ซึ่งคำนวณจากจำนวนข้อมูลที่ทำนายคลาสถูกต้อง หารด้วยจำนวนการทำนายทั้งหมดดังสมการที่ 2.10

$$Accuracy = \frac{a+d}{a+b+c+d} \qquad 2.10$$

2.6.2 ค่าความแม่นยำ (Precision)

ค่าความแม่นยำ คือ ความสามารถของแบบจำลองที่สามารถระบุเฉพาะข้อมูลที่เกี่ยวข้องกับคลาสที่พิจารณา การวัดค่าความแม่นยำของการจำแนกเป็นการวัดโดยแยกพิจารณาทีละคลาส คำนวณจากจำนวนข้อมูลที่ทำนายคลาสที่พิจารณาถูกต้อง หารด้วยจำนวนข้อมูลที่ทำนายคลาสที่พิจารณาถูกต้องบวกกับจำนวนครั้งที่ทายคลาสที่พิจารณาเป็นคลาสอื่น ๆ ดังสมการที่ 2.11 และ 2.12 ตามลำดับ

$$Precision_X = \frac{a}{a+b} \qquad 2.11$$

$$Precision_Y = \frac{d}{c+d} \quad 2.12$$

2.6.3 ค่าความระลึก (Recall)

ค่าความระลึก แสดงถึง ความสามารถของแบบจำลองในการค้นหากรณีที่เกี่ยวข้องทั้งหมดกับคลาสที่พิจารณา การวัดค่าความความระลึกของการจำแนกเป็นการวัดโดยแยกพิจารณาทีละคลาส คำนวณจากจำนวนครั้งการทำนายถูกในคลาสที่พิจารณา หารด้วยผลรวมของจำนวนครั้งการทำนายถูกในคลาสที่พิจารณากับจำนวนครั้งที่ทำนายผิดในคลาสที่ไม่ได้พิจารณา ดังสมการที่ 2.13 และ 2.14 ตามลำดับ

$$Recall_X = \frac{a}{a+c} \quad 2.13$$

$$Recall_Y = \frac{d}{b+d} \quad 2.14$$

2.6.4 ค่าเฉลี่ยประสิทธิภาพโดยรวม (F-Measure)

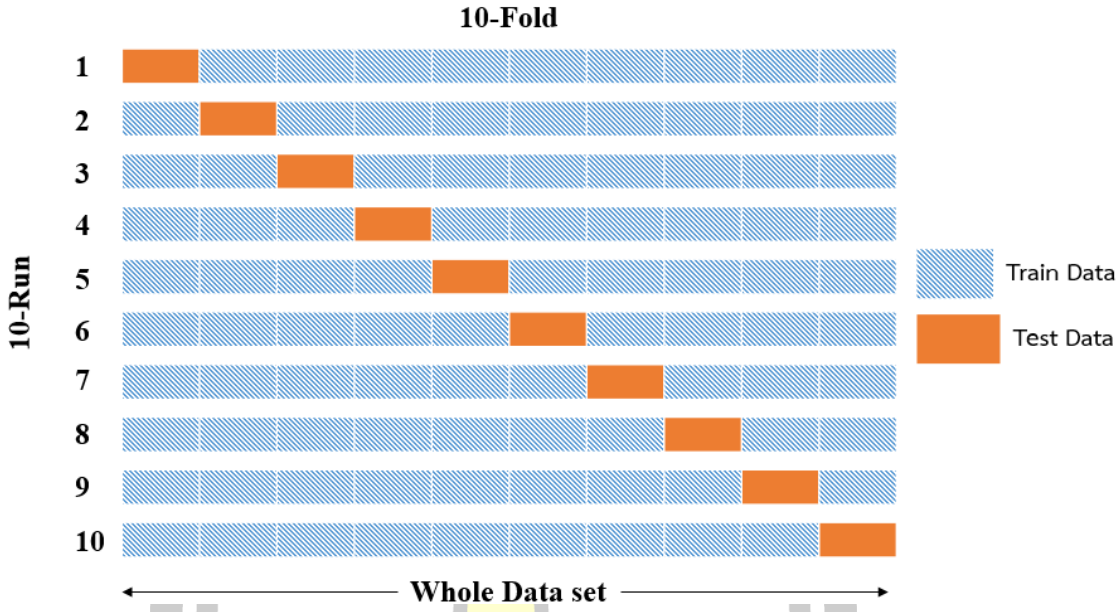
ค่าเฉลี่ยประสิทธิภาพโดยรวมหาค่าเฉลี่ยจากค่าความแม่นยำและค่าความระลึก แบบจำลองที่มีประสิทธิภาพดีจะต้องมีค่าความระลึกและค่าความแม่นยำสูงใกล้เคียงกัน การค่าค่าเฉลี่ยประสิทธิภาพโดยรวม แสดงดังสมการ 2.15 และ 2.16 ตามลำดับ

$$F-measure_X = 2 \times \frac{Precision_X \times Recall_X}{Precision_X + Recall_X} \quad 2.15$$

$$F-measure_Y = 2 \times \frac{Precision_Y \times Recall_Y}{Precision_Y + Recall_Y} \quad 2.16$$

2.6.5 การแบ่งข้อมูลเพื่อวัดประสิทธิภาพแบบ K-Fold Cross-Validation

K-Fold Cross-Validation Test คือ การแบ่งข้อมูลออกเป็น K ชุด ทำการทดสอบประสิทธิภาพจำนวน K รอบ โดยแต่ละรอบจะมีข้อมูล K-1 ชุดใช้เป็นชุดสอน และข้อมูล 1 ชุดใช้เป็นชุดทดสอบ เริ่มทดสอบโดยใช้ชุดข้อมูลแรกเป็นตัวทดสอบและข้อมูลชุดที่เหลือเป็นชุดข้อมูลเรียนรู้ดำเนินวิธีการซ้ำจนครบจำนวน K ชุด



รูปที่ 2.9 การแบ่งข้อมูลแบบ 10-Fold Cross-Validation

การแบ่งข้อมูลแบบ 10-Fold Cross-Validation คือ แบ่งข้อมูลออกเป็น 10 ชุด และทำการทดสอบ 10 รอบ โดยรอบแรกข้อมูลชุดที่ 1 จะถูกนำมาทดสอบประสิทธิภาพของแบบจำลองและข้อมูลชุด 2 ถึง 10 จะเป็นชุดสอน ในรอบที่สองข้อมูลชุดที่ 2 จะเป็นชุดทดสอบ ในขณะที่ข้อมูลชุดที่ 1 3 4 จนถึง 10 เป็นชุดสอน กระบวนการนี้จะถูกทำซ้ำจนครบ 10 รอบ แล้วนำค่าประสิทธิภาพที่ได้ในแต่ละรอบมาคำนวณค่าเฉลี่ย วิธีการประเมินผลด้วย K-Fold Cross Validation มีข้อดี คือ ข้อมูลทุกชุดจะถูกนำมาทดสอบประเมินผลแต่มีข้อเสีย คือ ใช้เวลานานในการทดสอบ โดยขึ้นอยู่กับจำนวนชุดที่นำมาทดสอบ

2.7 งานวิจัยที่เกี่ยวข้อง

2.7.1 งานวิจัยที่ใช้พื้นฐานเทคนิค Apriori

Lui และคณะ [1] นำเสนอการรวมกฎความสัมพันธ์และการจำแนกข้อมูลเป็นครั้งแรกด้วย CBA (Classification Based On Associations) โดยนำกฎความสัมพันธ์เข้ามาเพื่อลดปัญหาการสร้างกฎจำนวนที่มากเกินไปของการจำแนก โดยการสร้างกฎรายการที่ผ่านค่าความเชื่อมั่นขั้นต่ำและค่าสนับสนุนขั้นต่ำนำไปสู่การสร้างกฎความสัมพันธ์ระบุคลาสที่มีประสิทธิภาพ อีกทั้งกฎรายการมีรูปแบบ “ถ้า-แล้ว” ซึ่งผู้ใช้งานเข้าใจได้ง่าย เช่น “ถ้าลูกค้าเป็นวัยกลางคนและมีเงินเดือนสูงแล้วจะซื้อสินค้าของบริษัท”โดยแบ่งการทำงานเป็น 2 ขั้นตอน 1) CBA-RG ซึ่งประยุกต์ใช้ขั้นตอนวิธี Apriori เพื่อค้นหากฎรายการทั้งหมด (Ruleitem) ซึ่งอยู่ในรูปแบบ *Itemset* → *y* เมื่อ *Itemset* คือ เซตรายการ และ *y* คือ คลาสพร้อมทั้งหาค่าสนับสนุน เลือกกฎรายการความถี่ที่ผ่านค่าความเชื่อมั่นขั้นต่ำและค่าสนับสนุนขั้นต่ำสร้างเป็นเซตของกลุ่มความสัมพันธ์ระบุคลาส 2) CBA-CB ใช้ขั้นตอนวิธี M2 สร้างตัวจำแนกจากกฎความสัมพันธ์ระบุคลาสที่ได้จากขั้นตอนแรก โดยเรียงลำดับกฎตามค่าความเชื่อมั่นตามด้วยค่าสนับสนุนและลำดับการเกิดของกฎ แต่ขั้นตอนการสร้างตัวจำแนกด้วยวิธีนี้ยัง

พบปัญหาการสิ้นเปลืองหน่วยความจำและเวลาประมวลผลช้า เมื่อต้องค้นหากฎรายการทั้งหมดจากระบบฐานข้อมูลขนาดใหญ่

Abdelhamid และคณะ [4] ศึกษาการตรวจจับเว็บไซต์หลอกลวง (Phishing Website) ด้วย การจำแนกข้อมูลเชิงความสัมพันธ์และนำเสนอขั้นตอนวิธี MCAC โดยให้ความเห็นว่าการจำแนกข้อมูลเชิงความสัมพันธ์ดั้งเดิมสำหรับการจำแนกเว็บไซต์หลอกลวง สามารถจำแนกได้เพียงเป็นเว็บไซต์หลอกลวงหรือไม่เป็น ในกรณีที่บางเว็บไซต์มีลักษณะที่อาจเป็นได้ทั้ง Legitimate และ Phishing ขั้นตอนวิธีดั้งเดิมจะตัดสินว่าเป็นเว็บไซต์หลอกลวงจากค่าสนับสนุนทันทีซึ่งอาจไม่ถูกต้องเสมอไป ขั้นตอนวิธี MCAC ใช้วิธีการกฎ Multi-Label Rule ด้วยการเพิ่มคลาส “น่าสงสัย” (Suspicious) ซึ่งแสดงถึงโอกาสเป็นเว็บไซต์หลอกลวงและแสดงอัตราส่วนระหว่างคลาส Legitimate และ Phishy ให้ผู้ใช้พิจารณาด้วยตนเอง การทดสอบโดยใช้ข้อมูลเว็บไซต์ตัวอย่างจำนวน 1,350 เว็บ จาก PhishTanks คัดเลือกแอททริบิวต์โดยใช้ค่าไคสแควร์ (Chi-Square) ขั้นตอนวิธีเริ่มจากสร้างกฎ Single Label ทั้งหมดแล้วผสานกฎที่มีข้อมูลทางซ้ายของกฎ (LHS) เหมือนกัน สร้างเป็นกฎ Multi-Label งานวิจัยนี้สร้างตัววัดประสิทธิภาพเรียกว่า Any rule และ Label weight โดย Label weight สามารถใช้บอกอัตราส่วนของคลาสที่ทำนายได้ เพื่อให้ผู้ตัดสินใจทราบว่ากฎนี้มีความใกล้เคียงเป็นเว็บไซต์หลอกลวงมากแค่ไหน ผลการทดลองแสดงค่าความถูกต้องเปรียบเทียบกับขั้นตอนวิธี RIPPER, PART, CBA และ MCAR พบว่า MCAC มีค่าความถูกต้องดีกว่า 1.86% 1.24% 4.46% 2.56% และ 0.8% ตามลำดับ ปัจจัยที่ทำให้ค่าความถูกต้องสูงเนื่องจากความสามารถในการระบุหลายคลาส อย่างไรก็ตามปัญหาของ MCAC คือ การผลิตกฎจำนวนมากกว่าขั้นตอนวิธี Rule Induction (Ripper) CBA หรือ C4.5 ซึ่งปัญหานี้สืบทอดจาก Association rules

Alwidian และคณะ [37] แสดงความเห็นว่าการจำแนกข้อมูลเชิงความสัมพันธ์ทั่วไปให้กำหนดความสำคัญของกฎโดยคำนึงจากค่าสนับสนุนและค่าความเชื่อมั่น โดยตั้งสมมุติฐานว่าทุกแอททริบิวต์มีความสำคัญเท่ากันโดยไม่คำนึงถึงการประยุกต์ใช้ในขอบเขตงานจริงโดยเฉพาะงานทางด้านการแพทย์ซึ่งสามารถระบุได้ว่าข้อมูลใดมีความสำคัญต่อความถูกต้องในการทำนาย ผู้วิจัยจึงนำเสนอขั้นตอนวิธี WCBA (Weighted classification based on association rules) โดยกำหนดค่าน้ำหนักให้กับแอททริบิวต์เพื่อบ่งบอกความสำคัญของแอททริบิวต์นั้น แอททริบิวต์ที่มีความสำคัญในการเกิดมะเร็งทรวงอกจะถูกกำหนดน้ำหนักความสำคัญโดยผู้เชี่ยวชาญ และใช้ค่าฮาร์โมนิก (Harmonic mean หรือ HM) ในการตัดกฎและสร้างกฎ ขั้นตอนวิธีเริ่มจากกำหนดน้ำหนักแอททริบิวต์ (Attribute Weight) โดยผู้เชี่ยวชาญ ค่าน้ำหนักสนับสนุนของกฎ (Weighted Support) คำนวณจากค่าเฉลี่ยน้ำหนักแอทริบิวต์ทั้งหมดในกฎ (Weight) คูณกับค่าสนับสนุนของกฎ หากค่าน้ำหนักสนับสนุนของกฎมากกว่าหรือเท่ากับค่าสนับสนุนขั้นต่ำ เมื่อสร้างกฎที่มี 1 เซตรายการได้แล้วจึงสร้างกฎลำดับชั้นถัดด้วยเทคนิค Apriori เมื่อได้กฎทั้งหมดแล้วจึงคำนวณค่าความเชื่อมั่นและค่าฮาร์โมนิก ซึ่งคำนวณดังสมการที่ 2.17

$$HM(r) = \frac{2 \times (WeightedSupport(r) \times conf(r))}{WeightedSupport(r) + conf(r)} \qquad 2.17$$

หลังจากนั้นเรียงลำดับกฎจากค่ามากไปค่าน้อยโดยใช้ค่าฮาร์โมนิก ค่าความเชื่อมั่น ค่าสนับสนุนและกฎที่เกิดขึ้นก่อน เมื่อเรียงลำดับเสร็จ เทคนิค M1 ถูกนำมาใช้เพื่อแบ่งกฎเป็น 2 กลุ่ม คือ กลุ่มกฎเข้มแข็ง (Strong Rule) และกลุ่มกฎสำรอง (Spare Rules) ด้วย วิธีการนี้เป็นวิธีการเดียวกับขั้นตอนวิธี CBA หลังจากนั้นขั้นตอนวิธี WCBA ใช้การตัดกฎแบบครอบคลุมฐานข้อมูล (Database Coverage) โดยนำตัวอย่างข้อมูลจับคู่กับกฎเฉพาะส่วนบรรพบุรุษของกฎ (Antecedent):กฎทั้งหมดที่จับคู่ได้แบ่งกฎตามคลาส คลาสใดมีค่าเฉลี่ยฮาร์โมนิก สูงกว่าจะถูกเลือกเป็นคำตอบ หากในกฎกลุ่มเข้มแข็งไม่พบกฎที่จับคู่กับตัวอย่างได้ ขั้นตอนวิธีจะค้นหาคลาสที่มีความถี่มากที่สุดในกลุ่มกฎสำรองเพื่อเลือกเป็นคลาสพื้นฐาน (Default class) ให้กับตัวอย่างข้อมูล การทดสอบเปรียบเทียบกับขั้นตอนวิธี CBA CMAR MCAR FACA และ ECBA โดยกำหนดค่าสนับสนุนขั้นต่ำอยู่ระหว่าง 10-30% ซึ่งอ้างอิงค่าจากผลการวิจัยที่ผ่านมา ค่าความเชื่อมั่นขั้นต่ำกำหนดค่า 50 % จากผลการทดลองกับชุดข้อมูลการเกิดซ้ำของมะเร็งเต้านม (Breast Cancer Recurrences) และชุดข้อมูลการวินิจฉัยโรคมะเร็ง (Breast Cancer Diagnosis) จาก UCI พบว่า WCBA มีค่าความถูกต้องมากกว่าขั้นตอนวิธีทั้ง 5 ที่นำมาเปรียบเทียบ เนื่องจากการกำหนดค่าน้ำหนักและเรียงจัดเรียงโดยอ้างอิงค่าฮาร์โมนิก นอกจากนั้นการจำแนกข้อมูลเชิงความสัมพันธ์อื่นปรากฏแอททริบิวต์ที่ไม่มีความสำคัญในการทำนายในกฎที่ถูกสร้างส่งผลให้ความถูกต้องในการทำนายลดลง ตัวอย่างเช่น แอททริบิวต์ Breast และ Breast Quad เมื่อพิจารณาในกฎที่ WCBA สร้างได้สองกฎแรกพบว่าเป็นกฎที่การจำแนกข้อมูลเชิงความสัมพันธ์ตัวอื่นไม่สามารถสร้างได้เนื่องจากกฎดังกล่าวไม่ผ่านค่าสนับสนุนขั้นต่ำหรือค่าความเชื่อมั่นขั้นต่ำ แต่สำหรับ WCBA ซึ่งอ้างอิงค่าฮาร์โมนิกพิจารณากฎนี้ว่ามีความสำคัญสูงและสามารถสร้างกฎที่สำคัญได้ ด้วยวิธีการนี้ WCBA ทำให้เกิดการใช้ค่าฮาร์โมนิกในการตัดกฎซึ่งยังไม่มีงานวิจัยใดใช้มาก่อน นอกจากนั้นกฎที่ไม่สามารถครอบคลุมฐานข้อมูลยังไม่ถูกตัดออกในทันทีแต่สามารถช่วยในการทำนายข้อมูลในกรณีที่กลุ่มกฎที่เข้มแข็งไม่สามารถทำนายข้อมูลได้ ผู้วิจัยทดลองปรับปรุงขั้นตอนวิธีที่นำมาเปรียบเทียบทั้ง 5 ด้วยประยุกต์การให้ค่าน้ำหนักกับแอททริบิวต์ ผลปรากฏว่าทั้ง 5 ขั้นตอนวิธีมีค่าความถูกต้องสูงขึ้น

RAJAB นำเสนอขั้นตอนวิธี APR ซึ่งนำเสนอการประเมินกฎแบบใหม่และการเรียงกฎแบบพลวัตที่เพิ่มประสิทธิภาพการจำแนก APR เรียงกฎโดย ลำดับกฎที่มีค่าความเชื่อมั่นสูงที่สุดแล้วเรียงโดยใช้ค่าสนับสนุนที่สูงกว่าหากค่าความเชื่อมั่นระหว่างกฎรายการเท่ากัน สุดท้ายหากค่าความเชื่อมั่นและค่าสนับสนุนของกฎสองกฎเท่ากัน ขั้นตอนวิธี APR เลือกเรียงลำดับกฎที่จำนวนเซตรายการที่มากกว่าขึ้นมาก่อน หลังจากนั้นชุดข้อมูลตัวอย่างถูกทดสอบกับกฎตามลำดับ กฎที่สามารถจับคู่กับตัวอย่างข้อมูลจะถูกเพิ่มไปยังตัวจำแนก ตัวอย่างข้อมูลที่เกี่ยวข้องกับกฎดังกล่าวจะถูกลบในทันที กฎอื่นที่เกี่ยวข้องกับตัวอย่างข้อมูลที่ถูกลบจะถูกปรับปรุงค่าสนับสนุนและค่าความเชื่อมั่นของกฎ หากกฎใดค่าสนับสนุนไม่ผ่านค่าสนับสนุนขั้นต่ำจะถูกตัดจากการประเมิน หลังจากนั้นขั้นตอนวิธีจะเรียงลำดับกฎใหม่ทั้งหมด ด้วยวิธีการประเมินกฎดังกล่าวส่งผลให้ตัวจำแนกมีขนาดเล็กลง ผลการทดลองแสดงว่าขั้นตอนวิธี APR สามารถสร้างจำนวนกฎที่น้อยกว่าและมีค่าความถูกต้องสูงกว่าขั้นตอนวิธี C4.5 RIPPER และ CBA อย่างไรก็ตามขั้นตอนวิธี APR ยังใช้วิธีการสร้างกฎคู่แข่งที่เป็นไปได้ทั้งหมดก่อนการตัดกฎซึ่งอาจพบอุปสรรคเช่นเดียวกับการจำแนกข้อมูลด้วยกฎความสัมพันธ์อื่นนั่นคือจำนวนกฎมหาศาลเมื่อต้องค้นหากฎจากชุดข้อมูลขนาดใหญ่หรือตั้งค่าสนับสนุนต่ำมาก

งานวิจัยที่โดดเด่นในกลุ่มงานการจำแนกเชิงความสัมพันธ์ ต่างนำเสนอวิธีการในการสร้างแบบจำลองที่น่าสนใจ เช่น ขั้นตอนวิธี CBA ใช้การจัดเรียงข้อมูลแนวนอนและวิธีการ Apriori เพื่อสร้างกฎ ขั้นตอนวิธี MCAC ทำนายคลาสของข้อมูลด้วยการนับจำนวนคลาสของกฎที่สามารถทำนายชุดข้อมูลได้ ขั้นตอนวิธี WCBA นำเสนอการสร้างค่าถ่วงน้ำหนักเพื่อหา ความสำคัญที่แท้จริงของแอททริบิวต์ ขั้นตอนวิธี APR นำเสนอวิธีการเรียงกฎที่มีประสิทธิภาพ อย่างไรก็ตามงานวิจัยเหล่านี้ใช้เทคนิค Apriori ซึ่งอุปสรรคสำคัญ คือ กฎคู่แข่งจำนวนมากจะถูกสร้างขึ้นเสียก่อน แล้วกฎที่ซ้ำซ้อนและไม่จำเป็นจะถูกตัดออกในภายหลัง

2.7.2 งานวิจัยที่ใช้พื้นฐานโครงสร้างต้นไม้

ขั้นตอนวิธี CBA ใช้วิธีการสร้างกฎคู่แข่งเพื่อตรวจสอบกฎที่ได้ว่าสามารถใช้ในการทำนายได้หรือไม่ เพื่อแก้ปัญหาการสร้างกฎคู่แข่งจำนวนมาก Li และคณะ [13] นำเสนอ ขั้นตอนวิธี CMAR (Classification Based On Multiple Association Rules) โดยให้ความเห็นว่าวิธีการที่มุ่งเน้นเลือกเฉพาะกฎที่มีค่าความเชื่อมั่นสูงอาจทำให้กฎที่สำคัญไม่ถูกนำไปใช้งานซึ่งส่งผลต่อความถูกต้องของการจำแนก CMAR ประยุกต์วิธีการเอฟพีโกรธ (FP Growth) และประยุกต์ใช้โครงสร้างต้นไม้ Frequent Pattern (FP-Tree) โดยอ่านฐานข้อมูลเพื่อค้นหารายการความถี่แล้วเรียงลำดับจากความถี่มากไปน้อยเรียกว่า F-list สร้าง FP-Tree โดยอ่านข้อมูลทีละรายการในแต่ละแทรนแซกชันหากพบรายการความถี่จึงนำรายการนั้นบรรจุในโหนดพร้อมคลาสแล้วเพิ่มลงใน FP-Tree พร้อมกับสร้างข้อมูล Linked-list เชื่อมต่อ Header table กับทุกโหนด หลังจากนั้นแบ่งเซตย่อยตามข้อมูลใน FP-list เพื่อสร้าง Projected Database สำหรับค้นหารายการความถี่ของแต่ละเซตย่อยเริ่มจากรายการความถี่น้อยที่สุด กระบวนการนี้สร้างกฎรายการความถี่ กฎและพร้อมนับค่าสนับสนุนในครั้งเดียว กฎทั้งหมดที่สร้างได้จะบรรจุไว้ใน CR-Tree เพื่อความรวดเร็วในการค้นหากฎที่ต้องการ กฎที่ใดที่มีเซตรายการเหมือนกันจะใช้โหนดร่วมกัน (Shared Prefix) ส่งผลให้ประหยัดพื้นที่ในการจัดเก็บข้อมูล ผลการทดลองพบว่า CR-Tree ประหยัดพื้นที่ 50-60% CMAR จัดการความซ้ำซ้อนของกฎโดยเรียงลำดับกฎจากมากไปน้อยด้วยค่าความเชื่อมั่น ค่าสนับสนุน และเรียงกฎที่มีจำนวนเซตรายการน้อยกว่าขึ้นก่อน วิธีนี้ทำให้กฎที่เฉพาะเจาะจงและความเชื่อมั่นต่ำมีโอกาสตัดออกมากขึ้น นอกจากนี้ CMAR ใช้ค่าไคสแควร์ถ่วงน้ำหนัก (Weighted Chi Square) เพื่อเลือกเฉพาะกฎความสัมพันธ์เชิงบวก สุดท้ายใช้วิธีการครอบคลุมฐานข้อมูล (Database Coverage Method) โดยจับคู่ตัวอย่างข้อมูลกับกฎตามลำดับการเรียง หากกฎใดไม่ครอบคลุมตัวอย่างจะถูกตัดออก กฎที่เหลืออยู่ถูกสร้างเป็นตัวจำแนกที่มีประสิทธิภาพจากผลการทดลองพบว่าขั้นตอนวิธีนี้มีความเร็วในการสร้างกฎมากกว่า CBA และยังมีความถูกต้องมากกว่า อย่างไรก็ตาม Thabtah [40] ให้ความเห็นว่าจุดอ่อนเพียงประการเดียวของ FP-Growth คือโครงสร้าง FP-Tree ที่อาจใหญ่เกินกว่าจะจัดเก็บในหน่วยความจำหลักได้ในระหว่างการประมวลผล โดยเฉพาะอย่างยิ่งในกรณีที่ขุดค้นฐานข้อมูลขนาดใหญ่

Deng และคณะ[41] นำเสนอขั้นตอนวิธี CBC (Condition-Based Classification) โดยเห็นว่าการจำแนกข้อมูลเชิงความสัมพันธ์ที่มีอยู่ในปัจจุบันมีกฎจำนวนมากยากต่อการแปลความหมายโดยผู้ใช้และการเรียงกฎตามลำดับสามารถแปลงให้อยู่รูปแบบต้นไม้ตัดสินใจได้ ซึ่งโครงสร้างต้นไม้

Condition-Based (CBT) เป็นโครงสร้างที่ลดขนาดต้นไม้ตัดสินใจได้ เริ่มต้นด้วยการแปลงกฎที่ได้จากการจำแนกทั้งหมดเป็นเซตของตัวชี้วัดที่แสดงว่าเงื่อนไขของกฎใดตรงกับตัวอย่างข้อมูลทดสอบใดบ้าง หลังจากนั้นจึงใช้เทคนิค Correlation Feature Selection (CFS) เพื่อการคัดเลือกคุณลักษณะ (Feature Selection) กับเซตตัวชี้วัดเพื่อให้ได้เซตย่อยที่ลดลง เงื่อนไขที่เหลืออยู่จะถูกเปลี่ยนให้อยู่ในรูปโครงสร้าง CBT ด้วยเทคนิคเดียวกับ C4.5 หลังจากนั้นนำโครงสร้าง CBT แปลงเป็นกลุ่มเรียงลำดับของกฎตามเงื่อนไข Condition-Based Classifier งานวิจัยใช้ชุดข้อมูลจาก UCI จำนวน 16 ชุด โดยตั้งค่าสนับสนุนขั้นต่ำ 5% และค่าความเชื่อมั่นขั้นต่ำที่ดีที่สุดจากการทดลองเท่ากับ 80% จากผลการทดลองพบว่าสร้างจำนวนกฎได้น้อยกว่า CBA C4.5 [42] และ GARC [43] โดย CBC ลดจำนวนกฎเหลือเพียง 10 กฎโดยเฉลี่ย นอกจากนั้นในชุดข้อมูลไททานิคมีผลการทดลองที่แตกต่างออกไป คือ จำนวนกฎในตัวจำแนกมากกว่าขั้นตอนวิธี C4.5 แต่อัตราผิดพลาดน้อยกว่า โดยกำหนดค่าสนับสนุนขั้นต่ำ 5% และค่าความเชื่อมั่นขั้นต่ำ 50% เนื่องจากคลาสของผู้เสียชีวิตมีข้อมูลตัวอย่างมากกว่าคลาสของผู้รอดชีวิต จึงแสดงให้เห็นว่า CBC มีความสามารถในการจัดการข้อมูลที่ไม่สมดุล (Imbalance Data) ได้ดีกว่า C4.5

อย่างไรก็ตามขั้นตอนวิธี CMAR พบอุปสรรคในการนำข้อมูลจำนวนมากไปสร้างโครงสร้างต้นไม้ในหน่วยความจำซึ่งใช้พื้นที่หน่วยความจำค่อนข้างมาก ในขณะที่ CBC ไม่ใช้กระบวนการคัดเลือกคุณลักษณะที่สำคัญก่อนการสร้างกฎแต่ใช้วิธีสร้างกฎทั้งหมดแล้วจึงคัดเลือกกฎในภายหลัง นอกจากนั้นเวลาในการประมวลยังเพิ่มขึ้นจากขั้นตอนการสร้างตัวจำแนก

2.7.3 งานวิจัยที่ใช้พื้นฐานการแสดงผลแนวตั้งและการอินเทอร์เซกชัน

Abdelhamid และคณะ [44] แสดงให้ความเห็นว่าขั้นตอนการตัดกฎด้วยการจับคู่ (Matching) ที่ใช้ในขั้นตอนวิธี CBA หรือ MCAR มีวิธีการเปรียบเทียบความเหมือนระหว่างกฎกับตัวอย่างข้อมูล โดยต้องตรงกันทั้งเซตรายการซึ่งอยู่ซ้ายมือของกฎ (LHS) และคลาสซึ่งอยู่ทางขวามือของกฎ (RHS) วิธีการนี้ทำให้ตัวจำแนกมีความถูกต้องสูงแต่อาจเกิด Overfitting นอกจากนั้นการเปรียบเทียบคลาสเป็นการจำกัดความสามารถในการพยากรณ์ข้อมูล ด้วยเหตุนี้ขั้นตอนวิธี MAC (Multiclass Associative Classification) จึงใช้การแทนค่าข้อมูลแบบตามแนวตั้ง โดยจัดเก็บรหัสแทรนแซกชันที่รายการนั้นบรรจุอยู่ซึ่งจำนวนแทรนแซกชันแสดงถึงค่าสนับสนุน กฎที่มี 1 รายการซึ่งผ่านค่าสนับสนุนขั้นต่ำ อินเทอร์เซคชัน (Intersection) ระหว่างกฎเพื่อสร้างกฎที่มี 2 รายการ และทำซ้ำจนกว่าจะไม่พบกฎที่ผ่านค่าสนับสนุนขั้นต่ำเป็นเทคนิคการสร้างกฎที่คล้าย Apriori ขั้นตอนวิธีคัดเลือกกฎที่ผ่านค่าความเชื่อมั่นขั้นต่ำ แล้วเรียงกฎตามค่าความเชื่อมั่นและค่าสนับสนุนจากค่ามากไปค่าน้อยหากทั้งสองค่าเท่ากัน กฎที่มีความถี่มากกว่าจะถูกพิจารณาก่อน ในกระบวนการตัดกฎใช้การจับคู่ด้วยการตรวจสอบความเหมือนเพียงแค่เปรียบเทียบ LHS ระหว่างกฎและตัวอย่างข้อมูลเท่านั้น ในขั้นตอนการทำนาย ขั้นตอนวิธี MAC เลือกกฎซึ่งสามารถจับคู่กับ LHS ของตัวอย่างได้ทั้งหมดแล้วจัดกลุ่มโดยแบ่งตามคลาส คลาสใดมีจำนวนกฎมากที่สุดจะถูกกำหนดให้เป็นคลาสของตัวอย่างข้อมูล การทดสอบกับชุดข้อมูล UCI 19 ชุด พบว่า จำนวนกฎที่ได้ลดลงเป็นเท่าตัวเมื่อเทียบกับขั้นตอนวิธี MCAR [45] แต่ค่าความถูกต้องต่ำกว่าเพียงเล็กน้อยเท่านั้น ขั้นตอนวิธี

MAC ให้ค่าความถูกต้องมากกว่า Ripper และ C4.5 0.94% และ 0.72% ตามลำดับ นอกจากนี้ MAC สร้างจำนวนกฎในตัวจำแนกน้อยกว่า MCAR เป็นจำนวน 9.79 กฎ โดยเฉลี่ยทุกชุดข้อมูล

หลังจากนำเสนอขั้นตอนวิธี MCAC แล้ว Abdelhamid และคณะนำการแทนค่าข้อมูลตามแนวตั้ง พัฒนาเป็นขั้นตอนวิธี eMCAC (Enhanted MCAC) [3] ความแตกต่างจาก MCAC นอกจากการจัดรูปแบบข้อมูลตามแนวตั้งแล้ว ยังมีการคำนวณค่าความเชื่อมั่นและค่าสนับสนุนที่แตกต่างกัน โดยค่าความเชื่อมั่นคิดค่าเฉลี่ยระหว่างรายการและคลาส ในขณะที่ค่าสนับสนุนนับรวมทั้งรายการและคลาส ขั้นตอนการค้นหากฎ เริ่มจากค้นหากฎรายการความยาว 1 เซตรายการ (F1) และคำนวณค่าสนับสนุนจากจำนวน TID ที่บรรจุเซตรายการกฎรายการใดปรากฏในหลายคลาส (Multiple Rules) ให้แยกสร้างกฎตามคลาสทั้งหมด หลังจากนั้นใช้การ Intersect F1 ทั้งหมดเพื่อหากฎรายการขนาด 2 เซตรายการ (F2) ขั้นตอนที่สองการสร้างกฎ eMCAC เลือกกฎเฉพาะกฎที่ผ่านค่าสนับสนุนขั้นต่ำและตัดกฎที่ไม่ผ่านออกไป หากกฎซึ่งมีเซตรายการเดียวกันแต่ปรากฏในหลายคลาส ให้พิจารณาว่ากฎนั้นผ่านเกณฑ์ค่าสนับสนุนขั้นต่ำและค่าความเชื่อมั่นขั้นต่ำหรือไม่ หากผ่านให้ยอมรับกฎทั้งสองข้อ ต่างกับขั้นตอนวิธี CBA ซึ่งพิจารณาเลือกกฎที่สามารถครอบคลุมตัวอย่างข้อมูลได้มากที่สุดเท่านั้น ขั้นตอนวิธี eMCAC ใช้การเรียงข้อมูลตามค่าความเชื่อมั่น ค่าสนับสนุน จำนวนกฎในเซตรายการซึ่งเลือกใช้กฎที่เซตรายการมีจำนวนน้อยก่อนสุดท้ายความถี่ของคลาสจึงถูกใช้เรียงในกรณีที่เงื่อนไขก่อนหน้าเท่ากัน การตัดกฎใช้วิธีครอบคลุมฐานข้อมูลโดยจับคู่ตัวอย่างข้อมูลกับกฎที่ตรงกันเฉพาะในส่วนเซตรายการเท่านั้นหากจับคู่กันได้ตัวอย่างข้อมูลจะถูกลบออก กฎจะถูกทำเครื่องหมายให้ทราบว่าสามารถจับคู่ตัวอย่างได้ กระบวนการนี้จะทำซ้ำจนกระทั่งตัวอย่างข้อมูลหรือกฎหมดไป กฎที่ยังเหลืออยู่จะไม่ถูกพิจารณา การทดลองประเมินผลแบบ Any label ซึ่งเลือกกำหนดคลาสอย่างใดอย่างหนึ่งให้กับชุดตัวอย่างและ Label-Weight ซึ่งแสดงอัตราส่วนโอกาสที่จะเป็นคลาสใด ๆ ในกรณีที่เซตรายการสามารถระบุคลาสได้ทั้งมากกว่าหนึ่งคลาสโดยตัดสินคลาสที่มีอัตราส่วนสูงกว่าให้กับชุดข้อมูล นอกจากนั้นยังแสดงอัตราส่วนสำหรับคลาสที่ถูกเลือกด้วย ข้อมูลเว็บไซต์ล่อลวงเหยื่อถูกใช้สำหรับการทดลองซึ่งรวบรวมจากเว็บไซต์ PhishTank.com อีกทั้ง Yahoo Directory และเว็บไซต์ Millersmiles.co.uk ด้วย รวมทั้งสิ้น 5,000 เว็บไซต์ แต่ละเว็บไซต์ประกอบด้วยข้อมูล 27 แอททริบิวต์ ใช้การเลือกแอททริบิวต์ที่มีนัยยะสำคัญด้วยวิธี Chi Square ขั้นตอนวิธีที่นำมาเปรียบเทียบได้แก่ MMAC CBA PART MCAR RIPPER C4.5 ผลการทดลองแสดงถึงความโดดเด่นในการจำแนกเว็บไซต์ล่อลวงเหยื่อของขั้นตอนวิธี eMCAC ในการทดลองแบบ Any Label มีค่าความถูกต้องสูงกว่า RIPPER C4.5 PART CBA และ MCAR ถึง 1.86% 1.24% 4.46% 2.56% และ 0.8% ตามลำดับ ผลการทดลองแบบ Multi-Label เปรียบเทียบกับ MMAC ปรากฏว่า eMCAC ให้ความถูกต้องสูงกว่าไม่ว่าจะเป็นการวัดผลด้วย Any Label หรือ Label-Weight

Hadi และคณะ [5] ศึกษาพบว่ากฎคู่แข่งที่สร้างจากชุดข้อมูลมีจำนวนมากซึ่งใช้เวลาและทรัพยากรสูง จึงสร้าง FACA (Fast Associative Classification Algorithm) เพื่อตรวจจับเว็บไซต์หลอกลวง ใช้แนวทางการจัดรูปแบบข้อมูลตามแนวตั้ง ซึ่งประยุกต์ใช้เซตผลต่าง [33] โดยเริ่มจากการค้นหากฎรายการทั้งหมดพร้อมกับคำนวณค่าสนับสนุนและค่าความเชื่อมั่นโดยคำนวณจากจำนวนผลต่างของแทรนแซกชันที่บรรจุกฎนั้นหักลบจากจำนวนแทรนแซกชันทั้งหมดในกลุ่มข้อมูล FACA นำ

กฎที่ผ่านเกณฑ์ค่าสนับสนุนขั้นต่ำและค่าความเชื่อมั่นขั้นต่ำในครั้งแรก (1-ruleitems) มาผสานรวมกันเป็น 2-ruleitem และค้นหากฎที่ผ่านเกณฑ์ขั้นต่ำต่อไป วิธีการนี้จะดำเนินการซ้ำจนกว่าจะไม่พบกฎใดที่ผ่านเกณฑ์ขั้นต่ำอีก จึงเรียงลำดับกฎเหล่านั้นตามจำนวนแอททริบิวต์ที่บรรจุในกฎจากน้อยไปมาก หากมีจำนวนแอททริบิวต์เท่ากัน กฎที่มีค่าความเชื่อมั่นสูงขึ้นมาก่อน หากยังมีกฎที่ความเชื่อมั่นเท่ากันจึงเลือกกฎที่มีค่าสนับสนุนสูงก่อนและท้ายที่สุดเรียงตามกฎที่ถูกสร้างก่อน ขั้นตอนการทำนายคลาสให้กับข้อมูลตัวอย่างใช้วิธีการ All Exact Match Prediction โดยนับจำนวนคลาสของทุกกฎที่จับคู่กับเซตรายการของชุดตัวอย่างได้ เลือกคลาสที่มีความถี่สูงที่สุดเป็นคลาสคำตอบให้กับตัวอย่างข้อมูล จากผลการทดสอบ FACA แสดงค่าความแม่นยำที่สูงกว่า CBA และ CMAR ถึง 4.2% และ 3.7% ในผลการวัดค่า F measure FACA มีค่าสูงกว่า CBA และ CMAR ถึง 1.2% และ 1.8% ในชุดข้อมูลข้อมูลเว็บไซต์หลอกลวง

Song และ Lee [10] เสนอขั้นตอนวิธี PCAR (Predictability-Based Collective Class Association Rule Mining) ใช้การ Cross Validation ระหว่างข้อมูลชุดสอนและข้อมูลชุดทดสอบเพื่อหาค่าความสามารถในการทำนาย (Predictability-value) ซึ่งผู้วิจัยให้ความเห็นว่าการจำแนกข้อมูลเชิงความสัมพันธ์อื่นให้ความสนใจค่าความเชื่อมั่นหรือค่าสนับสนุนเท่านั้น ไม่ได้สนใจข้อมูลทางสถิติของกฎที่สร้างขึ้นมาว่ามีความสามารถในการทำนายข้อมูลมากเพียงใด เริ่มจากการประเมินกฎรายการด้วยวิธี K-fold validation โดยแบ่งกลุ่มชุดข้อมูลเป็นชุดทดสอบและชุดสอน หลังจากนั้นหากฎที่มีในข้อมูลชุดทดสอบด้วยขั้นตอนวิธี Eclat แล้ว ในขั้นตอนนี้ PCAR จะตัดกฎไม่มีคุณภาพออกด้วยการเปรียบเทียบกฎกับข้อมูลชุดสอน หากกฎไม่สามารถจำแนกข้อมูลใด ๆ ได้จะถูกตัดออก นอกจากนั้นระหว่างกระบวนการนี้กฎจะถูกตรวจสอบความซ้ำซ้อนจากกฎที่ผ่านการตรวจสอบแล้ว เพื่อผลการทดสอบที่หลากหลายผู้วิจัยสร้างขั้นตอนวิธี PCAR2 ซึ่งไม่รวมกระบวนการตัดกฎเข้าไปด้วย เมื่อได้กฎที่ต้องการจึงคำนวณค่าความสามารถในการทำนายสำหรับทุกกฎจากสัดส่วนค่าสนับสนุนของกฎต่อจำนวนข้อมูลในชุดทดสอบ หากกฎใดปรากฏอยู่ในหลายชุดทดสอบ ให้เฉลี่ยค่าความสามารถในการทำนายของกฎ หลังจากนั้นจึงเรียงกฎโดยเริ่มจากค่าความสามารถในการทำนายจาก ค่าความเชื่อมั่น ค่าสนับสนุน จำนวนกฎรายการและความถี่ของกฎ โดยทุกเงื่อนไขการเรียงใช้วิธีเรียงจากค่ามากไปหาค่าน้อย ผลการทดลองด้วยชุดข้อมูลมาตรฐานจาก UCI 16 ชุด พบว่าขั้นตอนวิธี PCAR และ PCAR2 ให้ค่าความถูกต้อง 83.21 และ 83.56 ตามลำดับ ในขณะที่ขั้นตอนวิธี C4.5 Ripper [46] CBA และ MCAR [45] ให้ค่าความถูกต้อง 81.39 81.43 80.00 และ 82.38 ตามลำดับ สรุปได้ว่าการเพิ่มค่าความสามารถในการทำนายทำให้ประสิทธิภาพของการจำแนกเพิ่มขึ้น แต่ในทางตรงกันข้ามด้วยการเพิ่มค่าความสามารถในการทำนายส่งผลให้เวลาในการประมวลผลค่อนข้างสูง ถึงอย่างไรก็ตามผู้วิจัยยังเห็นว่าวิธีนี้เป็นวิธีสากลที่สามารถประยุกต์ใช้กับการจำแนกข้อมูลเชิงความสัมพันธ์ทั่วไปโดยเฉพาะงาน Multi-Class

ขั้นตอนวิธีข้างต้นนำเสนอเทคนิคการสร้างกฎจากวิธีการอินเทอร์เซกชันระหว่างหมายเลขแทรนเซกชันซึ่งสามารถลดปัญหาการอ่านชุดข้อมูลหลายครั้งลงไปได้ อย่างไรก็ตามอุปสรรคของขั้นตอนวิธีเหล่านี้ คือ การสร้างกฎจำนวนมากเพื่อคัดเลือกกฎที่มีประสิทธิภาพในภายหลัง เช่นเดียวกับวิธีการที่ประยุกต์จากเทคนิค Apriori

2.7.4 งานวิจัยที่เน้นการเพิ่มประสิทธิภาพการค้นหากฎรายการ

Sarah และคณะ [28] ให้ความเห็นว่าสิ่งที่ต้องให้ความสำคัญกับเทคนิคการค้นหากฎความสัมพันธ์แบบ Apriori คือ การกำหนดค่าสนับสนุนขั้นต่ำ ซึ่งหากกำหนดค่าสูงกฎจะถูกสร้างน้อยเกินไป หากกำหนดค่าต่ำกฎจะถูกสร้างเป็นจำนวนมากจนการค้นหากฎที่ดีที่สุดอาจเป็นปัญหา ในขณะที่เทคนิค FP-growth มีปัญหาในการค้นหากฎรายการความถี่ในโครงสร้างต้นไม้ FP หลายครั้ง Sarath และคณะ นำเสนอ วิธีการค้นหากฎความสัมพันธ์ด้วยวิธีการ Binary Particle Swarm Optimization โดยไม่จำเป็นต้องกำหนดค่าสนับสนุนขั้นต่ำและค่าความเชื่อมั่นขั้นต่ำ โดยใช้วิเคราะห์ชุดข้อมูลจริงจาก Bank of India ขั้นตอนวิธี BPSO (Binary Particle Swarm Optimization) สำหรับกฎความสัมพันธ์ ค้นหากฎที่ดีที่สุดจำนวน M ซึ่งค่า M ถูกกำหนดโดยผู้ใช้ ก่อนการค้นหากฎความสัมพันธ์ขั้นตอนวิธีเริ่มต้นด้วยการแปลงข้อมูล แทรนแซกชันให้อยู่ในรูปไบนารีซึ่งสามารถเพิ่มความเร็วในการคำนวณ แล้วแทนค่าข้อมูลในรูปแบบ Michigan ด้วยข้อมูล 2 บิต โดยบิตที่ 1 หากเท่ากับ 1 แสดงว่าข้อมูลรายการนี้อยู่ในกฎ หากเท่ากับ 0 แสดงว่ารายการนี้ไม่อยู่ในกฎความสัมพันธ์ ข้อมูลบิตที่ 2 หากมีค่าเท่ากับ 1 แสดงว่ารายการนี้ควรอยู่ทางซ้ายของกฎ (Antecedent) หากมีค่าเท่ากับ 0 จะอยู่ทางขวาของกฎ (Consequence) สิ่งที่แสดงว่ากฎใดดีที่สุดในแต่ละรอบการค้นหา คือ ค่า Fitness ของแต่ละอนุภาค (Particle) จากงานวิจัยนี้คำนวณจาก $sup(A \rightarrow B) \times conf(A \rightarrow B)$ การค้นหากฎความสัมพันธ์ดำเนินการซ้ำ M รอบ กฎที่ดีที่สุดจะถูกค้นพบในแต่ละรอบ การทดลองใช้ชุดข้อมูลจริงจากธนาคารพาณิชย์ในประเทศอินเดียและฐานข้อมูลหนังสือ อาหารและร้านค้าปลีก BPSO ตั้งค่า M=10 ซึ่งแทนค่าจำนวนกฎที่ดีที่สุดที่ต้องการ ผลการทดลองแสดงว่า BPSO สร้างกฎที่ไม่ซ้ำซ้อนและมีความยาวหลากหลาย ในขณะที่ Apriori สร้างกฎที่มีความยาว 3 ขึ้นไปเท่านั้น และสร้างกฎที่มีความซ้ำซ้อนขึ้น นอกจากนั้นเทคนิค FP-Growth สร้างกฎขึ้นมาจำนวนมากและบางกฎไม่มีความจำเป็น ข้อจำกัดของวิธีการนี้ คือ ผู้ใช้ต้องกำหนดตัวเลขของกฎที่ต้องการ จากผลการทดลองกับชุดข้อมูลอาหาร บางกฎมีค่าสนับสนุนและค่ารับรอง

ในกระบวนการสร้างกฎคู่แข่งการจำแนกข้อมูลเชิงความสัมพันธ์ต้องการเวลาและหน่วยความจำหลักสำหรับการประมวลผลสูง ซึ่งเป็นข้อด้อยที่ได้รับสืบทอดจากวิธีการกฎความสัมพันธ์ Thabtah และคณะ นำเสนอการสร้างกฎคู่แข่งแบบคู่ขนาน (Parallel) ด้วยแนวทาง Map Reduce (MR) และพัฒนาขั้นตอนวิธี MRMCAR [38] โดยใช้วิธีการสร้างกฎคู่แข่งโดยการแปลงข้อมูลให้อยู่ในรูปแบบ Line Space ก่อนเข้าสู่ฟังก์ชัน ToFrequent.Mapper เพื่อจัดข้อมูลให้อยู่ในรูปแบบ Item Space แล้วนำเข้าฟังก์ชัน ToFrequent.Reduce ผลลัพธ์ที่ได้ คือ Item ID ซึ่งประกอบด้วย Line ID และ Class ที่เกี่ยวข้องกับ Item ID นั้น โดยสามารถคำนวณค่าสนับสนุนและค่าความเชื่อมั่นได้ในขั้นตอนนี้ Item ที่ผ่านค่าสนับสนุนขั้นต่ำจะถูกนำเข้าฟังก์ชัน ToLine.Mapper และ ToLine.Reduce เพื่อแปลงข้อมูลกลับให้อยู่ในรูปแบบ Line Space ขั้นตอนวิธีตรวจสอบว่ากฎรายการใดที่ปรากฏใน Line อย่างน้อย 1 ครั้ง จะถือว่าเป็นกฎรายการความถี่แน่นอน ดังนั้นขั้นตอนวิธี MRMCAR สร้างกฎคู่แข่งโดยการใช้ประโยชน์จากฟังก์ชัน Reducer ซึ่งต่างจากวิธีการ Apriori กระบวนการข้างต้นนี้จะถูกทำซ้ำจนกว่าจะไม่มีรายการที่ผ่านค่าสนับสนุนขั้นต่ำ การตัดกฎของ MCMCAR มี 3 ขั้นตอนด้วยกันเริ่มจากแปลงข้อมูลให้อยู่ในรูป Line Space พร้อมกับป้อนข้อมูลค่า

น้ำหนักการเรียงข้อมูลสำหรับทุกกฎ แล้วเรียงลำดับกฎโดยผู้ใช้สามารถเลือกวิธีการเรียงข้อมูลได้ถึง 5 ขั้นตอนต่อไป คือ การจับคู่กับข้อมูลตัวอย่าง กฎลำดับสูงสุดที่จับคู่และทำนายข้อมูลได้ถูกต้อง หากไม่มีกฎใดทำนายได้ถูกต้องขั้นตอนวิธีจะเลือกกฎที่ตรงกับรูปแบบข้อมูลมากที่สุด (Partial Matching) ขั้นตอนสุดท้าย คือ การแปลงข้อมูลกลับไปในรูปแบบ Item Space ขั้นตอนวิธี MRMCAR สร้างตัวจำแนกเป็นชุด โดยกำหนดค่าสนับสนุนขั้นต่ำเพียงค่าเดียว แต่ค่าความเชื่อมั่นขั้นต่ำมีมากกว่า 1 ค่า ซึ่งเวลาในการสร้างไม่ได้แตกต่างจากการสร้างตัวจำแนกเพียงชุดเดียว ขั้นตอนการทำนายสามารถใช้การทำนายแบบ Single Rule และ Multiple Rules ได้ โดย Multiple Rules ใช้วิธีการเช่นเดียวกับขั้นตอนวิธี MAC คือ เลือกกฎที่มีส่วน Antecedent ของกฎตรงกับข้อมูลทดสอบแล้วแบ่งกลุ่มตามคลาส คลาสใดมีจำนวนกฎมากกว่าจะถูกเลือกให้เป็นคลาสของข้อมูลทดสอบ ในการทดลองแบ่งข้อมูลแบบ 10 Fold Cross-Validation ประมวลผลแบบลำดับ และประมวลผลแบบกระจาย โดยในการประมวลผลแบบกระจาย แบ่งการทำงานออก 1 เทรด (Thread) ใช้เวลา 93 วินาที แปลงข้อมูล 3 กิกะไบต์ให้อยู่ในรูป Line Space โดยมีจำนวน 10 ล้าน Line เปรียบเทียบผลการทดลองกับ C4.5 C5 Ripper CBA และ MCAR การทดลองโดยใช้ข้อมูล Wine จาก UCI พบว่าเมื่อสร้างตัวจำแนกโดยกำหนดค่าความเชื่อมั่นขั้นต่ำ 20 ค่าที่แตกต่างกันโดยมีค่าอยู่ระหว่าง 0% - 100% ความห่างกันของค่าสนับสนุนเท่ากับ 5% ใช้เวลาในการสร้างตัวจำแนกเฉลี่ยเพียง 294.31 มิลลิวินาที ผลการทดลองแสดงให้เห็นว่าค่าความเชื่อมั่น 5% สร้างกฎรายการได้มากที่สุด อย่างไรก็ตามค่าความเชื่อมั่นที่ลดลงไม่มีผลต่อความถูกต้องของแบบจำลองแต่อย่างใด เมื่อพิจารณาค่าอัตราความผิดพลาดจากชุดข้อมูล UCI 20 ชุด พบว่าขั้นตอนวิธี MRMCAR มีอัตราความผิดพลาดต่ำที่สุด 8 ชุดข้อมูล เมื่อเลือกการเรียงกฎจากค่าสนับสนุนสูง จำนวนกฎน้อย และค่าสนับสนุนสูง โดยค่าเฉลี่ยอัตราความผิดพลาดต่ำกว่า C5, J48, RIPPER และ CBA 3.66% 2.14% 3.80% และ 2.55% ตามลำดับ

Tayal และคณะ [47] นำเสนอขั้นตอนวิธี PSOCARM (Particle Swarm Optimization Class Association Rules Miner) ซึ่งเป็นการจำแนกข้อมูลเชิงความสัมพันธ์ที่ประยุกต์ใช้การเพิ่มประสิทธิภาพฝูงอนุภาคแบบไบนารี (Binary Particle Swarm Optimization) ในการค้นหากฎรายการโดยไม่ต้องกำหนดค่าสนับสนุนขั้นต่ำเนื่องจากขั้นตอนวิธีใช้หลักการของฝูงอนุภาคเพื่อค้นหากฎรายการที่ดีที่สุด อนุภาคประกอบด้วยความเร็ว (Velocity) และตำแหน่ง (Position) ซึ่งสามารถหาค่าได้จากสมการที่ 2.18 และสมการที่ 2.19 ตามลำดับ

$$V_{i+1} = W \times V_i + C_1 \times r_1 \times (P_i - X_i) + C_2 \times r_2 \times (G - X_i) \quad 2.18$$

$$\begin{aligned} &if\ (rand() < S(V_{i+1})) \\ &then \\ &\quad X_{i+1} = 1 \\ &else \\ &\quad X_{i+1} = 0 \end{aligned} \quad 2.19$$

ค่า X (Position) สำหรับทุกรายการในอนุภาคเพื่อบอกว่ารายการนี้จะปรากฏในกฎรายการหรือไม่ (0 คือ ไม่ปรากฏ 1 ปรากฏ) โดยค่า X ถูกกำหนดจากอัตราการเปลี่ยนแปลงของค่า

ความเร็ว (V) ซึ่งคำนวณจากสูตร โดยค่า V จะถูกนำเข้าฟังก์ชัน Sigmoid ซึ่งทำหน้าที่คล้าย Activate Function เพื่อให้ค่าจำนวนจริงระหว่าง 0 ถึง 1 ค่าจากฟังก์ชันที่ได้ถูกนำไปเปรียบเทียบกับค่าสุ่มจาก 0 และ 1 เพื่อทำให้โอกาสในการปรากฏ (Position) ของรายการ เท่ากับ 50% ค่า W คือ ค่าน้ำหนักความเฉื่อย เพื่อป้องกันการวนซ้ำการทำงานหาค่าความเร็วแบบเชิงเส้นมากเกินไป หาค่า W มีค่าน้อยกว่า 1 อนุภาคจะให้ความสำคัญกับตำแหน่งที่ดีที่สุดในปัจจุบัน หากค่า W มากกว่า 1 อนุภาคจะให้ความสำคัญในการหาค่าตำแหน่งใหม่มากกว่าค่าปัจจุบัน ค่า C1 C2 ค่าน้ำหนักคงที่ซึ่งค่าที่แนะนำ คือ 0.729 และ 1.494 ตามลำดับ ค่าความเร็วและตำแหน่งสำหรับทุกอนุภาคจะถูกปรับปรุงค่าให้มีตำแหน่งที่ดีขึ้นทุกรอบการประมวลผลโดยพิจารณาจากค่าฟิตเนสที่ดีขึ้นทั้งของตัวอนุภาคเองและอนุภาคข้างเคียง ขั้นตอนวิธี PSOCARM แบ่งการทำงานเป็น 3 ขั้นตอน โดยขั้นตอนที่ 1 เริ่มต้นโดยการให้ผู้ใช้กำหนดจำนวนกฎ N กฎ แล้วใช้วิธีการฝูงอนุภาคแบบไบนารี (Binary Particle Swarm Optimization) เพื่อหาความสัมพันธ์ระหว่างเซตรายการและคลาส โดยแปลงข้อมูลทุกกฎภายในแทรนแซกชันให้อยู่ในรูปแบบไบนารี การแทนที่กฎใช้วิธีการมิชิแกน (Michigan Approach) ซึ่งแต่ละอนุภาคแทนกฎเพียงกฎเดียวเท่านั้น ค่าตำแหน่งและความเร็วจะถูกสุ่มค่าในครั้งแรก กระบวนการหาค่าตำแหน่งและความเร็วใหม่ของแต่ละอนุภาคจะถูกทำซ้ำทั้งสิ้น 10 รอบ เพื่อหากฎที่มีตำแหน่งดีที่สุดของฝูงอนุภาค (Global Best Position) จำนวน 10 กฎโดยกฎที่ดีที่สุด คือ กฎที่มีค่า Fitness สูงที่สุด ขั้นตอนนี้จะถูกทำซ้ำ N ครั้งเพื่อให้ได้กฎจำนวน N กฎ ขั้นตอนที่ 2 กฎสากลที่มีตำแหน่งดีที่สุด (Global Best Position) จำนวน N กฎจะถูกเรียงลำดับโดยใช้วิธีเรียงตามค่าความเชื่อมั่นแล้วเรียงตามค่าสนับสนุนและสุดท้าย คือ จำนวนกฎที่สั้นที่สุด ขั้นตอนวิธีจะคัดเลือกเพียง 5 กฎเท่านั้น เพื่อสร้างแบบจำลองทำนายข้อมูล ขั้นตอนที่ 3 ประเมินความถูกต้องของแบบจำลอง การทดลองทำกับชุดข้อมูล E-mail 2,500 ฉบับ จาก Phishing Corpus [48] และ Spam Assassin [49] ซึ่งมีอีเมล์ล่อลวงเหยื่อจำนวน 1,260 ฉบับ ใช้การสกัดคุณลักษณะโดยความถี่ของคุณลักษณะตามวิธีการขุดค้นเหมืองข้อความบนข้อความไม่มีโครงสร้างซึ่งสามารถสกัดได้ 23 คุณลักษณะ หลังจากนั้นจึงแปลงข้อมูลให้อยู่ในรูปไบนารี นอกจากนั้นยังรวมรวมชุดข้อมูลเว็บไซต์ล่อลวงเหยื่อจำนวน 200 เว็บไซต์ซึ่งแบ่งเป็นเว็บไซต์ล่อลวงเหยื่อจำนวน 100 เว็บไซต์ จาก Phis Tank สกัดคุณลักษณะจำนวน 7 คุณลักษณะ ขั้นตอนวิธีนำมาเปรียบเทียบผลการทดลองได้แก่ CBA CMAR PRM CPAR และ FOIL โดยกำหนดค่าสนับสนุนขั้นต่ำ และค่าความเชื่อมั่นขั้นต่ำที่ 20% และ 80% ตามลำดับ สำหรับขั้นตอนวิธี PSOCARM ไม่มีการกำหนดค่าขั้นต่ำใด ๆ ผลการทดลองแสดงว่าขั้นตอนวิธี PSOCARM มีความถูกต้องสูงกว่าทุกขั้นตอนวิธีอย่างน้อย 10% สำหรับชุดข้อมูลอีเมล์ล่อลวงเหยื่อ โดย PSOCARM ใช้จำนวนกฎโดยเฉลี่ยเพียงแค่ 5 กฎเท่านั้น ผลการทดลองกับชุดข้อมูลเว็บไซต์ล่อลวงเหยื่อพบว่าขั้นตอนวิธี PSOCARM มีค่าความถูกต้อง 88% ซึ่งสูงกว่าทุกขั้นตอนวิธีและจำนวนกฎที่สร้างในแต่ละ Fold เท่ากับ 5 กฎ ในขณะที่ CMAR สร้างกฎเฉลี่ย 97 กฎ แสดงให้เห็นความการใช้กฎที่น้อยแต่มีประสิทธิภาพของขั้นตอนวิธี PSOCARM การทดลองกำหนดจำนวนกฎที่ต้องการในแต่ละรอบการทดสอบเท่ากับ 5 กฎ ซึ่งขั้นตอนวิธีจะวนซ้ำให้ได้กฎที่ดีที่สุดเพียง 1 กฎ ซึ่งทำซ้ำทั้งสิ้น 5 รอบ ภายใน 1 รอบการทำงานจะกำหนดจำนวนรอบการประมวลผล BSO เพื่อค้นหาอนุภาคที่ดีทีสุด (Best Global Particle) โดยกำหนดจำนวน 10 รอบ ในแต่ละรอบจะพิจารณาทุกอนุภาคที่มีในชุดข้อมูล โดยทำการปรับตำแหน่งและทิศทางตามสมการ แล้วจึงปรับปรุงค่าอนุภาค

สากลดีที่สุดจากอนุภาคทั่วไปที่ค้นหาได้จากภายในรอบการทำงาน จุดเด่นของขั้นตอนวิธี PSOCARM คือ จำนวนรายการในกฎ (Antecedent) ที่มีหลากหลายมากกว่าขั้นตอนวิธีอื่น ๆ อย่างไรก็ตามผู้ใช้ต้องกำหนดค่า N จำนวนกฎเริ่มต้นจำนวน N ขั้นตอนวิธีคล้ายกับวิธีการ Top-K Rules และท้ายที่สุดขั้นตอนวิธีจะเลือกกฎจำนวน 5 ข้อเท่านั้นในขั้นตอนการตัดกฎ ซึ่งงานวิจัยไม่ได้อธิบายเหตุผลการใช้กฎจำนวน 5 ข้อ และการกำหนดจำนวนคงที่เช่นนี้ จะสามารถทำให้แบบจำลองสามารถทำนายข้อมูลได้ดีมากแค่ไหน

Lakshmi [50] นำเสนอขั้นตอนวิธี PSTMiner และ PSToSWMine เพื่อจัดการข้อมูล Data Stream ด้วยการใช้ Prefix Streaming Tree (PS Tree) [51] โดยอธิบายว่าในงาน Data Stream Data Stream มีลักษณะข้อมูลที่ต่อเนื่องและไม่สามารถคาดเดาลำดับข้อมูลได้ หน่วยความจำถูกใช้งานอย่างจำกัด และต้องการสกัดข้อมูลรวดเร็วที่สุดเท่าที่เป็นไปได้ ทำให้ Data Stream ต้องการการจำแนกที่แตกต่างจากเดิม ขั้นตอนวิธี PSTMiner ใช้สำหรับ Landmark Windows ซึ่งจัดเก็บข้อมูลตั้งแต่เริ่มต้นจนถึงเวลาปัจจุบันไว้ใน PS Tree แล้วหากฎรายการความถี่ซึ่งผ่านค่าสนับสนุนขั้นต่ำและสกัดกฎที่ผ่าน ค่าความเชื่อมั่นขั้นต่ำ กฎที่ซ้ำซ้อนถูกตัดออกด้วยการประเมินด้วย Chi-square กลุ่มของกฎถูกสร้างเป็นตัวจำแนกและเรียงกฎตามค่าความเชื่อมั่นและค่าสนับสนุนโดยเรียงจากค่ามากไปน้อย ขั้นตอนวิธี PSToSWMine จัดการข้อมูล Sliding windows ซึ่งจัดเก็บข้อมูลในช่วงเวลาหนึ่ง จัดเก็บข้อมูลสตรีมใน PS Tree แล้วค้นหารายการความถี่และกฎที่ผ่านค่าความเชื่อมั่นขั้นต่ำ หากกฎยังไม่เพียงพอสำหรับการทำนาย จึงนำข้อมูลชุด Windows ถัดไปเข้ากระบวนการซ้ำ การทดลองใช้ชุดข้อมูลจาก UCI จำนวน 14 ชุด และข้อมูลสังเคราะห์ 4 ชุด โดยผลการทดลองแสดงให้เห็นว่าขั้นตอนวิธี PSTMiner มีค่าความถูกต้องมีค่าเฉลี่ย 91.2 สูงกว่าเมื่อเปรียบเทียบกับ C4.5 CBA CMAR และ L3 ซึ่งมีค่าความถูกต้อง 84.3 84.6 85.7 และ 89.2 ตามลำดับ ขั้นตอนวิธี PSToSWMine เปรียบเทียบกับ STREAMGEN และ DDPMine ให้ผลการทดลองดีกว่าถึง 10% อย่างไรก็ตามงานวิจัยไม่ได้กล่าวถึงจำนวนกฎที่แต่ละขั้นตอนวิธีสร้างได้ รวมถึงไม่แสดงความเร็วในการประมวลผล

แนวทางการเพิ่มประสิทธิภาพสำหรับการจำแนกข้อมูลเชิงความสัมพันธ์ไม่จำเป็นต้องกำหนดค่าสนับสนุนขั้นต่ำ และดำเนินการซ้ำจนกว่าจะสามารถสร้างกฎที่มีประสิทธิภาพสูง ถึงแม้การกำหนดค่าสนับสนุนขั้นต่ำจะไม่มีความจำเป็นแต่ผู้ใช้งานต้องกำหนดจำนวนกฎที่ต้องการให้ขั้นตอนวิธีซึ่งหากกำหนดจำนวนกฎน้อยเกินไปอาจส่งผลให้ตัวจำแนกมีประสิทธิภาพไม่ดี นอกจากนั้นกระบวนการทำซ้ำเพื่อค้นหากฎส่งผลให้เกิดอุปสรรคในด้านเวลาการประมวลผล

2.7.5 งานวิจัยที่เน้นการขุดค้นเซตรายการความถี่

Djenouri และคณะ [52] ให้ความเห็นว่าวิธีการหากฎความสัมพันธ์ด้วยเทคนิค Apriori สแกนฐานข้อมูลหลายครั้ง นอกจากนั้นการตั้งค่าสนับสนุนขั้นต่ำเป็นประเด็นที่ส่งผลต่อจำนวนของกฎคู่แข่ง จึงนำเสนอขั้นตอนวิธี SS-FIM (Single Scan for Frequent Itemsets Mining) อ่านฐานข้อมูลเพียงครั้งเดียวและเก็บกฎรายการคู่แข่งและค่าสนับสนุนไว้ใน Hash Table โดยนำชุดข้อมูลจาก Bilkent University Function Approximation Repository จำนวน 10 ชุด แบ่งเป็น

ชุดข้อมูลขนาดเล็ก 5 ชุด ขนาดกลาง 4 ชุด และขนาดใหญ่ 1 ชุด ขั้นตอนวิธี SS-FIM สร้างกฎคู่แข่งที่เป็นไปได้ทั้งหมดในแต่ละแทรนแซกชันของชุดข้อมูล หลังจากนั้นใช้ Hashing algorithm ค้นหากฎคู่แข่งใน Hash Table หากไม่พบเพิ่มกฎคู่แข่งลงไปใน Hash Table และตั้งค่าสนับสนุนเท่ากับ 1 หากค้นหากฎพบ ขั้นตอนวิธีจะบวกค่าสนับสนุนเพิ่ม ขั้นตอนวิธีมีประสิทธิภาพ $O(m2^p)$ จากผลการทดลองแสดงให้เห็นว่า สำหรับชุดข้อมูลขนาดเล็กวิธีการ Apriori แบบดั้งเดิมใช้ต้นทุน (Cost) ซึ่งหมายถึง จำนวนขั้นตอนการทำงานของ CPU ได้ดีกว่า SS-FIM แต่สำหรับชุดข้อมูลขนาดกลางและใหญ่ SS-FIM ใช้ Cost น้อยกว่า เช่นเดียวกับในการใช้เวลาประมวลผลซึ่ง SS-FIM ใช้เวลาน้อยกว่า Apriori สำหรับชุดข้อมูลขนาดกลางและขนาดใหญ่ แสดงให้เห็นถึงความสามารถในการรองรับฐานข้อมูลขนาดใหญ่และมีความหนาแน่นของข้อมูลน้อย การทดลองเปลี่ยนค่าสนับสนุนขั้นต่ำไปหลากหลายค่า พบตั้งแต่ 10-100% พบว่า SS-FIM ใช้เวลาประมวลผลในแต่ละค่าสนับสนุนแทบไม่ต่างกัน แสดงให้เห็นว่าการใช้ Hash Table ทำให้ข้อจำกัดของการตั้งค่าสนับสนุนขั้นต่ำหมดไป อย่างไรข้อตามงานวิจัยไม่ได้ระบุจำนวนแทรนแซกชันที่ Hash Table สามารถรองรับได้ โดยในการทดลองชุดข้อมูล BM-POS มีจำนวนข้อมูล 515,597 แทรนแซกชัน จำนวนรายการ 1,657 ชุด เฉลี่ย 2.5 รายการต่อแทรนแซกชัน แต่ไม่ได้กล่าวถึงจำนวนกฎที่สร้างได้ หากจำนวนกฎที่สร้างได้มากกว่านี้ Hash Table จะสามารถรองรับได้หรือไม่ อาทิชุดข้อมูล Lymph ที่มีกฎรายการทั้งสิ้น 4 ล้านข้อเมื่อตั้งค่าสนับสนุนขั้นต่ำ 1% [12]

Linh และคณะ [53] มีความเห็นว่าการให้ผู้ใช้กำหนดค่าสนับสนุนขั้นต่ำเพื่อสร้างกฎความสัมพันธ์นั้นไม่เป็นธรรมชาติสำหรับมนุษย์จึงเสนอขั้นตอนวิธี ETARM โดยปรับปรุงเพิ่มเติมจากขั้นตอนวิธี TopKRules ซึ่งเป็นแนวทางการขุดค้นเซตรายการที่มีความถี่สูงสุด K ลำดับแรก (Mining Top-K Association Rules) โดย K คือ จำนวนกฎที่ผู้ใช้งานต้องการวิเคราะห์ซึ่ง โดยกฎจะถูกเรียงลำดับตามค่าสนับสนุนจากมากไปน้อย ETARM หา 1-itemset ที่มีความถี่สูงสุด K ลำดับแล้วจึงขยายไปสู่ 2-itemset โดยการขยายสามารถทำได้ทั้งสองข้างของกฎ ก่อนการขยายจะพิจาณาลำดับตามตัวอักษร (Lexicon) หากเซตรายการใหม่มีลำดับตัวอักษรที่ต่ำกว่าเซตรายการที่มีอยู่แล้วในกฎจะไม่สามารถขยายลงไปในกฎเพื่อป้องกันการซ้ำซ้อนของกฎ โดยแนวทางนี้ให้ผลการทดลองที่ดีกว่า TopKRules ในด้านของเวลาประมวลผลและการใช้หน่วยความจำที่น้อยลง แต่อย่างไรก็ตามแนวทางที่กล่าวมานั้นใช้สำหรับการหากฎความสัมพันธ์เท่านั้น การประยุกต์ใช้สำหรับการจำแนกข้อมูลเชิงความสัมพันธ์จะทำให้จุดเด่นของ ETARM ในการขยายข้อมูลไปขวาของกฎความสัมพันธ์ไม่สามารถใช้งานได้

Deng [54] นำเสนอ ขั้นตอนวิธี Fin (Fast Mining Frequent Itemsets using Nodesets) คิดค้นวิธีการเข้ารหัสโหนดในโครงสร้างต้นไม้ เพื่อลดขนาดของโครงสร้างข้อมูลขนาดใหญ่เกินกว่าการจัดเก็บในหน่วยความจำซึ่งเกิดปัญหาใน FP-Tree ด้วยการนำเสนอโครงสร้าง Nodelist ซึ่งจัดเก็บกฎรายการ พร้อมกับความถี่และลำดับ Pre-order ไว้ที่โหนดในโครงสร้างต้นไม้ POC หลังจากนั้นจึงท่องไปในโครงสร้างข้อมูลแบบวิธี Pre-order เพื่อสร้าง Nodesets หลังจากนั้นจัดทำโครงสร้างต้นไม้ Set-enumeration สำหรับค้นหาเซตรายการความถี่ด้วยยุทธ์ศาสตร์ผสมผสาน (Hybrid Strategy) เพื่อลดการสร้างรายการคู่แข่งและลดพื้นที่การค้นหา เพื่อค้นหาเซต

รายการความถี่ การทดลองใช้ชุดข้อมูล Mushroom Connect13 จาก UCI และ T25I10D100K จาก IBM พบว่าเวลาในการสร้างเซตรายการความถี่เร็วกว่าขั้นตอนวิธี Prepost และ FP-Growth* ตัวอย่างเวลาการประมวลผลในชุดข้อมูล Mushroom เมื่อตั้งค่าสนับสนุนขั้นต่ำ 5% ขั้นตอนวิธี Fin FP-growth* PrePost ใช้เวลา 0.14 0.2 และ 0.38 ตามลำดับ เมื่อเปรียบเทียบประสิทธิภาพในการใช้หน่วยความจำกับ PrePost ด้วยชุดข้อมูล Connect ค่าสนับสนุนขั้นต่ำ 60% พบว่า Fin มีปริมาณการใช้งานหน่วยความจำน้อยกว่า PrePost อย่างไรก็ตาม Fin ให้ความสำคัญกับความเร็วและปริมาณการใช้หน่วยความจำที่ลดลงในการสร้างกลุ่มรายการปรากฏบ่อยเท่านั้น

Deng [34] ได้นำเสนอขั้นตอนวิธี dFin (Different Fast Mining Frequent Itemsets using Nodesets) ซึ่งประยุกต์จาก Fin โดยใช้เซตผลต่าง (Different Set) ร่วมกับ Nodeset เพื่อเพิ่มความเร็วการผลิตเซตรายการความถี่ (Frequent Itemset) โดยทดลองกับกลุ่มข้อมูล 4 ชุด จาก UCI และ IBM เทียบการทดลองกับวิธีการ FIN PrePost FP-Growth และ Eclat_g โดยเริ่มจากหารายการความถี่จากชุดข้อมูลแล้วสร้างโครงสร้างต้นไม้ PPC จากชุดรายการความถี่ซึ่งเก็บลำดับ Pre-Order Post-Order และความถี่ เพื่อนำไปสู่การสร้าง Nodesets ขั้นตอนต่อไป คือ การคำนวณค่าสนับสนุนจากเซตผลต่างแล้วสร้างโครงสร้างต้นไม้ Set-enumeration สำหรับค้นหาเซตรายการความถี่ด้วยยุทธ์ศาสตร์ผสมผสาน (Hybrid Strategy) เพื่อลดการสร้างรายการคู่แข่งและลดพื้นที่การค้นหา ผลการทดลองพบว่า dFin ใช้เวลาในการประมวลผลได้เร็วกว่าทุกขั้นตอนวิธีที่นำมาเปรียบเทียบ ได้แก่ Fin PrePost+[55] FP_Growth* และ Eclat_g ไม่ว่าค่าสนับสนุนขั้นต่ำจะเปลี่ยนแปลงไปอย่างไร แต่ถึงกระนั้น dFin แสดงผลเปรียบเทียบเพียงในด้านความเร็วและการใช้หน่วยความจำที่ลดลงในการสร้างกลุ่มรายการปรากฏบ่อยเท่านั้นและยังจำเป็นต้องใช้การสร้างเซตรายการทั้งหมดขึ้นมา

งานวิจัยที่เกี่ยวข้องกับการขุดค้นเซตรายการความถี่นำเสนอแนวทางพัฒนาค้นหาเซตรายการซึ่งมีค่าสนับสนุนมากกว่าหรือเท่ากับค่าสนับสนุนขั้นต่ำ ด้วยวิธีการหลากหลายและโครงสร้างข้อมูลที่น่าสนใจ อย่างไรก็ตามเนื่องจากการขุดค้นเซตรายการความถี่ไม่ได้ให้ความสำคัญกับการกำจัดกฎที่ซ้ำซ้อนและไม่มีประโยชน์ ทำให้หากต้องประยุกต์ใช้กับการจำแนกข้อมูลเชิงความสัมพันธ์จำเป็นต้องมีการปรับปรุงวิธีการเพื่อให้ลดการสร้างกฎคู่แข่งที่ไม่จำเป็น

2.7.6 งานวิจัยด้านการขุดค้นกฎรายการระบุคลาสและการค้นหาด้วยเงื่อนไข

Nguyen และคณะ [11] นำเสนอขั้นตอนวิธี CAR-Miner เนื่องจากเห็นว่าการสร้าง CARs เป็นขั้นตอนที่ใช้เวลานานจึงโครงสร้างต้นไม้เพื่อเก็บกฎรายการความถี่โดยอ่านฐานข้อมูลเพียงครั้งเดียวชื่อว่าขั้นตอนวิธี MERC-Tree (Modification Equivalence Class Rule-Tree) โดยปรับปรุงเพิ่มเติมจาก ECR-Tree โครงสร้าง MERC-Tree บรรจุโหนดของ Obidset ซึ่งประกอบด้วยเซตรายการ รหัส แทรนแซกชันที่มีเซตรายการดังกล่าว จำนวนแทรนแซกชันในแต่ละคลาสที่เซตรายการระบุ และคลาสที่มีจำนวนแทรนแซกชันมากที่สุด วิธีการเริ่มต้นด้วยการอ่านฐานข้อมูลเพื่อสร้างโหนดสำหรับ 1-itemset Obidset ทั้งหมด ท่องไปในโครงสร้างเพื่อรวมโหนดปัจจุบันกับโหนดถัดไปซึ่งมีแอททริบิวต์ที่แตกต่างกัน โดยประมวลผลแทรนแซกชันที่มีรหัสเหมือนกัน (Intersect) ระหว่าง 2

โหนดเพื่อสร้างโหนดใหม่ที่ผ่านค่าสนับสนุนขั้นต่ำ วิธีการนี้จะดำเนินซ้ำจนไม่มีกฎใดที่ผ่านเกณฑ์ค่าสนับสนุนขั้นต่ำอีกต่อไป ค่าความเชื่อมั่นคำนวณจากจำนวนแทรนแซกชันในคลาสที่มีจำนวนสูงสุดใน Obidset นั้นหารด้วยจำนวนสมาชิกของแทรนแซกชันที่บรรจุ Obidset ผลการทดลองกับชุดข้อมูลจาก UCI 4 ชุด โดยเฉพาะ Breast ที่มีกฎถึง 4 แสนข้อเมื่อค่าสนับสนุนเป็น 1% ใช้เวลาสร้างกฎ 1.517 วินาที ในขณะที่ ECR-CARM [56] ใช้เวลา 17.136 วินาที อย่างไรก็ตามผลการทดลองไม่ได้แสดงปริมาณการใช้หน่วยความจำและขั้นตอนวิธีนี้ยังสร้างกฎคู่แข่งจำนวนมาก

Nguyen และ Nguyen [20] ให้ความเห็นว่าขั้นตอนวิธี CAR-Miner [11] ใช้หน่วยความจำมากจากการเก็บข้อมูล Obidset และต้องใช้เวลามากในการประมวลผลจุดตัดระหว่าง 2 Obidset เพื่อหากฎในลำดับถัดไป จึงนำเสนอขั้นตอนวิธี CAR-Miner-Diff ใช้แทรนแซกชันที่แตกต่างระหว่าง 2 Obidset (d2O) เพื่อคำนวณค่าสนับสนุนและค่าความเชื่อมั่น ส่งผลให้ประหยัดการใช้หน่วยความจำและเวลาประมวลผล นอกจากนี้เพื่อให้ประหยัดเวลามากขึ้นผู้วิจัยจึงเพิ่มขั้นตอนวิธี CAR-Miner-Diff-Sort เพื่อเรียง Obidset ตามจำนวนของเซตที่แตกต่างจากน้อยไปมาก ผลการทดลองด้วยชุดข้อมูลจาก UCI 11 ชุด แสดงให้เห็นว่าเวลาประมวลผลลดลงอย่างมาก ยกตัวอย่างชุดข้อมูล Connect ที่กำหนดค่าสนับสนุน 88% ขั้นตอนวิธี MAC CAR-Miner CAR-Miner-Diff และ CAR-Miner-Diff-Sort ใช้เวลา 1,123.5498 วินาที 427.84 วินาที 8.944 วินาที และ 3.188 วินาทีตามลำดับ เมื่อพิจารณาการใช้หน่วยความจำ ขั้นตอนวิธี CAR-Miner CAR-Miner-Diff และ CAR-Miner-Diff-Sort ใช้หน่วยความจำ 6,980.772 เมกะไบต์ 103.6018 เมกะไบต์ และ12.4101 เมกะไบต์ตามลำดับ นอกจากนี้เมื่อใช้ชุดข้อมูล Lymph เมื่อกำหนดค่าสนับสนุน 10% จะมีจำนวนกฎ 4 ล้านข้อ ขั้นตอนวิธี MAC CAR-Miner-Sort และ CAR-Miner-Diff-Sort ใช้เวลาประมวลผล 459.132 วินาที 37.609 วินาที และ 29 วินาที ตามลำดับ เห็นได้ว่า CAR-Miner-Diff และ CAR-Miner-Diff-Sort เหมาะกับฐานข้อมูลขนาดใหญ่ อย่างไรก็ตามงานวิจัยนี้มุ่งเน้นการสร้างกฎอย่างรวดเร็ว ไม่ได้ให้ความสำคัญกับความซ้ำซ้อนของกฎ และผลการทดลองไม่ได้กล่าวถึงการวัดผลด้วยค่าความถูกต้องหรือค่าความผิดพลาด

Nguyen และคณะ [14] นำเสนอขั้นตอนวิธี CCAR (Class Association Rules) โดยเห็นว่าผู้ใช้ไม่ต้องการทราบกฎทั้งหมดแต่สนใจเฉพาะกฎที่เกี่ยวข้องกับบางแอททริบิวต์และบางค่าเท่านั้น ผู้วิจัยจึงใช้โครงสร้างต้นไม้ CCR (CCR-Tree) ซึ่งแต่ละโหนดเก็บโครงสร้างข้อมูล Obidset ซึ่งประกอบด้วยแอททริบิวต์ ค่าที่จัดเก็บ รหัสแทรนแซกชันที่บรรจุแอททริบิวต์และค่าที่จัดเก็บในแต่ละคลาส จำนวนแทรนแซกชันในแต่ละคลาส ซึ่งนอกจากจะจัดเก็บข้อมูล 1-itemsets ในระดับที่ 1 แล้วยังจัดเก็บเงื่อนไขที่ผู้ใช้กำหนดลงไปด้วย หลังจากนั้นจึงใช้ข้อมูลที่จัดเก็บสร้างโหนดใหม่ซึ่งเกิดจากการรวมโหนดในระดับที่ 1 โดยใช้โหนดซึ่งเก็บค่าเงื่อนไขเป็นหลัก และกฎใหม่ที่เกิดขึ้นต้องผ่านค่าสนับสนุนขั้นต่ำ วิธีการจะถูกทำซ้ำจนกว่าไม่สามารถรวมโหนดได้อีก ผลลัพธ์จากกระบวนการจะสร้างกฎทั้งหมดไว้ที่โหนดใบ ผลการทดลองพบว่าการใช้หน่วยความจำปรากฏของ CCAR ลดลงถึง 50% และ 89% เมื่อเทียบกับ Pre-CAR-Miner และ CAR-Miner-Post [57] ตามลำดับ โดยใช้ชุดข้อมูล Lymph เมื่อพิจารณาด้านความเร็วในการประมวลผลพบว่า CCAR เร็วกว่า CAR-Miner-Post 12

เท่า และเร็วกว่า Pre-CAR-Miner 3 เท่า อย่างไรก็ตามขั้นตอนวิธีนี้ต้องการสร้างกฎทั้งหมดขึ้นมาก่อนแล้วคัดกรองในภายหลัง

Nguyen และคณะ [15] พบว่าผู้ใช้ตัวจำแนกสนใจเพียงรายการที่เฉพาะเจาะจงเท่านั้น จึงคิดค้นขั้นตอนวิธี LD-CARM-IC โดยใช้โครงสร้างตาข่าย (Lattice Structure) ร่วมกับ Obidset ซึ่งใช้เทคนิคเซตผลต่าง (Different Set) เพื่อสร้างเซตรายการความถี่ตามเงื่อนไขที่ผู้กำหนดขึ้นมาได้ โดยมีขั้นตอน คือ 1) เก็บรายการ คลาส และหมายเลขแทรนแซกชันที่บรรจุรายการเหล่านั้นให้อยู่ในรูปแบบ Obidset โดยจัดเก็บไว้ที่โครงสร้างตาข่ายระดับที่หนึ่ง 2) ท่องไปในโครงสร้างและผสานโหนดกับโหนดถัดไปด้วยเทคนิค d2O เช่นเดียวกับขั้นตอนวิธี CAR-Miner-Diff ค้นหากฎที่ผ่านเกณฑ์ค่าสนับสนุนขั้นต่ำเพื่อสร้างโหนดใหม่ 3) ค้นหากฎโดยท่องไปยังโหนดในโครงสร้างทีละระดับเพื่อเปรียบเทียบกับเงื่อนไข หากโหนดใดเก็บรายการที่ตรงกับเงื่อนไขจึงทำการสร้างกฎจากโหนดนั้น พร้อมทั้งทำเครื่องหมายแสดงเพื่อให้ทราบว่าโหนดนั้นได้สร้างกฎแล้ว ผลการทดลองกับ 14 กลุ่มข้อมูลจาก UCI ระบุว่าวิธีการ LD-CARM-IC มีประสิทธิภาพดีกว่าวิธีการ MAC CMAR CAR-Miner และ CCAR ในด้านความเร็วและการประหยัดหน่วยความจำ เมื่อพิจารณาในฐานข้อมูลที่มีความหนาแน่นสูงอาทิ Chess ด้วยค่าสนับสนุน 40% และค่าความเชื่อมั่น 50% LD-CARM-IC ใช้เวลาประมวลผล 6.782 วินาที ในขณะที่ขั้นตอนวิธี Pre-CAR-Miner CCAR ใช้เวลา 133.678 วินาที และ 60.809 วินาทีตามลำดับ แสดงให้เห็นถึงความสามารถในการสร้างกฎในฐานข้อมูลขนาดใหญ่จากประโยชน์ของเทคนิคเซตผลต่างอย่างไรก็ตามข้อจำกัดของวิธีนี้ คือ ผู้ใช้ต้องกำหนดเงื่อนไขให้อยู่ในรูปแบบแอททริบิวต์และค่าที่ต้องการ ซึ่งผู้ใช้งานต้องระบุค่าภายในแอททริบิวต์ที่เฉพาะเจาะจงเท่านั้น

Nguyen และคณะ [58] ให้ความเห็นว่าขั้นตอนวิธี LD-CARM-IC ทำงาน 2 ขั้นตอนเริ่มจากค้นหากฎที่เซตรายการตรงกับเงื่อนไข หลังจากนั้นจึงหากฎที่คลาสตรงกับเงื่อนไข กระบวนการนี้อัลกอริทึมทำงานสองรอบ ซึ่งหากสามารถลดจำนวนรอบการทำงานได้จะเพิ่มประสิทธิภาพให้สูงขึ้น จึงได้นำเสนอขั้นตอนวิธี GCSC (Generate CARs with Synthetic Constraints) ซึ่งสามารถค้นหากฎเซตรายการตรงกับเงื่อนไขที่ผู้ใช้กำหนดด้วยการทำงานขั้นตอนวิธีเพียงรอบเดียว โดยใช้ชุดข้อมูลจาก UCI 4 ชุดข้อมูล กำหนดจำนวนเงื่อนไขแบบซุ่มตามอัตราส่วนร้อยละขึ้นกับชุดข้อมูลนั้น ๆ ในบางชุดข้อมูลที่มีคลาสจำนวนมากกว่า 2 คลาส การทดสอบจะเลือกคลาสสองลำดับแรกที่ค้นพบ ขั้นตอนวิธี GCSC เริ่มจากสร้างโครงสร้างตาข่ายประกอบด้วย Obidset จากชุดข้อมูลฝึกสอนเช่นเดียวกับขั้นตอนวิธี LD-CARM-IC แล้วค้นหากฎที่เซตรายการ กฎรายการที่ตรงกับเงื่อนไขจะจัดเก็บไว้ในตัวแปร temp ขั้นตอนนี้เพิ่มความเร็วในการเปลี่ยนเงื่อนไขการค้นหาซึ่งแตกต่างจาก LD-CARM-IC ที่ใช้วิธีการทำเครื่องหมายภายในโหนดและหากต้องการค้นหาเงื่อนไขใหม่ต้องท่องไปในโครงสร้างตั้งแต่โหนดรากเพื่อลบเครื่องหมาย กฎที่เซตรายการตรงกับเงื่อนไขจะนำเข้าสู่ฟังก์ชัน TRAVERSE-LATTICE เพื่อค้นหาคลาสที่ตรงกับเงื่อนไขทันที เพื่อป้องกันการสร้างโหนดซ้ำซ้อนดังเช่นในขั้นตอนวิธี CCAR กฎจะถูกตรวจสอบค่าสนับสนุนกับเกณฑ์ค่าสนับสนุนขั้นต่ำหากไม่ผ่าน ขั้นตอนวิธีจะไม่สำรวจไปยังโหนดลูกของกฎนั้น แต่หากกฎนั้นผ่านเกณฑ์ จึงตรวจสอบว่ากฎนั้นมีคลาสตรงกับเงื่อนไขหรือไม่ กฎที่คลาสตรงกับเงื่อนไขจะถูกสร้างเป็น CARs แล้วจึงสำรวจไปยังโหนดลูกของกฎนั้นต่อไป เมื่อค้นหากฎที่ตรงกับเงื่อนไขทั้งหมดแล้วขั้นตอนวิธีจะลบค่าใน temp เพื่อรอรับเงื่อนไข

ใหม่ จากผลการทดลองแสดงว่า GCSC ใช้เวลาค้นหากฎที่ตรงกับเงื่อนไขเร็วกว่า LD-CARM-IC ตัวอย่างในชุดข้อมูล German พบว่า LD-CARM-IC ใช้เวลาประมวลผล 10.836 วินาที แต่ GCSC ใช้เวลา 2.852 วินาที

Nguyen [16] นำเสนอขั้นตอนวิธี Top-k CARs เพื่อค้นหา CARs ลำดับสูงสุด k ลำดับแรกด้วยวิธีการเรียงข้อมูลแบบเร็ว (Quick Sort) โดยมีแนวคิดจาก TopKRules [17] ซึ่งเห็นว่าค่าสนับสนุนขั้นต่ำยากต่อความเข้าใจของผู้ใช้ จึงสร้างตัวแปร k เพื่อให้ผู้ใช้กำหนดจำนวนกฎที่ต้องการให้เกิดขึ้นในแทนการค่าสนับสนุนขั้นต่ำ โดยใช้วิธีการสุ่มเลือกกฎเพื่อเป็นจุดหมุน (Pivot) สำหรับแบ่งกฎเป็นสองกลุ่มโดยกลุ่มแรก s1 เป็นกลุ่มของกฎที่มีค่าสนับสนุนสูงกว่าหรือเท่ากับจุดหมุน กลุ่มที่สอง s2 เป็นกลุ่มของกฎที่มีค่าสนับสนุนน้อยกว่าจุดหมุน หากจำนวนกฎใน s1 น้อยกว่าค่า k จึงทำ Quicksort เพื่อหากฎส่วนที่เหลือจาก s2 จำนวน k-|s1| ในทางตรงกันข้ามหากกฎใน s1 มากกว่าค่า k ให้ค้นหากฎ |s1| - k ด้วยวิธีการ QuickSort อีกครั้ง จากผลการทดลองพบว่าวิธีการนี้มีความเร็วมากกว่าวิธี InsertionSort-Based Classification [59] ตัวอย่างการทดลองด้วยชุดข้อมูล German แล้วพบว่า InsertionSort ใช้เวลา 10.03 วินาที ในขณะที่ QuickSort ใช้เวลา 0.03 วินาที อุปสรรคของวิธีการนี้ คือ จำนวนกฎที่ผู้ใช้เลือก (ค่า k) อาจส่งผลต่อความถูกต้องให้การทำนายข้อมูลลดลง และผู้ใช้จำเป็นต้องใช้วิธีลองผิดลองถูกเพื่อกำหนดค่า k ที่สามารถสร้างตัวจำแนกที่มีประสิทธิภาพสูง

กลุ่มงานวิจัยที่เกี่ยวข้องกับการขุดค้นกฎรายการระบุคลาสและการค้นหาด้วยเงื่อนไขมุ่งเน้นการสร้างกฎรายการระบุคลาสซึ่งเป็นกฎรายการที่มีค่าสนับสนุนมากกว่าหรือเท่ากับค่าสนับสนุนขั้นต่ำให้เร็วที่สุดและครอบคลุมกฎที่เป็นไปได้มากที่สุด นอกจากนั้นยังให้ความสำคัญกับเงื่อนไขที่ผู้ใช้ต้องการโดยสามารถสร้างกฎรายการตามเงื่อนไขเซตรายการที่ผู้ใช้ให้ความสนใจ อย่างไรก็ตามงานวิจัยกลุ่มนี้ไม่ให้ความสนใจกับความซ้ำซ้อนของกฎ ส่งผลให้อาจมีกฎรายการจำนวนมากถูกสร้างขึ้นเมื่อมีการกำหนดค่าสนับสนุนขั้นต่ำให้มีค่าน้อยเกินไป นอกจากนั้นงานวิจัยไม่ได้ให้ความสนใจกับค่าความถูกต้องสำหรับตัวจำแนกโดยเห็นได้จากการไม่ปรากฏการแสดงผลการทดลองที่เกี่ยวกับค่าความถูกต้องและตัวประเมินประสิทธิภาพด้านอื่น

2.7.7 งานวิจัยที่เน้นการลดการสร้างกฎคู่แข่ง

การเหนี่ยวนำกฎ (Rule Induction) เป็นการจำแนกประเภทหนึ่งที่สามารถใช้กับการค้นหา CARs ได้อย่างมีประสิทธิภาพ โดย Thabtah และคณะ [20] นำเสนอขั้นตอนวิธี eDRI (Enhance Dynamic Rule Induction) ซึ่งจัดเป็นการเหนี่ยวนำกฎที่มีลักษณะคล้ายการจำแนกข้อมูลเชิงความสัมพันธ์โดยใช้ค่าความถี่ (Freq) และความแข็งแรงของกฎ (Rule_strength) ซึ่งทำหน้าที่เหมือนค่าสนับสนุนและค่าความเชื่อมั่นในการจำแนกข้อมูลเชิงความสัมพันธ์เพื่อการคัดเลือกกฎที่มีคุณภาพ ขั้นตอนวิธีเริ่มจากค้นหาเซตรายการที่มีค่าความแข็งแรง 100% เพื่อสร้างเป็นกฎรายการแล้วลบแทรนแซกชันที่มีความเกี่ยวข้องกับกฎนั้นพร้อมปรับปรุงค่าความถี่และความแข็งแรงของเซตรายการทุกครั้งแล้วค้นหาเซตรายการที่มีความแข็งแรงสูงสุดต่อไป หากเซตรายการที่มีความแข็งแรงของกฎไม่ถึง 100% สามารถนำไปสร้างกฎโดยถูกพิจารณาในภายหลังได้ โดยการนำเซต

รายการที่มีความแข็งแรงลำดับถัดไปเข้าร่วมด้วยในกฎด้วย ซึ่งผลการทดลองกับชุดข้อมูลเว็บไซต์ล่อลวง (Phishing Websites) และชุดข้อมูลมาตรฐาน พบว่าจำนวนกฎที่สร้างได้น้อยกว่าขั้นตอนวิธี PRISM และให้ความแม่นยำสูงกว่า แต่ในขณะเดียวกันวิธีการปรับปรุงความถี่ของรายการที่เกี่ยวข้องกับกฎที่ถูกพิจารณาแล้ว ทำให้กฎสำคัญอาจไม่ได้ถูกนำมาใช้งาน

Thabtah และคณะ [60] ให้ความเห็นว่าตัวจำแนกประเภท Rule-based เป็นวิธีที่เหมาะสมกับการตรวจจับเว็บไซต์ล่อลวงเหยื่อ เนื่องจากเนื้อหาของแบบจำลองที่ได้คือความรู้ของมนุษย์ที่ตรงไปตรงมาซึ่งผู้ใช้มือใหม่สามารถเข้าใจได้ง่ายและปรับแต่งเมื่อจำเป็นนอกจากนั้นรูปแบบของกฎ “If-then” ง่ายต่อการควบคุมโดยผู้ใช้ นอกจากนั้นตัวจำแนกที่มีพื้นฐานโดยกฎได้รับการพิสูจน์ในหลายงานวิจัยถึงความถูกต้องในการทำนายข้อมูล แต่โดยส่วนมากเครื่องมือตรวจจับเว็บไซต์ล่อลวงเหยื่อมักใช้วิธีการทั่วสำหรับงานเรียนรู้ของเครื่องจักร (Machine Learning) ในทางกลับกันตัวจำแนกประเภท Rule-based ถูกประยุกต์ใช้กับงานประเภทนี้น้อยมาก งานวิจัยนี้จึงนำเสนอการเปรียบเทียบขั้นตอนวิธีประเภท Rule-based ซึ่งใช้วิธีการพื้นฐานที่แตกต่างกันได้แก่ วิธีการละโมบ (ต้นไม้ตัดสินใจ) วิธีการอนุมานกฎ การจำแนกด้วยกฎความสัมพันธ์ เพื่อการตรวจจับเว็บไซต์ล่อลวงเพื่อหาประสิทธิภาพที่แท้จริง โดยใช้ข้อมูลจริงของเว็บไซต์ล่อล่วงเหยื่อมากกว่า 11,000 เว็บไซต์ซึ่งสะสมจากเครื่องมือออนไลน์ ขั้นตอนวิธีที่นำมาเปรียบเทียบได้แก่ eDRI RIPPER C4.5 และ RIDOR เปรียบเทียบประสิทธิภาพโดยอัตราความผิดพลาด เวลาที่ใช้ในการสร้างแบบจำลองและจำนวนกฎภายในแบบจำลอง โดยแบ่งการทดลอง 2 แบบ คือ ทดลองโดยคัดเลือกคุณสมบัติโดยใช้ Correlation Feature Set (CFS) และทดลองโดยไม่คัดเลือกคุณสมบัติ ผลการทดลองแสดงให้เห็นหน้าที่ของการคัดเลือกคุณสมบัติซึ่งมีผลต่ออัตราการตรวจจับเว็บไซต์ล่อลวง ผลการทดลองแสดงว่า C4.5 โดดเด่นกว่าตัวจำแนกอื่น โดยมีค่าความถูกต้องสูงกว่า RIPPER RIDOR และ eDRI เป็นอัตราส่วน 0.86% 3.03% และ 3.33% ตามลำดับ แต่แบบจำลองของ C4.5 มีจำนวนกฎมากที่สุด โดยมีจำนวนกฎมากกว่า 140 130 144 ข้อ เมื่อเปรียบเทียบกับ RIPPER RIDOR และ eDRI จำนวนกฎที่มากขึ้นสร้างความลำบากในการใช้งานกับผู้ใช้ ขั้นตอนวิธีที่สร้างแบบจำลองได้เร็วที่สุด คือ eDRI ซึ่งใช้วิธีการลบข้อมูลชุดสอนที่เกี่ยวข้องทันทีเมื่อกฎที่ถูกสร้าง การปรับลำดับกฎด้วยความถี่ทันทีเมื่อกฎถูกสร้างอนุญาตให้มีการแบ่งแยกกฎที่อ่อนแอก่อนที่ขั้นตอนการประเมินกฎจะเริ่มต้นขึ้นส่งผลให้ประหยัดเวลาและทรัพยากรในการประมวลผล การสร้างแบบจำลองได้อย่างรวดเร็วทำให้ eDRI เหมาะกับการเป็นวิธีการตรวจจับเว็บไซต์ล่อหลวงเหยื่อนอกจากนั้นยังสร้างกฎจำนวนน้อยแลกเปลี่ยนกับค่าความถูกต้องที่น้อยกว่า C4.5 อัตราข้อผิดพลาดระหว่างตัวจำแนกประเภทพื้นฐานจากกฎดีกว่า Bayes Net และ Simple Logistics คิดเป็น 1.08% และ 0.47% ตามลำดับ โดยเฉพาะ C4.5 และ RIPPER ให้อัตราข้อผิดพลาดต่ำที่สุด เมื่อใช้วิธีการ CFS คัดเลือกแอททริบิวต์สามารถคัดเลือก 9 แอททริบิวต์ที่ส่งผลต่อแบบจำลอง ผลการทดลองแสดงให้เห็นว่าค่าประสิทธิภาพของแบบจำลองลดลงเพียง 1% ในบางขั้นตอนวิธี นอกจากนั้นตัวจำแนกพื้นฐานกฎมีประสิทธิภาพการทำนายที่ดีกว่าตัวจำแนกดั้งเดิม ในขณะที่ RIPPER ให้ความสำคัญกับคลาส Phishing มากกว่าหากกฎใดไม่สามารถเป็น Phishing ได้จะจัดเป็น Legitimate ซึ่งตรงข้ามกับขั้นตอนวิธี RIDOR ที่ให้ความสำคัญกับคลาส Legitimate มากกว่า ทั้งสองขั้นตอนวิธีแสดงให้เห็นความไม่สมดุลในการทำนายดังนั้นกฎที่ได้มาอาจไม่เพียงพอต่องานด้านตรวจจับเว็บไซต์ล่อลวงเหยื่อ ในทางตรงกันข้าม

ขั้นตอนวิธี eDRI สร้างแบบจำลองซึ่งมีกฎเพื่อตรวจจับคลาส Phishing จำนวน 9 ข้อและ 16 กฎ สำหรับตรวจจับข้อมูลคลาส legitimate แสดงให้เห็นว่า eDRI มีจำนวนกฎที่แน่นอนในทั้งสองคลาส นอกจากนั้นจำนวนกฎที่ได้ยังน้อยกว่า 2 ขั้นตอนวิธีข้างต้น

ในงานวิจัยที่ผ่านมาเซตรายการแบบปิด (Closed) และเซตรายการความยาวสูงสุด (Maximal) สามารถลดจำนวนของกฎที่ซ้ำซ้อนจากการหาเซตรายการความถี่ (Frequent Itemsets) ลงได้มาก แต่ยังไม่มีงานวิจัยที่แสดงการทดลองเปรียบเทียบในทางสถิติอย่างชัดเจน Antonie และ คณะ [61] จึงนำเสนอการทดลองเพื่อแสดงการลดลงของกฎอย่างเป็นรูปธรรมและผลกระทบจากการ ลดลงของกฎต่อความถูกต้องของตัวจำแนกเพื่อค้นหาว่าเซตรายการแบบใดเหมาะสมกับการจำแนก ข้อมูลเชิงความสัมพันธ์มากที่สุด กรอบการทำงานจึงถูกสร้างขึ้นโดยสร้างตัวจำแนกที่ประกอบด้วยเซต ของกฎรายการซึ่งสกัดจากวิธีการที่แตกต่างกันได้แก่ เซตรายการความถี่ เซตรายการแบบปิด และ เซตรายการความยาวสูงสุด เพื่อเปรียบเทียบจำนวนกฎรายการที่ลดลงระหว่างทั้ง 3 วิธีการ และ ระดับประสิทธิภาพของตัวจำแนกที่เปลี่ยนไป กระบวนการสร้างตัวจำแนกเริ่มโดยสกัดกฎจากทั้ง 3 วิธีการ คัดเลือกกฎที่ผ่านเกณฑ์ค่าสนับสนุนขั้นต่ำและค่าความเชื่อมั่นขั้นต่ำ แล้วเรียงกฎตามลักษณะ CSC (Confidence Support และ Cardinality) ซึ่งจะสร้างตัวจำแนกทั้งหมด 3 รูปแบบ หลังจาก นั้นวัดประสิทธิภาพการจำแนกข้อมูลด้วยวิธีการที่แตกต่างกัน 3 วิธี ได้แก่ วิธีที่ 1 จำแนกคลาสของ ตัวอย่างข้อมูลด้วยกฎแรกที่พบ (First Rule) หรือ FR ซึ่งเป็นลักษณะการจำแนกของขั้นตอนวิธีการ จำแนกข้อมูลเชิงความสัมพันธ์ส่วนใหญ่ ตัวอย่างเช่นขั้นตอนวิธี CBA วิธีที่ 2 จำแนกจากค่าความ เชื่อมั่นโดยเฉลี่ย (AvR) โดยแบ่งกฎทั้งหมดที่สามารถทำนายตัวอย่างข้อมูลที่ได้ตามคลาส แล้วหา คำนวณค่าเฉลี่ยของค่าความเชื่อมั่นในแต่ละคลาส คลาสใดมีค่าเฉลี่ยสูงสุดจะเป็นคำตอบของตัวอย่าง ข้อมูล วิธีที่ 3 ทำนายโดยใช้วิธีการจำแนก 2 ระดับ (2SARC) [62] ซึ่งใช้คุณลักษณะทั้งในรูปแบบ คุณลักษณะของคลาสและคุณลักษณะของกฎ โดยใช้ขั้นตอนวิธี KNN การทดลองใช้ข้อมูลจากชุด ข้อมูลจาก UCI จำนวน 20 ชุดข้อมูล โดยใช้ขั้นตอนวิธี Apriori-like เพื่อสร้างกฎรายการความถี่ กำหนดค่าสนับสนุนขั้นต่ำ 1% 5% และ 10% ค่าความเชื่อมั่นขั้นต่ำ 50% วัดประสิทธิภาพแบบ 10-Fold Cross Validation ด้วยค่าความถูกต้องและใช้ Cost curve [18] ซึ่งเป็นทางเลือก นอกเหนือจากกราฟแบบ ROC การแปลงข้อมูลต่อเนื่องให้เป็นข้อมูลไม่ต่อเนื่องใช้วิธีการจาก [63] ผล การทดลองในด้านการลดจำนวนกฎพบการลดลงของจำนวนกฎโดยเฉลี่ยในทุกชุดข้อมูลและค่า สนับสนุนขั้นต่ำเปรียบเทียบวิธีการเซตรายการความถี่ เซตรายการแบบปิด และเซตรายการความยาว สูงสุด คือ 18% และ 34.36 ถึง 46.32% ตามลำดับ ผลการทดลองในด้านประสิทธิภาพการจำแนก พบว่า เมื่อทดสอบด้วยวิธี FR เซตรายการแบบปิด และเซตรายการความยาวสูงสุด เพิ่มประสิทธิภาพ ของตัวจำแนกเล็กน้อยโดยตัวเลขอยู่ระหว่าง -0.27% ถึง 0.93% และ -5.36% ถึง 5.32% ตามลำดับ สำหรับวิธี AvR เซตรายการแบบปิดเพิ่มประสิทธิภาพการจำแนกเล็กน้อยระหว่าง -2.29% ถึง 0.66% ในขณะที่เซตรายการความยาวสูงสุด ลดประสิทธิภาพการจำแนกเล็กน้อยระหว่าง -5.33% ถึง 1.11% วิธี 2ARC-CF ประสิทธิภาพการจำแนกเพิ่มขึ้นเล็กน้อยเมื่อใช้เซตรายการแบบปิด ในขณะที่ เซตรายการความยาวสูงสุด ทำให้ประสิทธิภาพลดลงเล็กน้อย สำหรับวิธี 2ARC-RF เซตรายการแบบ ปิด และเซตรายการความยาวสูงสุด ลดประสิทธิภาพของตัวจำแนกอย่างไม่มีนัยยะเนื่องจากอัตราการ เพิ่ม/ลดของความถูกต้องค่อนข้างแปรปรวนโดยตัวเลขอยู่ระหว่าง -1.33% ถึง 1.8% และ -3.99%

ถึง 3.67% ตามลำดับ สรุปได้ว่าการลดลงของจำนวนกฎมีนัยยะน้อยมากต่อความถูกต้องของตัวจำแนก การทดลองโดยใช้ชุดข้อมูล Microarray ซึ่งใช้ในการวิเคราะห์การเกิดโรคมะเร็งประกอบด้วยข้อมูลยีนส์ ซึ่งประกอบด้วยคุณลักษณะจำนวนมากในชุดข้อมูล แต่ข้อมูลที่รวบรวมได้มีจำนวนน้อยซึ่งยากต่อการสร้างตัวจำแนกที่ดี ผลการทดลองแสดงให้เห็นว่าอัตราการลดลงของจำนวนเซตรายการมีค่าเท่ากันสำหรับเซตรายการแบบปิด และเซตรายการความยาวสูงสุด คือ 14.28% โดยเฉลี่ย อย่างไรก็ตามเมื่อพิจารณาค่าความถูกต้องปรากฏว่าประสิทธิภาพของตัวจำแนกไม่ได้ลดลงแต่อย่างใดแสดงให้เห็นว่าการใช้เซตรายการแบบปิด และเซตรายการความยาวสูงสุด ลดจำนวนของกฎลงไม่ทำให้ประสิทธิภาพของตัวจำแนกลดลงแต่กฎที่น้อยลงทำให้ผู้ใช้งานวิเคราะห์ได้กฎได้ง่ายขึ้น การวัดประสิทธิภาพด้วยกราฟเส้นโค้งต้นทุน (Cost Curve) [64] ซึ่งเป็นกราฟจำลองแสดงประสิทธิภาพสำหรับการจำแนก การใช้ Cost Curve เพื่อหาเห็นว่าเงื่อนไขใดที่ส่งผลต่อการเลือกใช้เซตรายการความถี่แบบปิด หรือเซตรายการความยาวสูงสุด ผลการทดลองโดยใช้การจำแนกด้วยวิธี FR AvR และ 2SARC แสดงให้เห็นว่าเซตรายการความยาวสูงสุด เหมาะกับการใช้งานเมื่อมีต้นทุนความน่าจะเป็นตั้งแต่ 0.65 ขึ้นไป ในขณะที่เมื่อความน่าจะเป็นต่ำเซตรายการแบบปิด และเซตรายการความถี่ให้ประสิทธิภาพที่ไม่ต่างกันเท่าใดนัก

Kamalov และคณะ [65] ได้พัฒนาวิธีการกรองคุณลักษณะบนพื้นฐานเวคเตอร์ลำดับ (Ranked Vector) โดยให้ความเห็นว่าการจำแนกข้อมูล คือ การนำคุณลักษณะ (Feature) หรือแอททริบิวต์เข้าเรียนรู้เพื่อสร้างแบบจำลองซึ่งสามารถทำนายคลาสของข้อมูลได้โดยหาความสัมพันธ์ของแต่ละคุณลักษณะกับคลาส ดังนั้นการเลือกคุณลักษณะจึงมีความสำคัญในการลดคุณลักษณะที่ซ้ำซ้อนต่อการทำนาย วิธีการที่นิยมใช้โดยทั่วไป อาทิ การวัดค่าโดยค่า Information Gain (IG) หรือ Chi-Square อย่างไรก็ตามจุดด้อยของการใช้วิธีการ IG และ Chi-Squared คือ ในชุดข้อมูลเดียวกันหากเลือกวิธีการกรองคุณลักษณะที่ต่างกันอาจได้คุณลักษณะที่ต่างกันด้วยเนื่องจากทั้งค่า IG และ Chi-squared ให้ความหมายทางสถิติที่แตกต่างกัน ซึ่งสามารถแก้ปัญหาโดยการนำค่าทั้งสองมารวมกันเพื่อให้ได้ค่าเวคเตอร์คะแนน (Vector Score) หรือ V-Score แต่อย่างไรก็ตามจุดด้อยของ V-Score คือ การไม่พิจารณาความเกี่ยวข้องกันระหว่างคุณลักษณะ ซึ่งบางครั้งกันนำคุณลักษณะที่มีค่า V-Score ต่ำ มารวมกันอาจได้กลุ่มของคุณลักษณะที่ช่วยให้แบบจำลองสามารถทำนายข้อมูลได้ถูกต้องมากขึ้น ตรงกันข้ามกับวิธีการ Correlation Feature Selection (CFS) ซึ่งลดบทบาทในการพิจารณาความสำคัญของคุณลักษณะเพียงคุณลักษณะเดียว อย่างไรก็ตามจุดด้อยของ CFS คือ คุณลักษณะที่ได้มักมีค่า IG และ V-Score ที่ต่ำ เพื่อแก้ปัญหาข้างต้นวิธีการกรองคุณลักษณะบนพื้นฐานเวคเตอร์ลำดับ (Ranked Vector) ได้ถูกพัฒนาขึ้นโดยใช้การรวมค่าที่ประเมินได้จากวิธีการ IG และ Chi -Squared เพื่อสร้างความเชื่อมั่นต่อคุณลักษณะที่ถูกเลือกให้สูงขึ้น โดยไม่กระทบต่อความแม่นยำของแบบจำลอง นอกจากนี้วิธีการ CFS ถูกนำมาใช้เพื่อคัดเลือกกลุ่มของคุณลักษณะที่เหมาะสม โดยประเมินความสามารถในการทำนายข้อมูลของแต่ละคุณลักษณะพร้อมกับระดับของความซ้ำซ้อนระหว่างกลุ่มคุณลักษณะเหล่านั้น วิธีการกรองคุณลักษณะบนพื้นฐานเวคเตอร์ลำดับ มี 3 ขั้นตอน ขั้นตอนที่ 1 คำนวณค่า V-Score สำหรับแต่ละคุณลักษณะ การหา V-Score คำนวณจากการรวมค่า IG และ CHI-Squared เข้าด้วยกันแต่เนื่องจากค่าทั้งสองมีความแตกต่างกันมากจึงต้องทำ

การปรับบรรทัดฐาน (Normalize) โดยค่าบรรทัดฐานของ IG สำหรับคุณลักษณะ a คำนวณจากสมการที่ 2.20

$$\frac{IG_a}{IG_{\max}} \quad 2.20$$

โดย $IG_{\max}$ คือ ค่า IG ที่สูงที่สุดภายในกลุ่มคุณลักษณะ ค่าบรรทัดฐานของ CHI สำหรับคุณลักษณะ a คำนวณดังสมการที่ 2.21

$$\frac{CHI_a}{CHI_{\max}} \quad 2.21$$

โดย $CHI_{\max}$ คือ ค่า Chi-Squared ที่สูงที่สุดภายในกลุ่มคุณลักษณะ หลังจากนั้นค่า V-Score คำนวณดังสมการที่ 2.22

$$|v_a| = \sqrt{(IG_a)^2 + (CHI_a)^2} \quad 2.22$$

ขั้นตอนที่ 2 เพื่อลดความซ้ำซ้อนในการเลือกคุณลักษณะที่อาจมีค่าV-score เท่ากัน วิธีการ Correlation Feature Selection (CFS) ถูกนำมาใช้เพื่อคัดเลือกกลุ่มของคุณลักษณะที่เหมาะสม โดยประเมินความในการทำนายข้อมูลของแต่ละคุณลักษณะพร้อมกับระดับของความซ้ำซ้อนระหว่างกลุ่มคุณลักษณะเหล่านั้น ขั้นตอนที่ 3 เพื่อกำจัดคุณลักษณะที่มีค่า V-Score ต่ำจึงมีการกำหนดเกณฑ์ตัด V-Score (Cut Off V-Score) โดยกำหนดค่าที่ 50% ของค่า V-Score ที่สูงที่สุด การทดลองใช้ชุดข้อมูลจาก UCI จำนวน 15 ชุด กำหนดค่าเกณฑ์ขั้นต่ำของ IG และ Chi-Squared ที่ 0.1 และ 10.83 ตามลำดับ วิธีการกรองคุณลักษณะที่ถูกนำมาเปรียบเทียบได้แก่ IG Chi-Squared CFS ขั้นตอนวิธีที่ถูกใช้ประเมินประสิทธิภาพของวิธีการกรองคุณลักษณะได้แก่ eDRI C4.5 PART เนื่องจากทั้ง 3 ขั้นตอนวิธีมีวิธีเรียนรู้กฎและสร้างแบบจำลองที่ต่างกัน ผลการทดลองแสดงว่าการวิธีการกรองข้อมูลแบบ Ranked Vector คัดเลือกคุณลักษณะได้น้อยกว่าทุกวิธีการในทุกชุดข้อมูล โดยคิดอัตราส่วนการลดลงของคุณลักษณะได้เป็น 44.81 50.76 8.66 เปรียบเทียบกับวิธีการ Chi IG CFS เมื่อพิจารณาค่าอัตราความผิดพลาดในการทำนาย (Error Rate) โดยทำการทดสอบร่วมกับขั้นตอนวิธี eDRI และ C4.5 พบว่าการกรองคุณลักษณะแบบ Ranked Vector ดีกว่า (ค่าความผิดพลาดต่ำกว่า) IG และ CHI-Squared และได้ค่าเท่ากับ CFS เมื่อทดสอบร่วมกับขั้นตอนวิธี PART พบว่าการกรองคุณลักษณะแบบ Ranked Vector ดีกว่าการกรองคุณลักษณะทุกวิธี

Alwidian และคณะ [66] ให้ความเห็นว่าถึงแม้การจำแนกข้อมูลเชิงความสัมพันธ์จะให้ค่าความถูกต้องสูงก็ตาม แต่ข้อด้อยสำหรับเทคนิคนี้ คือ การสร้างกฎจำนวนมาก ทำให้การประมวลผลช้าและสิ้นเปลืองหน่วยความจำ ในทางกลับกันขั้นตอนวิธีที่สามารถสร้างกฎได้อย่างรวดเร็วแต่ความแม่นยำกลับไม่สูงมากนัก นอกจากนั้นขั้นตอนวิธีต่างมีประสิทธิภาพการทำนายไม่คงที่เมื่อชุดข้อมูลชุดใหม่เกิดขึ้นตลอดเวลา อาทิ ข้อมูลธุรกรรมออนไลน์ ข้อมูลการซื้อขายหุ้น การซื้อสินค้าออนไลน์ ด้วยเหตุนี้ผู้วิจัยจึงนำเสนอขั้นตอนวิธี FCBA (Fast Classification Based on Association Rules) ซึ่งปรับปรุงประสิทธิภาพขั้นตอนวิธี CBA ด้วยวิธีการตัดกฎใหม่เรียกว่า การตัดกฎภายใน (Internal

Pruning) ซึ่งทำให้ความเร็วในการสร้างตัวจำแนกด้วยเทคนิค Apriori เร็วขึ้นและยังมีความถูกต้องสูง โดยทดสอบกับชุดข้อมูลมาตรฐาน 11 ชุดจาก UCI วิธีการตัดกฎภายในจะเกิดขึ้นพร้อมกับการสร้างกฎ ขั้นตอนวิธีจะคัดเลือกกฎที่มีค่าสนับสนุนมากกว่าหรือเท่ากับค่าสนับสนุนขั้นต่ำ กฎใดที่ค่าความเชื่อมั่นมากกว่าค่าความเชื่อมั่นขั้นต่ำจะถูกจัดเป็น CARs เพื่อสร้างตัวจำแนก ในทางตรงกันข้ามกฎที่ไม่ผ่านค่าความเชื่อมั่นขั้นต่ำจะถูกจัดกลุ่มเป็นกฎคู่แข่ง จากนั้นผสานกฎคู่แข่งเข้าด้วยกันเพื่อตรวจสอบค่าสนับสนุนและค่าความเชื่อมั่น กฎที่ได้ทั้งหมดจะถูกเรียงลำดับจากค่าสนับสนุนที่มากที่สุดก่อนหากค่าสนับสนุนเท่ากันจะเลือกเรียงกฎจากลำดับการสร้าง หากผ่านเกณฑ์จะถูกบรรจุเข้าเป็นตัวจำแนกต่อไป เมื่อเรียงกฎการตัดกฎภายนอก (External Pruning) จะตัดกฎที่มีสำคัญน้อยอีกครั้งด้วยเทคนิค M1 เช่นเดียวกับ CBA เห็นได้ว่า FCBA มุ่งเน้นเพียงการสร้างกฎที่มีเพียง 1 เซตรายการเท่านั้น ด้วยวิธีการดังกล่าวทำให้กฎที่มีค่าสนับสนุนผ่านเกณฑ์แต่ค่าความเชื่อมั่นต่ำ ไม่ถูกนำมาสร้างกฎโดยไม่จำเป็น FCBA จึงมีใช้เวลาในการสร้างกฎที่ต่ำมาก จากผลการทดลอง โดยใช้ชุดข้อมูลมาตรฐาน 11 ชุดจาก UCI เปรียบเทียบกับขั้นตอนวิธี CBA MCAR CMAR และ FACA ตั้งค่าสนับสนุนขั้นต่ำ 5% และค่าความเชื่อมั่นขั้นต่ำ 50% เมื่อวัดค่าความถูกต้อง FCBA ได้อันดับดีที่สุดใน 7 ชุดข้อมูลจาก 11 ชุดข้อมูล โดยมีค่าเฉลี่ยความถูกต้องอยู่ที่ 79.59 % การวัดผลในเชิงเวลาการประมวลผลใช้เวลาน้อยกว่าทุกขั้นตอนวิธีที่นำมาเปรียบเทียบนอกจากนั้นยังสามารถสร้างกฎจำนวนน้อยกว่าด้วย

Hadi และคณะ [8] นำเสนอ ACPRISM ซึ่งถือเป็นการประยุกต์ใช้วิธีการเหนี่ยวนำกฎร่วมกับการจำแนกข้อมูลเชิงความสัมพันธ์ซึ่งให้ความเห็นว่าการจำแนกข้อมูลเชิงความสัมพันธ์มีการสร้างกฎจำนวนมากจึงใช้วิธีการ 1) หาแอททริบิวต์ที่มีค่า IG ที่ดีที่สุด เพื่อตัดกฎที่ซ้ำซ้อนและกฎที่ไม่ปกติออกไป 2) ใช้หลักการแบ่งแยกและเอาชนะ แบ่งกลุ่มตามข้อมูลที่บรรจุภายในแอททริบิวต์จากข้อที่ 1 แล้วพิจารณาเป็นกลุ่ม 3) สร้างกฎโดยเลือกกฎที่มีค่าความเชื่อมั่นเท่ากับ 100% หากไม่มีกฎที่ความเชื่อมั่น 100% สามารถนำเซตรายการอื่นเข้ามาร่วมด้วย 4) สร้างแบบจำลองเพื่อทำนาย ด้วยขั้นตอนเหล่านี้ทำให้ ACPRISM สามารถสร้างกฎได้อย่างรวดเร็ว นอกจากนั้นยังให้ค่าความแม่นยำที่ค่อนข้างสูงกว่าเมื่อเปรียบเทียบกับ CBA PRISM FACA หรือ RIPPER เมื่อใช้ชุดข้อมูลจาก UCI 16 ชุดในการทดลอง ผลการทดลองระบุว่า ACPRISM ให้ค่าความแม่นยำเฉลี่ยถึง 82.50% และถึงแม้จะยังน้อยกว่า FACA เพียง 0.54% แต่เมื่อนับตามวิธีการจัด Ranking พบว่า ACPRISM มีอันดับเฉลี่ย 2.81 ซึ่งถือว่าดีที่สุด นอกจากนั้นผู้วิจัยยังได้ทดลองกับชุดข้อมูลแหล่งน้ำใต้ดินพบว่ามีความแม่นยำสูงถึง 86.11% และใช้เวลาน้อยกว่าขั้นตอนวิธีอื่น แต่จำนวนกฎที่สร้างได้ยังเป็นรองเพียงแค่ RIPPER เท่านั้น อย่างไรก็ตามหากชุดข้อมูลมีจำนวนแอททริบิวต์มากจะทำให้เกิดการประมวลผลนานมากขึ้นเนื่องจากพื้นที่สำหรับค้นหากฎที่เพิ่มขึ้น

Schmid และคณะ [67] นำเสนอขั้นตอนวิธี CMARAA (Classification Based On Multiple Class-Association Rules for Authorship Attribution) เพื่อใช้จำแนกหาแหล่งที่มาของผู้เขียนอีเมล์ที่มีแนวโน้นในการก่ออาชญากรรมบนโลกออนไลน์ การจำแนกแหล่งที่มาของผู้เขียนจัดเป็นปัญหาการแบ่งประเภทข้อความโดยเน้นการค้นหาลักษณะการเขียนเฉพาะ (Write Print) ที่เป็นเอกลักษณ์ของแต่ละบุคคลโดยเรียนรู้จากการเขียนในอดีตของบุคคล โดยปกติอีเมล์มีรูปแบบการ

เขียนที่ไม่เป็นทางการ จึงปรากฏคำศัพท์ที่ผิดและการเขียนที่ผิดหลักไวยากรณ์จำนวนมาก ซึ่งผู้อ่านมองข้ามความผิดพลาดไป แต่ทำให้การใช้วิธีการตรวจจับคำและจำแนกคุณลักษณะของอีเมล์มีความลำบากอย่างยิ่ง หลักการจำแนกด้วยคุณลักษณะแบบ Lexical และ Syntactic ที่ผ่านมาไม่สามารถสร้างการจำแนกอีเมล์ที่แม่นยำสูงได้ ดังนั้นการตรวจจับลักษณะการเขียนอีเมล์เป็นทางเลือกที่มีประสิทธิภาพ ลักษณะการเขียนพิจารณาโดยผสมหลายองค์ประกอบ ได้แก่ คำศัพท์ (Lexical) การสร้างประโยค (Syntactical) โครงสร้าง (Structural) ความหมาย (Semantic) และเนื้อหาเฉพาะเจาะจง (Content-specific) ซึ่งจากผลการทดลองโดย de Vel และคณะ [68, 69] แสดงให้เห็นผลลัพธ์ที่ดีกว่าการแยกพิจารณาแต่ละองค์ประกอบ ขั้นตอนวิธี CMARAA ซึ่งเป็นขั้นตอนวิธีที่ให้ความสนใจกับคุณสมบัติ (Features) วิธีการเขียนเฉพาะตัวบุคคล วัดความสัมพันธ์ของคุณสมบัติเหล่านี้แล้วสร้างตัวจำแนกที่ง่ายต่อการใช้งาน การทดลองดำเนินการโดยรวบรวมอีเมล์จำนวน 100 ฉบับ ที่ถูกเขียนโดยบุคคลต่าง ๆ จำนวน 14 คน เพื่อทดสอบว่าขั้นตอนวิธีจะสามารถแยกแยะอีเมล์ของแต่ละบุคคลได้หรือไม่ การสกัดคุณลักษณะในรูปแบบ ARFF (Weka Input Format) อีเมล์ทุกฉบับถูกแทนที่ในรูปแบบ ARFF 1 บรรทัด ทำการปรับบรรทัดฐาน (Normalize) คุณลักษณะตามเทคนิค Equal-Frequency Discretization และสกัดรูปแบบความถี่ที่ผ่านเกณฑ์ค่าสนับสนุนขั้นต่ำที่กำหนด ในการตัดกฎขั้นตอนที่ 1 การเรียงกฎทำงานตรงข้ามกับ CPAR ซึ่งจะเรียงกฎทั่วไปก่อนกฎที่เฉพาะ แต่สำหรับ CMARAA นั้นกฎทั่วไปแสดงถึงความไม่ชัดเจนในรูปแบบการเขียนเอกสารซึ่งส่งผลให้ความถูกต้องลดลง ดังนั้น CMARAA จึงเลือกเรียงกฎที่เฉพาะก่อนกฎทั่วไปเพื่อแสดงถึงรูปแบบการเขียนที่ปรากฏบ่อยของผู้เขียน การตัดกฎขั้นตอนที่ 2 ทำกับกฎที่มีค่าความสัมพันธ์เชิงบวกที่วัดผลโดยค่า Chi-Square ซึ่งทำในขณะที่กฎถูกเพิ่มลงใน CR-Tree การตัดกฎขั้นที่ 3 ใช้รูปแบบการครอบคลุมฐานข้อมูลแบบมีเกณฑ์ซึ่งแตกต่างจากขั้นตอนวิธีอื่น คือเมื่อกฎสามารถจับคู่กับข้อมูลตัวอย่างใดได้ กฎจะถูกเลือกและข้อมูลตัวอย่างนั้นจะไม่ถูกลบทันที แต่จะนับค่าสะสมจนกระทั่งมีกฎที่สามารถจับคู่กับข้อมูลตัวอย่างนั้นครบ 3 กฎจึงจะสามารถลบข้อมูลตัวอย่างนั้นได้ การทำเช่นนี้ทำให้ขั้นตอนวิธี CMARAA มีจำนวนกฎที่สามารถพิจารณาได้มากขึ้นส่งผลให้ชุดข้อมูลตัวอย่างสามารถถูกทำนายได้มากขึ้น ขั้นตอนการทำนายในกรณีที่ปรากฏว่าแบบจำลองสามารถทำนายคลาสข้อมูลตัวอย่างได้มากกว่า 1 คลาส ขั้นตอนวิธีจะแบ่งกฎออกเป็นกลุ่มตามคลาส แล้วเลือกกลุ่มกฎที่มีความเข้มแข็งของกฎสูงสุด (Strongest rules) วิธีการหาความเข้มแข็งของคำนวณจากค่า Weighted Chi-Squared สำหรับแต่ละกลุ่มของกฎเช่นเดียวกับขั้นตอนวิธี CMAR โดย กำหนด sup(P) คือ จำนวนรายการ P ทั้งหมดที่เกิดปรากฏในฐานข้อมูล sup(c) คือ จำนวนแทรนแซกชันที่เกี่ยวข้องกับคลาส c และ |T| คือ จำนวนแทรนแซกชันทั้งหมดในฐานข้อมูล ค่า Max Chi-Squared คำนวณจากสมการที่ 2.23 เมื่อ e คำนวณจากสมการที่ 2.24

$$max\chi^2 = \left(min\left(sup(P), sup(c)\right) - \frac{sup(P)sup(c)}{T} \right)^2 |T|e \qquad 2.23$$

$$e = \frac{1}{sup(P)sup(c)} + \frac{1}{sup(P)(|T| - sup(c))} + \frac{1}{|T| - (sup(P)sup(c))} + \frac{1}{(|T| - (sup(P))(|T| - sup(c))} \quad 2.24$$

$$\sum \frac{(\chi^2)}{max\chi^2} \quad 2.25$$

ค่า Weighted ของแต่ละกลุ่มคำนวณจากสมการที่ 2.25 การหาค่า WCS เป็นการลดอคติที่เกิดขึ้นกับกลุ่มกฎที่มีจำนวนกฎน้อยกว่า เปรียบเทียบขั้นตอนวิธีกับ CBA AM J4.8 BayesNet และ END โดยใช้รวบรวมข้อมูลอีเมล์จากชุดข้อมูล Enron e-mail แบ่งข้อมูลเป็นชุดสอน 90% และชุดทดสอบ 10% โดยกระจายอีเมล์ของผู้แต่งแต่ละคนออกไปตามสัดส่วน กำหนดค่าสนับสนุนขั้นต่ำสำหรับขั้นตอนวิธี CBA CMAR และ CMARAAที่ 10% และค่าความเชื่อมั่นขั้นต่ำ 0% เพื่อจำลองประสิทธิภาพของการตัดกฎในช่วงที่ 3 โดยให้ความเห็นว่า CMARAA ให้ความสำคัญกับการหาสัดส่วน ของความสัมพันธ์เชิงบวกและ WCS มากกว่าจึงกำหนดค่าความเชื่อมั่นไว้ที่ 0% เพื่อความยุติธรรม ผลการทดลองโดยใช้ผู้แต่ง 10 คน ใช้เวลาในการประมวลผลมากกว่าขั้นตอนวิธีอื่น เนื่องจาก CMARAA มุ่งเน้นการตรวจจับอาชญากรรมบนโลกออนไลน์มากกว่าประเด็นทางด้านการใช้หน่วยความจำและเวลาในการประมวลผลจึงเป็นเรื่องที่ให้ความสำคัญรองลงมา

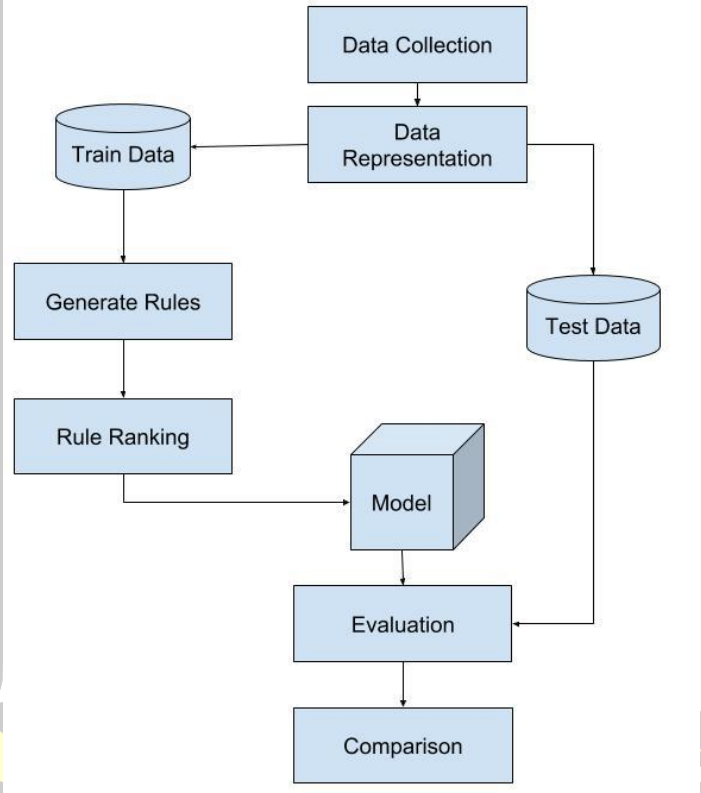
งานวิจัยที่กล่าวมาข้างต้น มีการนำเสนอวิธีการที่มีประสิทธิภาพในด้านต่าง ๆ เช่น ด้านการลดเวลาการประมวลผล สามารถสร้างตัวจำแนกขนาดเล็ก ตลอดจนสามารถจำแนกข้อมูลให้ค่าความถูกต้องที่สูงมาก วิธีการที่น่าสนใจสำหรับการลดการสร้างกฎคู่แข่ง คือ การลดพื้นที่การค้นหากฎลง ซึ่งถูกนำเสนอในขั้นตอนวิธี eDRI และ ACPRISM ผลการทดลองของทั้ง 2 ขั้นตอนวิธีแสดงให้เห็นว่าถึงแม้จะสามารถลดจำนวนกฎได้ลงเป็นอย่างมากเมื่อเทียบกับขั้นตอนวิธีอื่น แต่ค่าความถูกต้องยังสูงมากอีกด้วย

งานวิจัยที่ผ่านมามีจุดเด่นและอุปสรรคตามที่ได้นำเสนอ อย่างไรก็ตามงานวิจัยส่วนใหญ่ ใช้วิธีการสร้างกฎจากจำนวนเซตรายการประกอบในกฎจำนวน 1 รายการ แล้วนำกฎที่สร้างได้มาขยายเพื่อสร้างกฎที่มีเซตรายการมากขึ้น โดยข้อมูลที่เกี่ยวข้องยังจัดเก็บในหน่วยความจำหลัก และจำเป็นต้องอ่านข้อมูลเหล่านั้นหลายครั้งเพื่อสร้างกฎคู่แข่ง กฎคู่แข่งจำนวนมากถูกสร้างขึ้นมาแล้วผ่านการคัดกรองด้วยกระบวนการตัดกฎเพื่อกำจัดกฎที่ซ้ำซ้อนและไม่มีประโยชน์ ซึ่งเป็นขั้นตอนที่ต้องอ่านชุดข้อมูลทั้งหมดอีกครั้ง หากสามารถลดการสร้างกฎคู่แข่งที่ไม่จำเป็นจะสามารถเพิ่มประสิทธิภาพของขั้นตอนวิธีได้ การลดข้อมูลที่ไม่จำเป็นในการสร้างกฎออกไปจัดเป็นการจำกัดกฎที่จะถูกสร้างในอนาคต ตลอดจนลดพื้นที่หน่วยความจำในการประมวลผลข้อมูล นอกจากนั้นการใช้ทฤษฎีเซตเข้าร่วมสามารถเพิ่มประสิทธิภาพในการสร้างกฎและการลดข้อมูลที่ไม่จำเป็น ซึ่งได้ถูกนำเสนอในงานวิจัยนี้

บทที่ 3

วิธีดำเนินการวิจัย

งานวิจัยนี้มีวัตถุประสงค์เพื่อพัฒนาวิธีการขุดค้นกฎที่มีประสิทธิภาพสำหรับการจำแนกเชิงความสัมพันธ์ ผู้วิจัยได้ทำการศึกษาทฤษฎี และงานวิจัยที่เกี่ยวข้องกับการจำแนกข้อมูลเชิงความสัมพันธ์ดังกล่าวในบทที่ 2 และสามารถออกแบบวิธีการดำเนินงานวิจัยเพื่อให้บรรลุตามวัตถุประสงค์ดังแสดงในรูปที่ 3.1 โดยขั้นตอนวิธีการดำเนินงานวิจัยประกอบไปด้วย 6 ขั้นตอน ได้แก่ 1) รวบรวมข้อมูล (Data Collection) 2) เตรียมข้อมูล (Data Preparation) 3) การสร้างกฎ (Rule Generation) 4) การเรียงกฎ (Rule Ranking)) 5) การวัดประสิทธิภาพอัลกอริทึม (Evaluation) 6) เปรียบเทียบประสิทธิภาพกับขั้นตอนวิธีอื่น ๆ (Comparison)



รูปที่ 3.1 วิธีดำเนินการวิจัย

3.1 การรวบรวมข้อมูล (Data Collection)

งานวิจัยนี้ใช้ข้อมูลมาตรฐานสำหรับการจำแนกจาก UCI (University of California, Irvine Machine Learning Repository) [70] จำนวน 14 ชุด ซึ่งชุดข้อมูลเหล่านี้ได้ถูกนำไปใช้ในการทดลองในงานวิจัยจำนวนมาก เช่น ขั้นตอนวิธี CBA [1] ขั้นตอนวิธี eMCAC [3] ขั้นตอนวิธี FACA [5] ขั้นตอนวิธี ACPRISM [8] ขั้นตอนวิธี CMAR [13] ขั้นตอนวิธี eDRI [20] ขั้นตอนวิธี MRMCAR [38] ขั้นตอนวิธี MAC [44] ขั้นตอนวิธี MCAR [45] และขั้นตอนวิธี CBC [41] รายละเอียดของชุดข้อมูลแสดงดังตารางที่ 3.1

ตารางที่ 3.1 รายละเอียดชุดข้อมูล

ลำดับ	ชุดข้อมูล	จำนวนแอททริบิวต์	จำนวนคลาส	จำนวนแถว
1	Anneal	38	6	898
2	Breast	11	2	286
3	Car	6	4	1,728
4	Contact-lenses	4	3	24
5	Diabetes	8	2	768
6	Iris	4	3	150
7	Labor	17	2	57
8	Lymph	18	4	148
9	Mushroom	22	2	8,214
10	Post-operative	9	4	90
11	Tic-tac-toe	9	2	958
12	Vote	16	2	435
13	Wined	13	3	178
14	Zoo	17	7	101

3.2 การเตรียมข้อมูล (Data Preparation)

3.2.1 การแปลงข้อมูล (Data Transformation)

ข้อมูลสำหรับการทดลองอยู่ในรูปแบบตารางข้อมูลเชิงสัมพันธ์ แอททริบิวต์สามารถมีค่าเป็นหมวดหมู่ (Categorical or Discrete) และต่อเนื่อง (Continuous) งานวิจัยนี้แปลงข้อมูลให้อยู่ในรูปแบบมาตรฐานเดียวกันเพื่อประสิทธิภาพในการประมวลผล สำหรับข้อมูลต่อเนื่องจะถูกจัดกลุ่มโดยแต่ละกลุ่มโดยใช้วิธีที่แตกต่างกันตามชุดข้อมูล เช่น วิธีการ Entropy-Based Binding สำหรับชุดข้อมูลจาก UCI ซึ่งวิธีการนี้เป็นการแบ่งช่วงย่อยข้อมูลแบบมีผลเฉลย โดยอ้างอิงช่วงข้อมูลกับคลาสที่เกี่ยวข้องเป็นวิธีการที่เหมาะกับการแบ่งช่วงข้อมูลเพื่อการจำแนก [71] สำหรับชุดข้อมูล Weather ซึ่งในสำหรับอธิบายขั้นตอนวิธีที่นำเสนอ ใช้วิธีการแบ่งข้อมูลแบบ Equal Interval Width และ Equal Frequency Width [72] ตัวอย่างชุดข้อมูลแสดงดังตารางที่ 3.2

ตารางที่ 3.2 ชุดข้อมูล Weather

TID	Outlook	Temperature	Humidity	Windy	Play
1	Sunny	85	85	False	No
2	Sunny	80	90	True	No
3	Overcast	83	86	False	Yes
4	Rainy	70	96	False	Yes

ตารางที่ 3.2 ชุดข้อมูล Weather (ต่อ)

TID	Outlook	Temperature	Humidity	Windy	Play
5	Rainy	68	80	False	Yes
6	Rainy	65	70	True	No
7	Overcast	64	65	True	Yes
8	Sunny	72	95	False	No
9	Sunny	69	70	False	Yes
10	Rainy	75	80	False	Yes
11	Sunny	75	70	True	Yes
12	Overcast	72	90	True	Yes
13	Overcast	81	75	False	Yes
14	Rainy	71	91	True	No

ตัวอย่างการแปลงข้อมูล Humidity โดยใช้วิธีการ Equal Interval Width แสดงดังสมการ 3.1

$$w = (max - min) / N \quad 3.1$$

เมื่อกำหนด *N* = 2 กลุ่ม *Min* คือ ค่าน้อยที่สุดในกลุ่มจากตัวอย่างข้อมูลคือค่า 65 และ *Max* คือ ค่าน้อยที่สุดในกลุ่มจากตัวอย่างข้อมูลคือค่า 96 ค่าความกว้างของแต่ละช่วงเท่ากับ 16 จึงสามารถแบ่งกลุ่มข้อมูลได้รูปที่ 3.2 ข้อมูล Weather เมื่อผ่านการแปลงแล้วแสดงดังตารางที่ 3.3

Bin 1 (65-81) = 65 70 75 และ 80 กำหนดเป็นค่า Normal

Bin 2 (82-96) = 85 86 90 91 95 และ 96 กำหนดเป็นค่า High

รูปที่ 3.2 ข้อมูล Humidity ที่ผ่านการแบ่งกลุ่ม

ตารางที่ 3.3 ข้อมูลที่ผ่านการเตรียมข้อมูล

TID	Outlook	Temperature	Humidity	Windy	Play
1	Sunny	Hot	High	False	No
2	Sunny	Hot	High	True	No
3	Overcast	Hot	High	False	Yes
4	Rainy	Mild	High	False	Yes
5	Rainy	Cool	Normal	False	Yes
6	Rainy	Cool	Normal	True	No
7	Overcast	Cool	Normal	True	Yes

ตารางที่ 3.3 ข้อมูลที่ผ่านการเตรียมข้อมูล (ต่อ)

TID	Outlook	Temperature	Humidity	Windy	Play
8	Sunny	Mild	High	False	No
9	Sunny	Cool	Normal	False	Yes
10	Rainy	Mild	Normal	False	Yes
11	Sunny	Mild	Normal	True	Yes
12	Overcast	Mild	High	True	Yes
13	Overcast	Hot	Normal	False	Yes
14	Rainy	Mild	High	True	No

3.2.2 การแทนค่าข้อมูล (Data Representation)

ข้อมูลที่ใช้ในงานวิจัยถูกแสดงอยู่ในรูปข้อมูลแนวตั้ง (Vertical Data Representation) ซึ่งแต่ละคอลัมน์เก็บเซตหมายเลขแทรนแซกชัน (Transaction ID) ที่มีรายการนั้นปรากฏอยู่ เช่น เซตหมายเลขแทรนแซกชันของ $\langle Outlook, Sunny \rangle$ คือ 1 2 8 9 และ 11 เนื่องจากค่า Sunny ปรากฏอยู่ในแทรนแซกชันดังกล่าว เป็นต้น นอกจากนี้แล้วเซตหมายเลขแทรนแซกชันของแต่คลาส เมื่อปรับรูปแบบการแทนค่าข้อมูลทั้งหมดแล้วจะได้ผลลัพธ์ดังตารางที่ 3.4

การแทนค่าข้อมูลแนวตั้งมีประสิทธิภาพสูงโดยเฉพาะอย่างยิ่งกับการคำนวณค่าสนับสนุนแต่ละรายการ รวมถึงสามารถรวม 2 รายการเข้าด้วยกันได้ง่ายและมีประสิทธิภาพ โดยการจัดข้อมูลแนวตั้งใช้เพียงการดำเนินการอินเทอร์เซกชันระหว่างเซตหมายเลขแทรนแซกชันของ 2 รายการ ตรงข้ามกับการแทนค่าข้อมูลแนวนอนซึ่งใช้วิธีการอ่านข้อมูลซ้ำ ซึ่งเสียเวลาในคำนวณค่าสนับสนุนและการรวมกฎรายการ [39]

ตารางที่ 3.4 ตัวอย่างแสดงข้อมูลแนวตั้ง

Outlook			Temperature			Humidity		Windy		Play	
Sunny	Overcast	Rainy	Hot	Mild	Cool	High	Normal	True	False	Yes	No
1	3	4	1	4	5	1	5	2	1	3	1
2	7	5	2	8	6	2	6	6	3	4	2
8	12	6	3	10	7	3	7	7	4	5	6
9	13	10	13	11	9	4	9	11	5	7	8
11		14		12		8	10	12	8	9	14
				14		12	11	14	9	10	
						14	13		10	11	
									13	12	
										13	

3.3 ขั้นตอนวิธีที่นำเสนอ (ECARG Algorithm)

งานวิจัยชิ้นนี้นำเสนอขั้นตอนวิธี ECARG หรือ Efficient Class Association Rule Generation algorithm ประกอบขั้นตอนการทำงาน 4 ขั้นตอน ได้แก่ 1) การสร้างกฎรายการความยาว 1 2) การลบกฎซ้ำซ้อน 3) การขยายกฎรายการ และ 4) การสร้างคลาสเริ่มต้น โดยมีรหัสเทียม (Pseudo code) แสดงดังรูปที่ 3.3

Algorithm: ECARG
Input: *dataset*, *minsup*
Output: *classifier*
1: *ruleItems* ← 1-ruleitem generation from *dataset*
2: **While** at least one rule's support meet minsup **do**
3: *R* ← maximum confidence rule from *ruleItems*
4: **If** *R*'s confidence < 100 and *R* is not null **then**
5: *R* ← extend *R* with the other *ruleitems*
6: **If** *R* is not null **then**
7: Insert *R* to *classifier*
8: Redundant rule removal
9: Update support and confidence for each *ruleItems*
10: **Else**
11: Exit while loop
12: Finding the default class
13: **Return** *classifier*

รูปที่ 3.3 The ECARG algorithm

3.3.1 การสร้างกฎรายการความยาว 1

การสร้างกฎรายการความยาว 1 ขั้นตอนวิธี ECARG เริ่มจากการสร้างกฎรายการความยาว 1 จากชุดข้อมูล (บรรทัดที่ 1) เพื่อความรวดเร็ว ECARG ใช้ประโยชน์จากการแทนค่าข้อมูลแนวตั้งเพื่อคำนวณค่าสนับสนุนของกฎรายการ ค่าสนับสนุนของกฎรายการสามารถคำนวณได้อย่างง่ายจาก $\left|g(itemset) \cap g(c_k)\right|$ หากกฎใดมีค่าสนับสนุนน้อยกว่าค่าสนับสนุนขั้นต่ำ กฎนั้นจะไม่ถูกนำมาขยายความยาวร่วมกับกฎอื่นในขั้นตอนถัดไป นอกจากนี้ค่าความเชื่อมั่นของกฎรายการสามารถคำนวณได้อย่างง่ายดังสมการที่ 2.7 โดยใช้ประโยชน์จากการแทนค่าข้อมูลแนวตั้ง ถ้าค่าความเชื่อมั่นของกฎรายการเท่ากับ 100% กฎรายการจะถูกเพิ่มเข้าไปในแบบจำลองเพื่อการจำแนกข้อมูล (บรรทัดที่ 7) หากค่าความเชื่อมั่นไม่เท่ากับ 100% กฎรายการดังกล่าวจะถูกนำไปพิจารณาเพื่อการขยายความยาวกฎในขั้นตอนต่อไป กฎรายการที่มีค่าความเชื่อมั่นของกฎมากกว่าหรือเท่ากับค่าความเชื่อมั่นขั้นต่ำจะถูกเรียกว่า กฎความสัมพันธ์ระบุคลาส (CAR)

3.3.2 การลบกฎซ้ำซ้อน

หลังจากค้นพบกฎความสัมพันธ์ระบุคลาสค่าความเชื่อมั่น 100% หมายเลขแทรนเซกชันที่เกี่ยวข้องกับกฎความสัมพันธ์ระบุคลาสจะถูกลบเพื่อลดการสร้างกฎที่ไม่จำเป็นหรือซ้ำซ้อน (บรรทัดที่ 8) การลบหมายเลขแทรนเซกชันที่ไม่จำเป็นสามารถทำได้อย่างง่ายด้วยวิธีการเซตผลต่าง โดยกำหนดให้ r_i เป็นกฎความสัมพันธ์ระบุคลาสค่าความเชื่อมั่น 100% และ T เป็นเซตของกฎรายการที่มีคลาสเช่นเดียวกับ r_i สำหรับทุก $r_j \in T$ หมายเลขแทรนเซกชันใหม่ของ r_j คือ $g(r_j) = g(r_j) - g(r_i)$ หลังจากนั้นหมายเลขแทรนเซกชันใหม่ ค่าสนับสนุน และค่าความเชื่อมั่น สำหรับทุกกฎจะถูกปรับปรุงด้วยวิธีการดังกล่าว (บรรทัดที่ 9)

3.3.3 การขยายกฎ

ทุกรอบการค้นหากฎ หากไม่ปรากฏกฎความสัมพันธ์ระบุคลาสค่าความเชื่อมั่น 100% กฎ r ที่มีค่าความเชื่อมั่นสูงที่สุดในรอบนั้นจะถูกพิจารณาให้ขยายความยาวกฎด้วยวิธีการค้นหาแนวกว้าง (Breadth first search) กฎดังกล่าวจะถูกรวมกับกฎรายการอื่นที่มีคลาสเดียวกันจนกระทั่งกฎความสัมพันธ์ระบุคลาสใหม่ที่ได้รับการขยายความยาวกฎจะมีค่าความเชื่อมั่นเท่ากับ 100% (บรรทัดที่ 5) ถ้า r_i ถูกขยายด้วย r_j เป็น r_{new} และ $g(r_j) \subseteq g(r_i)$ แล้ว $conf(r_{new})$ =100% หลังจากกฎความสัมพันธ์ระบุคลาสซึ่งถูกขยายความยาวถูกเพิ่มลงในแบบจำลองแล้วหมายเลขแทรนเซกชันที่เกี่ยวข้องกับกฎความสัมพันธ์ระบุคลาสจะถูกลบ จนกระทั่งหากไม่มีกฎรายการใดมีค่าสนับสนุนมากกว่าหรือเท่ากับค่าสนับสนุนขั้นต่ำแล้ว การสร้างกฎความสัมพันธ์ระบุคลาสจะหยุดการทำงาน

3.3.4 การสร้างคลาสเริ่มต้น

ขั้นตอนวิธี ECARG จะดำเนินการต่อเพื่อค้นหาคลาสเริ่มต้น (a default class) เพื่อไปยังแบบจำลอง คลาสที่ปรากฏในแทรนเซกชันที่เหลืออยู่มากที่สุดจะถูกกำหนดเป็นคลาสเริ่มต้น (บรรทัดที่ 12) ในกรณีที่ไม่เหลือแทรนเซกชันในข้อมูลชุดสอน ขั้นตอนวิธีจะสร้างคลาสเริ่มต้นโดยนับจำนวนคลาสที่ปรากฏในกฎความสัมพันธ์ระบุคลาสภายในแบบจำลอง คลาสใดมีจำนวนมากกว่าคลาสนั้นจะถูกจัดเป็นคลาสเริ่มต้น

เพื่อแสดงให้เห็นการทำงานของขั้นตอนวิธี ECARG ข้อมูลในตารางที่ 3.3 จะถูกใช้เป็นข้อมูลชุดตัวอย่าง โดยกำหนดค่าสนับสนุนขั้นต่ำและค่าความเชื่อมั่นขั้นต่ำเท่ากับ 3 และ 50% ตามลำดับ ข้อมูลตัวอย่างถูกเปลี่ยนแปลงเป็นข้อมูลแนวตั้งได้ผลลัพธ์ดังตารางที่ 3.4 หลังจากนั้นวิธีการอินเทอร์เซกชันถูกนำมาใช้เพื่อสร้างกฎรายการความยาว 1 ดังตารางที่ 3.5 โดยข้อมูลสองแถวล่างสุด แสดงค่าสนับสนุนและค่าความเชื่อมั่นของกฎรายการตามลำดับ จากตารางที่ 3.5 ค่า $Sunny$ ใน $Outlook$ ปรากฏในแทรนเซกชันหมายเลข 1 2 8 9 และ 11 แสดงเป็น $g(\langle Outlook, Sunny \rangle) = \{1,2,8,9,11\}$ คลาส $Yes\ (Y)$ ปรากฏในแทรนเซกชันหมายเลข 3 4 5 7 9 10 11

12 และ 13 แสดงเป็น $g(Yes)=\{3,4,5,7,9,10,11,12,13\}$ ในขณะที่คลาส $No\ (N)$ ปรากฏในแทรนเซกชันหมายเลข 1 2 6 8 และ 14 แสดงเป็น $g(No)=\{1,2,6,8,14\}$ หมายเลขแทรนเซกชันที่มีกฎรายการ $\langle Outlook, Sunny\rangle \rightarrow Yes$ คือ $g(\langle Outlook, Sunny\rangle) \cap g(Yes)=\{1,2,8,9,11\} \cap\{3,4,5,7,9,10,11,12,13\}=\{9,11\}$ ดังนั้นค่าสนับสนุนของ $\langle Outlook, Sunny\rangle \rightarrow Yes$ เท่ากับ 2 หมายเลขแทรนเซกชันที่มีกฎรายการ $\langle Outlook, Sunny\rangle \rightarrow No$ คือ $g(\langle Outlook, Sunny\rangle) \cap g(No)=\{1,2,8,9,11\} \cap\{1,2,6,8,14\}=\{1,2,8\}$ ดังนั้นค่าสนับสนุนของ $\langle Outlook, Sunny\rangle \rightarrow No$ เท่ากับ 3 จากข้อมูลข้างต้นกฎรายการ $\langle Outlook,\ Sunny\rangle \rightarrow Yes$ จะไม่ถูกนำไปขยายร่วมกับกฎรายการอื่น เนื่องจากค่าสนับสนุนต่ำกว่าค่าสนับสนุนขั้นต่ำ ขั้นตอนวิธีจะตัดกฎที่ค่าสนับสนุนน้อยกว่าค่าสนับสนุนขั้นต่ำถึงแม้กฎจะมีค่าความเชื่อมั่นสูง เพื่อลดการสร้างกฎที่ซ้ำซ้อน ซึ่งในตัวอย่างนี้กำหนดค่าสนับสนุนขั้นต่ำเท่ากับ 3 ผลลัพธ์หลังการตัดกฎที่ไม่ผ่านค่าสนับสนุนขั้นต่ำแสดงดังตารางที่ 3.6 กฎที่ผ่านค่าสนับสนุนแสดงในตารางพื้นที่สีขาว

ตารางที่ 3.5 กฎรายการความยาว 1

Outlook						Temperature						Humidity				Windy			
Sunny		Overcast		Rainy		Hot		Mild		Cool		High		Normal		True		False	
Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N
9 11	1 2 8	3 7 12 13		4 5 10	9	3 13	1 2	4 10 11 12	8 1 4	5 7 9	6	3 4 12	1 2 8 14	5 7 9 10 11 13	6	7 11 12	2 6 14	3 4 5 9 10 13	1 8
2	3	4	0	3	1	2	2	4	2	3	1	3	4	6	1	3	3	6	2
	60	100		60				67		75		43	57	86		50	50	75	

ตารางที่ 3.6 กฎที่ผ่านค่าสนับสนุนขั้นต่ำ

Outlook						Temperature						Humidity				Windy			
Sunny		Overcast		Rainy		Hot		Mild		Cool		High		Normal		True		False	
~~Y~~	N	Y	~~N~~	Y	~~N~~	~~Y~~	~~N~~	Y	~~N~~	Y	~~N~~	Y	N	Y	~~N~~	Y	N	Y	~~N~~
9 11	1 2 8	3 7 12 13		4 5 10	9	3 13	1 2	4 10 11 12	8 1 4	5 7 9	6	3 4 12	1 2 8 14	5 7 9 10 11 13	6	7 11 12	2 6 14	3 4 5 9 10 13	1 8
2	3	4	0	3	1	2	2	4	2	3	1	3	4	6	1	3	3	6	2
	60	100		60				67		75		43	57	86		50	50	75	

ค่าความเชื่อมั่นของกฎรายการสามารถคำนวณได้อย่างง่ายจาก $\left|g\left(itemset \rightarrow c_k\right)\right| \div \left|g\left(itemset\right)\right| \times 100$ ถ้าค่าความเชื่อมั่นของกฎรายการเท่ากับ 100% กฎรายการจะถูกเพิ่มลงในแบบจำลอง ถ้าไม่กฎรายการจะถูกพิจารณาให้ขยายความยาวกฎร่วมกับกฎรายอื่นในขั้นตอนถัดไป ยกตัวอย่างค่าความเชื่อมั่น $\langle Outlook, Sunny\rangle \rightarrow No$ สามารถคำนวณจาก $\left|g\left(1,2,8\right)\right| \div \left|g\left(1,2,8,9,11\right)\right| \times 100 = 3 \div 5 \times 100 = 60\%$ ค่าความเชื่อมั่นของกฎรายการ $\langle Outlook, Sunny\rangle \rightarrow No$ ไม่เท่ากับ 100% ดังนั้นกฎจะถูกขยาย ในขณะที่ค่าความเชื่อมั่นของกฎรายการ $\langle Outlook, Overcast\rangle \rightarrow Yes$ คือ $\left|g\left(3,7,12,13\right)\right| \div \left|g\left(3,7,12,13\right)\right| \times 100 = 4 \div 4 \times 100 \quad = 100\%$ ดังนั้นกฎรายการนี้จะถูกเพิ่มลงในแบบจำลอง

หลังจากค้นพบกฎความสัมพันธ์ระบุคลาสกฎแรก หมายเลขแทรนเซกชันที่เกี่ยวข้องกับกฎความสัมพันธ์ระบุคลาสจะถูกลบ จากตารางที่ 3.6 ถ้า $\langle Outlook, Overcast\rangle$ ถูกพบในแทรนเซกชันใด แทรนเซกชันนั้นจะมีคลาสเป็น Yes เสมอ ดังนั้นกฎรายการ $\langle Outlook, Overcast\rangle \rightarrow Yes$ ไม่มีความจำเป็นต้องขยายและแทรนเซกชันหมายเลข 3 7 12 และ 13 ควรถูกลบออกจากชุดข้อมูล ขั้นตอนวิธี ECARG ประยุกต์วิธีการเซตผลต่างทำให้สามารถลบข้อมูลหมายเลขแทรนเซกชันได้อย่างง่ายดาย

ยกตัวอย่างกฎรายการ $g\left(\langle Outlook, Overcast\rangle\right) \rightarrow Yes = \{3,7,12,13\}$ และ กฎรายการ $g\left(\langle Humidity, High\rangle\right) \rightarrow Yes = \{3,4,12\}$ หมายเลขแทรนเซกชันใหม่ของกฎรายการ $g\left(\langle Humidity, High\rangle\right) \rightarrow Yes = g\left(\langle Humidity, High\rangle \rightarrow Yes\right) - g\left(\langle Outlook, Overcast\rangle \rightarrow Yes\right) = \{3,4,12\} - \{3,7,12,13\} = \{4\}$ หมายเลขแทรนเซกชันใหม่ของกฎรายการทั้งหมดหลังค้นพบกฎความสัมพันธ์ระบุคลาสกฎที่ 1 แสดงดังตารางที่ 3.7

ตารางที่ 3.7 กฎและเซตหมายเลขแทรนแซกชันหลังการสร้างกฎที่ 1

Outlook						Temperature						Humidity				Windy			
Sunny		Overcast		Rainy		Hot		Mild		Cool		High		Normal		True		False	
~~Y~~	N	Y	~~N~~	Y	~~N~~	~~Y~~	~~N~~	Y	~~N~~	Y	~~N~~	Y	N	Y	~~N~~	Y	N	Y	~~N~~
9 11	1 2 8			4 5 10	9		1 2	4 10 11	8 14	5 9	6	4	1 2 8 14	5 9 10 11	6	11	2 6 14	4 5 9 10	1 8
2	3	0	0	3	1	2	2	3	2	2	1	1	4	4	1	1	3	4	2
	60			60				60					80	80			75	67	

จากตารางที่ 3.7 ไม่มีกฎรายการใดที่มีค่าความเชื่อมั่นเท่ากับ 100% ขั้นตอนวิธีเลือกกฎรายการ $g(\langle Humidity, High\rangle) \rightarrow No$ ซึ่งค่าความเชื่อมั่นสูงที่สุด (80%) และ $g(\langle Outlook, Sunny\rangle) \rightarrow No = \{1,2,8\}$ เป็นซับเซตของ $g(\langle Humidity, High\rangle) \rightarrow No = \{1,2,8,14\}$ ดังนั้นกฎรายการใหม่ $g(\langle Outlook, Sunny\rangle, \langle Humidity, High\rangle) \rightarrow No$ ซึ่งมีค่าความเชื่อมั่น 100% ถูกค้นพบ ทำให้กฎรายการ $g(\langle Outlook, Sunny\rangle, \langle Humidity, High\rangle) \rightarrow No$ หยุดการขยายกฎต่อไป สำหรับการขยายกฎที่เริ่มจากกฎรายการ $g(\langle Humidity, High\rangle) \rightarrow No$ มีเพียงกฎเดียวที่มีค่าความเชื่อมั่น 100% และกฎรายการดังกล่าวได้ถูกเพิ่มลงในแบบจำลองซึ่งถูกเรียกว่า กฎความสัมพันธ์ระบุคลาสกฎที่ 2

หลังจากกฎความสัมพันธ์ระบุคลาสกฎที่ 2 ถูกเพิ่มลงในแบบจำลอง หมายเลขแทรนเซกชันที่เกี่ยวข้องกับกฎดังกล่าวจะถูกลบออกจากชุดข้อมูลด้วยวิธีการเช่นเดียวกับที่อธิบายข้างต้น หมายเลขแทรนเซกชันที่เหลืออยู่แสดงดังตารางที่ 3.8

ตารางที่ 3.8 กฎและเซตหมายเลขแทรนแซกชันหลังการสร้างกฎที่ 2

Outlook						Temperature						Humidity				Windy			
Sunny		Overcast		Rainy		Hot		Mild		Cool		High		Normal		True		False	
~~Y~~	N	Y	~~N~~	Y	~~N~~	~~Y~~	~~N~~	Y	~~N~~	Y	~~N~~	Y	N	Y	~~N~~	Y	N	Y	~~N~~
9 11				4 5 10	9			4 10 11	14	5 9	6	4	14	5 9 10 11	6	11	6 14	4 5 9 10	
2	0	0	0	3	1	0	0	3	1	2	1	1	1	4	1	1	2	4	0
				75				75						80				100	

จากตารางที่ 3.8 กฎรายการ $\langle Windy, False\rangle \rightarrow Yes$ มีค่าความเชื่อมั่นเท่ากับ 100% กฎรายการดังกล่าวถูกเพิ่มลงในแบบจำลองและถูกเรียกว่ากฎความสัมพันธ์ระบุคลาสกฎที่ 3 แล้วลบหมายเลขแทรนเซกชันที่เกี่ยวข้องกับกฎดังกล่าวออกจากชุดข้อมูล จนกระทั่งไม่มีกฎรายการใดที่มีค่าสนับสนุนผ่านเกณฑ์ค่าสนับสนุนขั้นต่ำดังตารางที่ 3.9 กระบวนการค้นหากฎความสัมพันธ์ระบุคลาสจึงสิ้นสุด

ขั้นตอนวิธี ECARG ดำเนินการสร้างคลาสเริ่มต้นเพื่อเพิ่มลงในแบบจำลอง ในขั้นตอนนี้คลาสที่ปรากฏในแทรนเซกชันที่เหลืออยู่มากที่สุดจะถูกเลือกเป็นคลาสเริ่มต้น จากตารางที่ 3.9 แทรนเซกชันที่ยังเหลืออยู่เกี่ยวข้องกับคลาส No มากที่สุด ดังนั้นคลาสเริ่มต้นสำหรับแบบจำลอง คือ คลาส No กฎความสัมพันธ์ระบุคลาสทั้งหมดถูกบรรจุในแบบจำลองดังตารางที่ 3.10

ตารางที่ 3.9 ข้อมูลที่ยังเหลืออยู่หลังจากสร้างกฎที่ 3

Outlook						Temperature						Humidity				Windy			
Sunny		Overcast		Rainy		Hot		Mild		Cool		High		Normal		True		False	
~~Y~~	N	Y	~~N~~	Y	~~N~~	~~Y~~	~~N~~	Y	~~N~~	Y	~~N~~	Y	N	Y	~~N~~	Y	N	Y	~~N~~
11								11	14		6		14	11	6	11	6 14		
1	0	0	0	0	0	0	0	1	1	0	1	0	1	1	1	1	2	0	0

ตารางที่ 3.10 กฎความสัมพันธ์ระบุคลาสทั้งหมดในแบบจำลอง

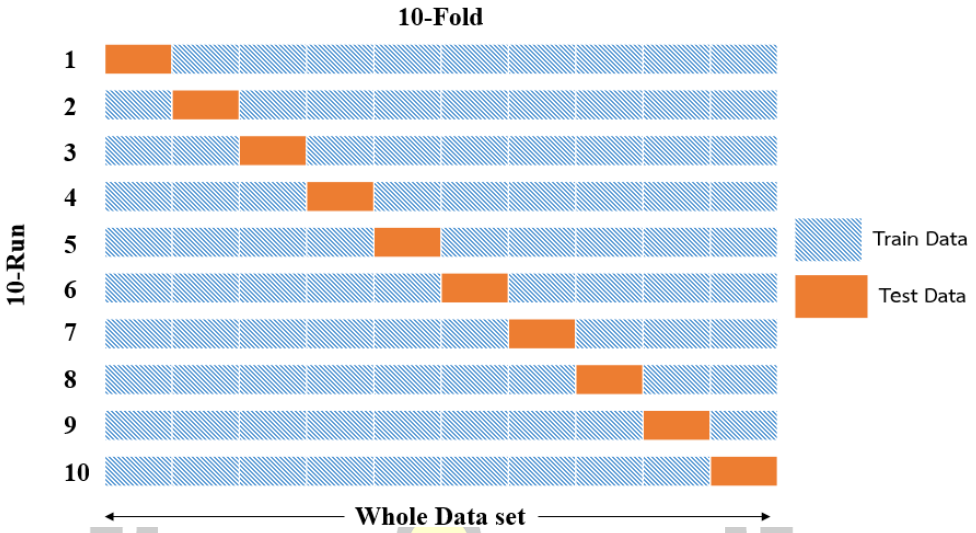
ลำดับ	กฎความสัมพันธ์ระบุคลาส
R1	$\langle Outlook, Overcast \rangle \rightarrow Yes$
R2	$\langle Outlook, Sunny \rangle, \langle Humidity, High \rangle \rightarrow No$
R3	$\langle Windy, False \rangle \rightarrow Yes$
Default Class	No

3.4 การวัดประสิทธิภาพ (Evaluation)

ผู้วิจัยได้ทำการวัดประสิทธิภาพด้านต่าง ๆ เพื่อทราบถึงประสิทธิภาพของขั้นตอนวิธีที่ได้พัฒนาขึ้นมา ได้แก่ ค่าความถูกต้อง ค่าความแม่นยำ ค่าความระลึก ค่าเฉลี่ยประสิทธิภาพโดยรวม จำนวนกฎเฉลี่ยที่ถูกสร้าง เวลาเฉลี่ยในการสร้างแบบจำลอง ปริมาณการใช้หน่วยความจำเฉลี่ยในการสร้างแบบจำลอง ซึ่งมีรายละเอียดดังต่อไปนี้

3.4.1 การแบ่งข้อมูลเพื่อวัดประสิทธิภาพ (Cross Validation)

การแบ่งข้อมูลและขั้นตอนการวัดประสิทธิภาพของการจำแนก เพื่อประเมินความสามารถของการทำนายคลาสคำตอบของตัวจำแนก ในขั้นตอนการแบ่งข้อมูลงานวิจัยนี้ใช้การแบ่งข้อมูลแบบ 10-Fold Cross-Validation คือ แบ่งข้อมูลออกเป็น 10 ชุด จำนวนข้อมูลในแต่ละชุดเท่ากัน และทำการทดสอบ 10 รอบ โดยรอบที่ 1 ข้อมูลชุดที่ 1 จะถูกเลือกเป็นข้อมูลชุดทดสอบและข้อมูลชุดที่ 2 ถึง 10 ถูกเลือกเป็นชุดสอน ในรอบที่สองข้อมูลชุดที่ 2 จะเป็นชุดทดสอบประสิทธิภาพในขณะที่ข้อมูลชุดที่ 1 ชุดที่ 3 จนถึงชุดที่ 10 เป็นชุดสอน กระบวนการนี้จะถูกทำซ้ำจนครบ 10 รอบ แล้วนำค่าประสิทธิภาพที่ได้ในแต่ละรอบมาคำนวณค่าเฉลี่ย วิธีการแบ่งข้อมูลแสดงดังรูปที่ 3.4



รูปที่ 3.4 ตัวอย่างขั้นตอนการทำงานของ 10-Fold Cross-Validation

3.4.2 การวัดประสิทธิภาพการจำแนก

งานวิจัยนี้ใช้วิธีการวัดประสิทธิภาพการจำแนกข้อมูลเชิงความสัมพันธ์ โดยใช้ 1) ค่าความถูกต้อง 2) ค่าความแม่นยำ 3) ค่าความระลึก 4) ค่าเฉลี่ยประสิทธิภาพโดยรวม ซึ่งอ้างอิงเมทริกซ์ความสับสน สำหรับแต่ละชุดข้อมูล

ตารางที่ 3.11 Confusion Matrix

Predict / Actual	Class C_1	Class C_2	Class C_3	...	Class C_n
Class C_1	X_{11}	X_{12}	X_{13}	...	X_{1n}
Class C_2	X_{21}	X_{22}	X_{23}	...	X_{2n}
Class C_3	X_{31}	X_{32}	X_{33}	...	X_{3n}
...	...	...	...	...	...
Class C_n	X_{n1}	X_{n2}	X_{n3}	...	X_{nn}

การวัดประสิทธิภาพการจำแนกอธิบายได้โดยใช้เมทริกซ์ความสับสน ซึ่งเป็นตารางที่จำนวนแถวเท่ากับจำนวนคอลัมน์ โดยจำนวนของคอลัมน์และแถวขึ้นกับจำนวนคลาสที่พิจารณาในชุดข้อมูล งานวิจัยนี้ใช้ชุดข้อมูลมาตรฐานจาก UCI จำนวน 14 ชุด ซึ่งแต่ละชุดข้อมูลมีจำนวนคลาสไม่เท่ากัน เช่น ชุดข้อมูล Glass มีจำนวน 7 คลาส ทำให้ตาราง Confusion Matrix มีขนาด 7x7 เป็นต้น โดยที่ข้อมูลด้านคอลัมน์ เป็นคลาสที่อยู่ในข้อมูลชุดสอน (Actual) และข้อมูลด้านแถวเป็นคลาสที่ทำนายได้จากแบบจำลอง (Predict) ตาราง Confusion Matrix สำหรับ n Class จะมีลักษณะดังตารางที่ 3.11

กำหนดให้ค่า TP (True Positive) คือ จำนวนข้อมูลที่ทำนายถูกว่าเป็นคลาสที่ *i* วิธีการคำนวณแสดงดังสมการที่ 3.2

$$TP_i = X_{ii} \tag{3.2}$$

กำหนดให้ค่า TTP (Total Numbers Of True Positive) คือ จำนวนข้อมูลที่ทำนายถูกว่าเป็นคลาสที่พิจารณาทั้งหมด วิธีการคำนวณแสดงดังสมการที่ 3.3

$$TTP = \sum_{i=1}^{n} x_{ii} \tag{3.3}$$

โดยที่ *n* คือ จำนวนคลาสทั้งหมด

กำหนดให้ค่า TFN (Total Numbers Of False Negative) คือ จำนวนข้อมูลที่ทำนายว่าเป็นคลาสที่ไม่ได้พิจารณาแต่คำตอบเป็นคลาสที่พิจารณา วิธีการคำนวณแสดงดังสมการที่ 3.4

$$TFN = \sum_{\substack{j=1,i=1 \\ j\neq i}}^{n} x_{ij} \tag{3.4}$$

กำหนดให้ค่า TFP (Total Numbers Of False Positive) คือ จำนวนข้อมูลที่ทำนายว่าเป็นคลาสที่พิจารณาแต่คำตอบคือคลาสที่ไม่ได้พิจารณา วิธีการคำนวณแสดงดังสมการที่ 3.5

$$TFP = \sum_{\substack{j=1,i=1 \\ j\neq i}}^{n} x_{ji} \tag{3.5}$$

ค่า TTN (Total Numbers Of True Negative) คือ จำนวนข้อมูลที่ทำนายถูกว่าเป็นคลาสที่ไม่ได้พิจารณา วิธีการคำนวณแสดงดังสมการที่ 3.6

$$TTN = \sum_{\substack{j=1 \\ j\neq i}}^{n} \sum_{\substack{k=1 \\ k\neq i}}^{n} x_{jk} \tag{3.6}$$

การวัดค่าความถูกต้องของการจำแนก (Accuracy) คำนวณจากค่าผลรวมระหว่างจำนวนครั้งที่ทำนายถูก หารด้วยจำนวนการทำนายทั้งหมด วิธีการคำนวณแสดงดังสมการที่ 3.7

$$Accuracy = \frac{TTP}{Total\ Number\ of\ Testing\ Entries} \tag{3.7}$$

การวัดค่าความแม่นยำของการจำแนก (Precision) เป็นการวัดโดยแยกพิจารณาทีละคลาส คำนวณจากจำนวนครั้งที่ทำนายถูกในคลาสที่พิจารณา หารด้วยจำนวนครั้งการทำนายทั้งหมดในคลาสที่พิจารณา ดังสมการที่ 3.8 งานวิจัยชิ้นนี้ทำนายข้อมูลหลายคลาส (Multi class) ดังนั้นค่าความแม่นยำเฉลี่ยของแบบจำลอง คำนวณจากผลรวมค่าความแม่นยำของทุกคลาสหารด้วยจำนวนคลาสทั้งหมด วิธีการคำนวณแสดงดังสมการที่ 3.9

$$Precision_i = \frac{TP_i}{TP_i + TFP_i} \tag{3.8}$$

$$Overall\ Precision = \frac{\sum_{i=1}^{n} Precision_i}{n} \quad 3.9$$

ค่าความระลึกของการจำแนก (Recall) คำนวณจากจำนวนครั้งที่ทำนายถูกในคลาสที่พิจารณา หารด้วยจำนวนครั้งที่ทำนายถูกในคลาสที่พิจารณาบวกจำนวนครั้งที่ทำนายผิดในคลาสที่ไม่ได้พิจารณา ดังสมการที่ 3.10 งานวิจัยชิ้นนี้ทำนายข้อมูลหลายคลาส (Multi class) ดังนั้นค่าความระลึกเฉลี่ยของแบบจำลอง คำนวณจากผลรวมค่าความระลึกของทุกคลาสหารด้วยจำนวนคลาส วิธีการคำนวณแสดงดังสมการที่ 3.11

$$Recall_i = \frac{TP_i}{TP_i + TFN_i} \quad 3.10$$

$$Overall\ Recall = \frac{\sum_{i=1}^{n} Recall_i}{n} \quad 3.11$$

ค่าเฉลี่ยประสิทธิภาพโดยรวม (F-Measure) หาจากค่าเฉลี่ยจากค่าความแม่นยำและค่าความระลึก ระบบที่มีประสิทธิภาพดีจะต้องมีค่าความระลึกและค่าความแม่นยำสูงใกล้เคียงกัน ดังสมการที่ 3.12 งานวิจัยชิ้นนี้ทำนายข้อมูลหลายคลาส (Multi class) ดังนั้นค่าประสิทธิภาพโดยรวมเฉลี่ยของแบบจำลอง คำนวณจากผลรวมค่าประสิทธิภาพโดยรวมของทุกคลาสหารด้วยจำนวนคลาส วิธีการคำนวณแสดงดังสมการที่ 3.13

$$F-measure_i = 2 \times \frac{Precision_i \times Recall_i}{Precision_i + Recall_i} \quad 3.12$$

$$Overall\ F-measure = \frac{\sum_{i=1}^{n} F-measure_i}{n} \quad 3.13$$

3.4.3 จำนวนกฎเฉลี่ยที่ถูกสร้าง

การพิจารณาจำนวนกฎที่ถูกสร้าง คือ การวิเคราะห์จำนวนกฎรายการที่ขั้นตอนวิธีสร้างได้และถูกบรรจุลงในตัวจำแนก หากกฎรายการมีจำนวนมากนอกจากจะทำแบบจำลองมีขนาดใหญ่ยากต่อการจัดการ ยังทำให้การจำแนกข้อมูลใช้เวลานานและยากต่อผู้ใช้งานที่ต้องการวิเคราะห์ความสำคัญของรายการที่เป็นส่วนประกอบของกฎซึ่งส่งผลต่อผลการทำนาย [73] หากแบบจำลองมีจำนวนกฎน้อยแต่ไม่กระทบถึงความถูกต้องในการจำแนกข้อมูลจะส่งผลให้ผู้ใช้งานสามารถวิเคราะห์ข้อมูลจากกฎได้อย่างมีประสิทธิภาพ [41, 44, 73] อย่างไรก็ตามความถูกต้องเป็นปัจจัยที่สำคัญซึ่งต้องพิจารณาควบคู่กับขนาดตัวจำแนกอีกด้วย

จำนวนกฎเฉลี่ยที่ถูกสร้างพิจารณาจากจำนวนกฎภายในแบบจำลองทั้งหมดที่ถูกสร้างขึ้นในแต่ละชุด (Fold) หารด้วยจำนวนชุดข้อมูล ดังสมการที่ 3.14

$$AVGrules = \frac{\sum_{i=1}^{n} R_i}{n} \quad 3.14$$

โดย *AVGRules* คือ จำนวนกฎเฉลี่ยที่ถูกสร้าง

R คือ จำนวนกฎภายในแบบจำลอง

n คือ จำนวนชุดข้อมูล

3.4.4 เวลาเฉลี่ยในการสร้างแบบจำลอง

การพิจารณาเวลาเฉลี่ยในการสร้างแบบจำลอง คือ การวัดผลเวลาตั้งแต่เริ่มทดสอบข้อมูลชุดสอนจนกระทั่งได้แบบจำลองสำหรับการทำนายข้อมูล เพื่อให้ทราบถึงความเร็วในการสร้างแบบจำลองของวิธีการที่นำเสนอในงานวิจัยเปรียบเทียบกับขั้นตอนวิธีที่โดดเด่น ซึ่งการพิจารณาเวลาสามารถใช้ประเมินประสิทธิภาพของขั้นตอนวิธีควบคู่กับการวัดประสิทธิภาพด้านอื่น

เวลาเฉลี่ยในการสร้างแบบจำลอง พิจารณาจากเวลาในแบบจำลองทั้งหมดที่ถูกสร้างขึ้นในแต่ละชุดข้อมูล (Fold) หารด้วยจำนวนชุดข้อมูล ดังสมการที่ 3.15

$$AVGtime = \frac{\sum_{i=1}^{n} T_i}{n} \quad 3.15$$

โดย *AVGTime* คือ เวลาเฉลี่ยในการสร้างแบบจำลอง

T คือ เวลาในการสร้างแบบจำลอง

n คือ จำนวนชุดข้อมูล

3.4.5 การวัดปริมาณการใช้หน่วยความจำเฉลี่ย

การวัดปริมาณการใช้หน่วยความจำเฉลี่ย เพื่อให้ทราบปริมาณการใช้งานหน่วยความจำหลัก (Main memory) ของวิธีการที่นำเสนอเปรียบเทียบกับขั้นตอนวิธีอื่น โดยวัดจากกระบวนการสร้างแบบจำลองทำนายข้อมูลในแต่ละขั้นตอนวิธีที่นำมาเปรียบเทียบ การวัดประสิทธิภาพนี้แสดงให้เห็นว่าขั้นตอนวิธีแนวโน้มของความสามารถรองรับการทำงานกับข้อมูลจริงซึ่งอาจมีขนาดมากกว่าข้อมูลที่ใช้ในการทดสอบ

การใช้หน่วยความจำเฉลี่ยในการสร้างแบบจำลอง พิจารณาจากการใช้หน่วยความจำในการสร้างแบบจำลองทั้งหมดในแต่ละชุดข้อมูล (Fold) หารด้วยจำนวนชุดข้อมูล ดังสมการที่ 3.16

$$AVGmemory = \frac{\sum_{i=1}^{n} M_i}{n} \quad 3.16$$

โดย *AVGMemory* คือ เวลาเฉลี่ยในการสร้างแบบจำลอง

M คือ ปริมาณหน่วยความจำที่ใช้ในการสร้างแบบจำลอง

n คือ จำนวนชุดข้อมูล

3.5 การเปรียบเทียบประสิทธิภาพ (Comparison)

เพื่อให้เห็นถึงผลการทดลองและประสิทธิภาพของขั้นตอนวิธีที่ชัดเจนจึงควรนำผลลัพธ์จากการทดลองที่ได้มาเปรียบเทียบประสิทธิภาพในด้านต่าง ๆ กับขั้นตอนวิธีอื่น โดยเลือกขั้นตอนวิธีที่มีลักษณะ ดังต่อไปนี้

1. ขั้นตอนวิธีต้องสร้างกฎที่มีลักษณะ “ถ้า-แล้ว” ซึ่งเป็นลักษณะของกฎที่ได้จากการจำแนกข้อมูลเชิงความสัมพันธ์

2. ขั้นตอนวิธีต้องวิธีการจัดรูปแบบข้อมูล สร้างกฎ และลดจำนวนแทรนแซกชันที่แตกต่างกับขั้นตอนวิธีที่นำเสนอในงานวิจัย

3. ขั้นตอนวิธีที่นำมาเปรียบเทียบต้องทำการทดลองโดยใช้ชุดข้อมูลมาตรฐานชุดเดียวกับงานวิจัยนี้

โดยในงานวิจัยนี้ได้ทำการเปรียบเทียบกับขั้นตอนวิธีต่อไปนี้

1. ขั้นตอนวิธี CBA นำเสนอการสร้างกฎโดยใช้วิธีการ Apriori ในการสร้างกฎ ใช้การจัดเรียงข้อมูลแบบแนวนอน และอ่านชุดข้อมูลซ้ำหลายครั้งเพื่อคำนวณค่าสนับสนุนและค่าความเชื่อมั่นของกฎ ซึ่งต่างจากการเรียงข้อมูลแนวตั้ง การเปรียบเทียบกับขั้นตอนวิธี CBA เพื่อเปรียบเทียบให้เห็นประสิทธิภาพของการจัดเรียงข้อมูลที่มีผลต่อการสร้างกฎคู่แข่ง

2. ขั้นตอนวิธี CMAR นำ FP-tree และ CR-tree มาใช้ในกระบวนการสร้างกฎและการจำแนก โดยการแบ่งเซตย่อยใน FP-Tree เพื่อค้นหากฎรายการความถี่และเพิ่มกฎรายการลงใน CR-Tree ตามลำดับความถี่ของกฎ ดังนั้นขั้นตอนวิธี CMAR ใช้การอ่านข้อมูลเพียงครั้งเดียวเพื่อสร้างกฎรายการความถี่ทั้งหมด กระบวนการจำแนกข้อมูลขั้นตอนวิธีนี้ใช้การทำนายด้วยหลายกฎ (Multiple rules) บนพื้นฐานของกระบวนการ Chi-Square

3. ขั้นตอนวิธี FACA ใช้การแทนค่าข้อมูลแนวตั้งร่วมกับการดำเนินการเซตผลต่างและใช้การคัดเลือกคุณสมบัติด้วย Chi-Square อย่างไรก็ตามขั้นตอนวิธี FACA ใช้วิธีการสร้างกฎคล้าย Apriori และไม่มีการลดพื้นที่ในการค้นหากฎลง การเปรียบเทียบด้วยจำนวนกฎที่สร้างได้และค่าความถูกต้องจะทำให้ทราบถึงความสำคัญของการลดจำนวนข้อมูลที่ไม่จำเป็นในการค้นหากฎ

บทที่ 4

ผลการวิจัยและการอภิปราย

ในบทนี้ผู้วิจัยได้นำเสนอผลการวิเคราะห์ข้อมูลที่ได้จากการทดสอบการจำแนกข้อมูลเชิงความสัมพันธ์ ประกอบด้วยการตั้งค่าการทดลอง และผลการประเมินประสิทธิภาพตัวจำแนก ผลการวิจัยมีดังนี้

4.1 การตั้งค่าการทดลอง

การทดลองทั้งหมดในงานวิจัยนี้ประมวลผลด้วยเครื่องคอมพิวเตอร์โน๊ตบุ๊ค หน่วยประมวลผลกลางใช้ชิปประมวลผลอินเทล คอร์ i-3 หน่วยความจำหลัก DDR4 ขนาด 8 กิกะไบต์ ขั้นตอนวิธีทั้งหมดพัฒนาด้วยภาษาจาวา (JAVA) กำหนดค่าสนับสนุนขั้นต่ำและค่าความเชื่อมั่นขั้นต่ำในการทดลองให้มีค่าเท่ากับ 2% และ 50% ตามลำดับ อ้างอิงตัวเลขจากงานวิจัย [1, 8] ซึ่งเป็นค่าที่เหมาะสมสำหรับการสร้างแบบจำลองที่ให้ค่าความถูกต้องสูง การทดลองดำเนินการกับชุดข้อมูล 14 ชุดจาก UCI Machine Learning Repository รายละเอียดชุดข้อมูลแสดงในตารางที่ 3.1

เพื่อให้เป็นประสิทธิภาพองแนวทางการคัดเลือกกฎที่ดีที่สุด ขั้นตอนที่นำเสนอได้ผู้วิจัยได้กำหนดให้ประเมินประสิทธิภาพแบบจำลอง 2 กรณี ได้แก่ 1) ขั้นตอนวิธียอมรับเฉพาะกฎที่มีค่าความเชื่อมั่นเท่ากับ 100% เท่านั้น (ECARG) 2) ขั้นตอนวิธียอมรับกฎที่มีค่าความเชื่อมั่นสูงสุด แม้ว่าค่าความเชื่อมั่นจะไม่ถึง 100% (ECARG2)

4.2 ผลการประเมินประสิทธิภาพ

4.2.1 ผลการประเมินค่าความถูกต้อง

จากผลการทดลองในตารางที่ 4.1 แสดงค่าความถูกต้องของการจำแนก ขั้นตอนวิธี CBA CMAR FACA ขั้นตอนวิธี ECARG และขั้นตอนวิธี ECARG2 มีค่าความถูกต้องเท่ากับ 79.24% 66.24% 82.67% 84.02% และ 84.42% ตามลำดับ เมื่อพิจารณาขั้นตอนวิธี ECARG พบว่ามีผลการทดลองที่โดดเด่นกว่าขั้นตอนวิธี CBA ในชุดข้อมูล Anneal Breast Contact Iris Labor Lymph Mushroom Post-operative Vote Wine และ Zoo โดยมีค่าความถูกต้องสูงกว่า 12.02% 3.17% 4.16% 2.66% 17.22% 10.75% 4.75% 13.33% 1.29% 8.90% และ 34.37% ตามลำดับ เมื่อเปรียบเทียบขั้นตอนวิธีที่นำเสนอกับขั้นตอนวิธี CMAR ในชุดข้อมูล Anneal Contact Diabetes Labor Lymph Mushroom Tic-tac-toe Vote Wine และ Zoo ปรากฏว่ามีค่าความถูกต้องสูงกว่า 21.94% 33.33% 10.29% 66.35% 45.27% 11.9% 12.31% 2.67% 35.95% และ 15.79% ตามลำดับ เมื่อเปรียบเทียบค่าความถูกต้องกับขั้นตอนวิธี FACA ในชุดข้อมูล Anneal Cars Contact Labor Lymph Mushroom Post-operative Vote Wine และ Zoo ปรากฏว่ามีค่าความถูกต้องสูงกว่า 7.90% 3.41% 7.5% 5.00% 6.08% 1.63% 2.22% 3.39% 6.71% และ 9% ตามลำดับ อย่างไรก็ตามเมื่อพิจารณาจากข้อมูลในตารางที่ 4.1 แสดงให้เห็นว่าขั้นตอนวิธี ECARG2

ให้ผลการทดลองที่ดีกว่าขั้นตอนวิธี CBA CMAR FACA และ ECARG โดยมีค่าความถูกต้องเฉลี่ยสูงกว่า 2.94% 19.17% 2.73% 1.40% และ 0.40% ตามลำดับ

ผลการทดลองดังกล่าว เนื่องจากขั้นตอนวิธี ECARG ค้นหากฎที่มีค่าความเชื่อมั่น 100% เพื่อสร้างแบบจำลอง เนื่องจากค่าความเชื่อมั่นสูงแสดงถึงความเป็นไปได้ของการปรากฏคลาสนั้นเมื่อพบเซตรายการในแทรนเซกชัน ดังนั้นขั้นตอนวิธี ECARG จึงสร้างแบบจำลองขนาดเล็กแต่ให้ค่าความถูกต้องสูง ในทางตรงกันข้ามขั้นตอนวิธี CBA CMAR และ FACA ค้นหากฎความสัมพันธ์ระบุคลาสที่ผ่านค่าความเชื่อมั่นขั้นต่ำ กฎความสัมพันธ์ระบุคลาสบางกฎมีค่าความเชื่อมั่นที่ไม่สูงนัก ซึ่งทำให้การจำแนกข้อมูลเกิดข้อผิดพลาดส่งผลให้ค่าความถูกต้องของขั้นตอนวิธีที่นำมาเปรียบเทียบนั้นต่ำกว่าขั้นตอนวิธีที่นำเสนอ

ตารางที่ 4.1 ผลการประเมินค่าความถูกต้อง (%)

ชุดข้อมูล	CBA	CMAR	FACA	ECARG	ECARG2
Anneal	83.19	73.27	87.31	95.21	**96.77**
Breast	67.16	**74.83**	72.44	70.33	73.02
Car	78.29	73.73	70.02	73.43	**87.79**
Contact-lenses	66.67	37.5	63.33	**70.83**	65
Diabetes	**74.47**	57.03	73.56	67.32	73.7
Iris	92.67	**97.33**	96.00	95.33	**96**
Labor	75.67	26.32	87.67	**92.67**	84
Lymph	77.76	43.24	82.43	**88.51**	81.81
Mushroom	93.4	86.25	96.52	98.15	**98.9**
Post-operative	56.67	**70**	67.78	**70**	60
Tic-tac-toe	**99.16**	53.03	90.23	65.34	88.94
Vote	94.02	92.64	91.92	**95.31**	95.17
Wined	89.97	62.92	86	**98.87**	97.16
Zoo	60.27	79.21	82.27	95.00	**96.00**
เฉลี่ย	79.24	66.24	82.67	84.02	**84.42**

4.2.2 ผลการประเมินจำนวนกฎเฉลี่ยที่สร้างได้

นอกจากการประเมินผลด้วยค่าความถูกต้องแล้ว การวิเคราะห์จำนวนกฎรายการที่ถูกสร้างเป็นสิ่งที่งานวิจัยนี้ให้ความสนใจ เนื่องจากกฎรายการที่มีจำนวนมากทำให้แบบจำลองการจำแนกข้อมูลมีขนาดใหญ่ จำนวนกฎในแบบจำลองที่มากส่งผลให้ยากต่อการจัดการโดยผู้ใช้งานหรือผู้วิเคราะห์ข้อมูลรวมไปถึงเวลาในการทำนายข้อมูลอีกด้วย ผลการประเมินแสดงดังตารางที่ 4.2

ตารางที่ 4.2 จำนวนกฎเฉลี่ยที่สร้างได้

ชุดข้อมูล	CBA	CMAR	FACA	ECARG	ECARG2
Anneal	3	165	15	**14**	17
Breast	16	127	23	**3**	35
Cars	25	272	9	**5**	18
Contact	9	25	**5**	7	8
Diabetes	56	115	24	**4**	38
Iris	11	38	7	**4**	9
Labor	**8**	297	15	9	9
Lymph	26	465	**15**	19	20
Mushroom	**8**	28	16	12	13
Post-operative	35	51	12	**11**	27
Tic-tac-toe	28	713	12	**6**	33
Vote	30	658	12	11	**10**
Wine	**5**	237	11	9	7
Zoo	**10**	97	10	**10**	**10**
เฉลี่ย	19	240	13	**8**	18

จากผลการทดลองในตารางที่ 4.2 ขั้นตอนวิธี ECARG สร้างกฎได้น้อยกว่าขั้นตอนวิธี CBA ในชุดข้อมูล Anneal Breast Cars Contact Diabetes Iris Labor Lymph Post-operative Tic-tac-toe Vote และ Wine โดยมีจำนวนกฎน้อยกว่า 14 กฎ 13 กฎ 20 กฎ 2 กฎ 42 กฎ 7 กฎ 5 กฎ 14 กฎ 24 กฎ 22 กฎ 19 กฎ และ 2 กฎ ตามลำดับ เมื่อเปรียบเทียบขั้นตอนวิธี ECARG กับขั้นตอนวิธี CMAR พบว่าขั้นตอนวิธีที่นำเสนอสร้างกฎน้อยกว่าในชุดข้อมูล Anneal Breast Cars Diabetes Iris Labor Lymph Mushroom Post-operative Tic-tac-toe Vote Wine และ Zoo โดยมีจำนวนกฎน้อยกว่า 151 กฎ 124 กฎ 267 กฎ 18 กฎ 111 กฎ 34 กฎ 288 กฎ 446 กฎ 16 กฎ 40 กฎ 707 กฎ 647 กฎ และ 228 กฎ ตามลำดับ เมื่อเปรียบเทียบกับขั้นตอนวิธี FACA พบว่าขั้นตอนวิธี ECARG สร้างกฎน้อยกว่าในชุดข้อมูล Anneal Breast Cars Diabetes Iris Labor Mushroom Post-operative Tic-tac-toe Vote และ Wine โดยมีจำนวนกฎน้อยกว่า 1 กฎ 20 กฎ 4 กฎ 20 กฎ 3 กฎ 6 กฎ 4 กฎ 1 กฎ 6 กฎ 1 กฎ และ 2 กฎ ตามลำดับ ความสำเร็จของขั้นตอนวิธี ECARG มาจากการค้นหากฎความสัมพันธ์ระบุคลาสที่มีประสิทธิภาพสูงสุดในแต่ละรอบการทำงาน และการตัดหมายเลขแทรนเซกชันที่ไม่จำเป็นซึ่งนำไปสู่กฎความสัมพันธ์ระบุคลาสที่ไม่จำเป็น

ผลการทดลองดังนี้ เนื่องจากขั้นตอนวิธี ECARG การคัดเลือกกฎความสัมพันธ์ระบุคลาสเพิ่มลงในแบบจำลอง ซึ่งคัดเลือกกฎความสัมพันธ์ระบุคลาสค่าความเชื่อมั่น 100% เท่านั้น หากกฎรายการความยาว 1 ไม่มีกฎใดที่มีค่าความเชื่อมั่น 100% ขั้นตอนวิธี ECARG จะขยายกฎที่มีค่าความเชื่อมั่นสูงที่สุดร่วมกับกฎรายการอื่นที่มีคลาสเดียวกัน เพื่อให้ค้นพบกฎรายการที่มีค่าความเชื่อมั่น

100% เท่านั้น เมื่อค้นพบกฎความสัมพันธ์ระบุคลาสแล้ว แทรนเซกชันที่เกี่ยวข้องกับกฎดังกล่าวจะถูกลบทันทีเพื่อลดโอกาสในการเกิดกฎที่ซ้ำซ้อน ด้วยเหตุผลนี้ขั้นตอนวิธี ECARG จึงมีจำนวนกฎความสัมพันธ์ระบุคลาสในแบบจำลองน้อยที่สุดเมื่อเปรียบเทียบกับขั้นตอนวิธีอื่น

เมื่อพิจารณากฎในชุดข้อมูล Zoo พบว่าขั้นตอนวิธี ECARG สามารถค้นหากฎได้จำนวน 10 กฎ เท่ากับขั้นตอนวิธี CBA และ FACA เมื่อพิจารณาเปรียบเทียบความสามารถในการค้นหาแล้วพบว่าขั้นตอนวิธีที่นำเสนอสามารถหากฎข้อ 5 6 8 และ 9 ดังรูปที่ 4.1 เมื่อเปรียบเทียบกับรูปที่ 4.2 ขั้นตอนวิธี FACA ไม่สามารถค้นพบกฎดังกล่าวได้ นอกจากนี้เมื่อพิจารณาชุดข้อมูล Iris ซึ่งขั้นตอนวิธีที่นำเสนอสามารถค้นหากฎได้จำนวน 4 กฎ ดังรูปที่ 4.3 จำนวนกฎน้อยกว่าขั้นตอนวิธี FACA ซึ่งค้นหากฎได้ทั้งสิ้น 7 กฎ ดังรูปที่ 4.4 นอกจากนี้ลักษณะของกฎทั้ง 4 กฎ เป็นกฎที่ประกอบด้วย 1 เซตรายการ แสดงความสามารถในการค้นหากฎสากล (Global rule) ซึ่งครอบคลุมชุดข้อมูลจำนวนมากส่งผลให้ค่าความถูกต้องสูง

Proposed	
1	feathers=true ->2
2	milk=true ->1
3	leg=8 ->6
4	fins=true -> 4
5	airbone=true ->5
6	eggs=true & breathes=false ->7
7	legs=6 -> 6
8	eggs=true & aquatic=true ->5
9	backbone=true ->3
10	default ->class 3

รูปที่ 4.1 กฎที่ค้นพบในชุดข้อมูล Zoo ด้วยขั้นตอนวิธีที่นำเสนอ

FACA	
1	milk=true ->1
2	feathers=true ->2
3	leg=8 ->7
4	legs=6 -> 6
5	fins=true -> 4
6	backbone=false->7
7	venomous=true ->3
8	legs=0 ->3
9	tail =true ->3
10	aquatic=true ->5
11	hair=false -> 3

รูปที่ 4.2 กฎที่ค้นพบในชุดข้อมูล Zoo ด้วยขั้นตอนวิธี FACA

Proposed
1. petallength=<2.45 ->Iris-setosa
2. petallength=2.45-4.75 ->Iris-versicolor
2. petalwidth=<0.8 ->Iris-setosa
3. petalwidth==>1.75 ->Iris-virginica
default => Iris-versicolor

รูปที่ 4.3 กฎที่ค้นพบในชุดข้อมูล Iris ด้วยขั้นตอนวิธีที่นำเสนอ

FACA
1. petallength=<2.45 ->Iris-setosa
2. petallength=2.45-4.75 ->Iris-versicolor
3. petalwidth==>1.75 ->Iris-virginica
4. petalwidth=0.8-1.75 ->Iris-versicolor
5. petallength==> 4.75 ->Iris-virginica
6. sepalLength=5.55-6.15 ->Iris-versicolor
7. sepalwidth=<2.95 ->Iris-virginica

รูปที่ 4.4 กฎที่ค้นพบในชุดข้อมูล Iris ด้วยขั้นตอนวิธี FACA

เมื่อพิจารณาจากข้อมูลในตารางที่ 4.2 แสดงให้เห็นว่าขั้นตอนวิธีที่นำเสนอแบบที่ 2 (คอลัมน์ ECARG2) สร้างกฎโดยเฉลี่ยได้มากกว่าขั้นตอนวิธีที่นำเสนอแบบที่ 1 จำนวน 10 กฎ อย่างไรก็ตามค่าความถูกต้องเฉลี่ยของขั้นตอนวิธี ECARG2 มากกว่า 3.90% นอกจากนี้ขั้นตอนวิธีที่นำเสนอแบบที่ 2 สร้างกฎน้อยกว่าขั้นตอนวิธี CBA CMAR และ FACA เท่ากับ 3 กฎ 222 กฎ และ 5 กฎ ตามลำดับ

4.2.3 ผลการประเมินเวลาเฉลี่ยในการสร้างแบบจำลอง

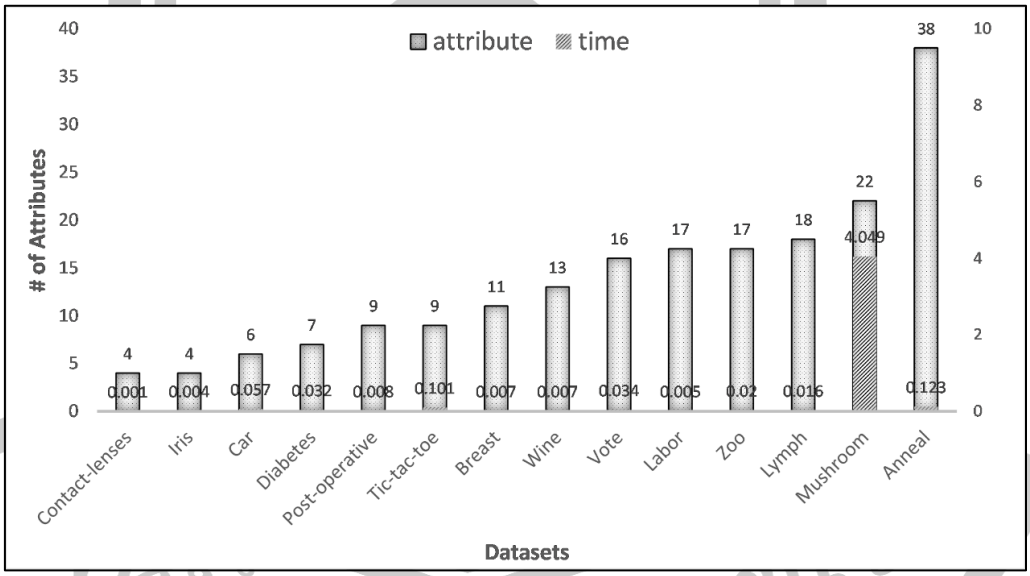
ตารางที่ 4.3 แสดงเวลาเฉลี่ยในการสร้างแบบจำลอง ซึ่งแสดงให้เห็นว่า ขั้นตอนวิธี ECARG สามารถสร้างแบบจำลองใช้เวลาเฉลี่ย 0.319 วินาที ในขณะที่ขั้นตอนวิธี CBA CMAR FACA และ ECARG2 ใช้เวลาเฉลี่ย 2.453 วินาที 0.623 วินาที และ 2.422 วินาที และ 0.335 วินาที ตามลำดับ นอกจากนี้ผลการทดลองแสดงให้เห็นว่าขั้นตอนวิธี ECARG2 ใช้เวลาสร้างแบบจำลองมากกว่าขั้นตอนวิธี ECARG เพียง 0.016 วินาทีเท่านั้น และสร้างกฎมากกว่า 10 กฎ โดยเฉลี่ย อย่างไรก็ตามค่าความถูกต้องเฉลี่ยของขั้นตอนวิธี ECARG2 สูงกว่าขั้นตอนวิธี ECARG เพียง 0.40% เหตุที่เป็นเช่นนี้เนื่องจาก ขั้นตอนวิธี ECARG ใช้วิธีการลดแทรนแซกชันที่เกี่ยวข้องกับกฎที่มีประสิทธิภาพทันทีเมื่อค้นพบกฎนั้น ส่งผลให้จำนวนของแทรนแซกชันที่ต้องพิจารณากฎลดลงอย่างรวดเร็ว ส่งผลต่อเวลาที่ใช้ในการสร้างแบบจำลอง

ตารางที่ 4.3 เวลาในการสร้างแบบจำลอง (วินาที)

ชุดข้อมูล	CBA	CMAR	FACA	ECARG	ECARG2
Anneal	1.05	**0.098**	0.877	0.123	0.164
Breast	0.67	0.169	0.185	**0.007**	0.027

ตารางที่ 4.3 เวลาในการสร้างแบบจำลอง (วินาที) (ต่อ)

ชุดข้อมูล	CBA	CMAR	FACA	ECARG	ECARG2
Cars	0.22	0.249	0.64	**0.057**	0.062
Contact	0.01	0.075	0.004	**0.001**	0.002
Diabetes	1.16	0.107	0.558	**0.032**	0.085
Iris	0.03	0.008	0.01	**0.004**	**0.004**
Labor	1.17	0.924	0.027	**0.005**	**0.004**
Lymph	1.32	3.782	3.7	**0.016**	**0.016**
Mushroom	25.83	**0.104**	21.5	4.049	4.128
Post-operative	0.09	0.041	0.063	**0.008**	0.012
Tic-tac-toe	0.23	0.235	0.8	**0.101**	0.135
Vote	1.54	2.601	5.3	**0.034**	**0.034**
Wine	0.12	0.273	0.19	**0.007**	**0.007**
Zoo	0.9	0.05	0.047	0.02	**0.013**
เฉลี่ย	2.453	0.623	2.422	**0.319**	0.335



รูปที่ 4.5 กราฟเปรียบเทียบจำนวนแอททริบิวต์และเวลาในการสร้างแบบจำลอง

ผลการทดลองการใช้เวลาเฉลี่ยในการสร้างแบบจำลองของขั้นตอนวิธี ECARG เปรียบเทียบกับจำนวนแอททริบิวต์แสดงดังรูปที่ 4.5 จากผลการทดลองแสดงว่า เมื่อจำนวนแอททริบิวต์เพิ่มขึ้น เวลาที่ใช้ในการสร้างแบบจำลองมีลักษณะแนวโน้มเป็นเส้นตรง (Linear) ยกเว้นชุดข้อมูล Mushroom ซึ่งมีจำนวน 22 แอททริบิวต์ ที่ใช้เวลาในการสร้างแบบจำลองสูงกว่าชุดข้อมูลอื่น ผลการทดลองนี้แสดงว่าจำนวนแอททริบิวต์ไม่ส่งผลต่อเวลาในสร้างแบบจำลองของขั้นตอนวิธี ECARG

เนื่องจากข้อมูลภายในแอททริบิวต์ที่มีค่าความเชื่อต่ำจะถูกกำจัดออกไปทันทีในขั้นตอนการลบข้อมูลที่เกี่ยวข้องกับกฎความสัมพันธ์ระบุคลาส

4.2.4 ผลการประเมินการใช้หน่วยความจำเฉลี่ยในการสร้างแบบจำลอง

ตารางที่ 4.4 แสดงผลการใช้หน่วยความจำเฉลี่ย ซึ่งจากผลการทดลองแสดงให้เห็นว่า ผลการทดลองแสดงว่าขั้นตอนวิธี ECARG ใช้หน่วยความจำโดยเฉลี่ย 4.61 เมกะไบต์ ต่ำกว่าขั้นตอนวิธี CBA CMAR FACA และ ECARG2 เท่ากับ 22.62 เมกะไบต์ 73.15 เมกะไบต์ 36.57 เมกะไบต์ และ 0.98 ตามลำดับ เมื่อพิจารณาอย่างละเอียดพบว่าขั้นตอนวิธีที่นำเสนอ ใช้หน่วยความจำน้อยที่สุดใน 12 จาก 14 ชุดข้อมูล เหตุที่เป็นเช่นนี้เนื่องจากวิธีการลดแทรนแซกชันที่เกี่ยวข้องกับกฎที่มีประสิทธิภาพทันที เมื่อค้นพบกฎนั้นของขั้นตอนวิธี ECARG สามารถลดจำนวนของแทรนแซกชันที่ต้องประมวลผลลงอย่างรวดเร็ว ส่งผลให้ขั้นตอนวิธีที่นำเสนอลดการใช้งานหน่วยความจำลงอย่างมาก

ตารางที่ 4.4 ผลการวัดปริมาณการใช้หน่วยความจำ (เมกะไบต์)

ชุดข้อมูล	CBA	CMAR	FACA	ECARG	ECARG2
Anneal	73.47	29.16	10.78	**10.68**	13.38
Breast	25.44	23.96	24.4	**1.92**	3.54
Cars	60.08	21.17	8.98	**3.05**	3.76
Contact	2.65	**0.99**	1.87	1.78	1.84
Diabetes	28.08	26.74	24.61	**3.01**	7.3
Iris	4.16	2.4	1.88	**1.17**	**1.17**
Labor	18.34	420.88	124.01	**1.95**	**1.95**
Lymph	27.31	250.93	231.75	**2.86**	**2.86**
Mushroom	28.89	29.12	24.52	**24.27**	24.31
Post-operative	15.17	8.78	16.38	**2.03**	2.61
Tic-tac-toe	31.76	62.23	12.76	**4.73**	8.44
Vote	23.57	2.65	3.13	**3.09**	3.15
Wine	20.87	175.36	59.52	**1.82**	**1.82**
Zoo	21.41	34.33	31.93	2.2	**2.13**
เฉลี่ย	27.23	77.76	41.18	**4.61**	5.59

4.2.5 ผลการประเมินค่าความแม่นยำ

ตารางที่ 4.5 แสดงค่าความแม่นยำของการจำแนก ซึ่งพบว่า ขั้นตอนวิธี CBA CMAR FACA ECARG และ ECARG2 มีค่าความแม่นยำโดยเฉลี่ยจาก 14 ชุดข้อมูล เท่ากับ 74.19% 55.93% 72.01% 82.59% และ 81.15% ตามลำดับ โดยขั้นตอนวิธี ECARG มีค่าความแม่นยำเฉลี่ยสูงที่สุด เนื่องจากขั้นตอนวิธี ECARG คัดเลือกกฎจากเซตรายการที่มีนัยสำคัญสูงสุดต่อการจำแนกคลาส

ภายในชุดข้อมูลเหล่านั้น ส่งผลให้อัตรา False Positive มีค่าน้อยกว่าขั้นตอนวิธีอื่น ส่งผลให้แบบจำลองที่สร้างโดยขั้นตอนวิธี ECARG มีค่าความแม่นยำสูงกว่าขั้นตอนวิธีที่นำมาเปรียบเทียบ

ตารางที่ 4.5 ผลการประเมินค่าความแม่นยำ (%)

ชุดข้อมูล	CBA	CMAR	FACA	ECARG	ECARG2
Anneal	69.85	67.29	42.36	61.85	**96.66**
Breast	65.17	72.13	**73.09**	70.26	68.4
Cars	69.88	33.33	40.51	**74.76**	**74.76**
Contact	49.1	31.82	51.67	**71.67**	61.67
Diabetes	**74.3**	50.98	73.3	65.1	71.26
Iris	92.7	**98.64**	91.85	91.41	95.83
Labor	66.62	31	75	**90.33**	85.5
Lymph	79.94	47.13	78.01	**86.23**	73.74
Mushroom	94.1	89.57	96.51	98.25	**98.93**
Post-operative	48.9	17.5	**70**	66.46	41.99
Tic-tac-toe	**98.9**	43.3	68.53	95.32	89.1
Vote	**95**	72.63	91.57	93.54	94.44
Wine	84.29	71.69	92.02	**98.97**	94.55
Zoo	49.92	56.07	63.75	**92.04**	89.33
เฉลี่ย	74.19	55.93	72.01	**82.59**	81.15

4.2.6 ผลการประเมินค่าความระลึก

ตารางที่ 4.6 แสดงค่าความระลึกของการจำแนก ซึ่งแสดงให้เห็นว่าขั้นตอนวิธี CBA CMAR FACA ECARG และ ECARG2 มีค่าความระลึกโดยเฉลี่ยจาก 14 ชุดข้อมูล เท่ากับ 78.68% 55.56% 64.44% 74.35% และ 79.05% ขั้นตอนวิธี ECARG2 มีค่าความระลึกเฉลี่ยสูงที่สุด สาเหตุที่ผลการทดลองเป็นดังนี้เนื่องจาก ขั้นตอนวิธี ECARG2 เลือกกฎรายการที่มีค่าความเชื่อมั่นสูงที่สุดในแต่ละรอบของกระบวนการค้นหากฎ โดยไม่จำกัดว่าต้องเป็นกฎรายการที่มีความเชื่อมั่น 100% เท่านั้น ส่งผลให้เมื่อทำนายชุดข้อมูลแล้วสามารถลดจำนวน False negative ลงไปได้อย่างมาก ในทางตรงกันข้ามขั้นตอนวิธี ECARG เลือกกฎรายการค่าความเชื่อมั่นเท่ากับ 100% เท่านั้น ส่งผลให้ในการค้นหากฎที่มีประสิทธิภาพสำหรับคลาสที่มีส่วนน้อย (Minority class) ไม่พบกฎรายการที่มีค่าความเชื่อมั่น 100% ด้วยเหตุผลนี้ในขั้นตอนการจำแนกข้อมูลสำหรับชุดข้อมูลที่มีคลาสส่วนน้อย จึงมีค่า False negative สูง ซึ่งค่า False negative ที่สูงทำให้ค่าความระลึกมีค่าต่ำ

ตารางที่ 4.6 ผลการประเมินค่าความระลึก

ชุดข้อมูล	CBA	CMAR	FACA	ECARG	ECARG2
Anneal	**83.18**	32.39	44.99	60.65	82.94

ตารางที่ 4.6 ผลการประเมินค่าความระลึก (ต่อ)

ชุดข้อมูล	CBA	CMAR	FACA	ECARG	ECARG2
Breast	67.15	61.38	60.81	50	**68.47**
Cars	**78.29**	33.29	25.16	33.02	68.64
Contact	58.3	66.67	48.33	**71.67**	61.67
Diabetes	74.5	47.18	**75.33**	50	71.17
Iris	92.7	**97.33**	95.33	89.44	96.5
Labor	75.66	26.01	75.92	**87.5**	86.25
Lymph	77.76	49.19	40.86	**78.06**	70.5
Mushroom	93.4	85.74	96.54	94.97	**98.88**
Post-operative	**56.7**	25	46.67	46.11	37.41
Tic-tac-toe	**98.9**	44.47	60.74	95.56	86.23
Vote	**94.9**	73.02	92.08	94.1	94.45
Wine	89.96	66.76	92.93	**98.33**	94.19
Zoo	60.27	69.34	46.43	**91.48**	89.42
เฉลี่ย	78.69	55.56	64.44	74.35	**79.05**

4.2.7 ผลการประเมินค่าประสิทธิภาพโดยรวม

ตารางที่ 4.7 แสดงค่าประสิทธิภาพโดยรวม ซึ่งแสดงให้เห็นว่า ขั้นตอนวิธี CBA CMAR FACA ECARG และ ECARG2 มีค่าประสิทธิภาพโดยรวมเท่ากับ 76.27% 54.71% 67.35% 78.25% และ 80.09% ตามลำดับ เมื่อพิจารณาโดยละเอียดพบว่าขั้นตอนที่นำเสนอมีค่าประสิทธิภาพโดยรวมสูงกว่าขั้นตอนวิธี CBA ในชุดข้อมูล Labor Lymph Mushroom Post-operative Wine และ Zoo โดยมีค่าสูงกว่า 18.04% 3.11% 2.83% 1.93% 11.62% และ 37.15% ตามลำดับ เมื่อเปรียบเทียบค่าประสิทธิภาพโดยรวมระหว่างขั้นตอนวิธีที่นำเสนอกับขั้นตอนวิธี CMAR พบว่ามีค่าสูงกว่า ในชุดข้อมูล Anneal Cars Contact Diabetes Labor Lymph Mushroom Post-operative Tic-tac-toe Vote Wine และ Zoo โดยมีค่าสูงกว่า 17.51% 12.50 28.59% 7.55% 60.61% 33.80% 8.97% 33.86% 51.56% 20.99% 29.51% และ 29.76% ตามลำดับ เมื่อเปรียบเทียบค่าประสิทธิภาพโดยรวมระหว่างขั้นตอนวิธีที่นำเสนอกับขั้นตอนวิธี FACA พบว่ามีค่าสูงกว่า ในชุดข้อมูล Anneal Cars Contact Labor Lymph Mushroom Tic-tac-toe Vote Wine และ Zoo โดยมีค่าสูงกว่า 17.61% 14.77% 21.73% 13.44% 28.31% 0.06% 31.04% 1.99% 6.18% และ 38.03% ตามลำดับ

ปัจจัยที่เป็นเช่นนี้เนื่องจากค่าประสิทธิภาพโดยรวมเป็นสัดส่วนผสมระหว่างค่าความแม่นยำและค่าความระลึก ขั้นตอนวิธี ECARG มีค่าความระลึกเฉลี่ยที่ต่ำกว่าขั้นตอนวิธี ECARG2 ส่งผลให้ผลวิจัยด้านค่าประสิทธิภาพโดยรวมต่ำกว่าขั้นตอนวิธี ECARG2

ตารางที่ 4.7 ผลการประเมินค่าประสิทธิภาพโดยรวม

ชุดข้อมูล	CBA	CMAR	FACA	ECARG	ECARG2
Anneal	75.93	43.73	43.64	61.24	**89.28**
Breast	66.15	66.32	66.39	58.42	**68.43**
Cars	**73.85**	33.31	31.04	45.81	71.57
Contact	53.31	43.08	49.94	**71.67**	61.67
Diabetes	**74.40**	49.01	74.30	56.56	71.21
Iris	92.70	**97.98**	93.56	90.41	96.16
Labor	70.85	28.29	75.46	**88.89**	85.87
Lymph	78.83	48.14	53.63	**81.94**	72.08
Mushroom	93.75	87.61	96.52	96.58	**98.90**
Post-operative	52.51	20.59	**56.00**	54.45	39.57
Tic-tac-toe	**98.90**	43.88	64.40	95.44	87.64
Vote	**94.95**	72.82	91.82	93.82	94.44
Wine	87.03	69.14	92.47	**98.65**	94.37
Zoo	54.61	62.00	53.73	**91.76**	89.37
เฉลี่ย	76.27	54.71	67.35	78.25	**80.09**

4.3 ผลการวิเคราะห์การใช้คลาสเริ่มต้น (Default class) สำหรับการจำแนกข้อมูล

ขั้นตอนวิธี ECARG มุ่งค้นหาเฉพาะกฎที่มีค่าความเชื่อมั่น 100% เท่านั้น สำหรับชุดข้อมูลที่ไม่มีคุณสมบัติใดที่มีค่าความเชื่อมั่น 100% ขั้นตอนวิธีจะไม่สามารถสร้างกฎที่เพียงพอต่อการทำนายได้ เมื่อตัวอย่างข้อมูลชุดทดสอบไม่สามารถถูกทำนายคลาสได้ด้วยกฎในตัวจำแนก คลาสเริ่มต้นจะถูกใช้เพื่อการจำแนกข้อมูลซึ่งส่งผลต่อค่าความถูกต้อง ผลลัพธ์จากตารางที่ 4.8 แสดงว่าขั้นตอนวิธี ECARG ซึ่งอนุญาตให้เพิ่มกฎที่มีค่าความเชื่อมั่นสูงที่สุดได้ในกรณีที่ไม่พบกฎที่มีค่าความเชื่อมั่น 100% มีอัตราการใช้กฎเริ่มต้นเฉลี่ยเพียง 0.4% จาก 14 ชุดข้อมูล ถึงแม้จะสร้างกฎมากกว่าขั้นตอนวิธีที่นำเสนอ แต่ให้ความถูกต้องสูงกว่า 1.4%

ตารางที่ 4.8 ผลการประเมินการใช้คลาสเริ่มต้น

ชุดข้อมูล	จำนวนแถว	จำนวนการใช้คลาสเริ่มต้น			
		ECARG		ECARG2	
		ครั้ง	%	ครั้ง	%
Anneal	898	8	9%	2	2%
Breast	286	25	87%	0	0%
Cars	1,728	71	41%	1	1%
Contact	24	0	0%	0	0%
Diabetes	768	67	87%	0	0%

ตารางที่ 4.8 ผลการประเมินการใช้คลาสเริ่มต้น (ต่อ)

ชุดข้อมูล	จำนวนแถว	จำนวนการใช้คลาสเริ่มต้น			
		ECARG		ECARG2	
		ครั้ง	%	ครั้ง	%
Iris	150	1	7%	0	0%
Labor	57	0	0%	0	0%
Lymph	148	0	0%	0	0%
Mushroom	8,214	37	5%	10	1%
Post-operative	90	3	33%	0	0%
Tic-tac-toe	958	77	80%	1	1%
Vote	435	5	11%	0	0%
Wine	178	0	0%	0	0%
Zoo	101	0	0%	0	0%
เฉลี่ย	996	21	25.8%	1	**0.4%**

เมื่อพิจารณาอย่างละเอียดโดยใช้ข้อมูลจากตารางที่ 4.2 พบว่าเมื่อใช้ขั้นตอนวิธีที่นำเสนอกับชุดข้อมูล Breast Diabetes และ Tic-tac-toe จะสามารถสร้างกฎได้เพียงแค่ 3 กฎ 4 กฎ และ 6 กฎ ตามลำดับ และมีอัตราการใช้คลาสเริ่มต้นสำหรับการจำแนกข้อมูลสูงถึง 87% สำหรับชุดข้อมูล Breast และ Diabetes และชุดข้อมูล Tic-tac-toe มีอัตราการใช้คลาสเริ่มต้น 80% แสดงให้เห็นถึงจุดด้อยของการมุ่งค้นหาเพียงเฉพาะกฎที่มีค่าความเชื่อมั่น 100% เท่านั้น เนื่องจากขั้นตอนวิธีที่นำเสนอเน้นการหากฎที่ให้ค่าความเชื่อมั่น 100% ซึ่งชุดข้อมูลทั้งสามชุด ประกอบด้วยข้อมูลที่มีเซตรายการที่ไม่สามารถระบุคลาสได้ 100% ส่งผลให้ไม่สามารถหากฎความสัมพันธ์ระบุคลาสค่าความเชื่อมั่น 100% ได้ หรือได้จำนวนกฎน้อย ทำให้ข้อมูลทดสอบส่วนใหญ่ไม่ตรงกับกฎที่อยู่ในแบบจำลอง คลาสเริ่มต้นจึงถูกใช้เพื่อการจำแนกสูง

ผลการประเมินความถูกต้องของกฎความสัมพันธ์ระบุคลาสและคลาสเริ่มต้นในแบบจำลองของขั้นตอนวิธี ECARG แสดงในตารางที่ 4.9 คอลัมน์ที่ 2 แสดงการกระจายตัวของข้อมูลในคลาส คลาสที่มีตัวอักษรหนาเป็นคลาสเริ่มต้นในแบบจำลอง คอลัมน์ที่ 3 แสดงความห่างระหว่างการกระจายตัวของคลาสที่ปรากฏในชุดข้อมูลมากที่สุดและคลาสที่ปรากฏในชุดข้อมูลน้อยที่สุด จากผลการทดลองพบว่ากฎความสัมพันธ์ระบุคลาสในแบบจำลองมีค่าความถูกต้องเฉลี่ย 67.06% และคลาสเริ่มต้นมีค่าความถูกต้องเฉลี่ย 16.97% ซึ่งแสดงให้เห็นว่ากฎที่สร้างขึ้นมีประสิทธิภาพในการจำแนก เมื่อพิจารณาโดยละเอียดพบว่าข้อมูล Breast Diabetes Post-operative และ Tic-tac-toe ค่าความถูกต้องของคลาสเริ่มต้นสูงกว่าความถูกต้องของกฎความสัมพันธ์ระบุคลาสถึง 49.95% 38.98% และ 18.78% ตามลำดับ

ตารางที่ 4.9 ผลการประเมินความถูกต้องของกฎความสัมพันธ์ระบุคลาสและคลาสเริ่มต้นในแบบจำลองของขั้นตอนวิธี ECARG

ชุดข้อมูล	การกระจายของคลาส	ระยะห่างคลาส (%)	ค่าความถูกต้อง (%)		
			รวม	CARs	คลาสเริ่มต้น
Anneal	**class 2=88** class 3=608 class 4=0 class 5=60 class U=34	1.32	95.21	88.64	6.57
Breast	**no-recurrence= 201** recurrence=85	42.29	70.33	10.19	60.14
Car	**unacc=1210** acc=384 vgood=65 good=69	5.37	73.43	58.50	14.93
Contact-lenses	hard lenses=4 **soft lenses=5** no lenses=15	26.67	70.83	70.83	0.00
Diabetes	**negative=500** positive=268	46.4	67.32	14.17	53.15
Iris	Setosa=50 **Versicolour=50** Virginica=50	100	95.33	85.33	10.00
Labor	**bad=20** good=37	54.05	92.67	92.67	0.00
Lymph	normal find=2 metastases=81 **malign lymph=61** fibrosis=4	2.47	88.51	86.51	2.00
Mushroom	**edible=4,208** poisonous=3,916	93.06	98.15	95.47	2.68
Post-operative	**A=64** S=24 I=2	3.13	70.00	33.33	36.67

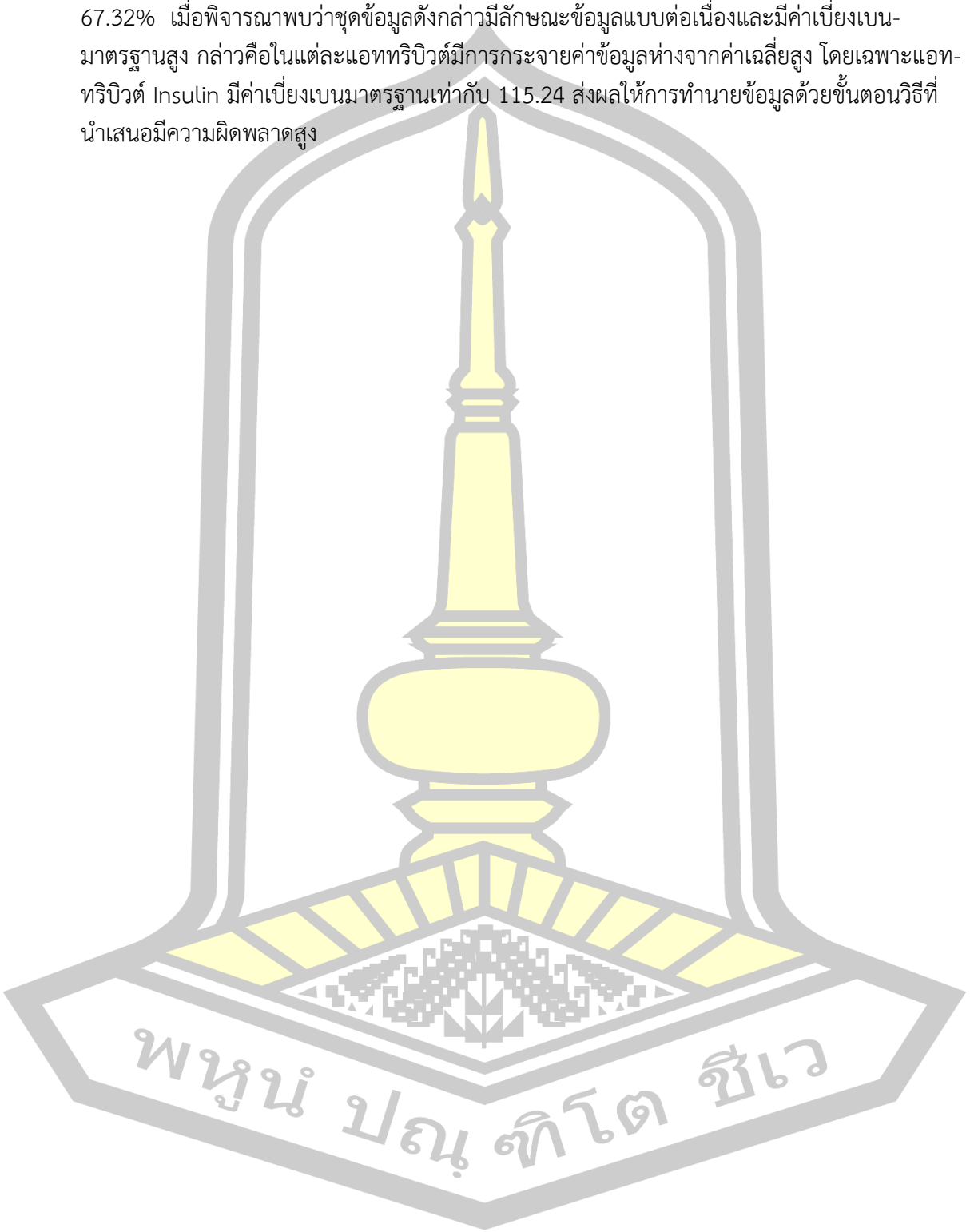
ตารางที่ 4.9 ผลการประเมินความถูกต้องของกฎความสัมพันธ์ระบุคลาสและคลาสเริ่มต้นในแบบจำลองของขั้นตอนวิธี ECARG (ต่อ)

ชุดข้อมูล	การกระจายของคลาส	ระยะห่างคลาส	ค่าความถูกต้อง (%)		
			รวม	CARs	คลาสเริ่มต้น
Tic-tac-toe	**positive=625** negative=333	53.28	65.34	23.28	42.06
Vote	**democrat=196** republican=239	82.01	95.31	88.00	7.32
Wined	**class 1=59** class 2=71 class 3=48	67.61	98.87	98.87	0.00
Zoo	mammal=41 bird=20 reptile=5 fish=13 amphibian=4 insect=8 **invertebrate=10**	9.76	95.00	93.00	2.00
เฉลี่ย			84.02	67.06	16.97

4.4 ผลการวิเคราะห์ลักษณะข้อมูล

ขั้นตอนวิธีที่นำเสนอในงานวิจัยนี้สามารถทำนายข้อมูลได้ค่าความถูกต้องสูงในชุดข้อมูลที่มีลักษณะหลากหลาย เช่น ชุดข้อมูล Zoo ซึ่งมีลักษณะข้อมูลแบบตัวเลข (Numeric) มีจำนวนตัวอย่าง 101 ประกอบด้วย 17 แอททริบิวต์ 7 คลาส และจำนวนรายการที่แตกต่างกันในชุดข้อมูล 36 รายการ (distinct) ผลการทดลองแสดงให้เห็นว่าขั้นตอนที่นำเสนอให้ผลลัพธ์ค่าความถูกต้องสูงที่สุดเท่ากับ 93.37% ในขณะที่ชุดข้อมูล Iris ซึ่งมีลักษณะข้อมูลแบบต่อเนื่อง ขั้นตอนวิธีที่นำเสนอโดยใช้การแบ่งกลุ่มข้อมูล (discretize) ด้วยวิธี Entropy based ประมวลผลได้ค่าความถูกต้อง 95.33% สูงกว่าขั้นตอนวิธี CBA 2.66% และเท่ากับขั้นตอนวิธี FACA แต่จำนวนกฎที่ถูกค้นพบโดยขั้นตอนที่นำเสนอมีจำนวนน้อยกว่า เมื่อพิจารณาชุดข้อมูล Contact ซึ่งกระจายข้อมูลอย่างสม่ำเสมอในทุกแอททริบิวต์ส่งผลให้ค่าความถูกต้องสูงกว่าขั้นตอนวิธีอื่น สำหรับข้อมูลที่มีลักษณะกระจายข้อมูลไม่สม่ำเสมออย่างเช่นชุดข้อมูล Lymph ประกอบด้วยข้อมูล 148 รายการ 18 แอททริบิวต์ 4 คลาส และจำนวนรายการในแอททริบิวต์ที่แตกต่างกัน 58 รายการ ขั้นตอนวิธีที่นำเสนอสามารถให้ค่าความถูกต้องสูงที่สุดเท่ากับ 88.51%

อย่างไรก็ตามผลการทดลองกับชุดข้อมูล Diabetes ปรากฏว่ามีค่าความถูกต้องเพียง 67.32% เมื่อพิจารณาพบว่าชุดข้อมูลดังกล่าวมีลักษณะข้อมูลแบบต่อเนื่องและมีค่าเบี่ยงเบนมาตรฐานสูง กล่าวคือในแต่ละแอททริบิวต์มีการกระจายค่าข้อมูลห่างจากค่าเฉลี่ยสูง โดยเฉพาะแอททริบิวต์ Insulin มีค่าเบี่ยงเบนมาตรฐานเท่ากับ 115.24 ส่งผลให้การทำนายข้อมูลด้วยขั้นตอนวิธีที่นำเสนอมีความผิดพลาดสูง



บทที่ 5

สรุปผล อภิปรายผล และข้อเสนอแนะ

งานวิจัยในหัวข้อการสร้างกฎที่มีประสิทธิภาพสำหรับการจำแนกข้อมูลเชิงความสัมพันธ์มีวัตถุประสงค์เพื่อพัฒนาวิธีการขุดค้นกฎที่มีประสิทธิภาพสำหรับการจำแนกเชิงความสัมพันธ์ ผู้วิจัยได้นำเสนอขั้นตอนวิธีด้วยกัน 2 วิธี ได้แก่ 1) ขั้นตอนวิธี ECARG ซึ่งยอมรับกฎความสัมพันธ์ระบุคลาสที่มีค่าความเชื่อมั่น 100% และ 2) ขั้นตอนวิธี ECARG2 ซึ่งยอมรับกฎความสัมพันธ์ระบุคลาสที่มีค่าความเชื่อมั่นที่สูงที่สุด โดยทำการทดลองกับชุดข้อมูลจาก UCI จำนวน 14 ชุด เปรียบเทียบกับขั้นตอนวิธี CBA CMAR และ FACA สามารถสรุปผลการวิจัย และอภิปรายผล ดังต่อไปนี้

5.1 สรุปผลการวิจัย

1. สรุปผลการประเมินด้านค่าความถูกต้อง พบว่าขั้นตอนวิธี ECARG มีความโดดเด่นกว่าขั้นตอนวิธี CBA CMAR และ FACA โดยมีค่าความถูกต้องสูงกว่า 4.78% 17.79% และ 1.35% ตามลำดับ ในขณะที่ขั้นตอนวิธี ECARG2 สามารถทำนายข้อมูลถูกต้องสูงกว่าขั้นตอนวิธี ECARG 0.40% นอกจากนี้เมื่อพิจารณาเปรียบเทียบค่าความถูกต้องในลักษณะ ชนะ-แพ้-เสมอ (win-lost-tie) ระหว่างขั้นตอนวิธี ECARG2 เปรียบเทียบกับขั้นตอนวิธี CBA CMAR FACA และ ECARG ได้ผลลัพธ์มีค่าเท่ากับ 11-3-0 11-3-0 9-4-1 และ 8-6-0 ตามลำดับ อย่างไรก็ตามเมื่อพิจารณาอย่างละเอียดพบว่า ขั้นตอนวิธี ECARG มีค่าความถูกต้องสูงที่สุด ถึง 6 จาก 14 ชุดข้อมูล ได้แก่ ชุดข้อมูล Contact Labor Lymph Post-operative Vote และ Wine เมื่อเปรียบเทียบกับทุกขั้นตอนวิธี ผลลัพธ์ดังกล่าวเนื่องจากขั้นตอนวิธี ECARG ค้นหากฎที่มีค่าความเชื่อมั่น 100% เพื่อสร้างแบบจำลอง เนื่องจากค่าความเชื่อมั่นสูงแสดงถึงความเป็นไปได้ของการปรากฏคลาสนั้นเมื่อพบเซตรายการในแทรนเซกชัน ดังนั้นขั้นตอนวิธี ECARG จึงสร้างแบบจำลองขนาดเล็กแต่ให้ค่าความถูกต้องสูง ในขณะที่ขั้นตอนวิธี CBA CMAR และ FACA สร้างแบบจำลองจากกฎความสัมพันธ์ระบุคลาสซึ่งผ่านค่าความเชื่อมั่นขั้นต่ำ กฎความสัมพันธ์ระบุคลาสบางกฎมีค่าความเชื่อมั่นที่ไม่สูงนักจึงทำให้การจำแนกข้อมูลเกิดข้อผิดพลาดส่งผลให้ค่าความถูกต้องของขั้นตอนวิธีที่นำมาเปรียบเทียบนั้นต่ำกว่าขั้นตอนวิธีที่นำเสนอในหลายชุดข้อมูล

2. สรุปผลการประเมินจำนวนกฎเฉลี่ยที่สร้างได้พบว่า ขั้นตอนวิธี ECARG สร้างกฎความสัมพันธ์ระบุคลาสเฉลี่ยเพียง 8 ข้อ ต่ำกว่าจำนวนกฎโดยเฉลี่ยจากขั้นตอนวิธี CBA CMAR FACA และ ECARG2 เท่ากับ 11 กฎ 232 กฎ 5 กฎ และ 10 กฎ ตามลำดับ เมื่อพิจารณาอย่างละเอียดพบว่า สร้างกฎน้อยที่สุด 8 จาก 14 ชุดข้อมูล ได้แก่ ชุดข้อมูล Anneal Breast Cars Diabetes Iris Post-operative Tic-tac-toe และ Zoo ผลการทดลองดังกล่าวแสดงถึงประสิทธิภาพในการค้นหากฎที่มีประสิทธิภาพของขั้นตอนวิธี ECARG สาเหตุเนื่องจากการคัดเลือกกฎความสัมพันธ์ระบุคลาสเพิ่มลงในแบบจำลอง ซึ่งคัดเลือกกฎรายการค่าความเชื่อมั่น 100% เท่านั้น หากกฎรายการความยาว 1 ไม่มีกฎใดที่มีค่าความเชื่อมั่น 100% ขั้นตอนวิธี ECARG จะขยายกฎที่มีค่าความเชื่อมั่นสูงที่สุดร่วมกับกฎรายการอื่นที่มีคลาสเดียวกัน ด้วยวิธีการขยายแนวกว้าง (Breath first search) เพื่อให้ค้นพบกฎรายการที่มีประสิทธิภาพสูงสุดเท่านั้น นอกจากนี้เมื่อค้นพบกฎ

ความสัมพันธ์ระบุคลาสแล้ว แทรนเซกชันที่เกี่ยวข้องกับกฎดังกล่าวจะถูกลบทันทีเพื่อลดโอกาสในการเกิดกฎที่ซ้ำซ้อน ด้วยเหตุผลนี้ทำให้ขั้นตอนวิธี ECARG มีจำนวนกฎความสัมพันธ์ระบุคลาสในแบบจำลองน้อยที่สุดเมื่อเปรียบเทียบกับขั้นตอนวิธีอื่น

3. สรุปผลการประเมินด้านเวลาเฉลี่ยที่ใช้ในการสร้างแบบจำลอง พบว่าขั้นตอนวิธี ECARG สามารถสร้างแบบจำลองใช้เวลาเฉลี่ย 0.319 วินาที เร็วกว่าขั้นตอนวิธี CBA CMAR FACA และ ECARG2 เท่ากับ 2.13 วินาที 0.3 วินาที 2.21 วินาที และ 0.016 วินาที ตามลำดับ นอกจากนี้สำหรับชุดข้อมูล Mushroom ซึ่งมีโอกาสสร้างกฎได้ถึง 41 ล้านกฎ [74] ขั้นตอนวิธีที่นำเสนอใช้ประโยชน์จากการลดจำนวนชุดข้อมูลที่ซ้ำซ้อนลง ส่งผลให้ใช้เวลาเพียง 4.049 วินาทีในการสร้างแบบจำลอง ในขณะที่ขั้นตอนวิธี CBA CMAR และ FACA .ใช้เวลา 25.83 วินาที 0.104 วินาที 21.5 และ 4.128 วินาที เพื่อสร้างแบบจำลอง เมื่อพิจารณาโดยละเอียดพบว่าขั้นตอนวิธี ECARG2 ใช้เวลาสร้างแบบจำลองมากว่าขั้นตอนวิธี ECARG 0.016 วินาทีเท่านั้น และสร้างกฎมากกว่า 10 ข้อ อย่างไรก็ตามค่าความถูกต้องเฉลี่ยของขั้นตอนวิธี ECARG2 สูงกว่าขั้นตอนวิธี ECARG เพียง 0.40% ปัจจัยความสำเร็จของขั้นตอนวิธี ECARG คือ การแทนค่าแนวตั้งที่สามารถลดเวลาในการประมวลผลได้อย่างมาก ขั้นตอนวิธีที่นำเสนอทำให้กระบวนการกำหนดค่ารายการความถี่มีประสิทธิภาพ โดยเฉพาะงานที่เกี่ยวข้องกับการนับค่าสนับสนุนและค่าความเชื่อมั่นของกฎรายการ นอกจากนี้การลดพื้นที่ในการค้นหากฎรายการซึ่งรายการที่ไม่จำเป็นจะถูกลบในทุกรอบการทำงานโดยใช้วิธีการเซตผลต่าง ส่งผลให้ลดขนาดของแบบจำลองลงได้

4. สรุปผลการประเมินการใช้หน่วยความจำเฉลี่ยในการสร้างแบบจำลองพบว่า ขั้นตอนวิธี ECARG ใช้หน่วยความจำในการสร้างแบบจำลองเฉลี่ยต่ำที่สุด โดยใช้หน่วยความจำเพียง 4.61 เมกะไบต์ ต่ำกว่าขั้นตอนวิธี CBA CMAR FACA และ ECARG2 เท่ากับ 22.62 เมกะไบต์ 73.15 เมกะไบต์ 36.57 เมกะไบต์ และ 0.98 เมกะไบต์ ตามลำดับ เมื่อพิจารณาอย่างละเอียดพบว่าขั้นตอนวิธี ECARG ใช้หน่วยความจำในขั้นตอนการสร้างแบบจำลองน้อยที่สุด ใน 12 จาก 14 ชุดข้อมูล ปัจจัยสำคัญที่ทำให้ขั้นตอนวิธี ECARG มีผลการทดลองดังกล่าว มาจากวิธีการลดพื้นที่ในการค้นหาโดยลบแทรนเซกชันที่ไม่จำเป็น เพื่อลดเวลาและการใช้หน่วยความจำในการสร้างกฎที่ไม่จำเป็นลง เมื่อชุดข้อมูลเหลือเพียงเซตรายการที่นัยสำคัญทางสถิติเท่านั้นนอกจากจะทำให้สามารถสร้างกฎที่มีประสิทธิภาพได้แล้ว ยังทำให้การประมวลผลของขั้นตอนวิธีใช้พื้นที่หน่วยความจำในน้อยลงจนได้ผลลัพธ์งานวิจัยดังกล่าว แตกต่างจากขั้นตอนที่นำมาเปรียบเทียบซึ่งสร้างกฎรายการโดยเริ่มขยายจากกฎรายความยาว 1 แล้วขยายไปเป็นกฎรายการความยาว 2 และขยายต่อไปจนกระทั่งไม่สามารถค้นพบกฎที่ผ่านเกณฑ์ขั้นต่ำ ซึ่งค่าสนับสนุนและค่าความเชื่อมั่นของกฎรายการจะคำนวณจากแทรนเซกชันทั้งหมดที่มีอยู่ ซึ่งแทรนเซกชันเหล่านั้นถูกจัดเก็บไว้ในหน่วยความจำตลอดการค้นหากฎ จนกระทั่งสร้างแบบจำลองเสร็จสิ้น ส่งผลให้การใช้งานหน่วยความจำของขั้นตอนวิธีที่นำมาเปรียบเทียบมีค่าสูงกว่าขั้นตอนวิธี ECARG

5. สรุปผลการประเมินค่าความแม่นยำพบว่าขั้นตอนวิธี ECARG มีค่าความแม่นยำเฉลี่ยจาก 14 ชุดข้อมูล เท่ากับ 82.59 เมื่อเปรียบเทียบกับขั้นตอนวิธี CBA CMAR FACA และ ECARG2 พบว่าค่าความแม่นยำเฉลี่ยสูงกว่า 8.39% 26.65% 10.57% และ 1.43% ตามลำดับ ปัจจัยที่ทำให้ขั้นตอนวิธี ECARG นำเสนอวิธีการเลือกคัดเลือกกฎจากเซตรายการที่มีนัยสำคัญสูงสุดต่อการจำแนกคลาสภายในในชุดข้อมูลเหล่านั้น ทำให้แบบจำลองสามารถจำแนกชุดข้อมูลและลดผลการ

ทำนายที่เป็นค่า False Positive จนกระทั่งมีค่าน้อยกว่าขั้นตอนวิธีอื่น ส่งผลให้ขั้นตอนวิธี ECARG สามารถสร้างแบบจำลองเพื่อจำแนกข้อมูลที่มีค่าความแม่นยำสูง

6. สรุปผลการประเมินด้านค่าความระลึกพบว่า ค่าความระลึกของขั้นตอนวิธี CBA CMAR FACA ECARG และ ECARG2 เท่ากับ 78.68% 55.56% 64.44% 74.35% และ 79.05% ตามลำดับ ส่งผลให้ขั้นตอนวิธี ECARG มีค่าความระลึกโดยเฉลี่ยเปรียบเทียบกับขั้นตอนวิธี CMAR และ FACA สูงกว่า 18.79% และ 9.91% ตามลำดับ เมื่อเปรียบเทียบกับขั้นตอนวิธี CBA และ ECARG2 พบว่ามีค่าความระลึกต่ำกว่า 4.34% และ 4.70% ตามลำดับ สาเหตุที่ผลการทดลองเป็นดังนี้เนื่องจาก ในชุดข้อมูลที่ไม่สมดุล เช่น Breast Cars Diabetes และ Post-operative ผู้วิจัยพบว่าแอททริบิวต์และค่าในชุดข้อมูลสำหรับข้อมูลส่วนน้อย (Minority) มีนัยยะสำคัญทางสถิติน้อย จนกระทั่งไม่สามารถค้นหากฎรายการที่มีค่าความเชื่อมั่นเท่ากับ 100% ได้ ขั้นตอนวิธี ECARG จึงไม่ได้สามารถสร้างกฎความสัมพันธ์ระบุคลาสสำหรับคลาสส่วนน้อย เป็นผลให้ในขั้นตอนการจำแนกข้อมูลส่วนน้อยซึ่งไม่สามารถจำแนกข้อมูลได้จากกฎที่มีในแบบจำลอง จึงถูกพิจารณาจำแนกข้อมูลเป็นคลาสเริ่มต้น โดยในชุดข้อมูล Breast Cars Diabetes และ Post-operative มีการจำแนกข้อมูลชุดทดสอบเป็นคลาสเริ่มต้นเท่ากับ 50% 33.02% 50% และ 46.67%. ตามลำดับ ส่งผลให้เกิดค่า False Negative จำนวนมาก ส่งผลให้ค่าความระลึกเฉลี่ยของขั้นตอนวิธี ECARG น้อยกว่าขั้นตอนวิธี ECARG2

7. สรุปผลการประเมินค่าประสิทธิภาพโดยรวมพบว่า ขั้นตอนวิธีที่นำเสนอมีค่าสูงกว่าขั้นตอนวิธี CBA CMAR และ FACA 1.98% 23.54% และ 10.90% แสดงถึงความสามารถของแบบจำลองซึ่งสามารถจำแนกข้อมูลได้ถูกต้องมากกว่าการทำนายไม่ถูกต้อง อย่างไรก็ตามเมื่อเปรียบเทียบกับขั้นตอนวิธี ECARG2 พบว่ามีค่าประสิทธิภาพโดยรวมต่ำกว่า 1.84% ปัจจัยที่เป็นเช่นนี้เนื่องจากค่าประสิทธิภาพโดยรวมเป็นสัดส่วนผสมระหว่างค่าความแม่นยำและค่าความระลึก ขั้นตอนวิธี ECARG มีค่าความระลึกเฉลี่ยที่ต่ำกว่าขั้นตอนวิธี ECARG2 ส่งผลให้ผลวิจัยด้านค่าประสิทธิภาพโดยรวมต่ำกว่าขั้นตอนวิธี ECARG2 อย่างไรก็ตามเมื่อพิจารณาผลการวิจัยด้านอื่น เช่น ค่าความถูกต้อง จำนวนกฎเฉลี่ยที่สร้างได้ เวลาเฉลี่ยในการสร้างแบบจำลอง และปริมาณหน่วยความจำที่ใช้ในการสร้างแบบจำลอง พบว่าขั้นตอนวิธี ECARG มีความโดดเด่นมากกว่าทุกขั้นตอนวิธีที่นำมาเปรียบเทียบ

8. ผลการทดลองแสดงว่าแบบจำลองที่ได้จากขั้นตอนวิธี ECARG พบว่าสามารถทำนายข้อมูลได้ถูกต้อง 84.02% เมื่อพิจารณาค่าความถูกต้องเฉพาะกฎความสัมพันธ์ระบุคลาสที่สร้างขึ้นจากขั้นตอนวิธีที่นำเสนอ พบว่าสามารถทำนายข้อมูลได้ถูกต้อง 67.07% โดยเฉลี่ย ในขณะที่คลาสเริ่มต้นทำนายถูกต้อง 16.97% โดยเฉลี่ย จากข้อมูลดังกล่าวพบว่าวิธีการหากฎความสัมพันธ์ระบุคลาสด้วยค่าความเชื่อมั่น 100% ถือเป็นกฎที่มีประสิทธิภาพในการจำแนกชุดข้อมูล

5.2 อภิปรายผลการวิจัย

งานวิจัยนี้นำเสนอขั้นตอนวิธีที่เพิ่มประสิทธิภาพการจำแนกเชิงความสัมพันธ์ ขั้นตอนวิธีที่นำเสนอไม่จำเป็นต้องดำเนินกระบวนการเรียงลำดับและตัดกฎ เนื่องจากขั้นตอนวิธีที่นำเสนอหลีกเลี่ยงการสร้างกฎคู่แข่งโดยเลือกกฎที่มีค่าความถูกต้องสูงที่สุดที่สามารถค้นพบเป็นลำดับแรกสุด

แตกต่างจากขั้นตอนวิธีการจำแนกเชิงความสัมพันธ์แบบดั้งเดิม ซึ่งต้องสร้างกฎความสัมพันธ์ระบุคลาสจำนวนมากในขั้นตอนการค้นหาและสร้างกฎ กระบวนการนี้ต้องสร้างกฎคู่แข่งให้ได้มากที่สุดเพื่อเพิ่มโอกาสค้นพบกฎที่มีประสิทธิภาพต่อการทำนาย ซึ่งส่งผลกระทบต่อเวลาและหน่วยความจำในการที่ใช้ในการสร้างแบบจำลอง ยิ่งไปกว่านั้นขั้นตอนวิธี ECARG มีกระบวนการลดพื้นที่ค้นหากฎโดยการพิจารณาแทรนแซกชันที่มีนัยสำคัญทางสถิติต่ำและลบออกไปอย่างรวดเร็ว นอกจากนี้การจัดรูปแบบการแทนค่าข้อมูลแนวตั้ง การดำเนินการเซตผลต่าง และการอินเทอร์เซกชันเซต ได้ถูกนำมาใช้ประโยชน์ช่วยให้ขั้นตอนวิธีสามารถคำนวณค่าสนับสนุน ค่าความเชื่อมั่น และการลบแทรน-แซคชันที่ไม่จำเป็น เป็นไปอย่างง่ายและมีประสิทธิภาพ ซึ่งนำไปสู่การลดเวลาและหน่วยความจำที่ใช้ในการสร้างแบบจำลองเพื่อทำนายข้อมูล

โดยสรุปแล้วขั้นตอนวิธี ECARG สามารถสร้างแบบจำลองเพื่อการจำแนกเชิงความสัมพันธ์ด้วยกฎที่มีประสิทธิภาพซึ่งสามารถทำนายผลและให้ค่าความถูกต้องสูง อีกทั้งมีจำนวนกฎที่มีประสิทธิภาพในแบบจำลองน้อยกว่าทุกขั้นตอนวิธีโดยเฉลี่ย ซึ่งง่ายต่อการตีความโดยผู้ใช้หรือนักวิเคราะห์ นอกจากนี้ขั้นตอนวิธี ECARG ให้ค่าเฉลี่ยของเวลาในการสร้างแบบจำลองและการใช้งานหน่วยความจำน้อยที่สุดเมื่อเทียบกับขั้นตอนวิธี CBA CMAR และ FACA

5.3 ข้อเสนอแนะ

5.3.1 การจำแนกข้อมูลที่ไม่สมดุล (Imbalance data)

ผลการทดลองด้านค่าความระลึกแสดงให้เห็นถึงข้อควรปรับปรุงของขั้นตอนวิธี ECARG ที่ทำนายชุดข้อมูลที่ไม่สมดุลได้ผลลัพธ์ที่ไม่น่าพอใจ นอกจากค่า False Negative มีจำนวนมากแล้วยังส่งผลต่อประสิทธิภาพโดยรวมของแบบจำลอง หากสามารถหาวิธีการค้นหากฎที่มีประสิทธิภาพสำหรับคลาสส่วนน้อยได้จะสามารถเพิ่มประสิทธิภาพโดยรวมของขั้นตอนวิธีได้ ประเด็นที่น่าสนใจคือ หลังจากลบหมายเลขแทรนเซกชันที่ไม่จำเป็นซึ่งเกี่ยวข้องกับกฎความสัมพันธ์ระบุคลาสที่ค้นพบแล้ว ทำอย่างไรจึงจะสามารถหาเซตรายการที่มีนัยสำคัญต่อข้อมูลส่วนน้อยจากแทรนเซกชันที่ยังเหลืออยู่

5.3.2 การจำแนกข้อมูลแบบกระจาย (Sparse data classification)

ขั้นตอนวิธี ECARG ออกแบบมาเพื่อจำแนกข้อมูลแบบหนาแน่น เมื่อนำข้อมูลแบบกระจาย เช่น ข้อมูลข่าวลวง มาการจำแนกข่าวลวง พบว่าเวลาในการประมวลผลค่อนข้างช้า เหตุที่เป็นเช่นนี้เนื่องจาก 1) แอททริบิวต์ที่มีจำนวนมาก และ 2) ลักษณะของข้อมูลในแอททริบิวต์ที่ไม่หลากหลาย ทำให้การใช้ประโยชน์จากการแทนค่าข้อมูลแนวตั้งและทฤษฎีเซตถูกใช้งานอย่างไม่มีประสิทธิภาพ ส่งผลให้ได้ค่าความถูกต้องที่ไม่น่าพึงพอใจ หากสามารถปรับปรุงโครงสร้างข้อมูลที่มีประสิทธิภาพดีให้กับขั้นตอนวิธี ECARG จะสามารถรองรับการทำงานกับข้อมูลประเภทกระจายได้ผลลัพธ์ที่ดีขึ้น

บรรณานุกรม

[1] Liu B, M Y, Hsu W. Integrating Classification And Association Rule Mining. Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining 1998;

[2] Thabtah F, Hadi W, Abdelhamid N, Issa A. Prediction phase in associative classification mining. International Journal of Software Engineering and Knowledge Engineering 2011; 21[06]: 855-876.

[3] Abdelhamid N. Multi-label rules for phishing classification. Applied Computing and Informatics 2015; 11[1]: 29-46.

[4] Abdelhamid N, Ayesh A, Thabtah F. Phishing detection based Associative Classification data mining. Expert Systems with Applications 2014; 41[13]: 5948-5959.

[5] Hadi We, Aburub F, Alhawari S. A new fast associative classification algorithm for detecting phishing websites. Applied Soft Computing 2016; 48[1]: 729-734.

[6] Singh J, Kamra A, Singh H. Prediction of heart diseases using associative classification. 5th International Conference on Wireless Networks and Embedded Systems (WECON); October 2016; 1-7.

[7] Jabbar M, Deekshatulu B, Chandra P. Heart Disease Prediction System using Associative Classification and Genetic Algorithm. arXiv:13035919 [cs, stat] 2013;

[8] Hadi We, Issa G, Ishtaiwi A. ACPRISM: Associative classification based on PRISM algorithm. Information Sciences 2017; 417[1]: 287-300.

[9] Wang D. Analysis and detection of low quality information in social networks. 2014 IEEE 30th International Conference on Data Engineering Workshops; March 2014; 350-354.

[10] Song K, Lee K. Predictability-based collective class association rule mining. Expert Systems with Applications 2017; 79[1]: 1-7.

[11] Nguyen L, Vo B, Hong T-P, Thanh HC. CAR-Miner: An efficient algorithm for mining class-association rules. Expert Systems with Applications 2013; 40[6]: 2305-2311.

[12] Nguyen L, Nguyen NT. An improved algorithm for mining class association rules

using the difference of Obidsets. Expert Systems with Applications 2015; 42[9]: 4361-4369.

[13] Li W, Han J, Pei J. CMAR: accurate and efficient classification based on multiple class-association rules. Proceedings 2001 IEEE International Conference on Data Mining; 2001; 369-376.

[14] Nguyen D, Vo B, Le B. CCAR: An efficient method for mining class association rules with itemset constraints. Engineering Applications of Artificial Intelligence 2015; 37[1]: 115-124.

[15] Nguyen D, Nguyen L, Vo B, Pedrycz W. Efficient mining of class association rules with the itemset constraint. Knowledge-Based Systems 2016; 103[1]: 73-88.

[16] Nguyen L. A Quick Method for Querying Top-k Rules from Class Association RuleSet. Journal of Universal Computer Science 2016; 22[6]: 822-835.

[17] Fournier-Viger P, Wu C-W, Tseng VS. Mining Top-K Association Rules. Canadian Conference on Artificial Intelligence; Springer, Berlin, Heidelberg; 61-73.

[18] Rokach L, Maimon O. Data Mining with Decision Trees: Theory and Applications. World Scientific; 2014.

[19] Cendrowska J. PRISM: An algorithm for inducing modular rules. International Journal of Man-Machine Studies 1987; 27[4]: 349-370.

[20] Thabtah F, Qabajeh I, Chiclana F. Constrained dynamic rule induction learning. Expert Systems with Applications 2016; 63[1]: 74–85.

[21] Ruggieri S. Efficient C4.5 [classification algorithm]. IEEE Transactions on Knowledge and Data Engineering 2002; 14[2]: 438-444.

[22] Gregor K, Danihelka I, Graves A, Rezende DJ, Wierstra D. DRAW: A Recurrent Neural Network For Image Generation. arXiv:150204623 [cs] 2015;

[23] Li W, Zhao R, Xiao T, Wang X. DeepReID: Deep Filter Pairing Neural Network for Person Re-Identification. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2014; 152-159.

[24] Eberhart R. Neural Network PC Tools: A Practical Guide. Academic Press; 2014.

[25] Elmehdwi Y, Samanthula BK, Jiang W. Secure k-nearest neighbor query over encrypted data in outsourced environments. 2014 IEEE 30th International Conference on Data Engineering; March 2014; 664-675.

[26] Bidder O, Campbell H, Gómez-Laich A, Urgé P, Walker J, Cai Y, et al. Love Thy Neighbour: Automatic Animal Behavioural Classification of Acceleration Data Using the K-Nearest Neighbour Algorithm. PLOS ONE 2014; 9[2]: e88609.

[27] Saini I, Singh D, Khosla A. QRS detection using K-Nearest Neighbor algorithm (KNN) and evaluation on standard ECG databases. Journal of Advanced Research 2013; 4[4]: 331-344.

[28] Sarath KNVD, Ravi V. Association rule mining using binary particle swarm optimization. Engineering Applications of Artificial Intelligence 2013; 26[8]: 1832-1840.

[29] Pradhan GN, Prabhakaran B. Association Rule Mining in Multiple, Multidimensional Time Series Medical Data. Journal of Healthcare Informatics Research 2017; 1[1]: 92-118.

[30] Borgelt C. Frequent item set mining. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 2012; 437-456.

[31] Agrawal R, Srikant R, others. Fast algorithms for mining association rules. 1994; 487–499.

[32] Han J, Pei J, Yin Y, Mao R. Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach. Data Mining and Knowledge Discovery 2004; 8[1]: 53-87.

[33] Zaki M, Gouda K. Fast Vertical Mining Using Diffsets. 2003; New York, USA: ACM; 326–335.

[34] Deng Z. DiffNodesets: An efficient structure for fast mining frequent itemsets. Applied Soft Computing 2016; 41[1]: 214-223.

[35] Uy RL, Marcos N. Fast 1-itemset frequency count using CUDA. 2016 IEEE Region 10 Conference (TENCON); November 2016; 210-213.

[36] Thabtah F, Cowling P, Peng Y. MMAC: a new multi-class, multi-label associative classification approach. Fourth IEEE International Conference on Data Mining, 2004 ICDM '04; November 2004; 217-224.

[37] Alwidian J, Hammo BH, Obeid N. WCBA: Weighted classification based on association rules algorithm for breast cancer disease. Applied Soft Computing 2018; 62[1]: 536-549.

[38] Thabtah F, Hammoud S, Abdel-Jaber H. Parallel Associative Classification Data Mining Frameworks Based MapReduce. Parallel Processing Letters 2015; 25[02]: 1550002.

[39] Abdelhamid N, Thabtah F. Associative Classification Approaches: Review and Comparison. Journal of Information & Knowledge Management 2014; 13[03]: 1450027.

[40] Thabtah F. A review of associative classification mining. The Knowledge Engineering Review 2007; 22[1]: 37-65.

[41] Deng H, Runger G, Tuv E, Bannister W. CBC: An associative classifier with a small number of rules. Decision Support Systems 2014; 59[1]: 163-170.

[42] Quinlan J. C4.5: Programs for Machine Learning. Elsevier; 2014.

[43] Chen G, Liu H, Yu L, Wei Q, Zhang X. A new approach to classification based on association rule mining. Decision Support Systems 2006; 42[2]: 674-689.

[44] Abdelhamid N, Ayesh A, Thabtah F, Ahmadi S, Hadi W. MAC: A Multiclass Associative Classification Algorithm. Journal of Information & Knowledge Management 2012; 11[02]: 1250011.

[45] Thabtah F, Cowling P, Peng Y. MCAR: multi-class classification based on association rule. The 3rd ACS/IEEE International Conference onComputer Systems and Applications, 2005; January 2005;

[46] Cohen W. Fast effective rule induction-RIPPER. Twelfth International Conference on Machine Learning 1995; 115–123.

[47] Tayal K, Ravi V. Particle Swarm Optimization Trained Class Association Rule Mining: Application to Phishing Detection. 2016; New York, NY, USA: ACM; 13:11–13:18.

[48] Phishing Corpus. [Online]. 1 Feb 2015 [cited 15 Apr 2018]; https://academictorrents.com/details/a77cda9a9d89a60dbdfbe581adf6e2df9197995a.

[49] SpamAssassin: Welcome to SpamAssassin. [Online]. 28 Jan 2020 [cited 15 Apr 2018]; https://spamassassin.apache.org.

[50] Lakshmi KP, Reddy CRK. Fast Rule-Based Prediction of Data Streams Using Associative Classification Mining. 2015 5th International Conference on IT

Convergence and Security (ICITCS); August 2015; 1-5.

[51] Lakshmi KP, Reddy CRK. Compact Tree for Associative Classification of Data Stream Mining. International Journal of Computer Science Issues 2012; 9[2]: 624-628.

[52] Djenouri Y, Comuzzi M, Djenouri D. SS-FIM: Single Scan for Frequent Itemsets Mining in Transactional Databases. Pacific-Asia Conference on Knowledge Discovery and Data Mining; 2017/05/23; Springer, Cham; 644-654.

[53] Nguyen L, Vo B, Nguyen L, Fournier-Viger P, Selamat A. ETARM: an efficient top-k association rule mining algorithm. Applied Intelligence 2017; 48[5]: 1-13.

[54] Deng Z, Lv S-L. Fast mining frequent itemsets using Nodesets. Expert Systems with Applications 2014; 41[10]: 4505-4512.

[55] Deng Z, Lv S-L. PrePost+: An efficient N-lists-based algorithm for mining frequent itemsets via Children–Parent Equivalence pruning. Expert Systems with Applications 2015; 42[13]: 5424-5432.

[56] Vo B, Le B. A Novel Classification Algorithm Based on Association Rules Mining. Pacific Rim Knowledge Acquisition Workshop; 2008/12/15; Springer, Berlin, Heidelberg; 61-75.

[57] Nguyen D, Vo B. Mining Class-Association Rules with Constraints. *Knowledge and Systems Engineering*: Springer, Cham 2014:307-318.

[58] Nguyen L, Vo B, Nguyen HS, Nguyen SH. Mining Class Association Rules with Synthesis Constraints. Asian Conference on Intelligent Information and Database Systems; 2017 Apr 3; Springer, Cham; 556-565.

[59] Nguyen L, Nguyen HT, Vo B, Nguyen N-T. A Method for Query Top-K Rules from Class Association Rule Set-Inserted-Based. Asian Conference on Intelligent Information and Database Systems; 2016/3/14; Springer, Berlin, Heidelberg; 644-653.

[60] Abdelhamid N, Thabtah F, Abdel-jaber H. Phishing detection: A recent intelligent machine learning comparison based on models content and features. 2017 IEEE International Conference on Intelligence and Security Informatics (ISI); July 2017; 72-77.

[61] Antonie L, Zaïane OR, Holte RC. Redundancy Reduction: Does It Help Associative

Classifiers? ; 2016; New York, NY, USA: ACM; 867–874.

[62] Antonie Ml, Zaiane OR, Holte RC. Learning to Use a Learned Model: A Two-Stage Approach to Classification. Sixth International Conference on Data Mining (ICDM'06); 33-42.

[63] Fayyad U, Irani K. Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning. 1993; https://trs.jpl.nasa.gov/handle/2014/35171

[64] Drummond C, Holte RC. Cost curves: An improved method for visualizing classifier performance. Machine Learning 2006; 65[1]: 95-130.

[65] Kamalov F, Thabtah F. A Feature Selection Method Based on Ranked Vector Scores of Features for Classification. Annals of Data Science 2017; 4[4]: 483-502.

[66] Alwidian J, Hammo B, Obeid N. FCBA : Fast Classification Based on Association Rules Algorithm. International Journal of Computer Science and Network Security (IJCSNS) 2016; 16[12]: 117–127.

[67] Schmid MR, Iqbal F, Fung BCM. E-mail authorship attribution using customized associative classification. Digital Investigation 2015; 14[1]: S116-S126.

[68] Iqbal F, Binsalleeh H, Fung BCM, Debbabi M. Mining writeprints from anonymous e-mails for forensic investigation. Digital Investigation 2010; 7[1-2]: 56-64.

[69] Iqbal F, Binsalleeh H, Fung BCM, Debbabi M. A unified data mining solution for authorship analysis in anonymous textual communications. Information Sciences 2013; 23198-112.

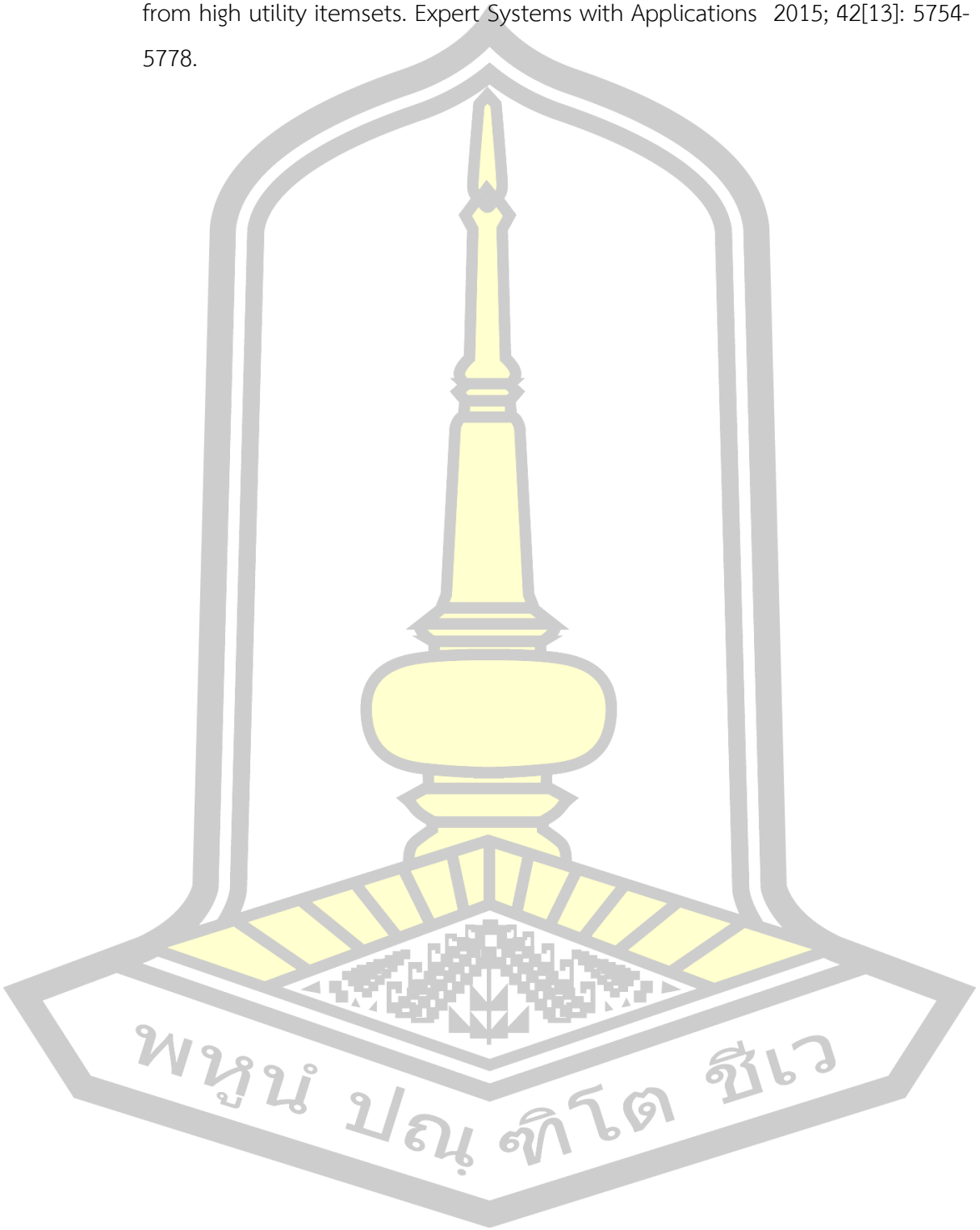
[70] Dua D, Graff C. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml/index.php.

[71] Dougherty J, Kohavi R, Sahami M. Supervised and Unsupervised Discretization of Continuous Features - ScienceDirect. Machine Learning Proceedings 1995; 9 july 1995; California: 194-202.

[72] Jiang S-y, Li X, Zheng Q, Wang L-x. Approximate Equal Frequency Discretization Method. 2009 WRI Global Congress on Intelligent Systems; 2009; Xiamen, China: IEEE; 514-518.

[73] Abdelhamid N, Jabbar AA, Thabtah F. Associative Classification Common Research Challenges. 2016 45th International Conference on Parallel Processing Workshops (ICPPW); August 2016; 432-437.

[74] Sahoo J, Das AK, Goswami A. An efficient approach for mining association rules from high utility itemsets. Expert Systems with Applications 2015; 42[13]: 5754-5778.



พหุน ปณฺ ฑิโต ชีเว

ประวัติผู้เขียน

ชื่อ	นายชาติวุฒิ ธนาจิรันธร
วันเกิด	วันที่ 15 เมษายน พ.ศ. 2522
สถานที่เกิด	
สถานที่อยู่ปัจจุบัน	บ้านเลขที่ 555/26 หมู่ 10 ตำบลอิสาณ อำเภอเมือง จังหวัดบุรีรัมย์ รหัสไปรษณีย์ 31000
ตำแหน่งหน้าที่การงาน	พนักงานมหาวิทยาลัย สายวิชาการ
สถานที่ทำงานปัจจุบัน	คณะวิทยาศาสตร์ มหาวิทยาลัยราชภัฏบุรีรัมย์ เลขที่ 439 ถนนจิระ อำเภอเมือง จังหวัดบุรีรัมย์ 31000 โทรศัพท์ 0-4461-1221 โทรสาร 0-4461-2858
ประวัติการศึกษา	พ.ศ. 2544 วิทยาศาสตรบัณฑิต (วท.บ.) สาขาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยมหาสารคาม พ.ศ. 2555 วิทยาศาสตรมหาบัณฑิต (วท.ม.) สาขาเทคโนโลยีสารสนเทศ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้า (ธนบุรี) พ.ศ. 2563 ปรัชญาดุษฎีบัณฑิต (ปร.ด.) สาขาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยมหาสารคาม