เทคนิคใหม่ในการสุ่มตัวอย่างแบบผสม สำหรับปัญหาความไม่สมดุลของประเภทข้อมูล

A NEW HYBRID SAMPLING TECHNIQUE FOR CLASS IMBALANCED PROBLEM

ยศธร สงวนมาก
YOTSATHON SANGUANMAK

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์
ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2560
KMITL-2017-SC-M-002-005

A NEW HYBRID SAMPLING TECHNIQUE FOR CLASS IMBALANCED PROBLEM

YOTSATHON SANGUANMAK

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE
DEGREE OF MASTER OF SCIENCE IN COMPUTER SCIENCE
DEPARTMENT OF COMPUTER SCIENCE
FACULTY OF SCIENCE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
2017

KMITL-2017-SC-M-002-005

คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองวิทยานิพนธ์

หัวข้อวิทยานิพนธ์ "เทคนิคใหม่ในการสุ่มตัวอย่างแบบผสมสำหรับปัญหาความไม่สมดุลของประเภทข้อมูล"
(A NEW HYBRID SAMPLING TECHNIQUE FOR CLASS IMBALANCED PROBLEM)

ชื่อนักศึกษา นายยศธร สงวนมาก
รหัสประจำตัว 58605075
ปริญญา วิทยาศาสตรมหาบัณฑิต (สาขาวิทยาการคอมพิวเตอร์)
ภาควิชา วิทยาการคอมพิวเตอร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์ ผศ.ดร.อนันตพร หรรษคุณาดัย

คณะกรรมการสอบวิทยานิพนธ์	ลายมือชื่อ
ผศ.ดร.ศรัณย์ อินทโกสุม ประธานกรรมการ	
ผศ.ดร.สายชล ใจเย็น อาจารย์บัณฑิตประจำ (ในสาขาวิชาที่เกี่ยวข้อง)	
ผศ.ดร.ธนัสนี เพียรตระกูล ผู้ทรงคุณวุฒิจากภายนอกสถาบันฯ	
ผศ.ดร.อนันตพร หรรษคุณาดัย อาจารย์ที่ปรึกษาวิทยานิพนธ์	

วัน/ เดือน/ ปี ที่สอบ 31 มีนาคม พ.ศ. 2560 เวลา 09.00 - 12.00 น.
สถานที่สอบ ณ ห้อง 301 อาคารพระจอมเกล้า

คณะวิทยาศาสตร์รับรองแล้ว

(รองศาสตราจารย์ ดร.ดุษณี ธนะบริพัฒน์)
คณบดีคณะวิทยาศาสตร์
วันที่ 31 **เดือน** มีนาคม **พ.ศ.** 60

หัวข้อวิทยานิพนธ์	เทคนิคใหม่ในการสุ่มตัวอย่างแบบผสม สำหรับปัญหาความไม่สมดุลของประเภทข้อมูล
ชื่อนักศึกษา	ยศธร สงวนมาก
รหัสประจำตัว	58605075
ปริญญา	วิทยาศาสตรมหาบัณฑิต
ภาควิชา	วิทยาการคอมพิวเตอร์
พ.ศ.	2560
อาจารย์ที่ปรึกษาวิทยานิพนธ์	ผศ.ดร. อนันตพร หรรษคุณาฒัย

บทคัดย่อ

ความไม่สมดุลของประเภทข้อมูลเป็นปัญหาหนึ่งที่สำคัญในกระบวนการเรียนรู้ของเครื่อง ซึ่งปัญหาดังกล่าวส่งผลกระทบต่อประสิทธิภาพการทำนายของโมเดล หนึ่งในวิธีการแก้ปัญหาความไม่สมดุลของประเภทข้อมูลที่ได้รับความนิยม คือ เทคนิคการสุ่มตัวอย่างซึ่งเป็นการแก้ปัญหาในระดับข้อมูล งานวิจัยนี้จึงทำการพัฒนาเทคนิคการสุ่มตัวอย่างแบบใหม่ขึ้นมาที่มีชื่อว่า “DBSM” ซึ่งเป็นเทคนิคผสมระหว่างการเพิ่มจำนวนข้อมูลร่วมกับการลดจำนวนข้อมูล นอกจากนี้ได้นำขั้นตอนวิธีเชิงพันธุกรรมมาประยุกต์ใช้กับอัลกอริทึม DBSM ในการหาคำตอบที่เหมาะสมในการแก้ปัญหาความไม่สมดุลของข้อมูล (GADBSM) จากผลการทดลอง เมื่อเปรียบเทียบเทคนิค DBSM กับเทคนิคการสุ่มตัวอย่างทั้ง 3 เทคนิค ได้แก่ SMOTE Tomek Links และ SMOTE+Tomek Links ในอัลกอริทึมการเรียนรู้ต้นไม้ตัดสินใจ ตัวจำแนกแบบเบย์อย่างง่าย และเพื่อนบ้านใกล้เคียงที่สุด k ตัว พบว่าเทคนิค GADBSM ให้ค่าเฉลี่ยของ F-measure และ AUC สูงที่สุดเมื่อเทียบกับเทคนิคการสุ่มตัวอย่างแบบอื่นในอัลกอริทึมการเรียนรู้ตัวจำแนกแบบเบย์อย่างง่ายและต้นไม้ตัดสินใจ ตามลำดับ นอกจากนี้เทคนิค GADBSM ยังสามารถเพิ่มประสิทธิภาพในการจำแนกประเภทข้อมูลในอัลกอริทึมการเรียนรู้ตัวจำแนกแบบเบย์อย่างง่าย ได้ถึง 7.18%

คำสำคัญ : ความไม่สมดุลของประเภทข้อมูล เทคนิคการสุ่มตัวอย่างแบบผสม เทคนิค SMOTE

Thesis Title	A New Hybrid Sampling Technique for Class Imbalanced Problem
Student Name	Yotsathon Sanguanmak
Student ID	58605075
Degree	Master of Science
Department	Computer Science
Year	2017
Thesis Advisor	Asst.Prof.Dr. Anantaporn Hanskunatai

Abstract

The class imbalance is a major problem in machine learning. This problem affects the performance of a model prediction. A popular technique to handle the class imbalance problem is a sampling technique that is solving in data level. Thus, this research proposes a new hybrid-sampling algorithm, called DBSM. This technique combines over-sampling and under-sampling techniques to deal with the class imbalance for two-classes classification problem. In addition, genetic algorithm is applied for parameters tuning in the DBSM algorithm and called GADBSM. The experimental results of DBSM are compared with three sampling techniques which are SMOTE, Tomek Links, and SMOTE+Tomek Links based on three learning algorithms which are decision tree, naivebayes, and k-nearest neighbors. The results show that the GADBSM algorithm yields the best in averages of F-measure and AUC when compared with other sampling techniques on naivebayes and decision tree learning algorithms. Moreover, GADBSM can improve the classification performance on naivebayes algorithm upto 7.18%.

Keywords : imbalanced dataset, hybrid-sampling, SMOTE

กิตติกรรมประกาศ

ในการทำวิทยานิพนธ์เล่มนี้สามารถสำเร็จได้ด้วยดีจากการช่วยเหลือและสนับสนุนจากบุคคลหลายท่าน ผู้จัดทำขอขอบพระคุณบุคคลดังต่อไปนี้

ผศ.ดร.อนันตพร หรรษคุณาฒัย อาจารย์ที่ปรึกษาปัญหาพิเศษที่กรุณาให้คำปรึกษาและตรวจสอบความเรียบร้อยของงานมาโดยตลอด

ผศ.ดร. ศรัณย์ อินทโกสุม ผศ.ดร.สายชล ใจเย็น และ ผศ.ดร. ธนัสนี เพียรตระกูล ประธานกรรมการและกรรมการ ที่เสียสละเวลาในการแนะแนวทางพัฒนา ชี้จุดบกพร่องที่ควรแก้ไข

และเพื่อนร่วมภาควิชาทุกคนในสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังที่ให้คำปรึกษาตลอดมา นอกจากนี้อาจมีบุคคลท่านอื่นที่ไม่ได้กล่าวไว้ ณ ที่นี้ จึงใคร่ขอขอบพระคุณทุกท่านที่ให้ความกรุณามีส่วนร่วมในการให้ความช่วยเหลือในการทำปัญหาพิเศษเล่มนี้

นายยศธร สงวนมาก

สารบัญ

หน้า

สารบัญตาราง

ตารางที่ **หน้า**

สารบัญรูป

รูปที่ **หน้า**

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญ

ในกระบวนการทางการทำเหมืองข้อมูล (data mining) หนึ่งในปัญหาที่พบคือปัญหาความไม่สมดุลของข้อมูล ซึ่งเกิดจากชุดข้อมูลที่มีจำนวนของกลุ่มข้อมูลกลุ่มหนึ่งมีจำนวนมากกว่ากลุ่มข้อมูลอีกกลุ่มหนึ่ง ซึ่งปัญหาความไม่สมดุลของข้อมูลส่งผลให้โมเดลที่ได้ไม่มีประสิทธิภาพและส่งผลเสียอย่างมากเมื่อโมเดลถูกนำไปใช้ เช่น โมเดลการตรวจหาผู้ป่วยที่เป็นโรคมะเร็ง พบว่าจำนวนผู้ป่วยที่เป็นโรคมะเร็งมีจำนวนน้อยมากเมื่อเทียบกับคนปกติทั่วไป ดังนั้นโมเดลที่ได้จึงมีค่าความผิดพลาดที่สูง ซึ่งหมายถึงโมเดลอาจทำนายผู้ป่วยที่เป็นโรคมะเร็งกลายเป็นคนปกติทั่วไปซึ่งส่งผลเสียเป็นอย่างมาก เนื่องจากผู้ป่วยที่เป็นโรคมะเร็งถูกทำนายเป็นคนปกติและไม่ได้รับการรักษาอย่างทันท่วงที หรือโมเดลการตรวจหาการรั่วไหลของน้ำมันซึ่งสามารถเกิดได้จากทั้งทางธรรมชาติหรือเกิดจากการขุดเจาะน้ำมัน น้ำมันที่รั่วไหลสู่ธรรมชาติทำให้ออกซิเจนในน้ำลดลงและส่งผลเสียกับสิ่งมีชีวิตภายในบริเวณนั้น ๆ นอกจากนี้ยังพบปัญหาความไม่สมดุลของข้อมูลในข้อมูลการพยากรณ์อากาศ ข้อมูลการทำนายเหตุแผ่นดินไหว ข้อมูลการระบุสแปมเมลล์ และข้อมูลการวินิจฉัยโรค เมื่อข้อมูลเหล่านี้ถูกนำไปใช้ในการสร้างโมเดลส่งผลให้โมเดลมีประสิทธิภาพลดลง ซึ่งปัจจุบันวิธีการแก้ปัญหาความไม่สมดุลของข้อมูล สามารถทำได้ 3 แบบด้วยกัน คือ ระดับขั้นตอนวิธี (algorithm level) เป็นเทคนิคการปรับแต่งอัลกอริทึมการเรียนรู้ให้มีประสิทธิภาพในการทำนายเพิ่มมากขึ้น ระดับข้อมูล (data level) เป็นเทคนิคการจัดการกับชุดข้อมูลให้มีความสมดุลมากที่สุด และ cost-sensitive เป็นเทคนิคการรวมการแก้ปัญหาในระดับขั้นตอนวิธีและระดับข้อมูลนำมาใช้ร่วมกัน

หนึ่งในวิธีการแก้ปัญหาความไม่สมดุลของข้อมูลที่ได้รับความนิยมเป็นจำนวนมาก คือ เทคนิคการสุ่มตัวอย่างซึ่งเป็นการแก้ปัญหาในระดับข้อมูล โดยเทคนิคการสุ่มตัวอย่างแบ่งเป็น 3 เทคนิค คือ เทคนิคการลดจำนวนข้อมูลของกลุ่มข้อมูล (undersampling) เป็นวิธีการลดจำนวนข้อมูลของกลุ่มข้อมูลที่เป็นเสียงส่วนมาก (majority class) ให้น้อยลงจนมีปริมาณพอ ๆ กับกลุ่มข้อมูลที่มีเสียงส่วนน้อย (minority class) เทคนิคการเพิ่มจำนวนข้อมูลของกลุ่มข้อมูล (oversampling) เป็นวิธีการเพิ่มจำนวนข้อมูลของกลุ่มข้อมูลที่เป็นเสียงส่วนน้อย (minority class) ให้เพิ่มขึ้นจนมีปริมาณพอ ๆ กับกลุ่มข้อมูลที่มีเสียงส่วนมาก (majority class) และเทคนิคผสมผสาน (hybridsampling) เป็นการรวมเทคนิคการลดจำนวนข้อมูลของกลุ่มข้อมูล และเทคนิคการเพิ่มจำนวนข้อมูลของกลุ่มข้อมูลนำมาใช้ร่วมกัน งานวิจัยนี้จึงทำการพัฒนาเทคนิคการสุ่มตัวอย่างแบบใหม่ขึ้นมาซึ่งเป็นเทคนิคผสมผสานระหว่างการเพิ่มจำนวนข้อมูลด้วยเทคนิค SMOTE ร่วมกับการลดจำนวนข้อมูลด้วยอัลกอริทึม DBSCAN ซึ่งเป็นเทคนิคการจัดกลุ่ม (clustering) และประยุกต์ขั้นตอนวิธีเชิงพันธุกรรมในการหาคำตอบที่เหมาะสมในการแก้ปัญหาความไม่สมดุลของข้อมูล

1.2 วัตถุประสงค์

พัฒนาขั้นตอนวิธีการสุ่มตัวอย่างแบบผสมผสานแบบใหม่ที่มีประสิทธิภาพ

1.3 ข้อจำกัดและขอบเขต

1. เปรียบเทียบประสิทธิภาพของขั้นตอนวิธีการสุ่มตัวอย่างแบบใหม่ที่นำเสนอกับเทคนิคการสุ่มตัวอย่าง 3 เทคนิค คือ เทคนิคการลดจำนวนข้อมูลของกลุ่มข้อมูล (undersampling) เทคนิคการเพิ่มจำนวนข้อมูลของกลุ่มข้อมูล (oversampling) และเทคนิคผสมผสาน (hybridsampling)

2. ชุดข้อมูลที่ใช้ในการทดลองเลือกใช้ชุดข้อมูลที่มีความไม่สมดุลของข้อมูลแบบ 2 คลาสเท่านั้น โดยใช้ทั้งหมด 12 ชุดข้อมูล จาก KEEL และในแต่ละชุดข้อมูลถูกแบ่งออกเป็นชุดข้อมูลฝึกสอนและชุดข้อมูลทดสอบโดยใช้ *5*-fold Crossvalidation

1.4 ขั้นตอนการดำเนินงาน

1. ศึกษาทฤษฎีการสุ่มตัวอย่าง
2. ศึกษาข้อมูลเกี่ยวกับการใช้งานโปรแกรม MatLab และ KEEL เพื่อใช้ในการทดลอง
3. พัฒนาขั้นตอนวิธีการสุ่มตัวอย่างแบบผสมผสานแบบใหม่
4. รวบรวมและทำความเข้าใจชุดข้อมูลที่ใช้ในการทดลอง
5. ทำการทดลอง
6. ประเมินและสรุปผลจากผลลัพธ์ที่ได้จากการทดลอง
7. จัดทำรูปเล่มปัญหาพิเศษ

1.5 ประโยชน์ที่คาดว่าจะได้รับ

ขั้นตอนวิธีการสุ่มตัวอย่างแบบผสมผสานแบบใหม่สามารถแก้ปัญหาความไม่สมดุลของข้อมูลทำให้โมเดลจำแนกประเภทที่สร้างได้มีประสิทธิภาพเพิ่มมากขึ้น

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

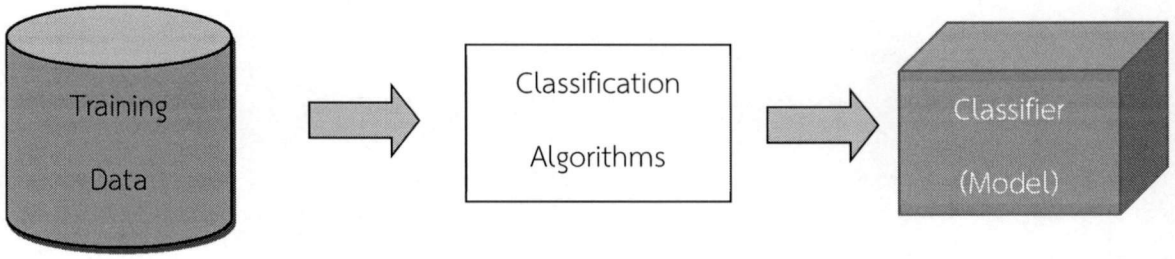
ในบทนี้จะกล่าวถึงทฤษฎีและงานวิจัยที่เกี่ยวข้องที่ใช้ในการแก้ปัญหาความไม่สมดุลของข้อมูล (class imbalance) สำหรับทฤษฎีที่เกี่ยวข้องในงานวิจัยนี้จะกล่าวถึงทฤษฎีการสร้างโมเดลจำแนกประเภทเพื่อใช้ในการจำแนกหรือทำนายข้อมูลที่จะเกิดขึ้น ทฤษฎีต้นไม้ตัดสินใจซึ่งเป็นหนึ่งในวิธีการจำแนกข้อมูลให้อยู่ในรูปต้นไม้ ทฤษฎีตัวจำแนกแบบเบย์อย่างง่าย ทฤษฎีการจำแนกประเภทข้อมูลด้วยอัลกอริทึมเพื่อนบ้านใกล้เคียงที่สุด k ตัว ทฤษฎีขั้นตอนวิธี DBSCAN เป็นขั้นตอนวิธีการจัดกลุ่มข้อมูลโดยใช้ความหนาแน่นของข้อมูล ทฤษฎีขั้นตอนวิธีเชิงพันธุกรรม ทฤษฎีการแก้ปัญหาความไม่สมดุลของข้อมูลโดยใช้เทคนิคการสุ่มตัวอย่าง (resampling) ซึ่งเป็นหนึ่งในเทคนิคการเตรียมข้อมูล (pre-processing) เพื่อเพิ่มประสิทธิภาพการทำนายของโมเดล โดยเทคนิคการสุ่มตัวอย่างที่นำมาใช้ในการทดลองครั้งนี้ คือ เทคนิคการเพิ่มจำนวนตัวอย่างของกลุ่มข้อมูล (oversampling) เทคนิคการลดจำนวนตัวอย่างของกลุ่มข้อมูล (undersampling) และเทคนิคแบบผสมผสานระหว่างการเพิ่มและลดจำนวนตัวอย่างของกลุ่มข้อมูล (hybridsampling) และในส่วนสุดท้ายอธิบายทฤษฎีการวัดประสิทธิภาพของโมเดล ด้วยเทคนิค *k*-fold Crossvalidation

นอกจากนี้ยังได้มีการศึกษางานวิจัยที่เกี่ยวข้องกับการแก้ปัญหาความไม่สมดุลของข้อมูล และการประยุกต์ใช้เทคนิคการสุ่มตัวอย่างเพื่อแก้ปัญหาความไม่สมดุลที่เกิดบนชุดข้อมูลของแอปพลิเคชันในปัจจุบัน

2.1 โมเดลจำแนกประเภท

โมเดลจำแนกประเภท (classification model) เป็นกระบวนในการสร้างโมเดลเพื่อแก้ปัญหาสำหรับการจำแนกหรือทำนายข้อมูลที่จะเกิดในอนาคต โดยข้อมูลที่ใช้ในการสำหรับจำแนกประเภทนั้นจะถูกแบ่งเป็นสองส่วน ส่วนแรกคือข้อมูลที่ใช้สำหรับฝึกสอน (training data) หรือใช้สำหรับการเรียนรู้ในการสร้างโมเดลการจำแนกประเภท ส่วนที่สองคือข้อมูลสำหรับทดสอบ (testing data) เพื่อทดสอบการจำแนกประเภทข้อมูลของโมเดล ในการสร้างโมเดลนั้นอาศัยการเรียนรู้จากข้อมูลที่ใช้สำหรับฝึกสอนและนำโมเดลที่ได้สร้างมานั้นจำแนกข้อมูลที่ใช้สำหรับทดสอบ ตัวอย่างเช่น จำแนกประเภทลูกค้า บัตรเครดิต ว่าเป็นลูกค้าชั้นพิเศษ หรือเป็นลูกค้าชั้นปกติ โดยพิจารณาข้อมูลรายได้ หรือข้อมูลฐานะทางการเงินของลูกค้า เป็นต้น สำหรับการสร้างโมเดลจำแนกประเภทข้อมูลแบ่งออกเป็น 2 ขั้นตอนมีดังนี้ [1]

1. Model Construction เป็นขั้นตอนการสร้างโมเดลจำแนกประเภทโดยอาศัยการเรียนรู้จากข้อมูลที่กำหนดคลาสไว้เรียบร้อยแล้วหรือเรียกว่า ข้อมูลสำหรับฝึกสอน (training data) ซึ่งโมเดลจำแนกประเภทนี้สามารถสร้างได้ด้วยอัลกอริทึมทางด้านการเรียนรู้ด้วยเครื่องจักร (machine learning) ตัวอย่างอัลกอริทึมที่สร้างโมเดลจากการเรียนรู้ด้วยเครื่องจักร เช่น ต้นไม้ตัดสินใจ (decision tree)



รูปที่ 2.1 ขั้นตอนการสร้างโมเดลการจำแนกประเภท

2. Model Usage เป็นขั้นตอนการนำโมเดลในการจำแนกที่สร้างขึ้นมาใช้กับข้อมูลที่ไม่เคยพบมาก่อน (unseen data) เพื่อทำนายและจำแนกประเภทให้กับข้อมูลนั้น และวัดค่าความถูกต้องจากข้อมูลสำหรับทดสอบ (testing data) ซึ่งเป็นข้อมูลคนละส่วนกับข้อมูลสำหรับฝึกสอน ในการวัดค่าความถูกต้องนั้นเมื่อโมเดลในการจำแนกให้ค่าความถูกต้องที่เหมาะสมแล้ว ก็จะนำโมเดลนั้นไปใช้จำแนกประเภทข้อมูลที่ไม่เคยพบมาก่อนได้

รูปที่ 2.2 ขั้นตอนการใช้งานโมเดลจำแนกประเภท

2.2 ต้นไม้ตัดสินใจ

ต้นไม้ตัดสินใจ (decision tree) เป็นวิธีการจำแนกข้อมูลในรูปของต้นไม้ที่แสดงความสัมพันธ์ระหว่างเงื่อนไขในแต่ละระดับ (level) ซึ่งต้นไม้ประกอบด้วยส่วนต่าง ๆ ดังนี้ คุณลักษณะ (attribute) คือส่วนที่เป็นโหนด (node) ค่าคุณลักษณะ (attribute value) คือส่วนที่เป็นกิ่ง (branch) และกลุ่มของข้อมูล (class level) คือส่วนที่เป็นใบ (leaves) และชุดข้อมูลในตารางที่ 2.1 เรียกว่าข้อมูลฝึกสอน (training set)

ตารางที่ 2.1 ข้อมูลของบัตรเครดิต

Customer	Savings	Assets	Income($1000s)	Credit Risk
1	Medium	High	75	Good
2	Low	Low	50	Bad
3	High	Medium	25	Bad
4	Medium	Medium	50	Good
5	Low	Medium	100	Good
6	High	High	25	Good
7	Low	Low	25	Bad
8	Medium	Medium	75	Good

ขั้นตอนวิธี Classification and Regression Trees

Classification and Regression Trees (CART) [2] คือ ขั้นตอนวิธีการจำแนกประเภทอย่างหนึ่งที่ใช้ในการสร้างโมเดลต้นไม้ตัดสินใจแบบสองกิ่งต่อหนึ่งโหนด (binary tree) ซึ่งสามารถแก้ปัญหาได้ทั้งแบบการจำแนกประเภท (classification) และการประมาณค่า (estimation) ในงานวิจัยนี้ได้เลือกใช้ขั้นตอนวิธี CART กับการแก้ปัญหาประเภทการจำแนกประเภท ซึ่งสามารถทำการคัดเลือกคุณลักษณะได้จากการจากสมการที่ 2.1

$$\Phi(s|t)=2P_L P_R \sum_{j=1}^{n} | P(j|t_L) - P(j|t_R) | \qquad (2.1)$$

โดยที่ $\Phi(s|t)$ คือ มาตรวัดที่ใช้ในการเลือกตัวแบ่ง (split) ที่ดีที่สุดที่โหนด t

n คือ จำนวนของคลาสทั้งหมด

t_L คือ โหนดลูกทางซ้ายของโหนด t

t_R คือ โหนดลูกทางขวาของโหนด t

P_L คือ $\frac{\text{จำนวนของตัวอย่างที่โหนด } t_L}{\text{จำนวนของตัวอย่างในชุดข้อมูลฝึกสอน}}$

P_R คือ $\frac{\text{จำนวนของตัวอย่างที่โหนด } t_R}{\text{จำนวนของตัวอย่างในชุดข้อมูลฝึกสอน}}$

$P(j|t_L)$ คือ $\frac{\text{จำนวนของตัวอย่างของคลาส j ที่โหนด } t_L}{\text{จำนวนของตัวอย่างที่โหนด t}}$

$P(j|t_R)$ คือ $\frac{\text{จำนวนของตัวอย่างของคลาส j ที่โหนด } t_R}{\text{จำนวนของตัวอย่างที่โหนด t}}$

จากตารางที่ 2.2 แสดงการคัดเลือกคุณลักษณะที่จะเป็นตัวแทนของโหนดราก (root node) โดยการแบ่งเป็นโหนดลูกทางซ้ายและโหนดลูกทางขวาของโหนด t ซึ่ง t คือโหนดราก

ตารางที่ 2.2 แบ่งโหนดทางซ้ายและทางขวาของโหนด t

Split	Left Child Node, t_L	Right Child Node, t_R
1	Savings = Low	Savings $\in$ {Medium,High}
2	Savings = Medium	Savings $\in$ {Low,High}
3	Savings = High	Savings $\in$ {Low,Medium}
4	Assets = Low	Assets $\in$ {Medium,High}
5	Assets = Medium	Assets $\in$ {Low,High}
6	Assets = High	Assets $\in$ {Low,Medium}
7	Income <= $25,000	Income > $25,000
8	Income <= $50,000	Income > $50,000
9	Income <= $75,000	Income > $75,000

จากสมการที่ 2.1 สามารถคำนวณ ค่า $\Phi(s|t)$ ของ split ที่ 1 ในรอบแรกได้ดังนี้

$$P_L = \frac{\text{จำนวนของตัวอย่างที่ } t_L \text{ (Savings = Low)}}{\text{จำนวนของตัวอย่างในชุดข้อมูลฝึกสอน}} = \frac{3}{8} = 0.375$$

$$P_R = \frac{\text{จำนวนของตัวอย่างที่ } t_R \text{ (Savings } \in \text{ \{Medium,High\})}}{\text{จำนวนของตัวอย่างในชุดข้อมูลฝึกสอน}} = \frac{5}{8} = 0.625$$

$$P(j|t_L) = \frac{\text{จำนวนของตัวอย่างของคลาส j ที่ } t_L \text{ (Savings = Low,Credit Risk = Good)}}{\text{จำนวนของตัวอย่างที่ t (Savings = Low)}} = \frac{1}{3} = 0.333$$

เมื่อ j คือคลาส Good

$$P(j|t_L) = \frac{\text{จำนวนของตัวอย่างของคลาส j ที่ } t_L \text{ (Savings = (Low,Credit Risk = Bad)}}{\text{จำนวนของตัวอย่างที่ t (Savings = Low)}} = \frac{2}{3} = 0.667$$

เมื่อ j คือคลาส Bad

$$P(j|t_R) \text{ คือ } \frac{\text{จำนวนของตัวอย่างของคลาส j ที่ } t_R \text{ (Savings } \in \{\text{Medium,High}\}\text{,Credit Risk = Good)}}{\text{จำนวนของตัวอย่างที่ t (Savings } \in \{\text{Medium,High}\})} = \frac{4}{5} = 0.8$$

เมื่อ j คือคลาส Good

$P(j|t_R)$ คือ $\frac{\text{จำนวนของตัวอย่างของคลาส j ที่ } t_R(\text{Savings} \in \{\text{Medium,High}\},\text{Credit Risk} = \text{Bad})}{\text{จำนวนของตัวอย่างที่ t (Savings} \in \{\text{Medium,High}\})} = \frac{1}{5} = 0.2$

เมื่อ j คือคลาส Bad

$2P_LP_R = 2*0.375*0.625 = 0.46875$

$\Phi(s|t) = \sum_{j=1}^{n} | P(j|t_L) - P(j|t_R) | = |0.333 - 0.8| + |0.667-0.2| = 0.934$

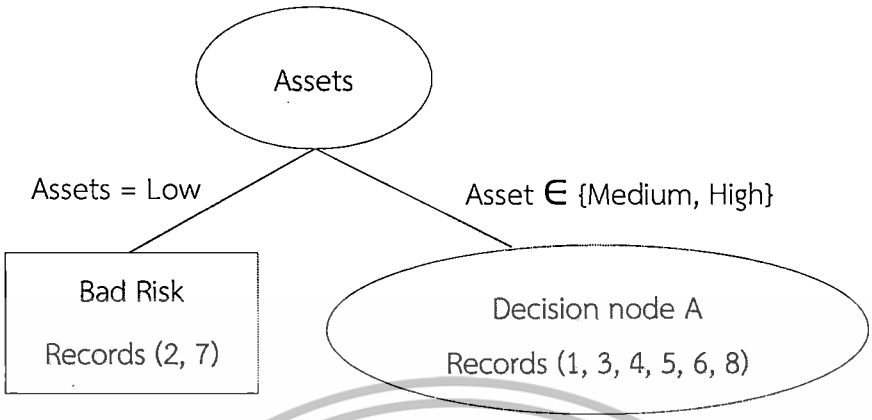
$\Phi(s|t) = 0.46875 * 0.934 = 0.4378$

ตารางที่ 2.3 ผลลัพธ์จากการคำนวณค่า $\Phi(s|t)$

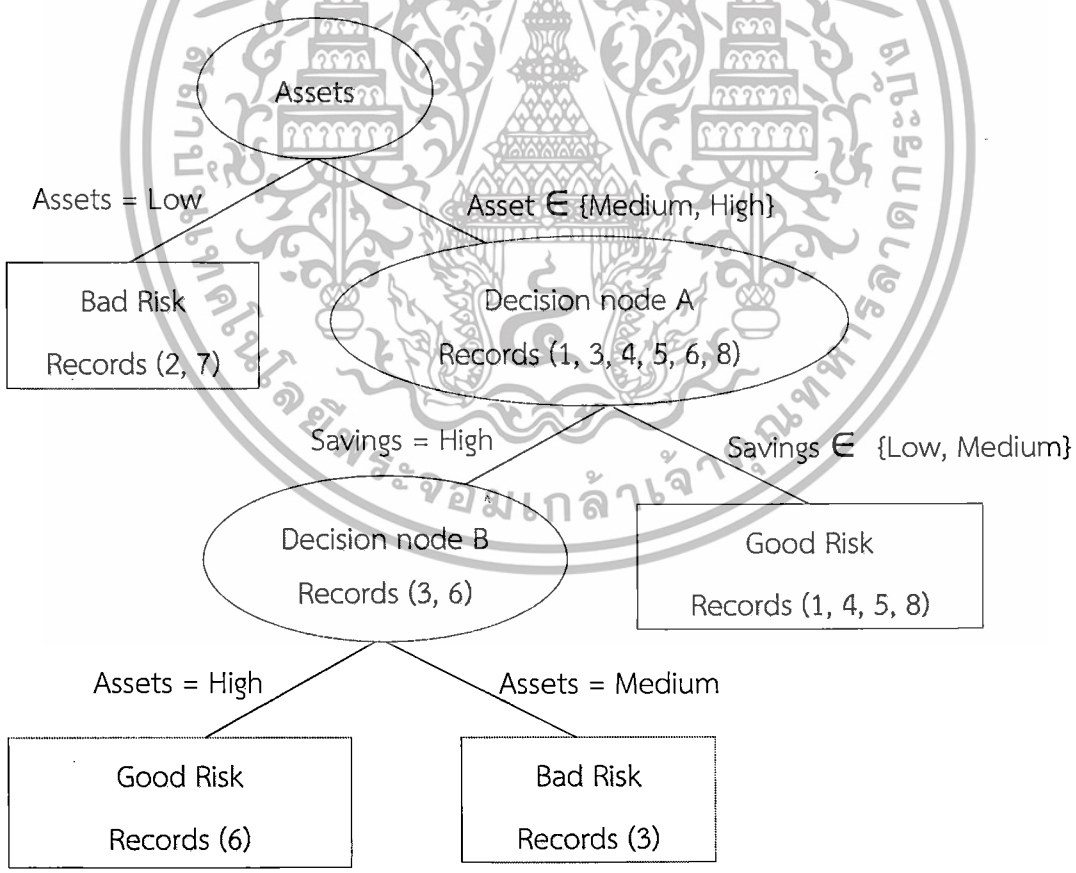
split	P_L	P_R	$P(j\|t_L)$	$P(j\|t_R)$	$2P_LP_R$	$\Phi(s\|t)$	$\Phi(s\|t)$
1	0.375	0.625	G : 0.333	G : 0.8	0.46875	0.934	0.4378
			B : 0.667	B : 0.2			
2	0.375	0.625	G : 1	G : 0.4	0.46875	1.2	0.5625
			B : 0	B : 0.6			
3	0.25	0.75	G : 0.5	G : 0.667	0.375	0.334	0.1253
			B : 0.5	B : 0.333			
4	0.25	0.75	G : 0	G : 0.833	0.375	1.667	**0.6248**
			B : 1	B : 0.167			
5	0.5	0.5	G : 0.75	G : 0.5	0.5	0.5	0.25
			B : 0.25	B : 0.5			
6	0.25	0.75	G : 1	G : 0.5	0.375	1	0.375
			B : 0	B : 0.5			
7	0.375	0.625	G : 0.333	G : 0.8	0.46875	0.934	0.4378
			B : 0.667	B : 0.2			
8	0.625	0.375	G : 0.4	G : 1	0.46875	1.2	0.5625
			B : 0.6	B : 0			
9	0.875	0.125	G : 0.571	G : 1	0.21875	0.858	0.1877
			B : 0.429	B : 0			

จากตารางที่ 2.3 แสดงผลลัพธ์การหาค่า $\Phi(s|t)$ ในแต่ละตัวแบ่งพบว่าตัวแบ่งที่ 4 มีโหนดทางซ้ายคือ Assets = Low และโหนดทางขวาคือ Assets = {Medium,High} ของ t ซึ่งให้ค่า $\Phi(s|t)$มากที่สุด ดังนั้นคุณลักษณะของ Assets จึงถูกเลือกเป็นตัวแทนของโหนดราก และมีค่าคุณลักษณะทางซ้ายเป็น Low และมีคลาสเป็น Bad Risk แสดงดังรูปที่ 2.3 ส่วนโหนดทางขวา

(node A) จะถูกทำซ้ำโดยการแบ่งโหนดทางซ้ายและโหนดทางขวาเพื่อหากลุ่มข้อมูล (class level) โดยพิจารณาเฉพาะตัวอย่างที่ 1, 3, 4, 5, 6 และ 8 ในรอบถัดไปดังรูปที่ 2.4



รูปที่ 2.3 ต้นไม้ตัดสินใจในรอบแรก



รูปที่ 2.4 ต้นไม้ตัดสินใจแบบโตเต็มที่

2.3 ตัวจำแนกแบบเบย์อย่างง่าย

ตัวจำแนกแบบเบย์อย่างง่าย (Naïve Bayes Classifier)[3] จัดเป็นการเรียนรู้แบบมีผู้สอน (supervised learning) ซึ่งกำหนดว่าจะต้องมีการจัดเตรียมประเภทของข้อมูลที่ต้องการให้เกิดการเรียนรู้เพื่อสร้างคอนเซปต์ของข้อมูลประเภทนั้น การประยุกต์ใช้ทฤษฎีตัวจำแนกแบบเบย์อย่างง่ายกับการจำแนกข้อมูลสามารถทำได้โดยการแทนค่าในสูตรซึ่งสามารถปรับให้สอดคล้องกับงานการจำแนกข้อมูลดังสมการที่ 2.2

$$P(\text{class}|\text{attribute}) = \frac{P(\text{class})P(\text{attribute}|\text{class})}{P(\text{attribute})} \quad (2.2)$$

จากสูตรข้างต้นสามารถอธิบายได้ว่าในการจำแนกประเภทของข้อมูลจะใช้การคำนวณค่าความน่าจะเป็นที่ข้อมูลนั้นอยู่ในประเภทของข้อมูลหนึ่งๆ เมื่อทราบค่าของคุณลักษณะของข้อมูลนั้น

อย่างไรก็ตามข้อมูลแต่ละตัวอย่างมีคุณลักษณะมากกว่าหนึ่งตัว ดังนั้น สูตรข้างต้นจึงเขียนใหม่ได้ดังสมการที่ 2.3

$$P(\text{class}|a_1,a_2,\ldots,a_n) = \frac{P(\text{class})P(a_1,a_2,\ldots,a_n|\text{class})}{P(a_1,a_2,\ldots,a_n)} \quad (2.3)$$

โดยที่ a_i คือ ค่าของคุณลักษณะใด ๆ ที่ปรากฏในตัวอย่างที่ต้องการจำแนก

สำหรับตัวแปร $P(\text{class}|a_1,a_2,\ldots,a_n)$ คือค่าความน่าจะเป็นของการเกิดคุณลักษณะที่มีค่า a_1 ร่วมกันกับ การเกิดคุณลักษณะที่มีค่า a_2 จนถึง การเกิดคุณลักษณะที่มีค่า a_n ซึ่งใช้วิธีการคำนวณหาค่าความน่าจะเป็นของการเกิดเหตุการณ์ต่าง ๆ ร่วมกัน ซึ่งสามารถคำนวณได้ดังสมการที่ 2.4 และ (2.5)

$$P(a_1,a_2,\ldots,a_n) = P(a_1)P(a_2|a_1)P(a_3|a_2a_1)\ldots P(a_n|a_1,\ldots,a_{n-1}) \quad (2.4)$$

$$P(a_1,a_2,\ldots,a_n|\text{Class}) = P(a_1|\text{Class})P(a_2|a_1,\text{Class})P(a_3|a_2,a_1,\text{Class})\ldots P(a_n|a_1..,a_{n-1},\text{class}) \quad (2.5)$$

สำหรับการคำนวณค่าความน่าเป็น $P(a_1,a_2,\ldots,a_n|\text{class})$ ซึ่งจะทำได้ยากและเป็นไปไม่ได้ที่จะได้ค่าที่ถูกต้อง เนื่องจากจะต้องเก็บข้อมูลฝึกที่มีจำนวนมาก เช่น สมมติว่าคุณลักษณะ a_i ใด ๆ มีค่าที่เป็นไปได้ 2 ค่าและกำหนดให้ข้อมูลฝึกนั้นมีจำนวนคุณลักษณะ 10 จำนวน ดังนั้นจึงต้องมีข้อมูลฝึกอย่างน้อยที่สุด 2^{10} ตัวอย่าง จึงจะทำให้มีโอกาสพบรูปแบบของการเกิดค่าคุณลักษณะร่วมกันประมาณหนึ่งครั้ง ซึ่งจำนวนข้อมูลฝึก 2^{10} ตัวอย่างนี้ยังคงขาดความน่าเชื่อถือทางสถิติเพราะจะต้องมีข้อมูลฝึกที่มีจำนวนมากกว่านี้หลายเท่าตัว อุปสรรคของการหาค่าความน่าจะเป็นข้างต้นทำให้ต้องมี

การสร้างสมมติฐานที่ทำให้เกิดการคำนวณมีความเป็นไปได้ในกรณีการเกิดคุณลักษณะ a_i ใด ๆ ไม่ขึ้นต่อกัน สามารถเปลี่ยนวิธีการคำนวณค่า$P(a_1,a_2,...,a_n|class)$ ได้ใหม่ ดังสมการที่ 2.6 และ 2.7

$$P(a_1,a_2,...,a_n|class) = \prod_{i=1}^{n} P(a_i|class) \quad (2.6)$$

$$P(a_1,a_2,...,a_n|class) = P(a_1|class)P(a_2|class)...P(a_n|class) \quad (2.7)$$

ดังนั้นการจำแนกข้อมูลแบบเบย์อย่างง่ายจึงสามารถทำได้โดยการหาค่าความน่าจะเป็น $P(class|a_1,a_2,...,a_n)$ โดยขึ้นอยู่กับประเภทของข้อมูลที่มีการเรียนรู้ในชุดข้อมูลนั้น ตัวอย่างเช่น บริษัทประกันภัยแห่งหนึ่ง ต้องการกำหนดค่าดอกเบี้ยประกันภัยให้เหมาะสมกับประเภทลูกค้า จึงได้ทำการเก็บข้อมูลลูกค้าไว้ในฐานข้อมูล บริษัทต้องการจำแนกประเภทของลูกค้าประกันภัยเป็น 2 ประเภท คือลูกค้าชั้นดีที่ไม่เคยเรียกให้บริษัทชดใช้ค่าสินไหมทดแทน ($class_a$) กับลูกค้าปกติที่รถเคยเกิดอุบัติเหตุ ($class_b$) สามารถทำได้โดยการคำนวณค่า $P(class_a|a_1,a_2,...,a_n)$ และ $P(class_b|a_1,a_2,...,a_n)$ แล้วดูว่าค่าใดมีค่ามากกว่ากัน ซึ่งสามารถเขียนได้ดังสมการที่ 2.8

$$Classify(data) = argmax_{c_j \in C} P(c_j)\prod_{i=1}^{n} P(a_i|c_j) \quad (2.8)$$

จากสมการที่ 2.8 อธิบายได้ถึงการหาความน่าจะเป็นของทั้ง 2 คลาส โดยพิจารณาจากค่าความน่าจะเป็นที่สูงที่สุด

2.4 เพื่อนบ้านใกล้เคียงที่สุด k ตัว

เพื่อนบ้านใกล้เคียงที่สุด k ตัว (*k*-Nearest Neighbor)[3] เป็นขั้นตอนวิธีจำแนกข้อมูลโดยเปรียบเทียบข้อมูลที่สนใจกับข้อมูลอื่น ผลลัพธ์ที่ได้จะเป็นค่าเฉลี่ยที่ใกล้ที่สุดจำนวน *k* ตัวในกรณีที่เป็นการทำนายเป็นค่าตัวเลข (regression) ในกรณีที่เป็นการทำการจำแนกกลุ่มข้อมูล (classification) จะใช้เสียงข้างมากในการกำหนดกลุ่มข้อมูล

การหาเพื่อนบ้านใกล้เคียงที่สุด k ตัวมีขั้นตอนวิธีดังต่อไปนี้

1. กำหนดขนาดของ *k* (ควรกำหนดให้เป็นเลขคี่)
2. คำนวณระยะห่าง (distance) ของข้อมูลที่ต้องการทำนายกับกลุ่มข้อมูลตัวอย่าง ซึ่งการคำนวณระยะห่างสามารถแสดงได้ดังสมการที่ 2.9

$$distance = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \quad (2.9)$$

โดยที่ p_i คือ ค่าคุณลักษณะที่ *i* ของข้อมูลตัวอย่าง
q_i คือ ค่าคุณลักษณะที่ *i* ของข้อมูลที่ต้องการทำนาย
n คือ คุณลักษณะทั้งหมดของตัวอย่าง

3. จัดเรียงลำดับของระยะห่างจากน้อยไปมากระหว่างตัวอย่างกับข้อมูลที่ต้องการทำนายตามจำนวน *k* ที่กำหนดไว้
4. พิจารณาข้อมูลจำนวน *k* ตัวอย่าง และสังเกตกลุ่มของข้อมูลใดที่ใกล้ตัวอย่างหรือข้อมูลที่ต้องการทำนายเป็นจำนวนมากที่สุด
5. กำหนดกลุ่มข้อมูล ให้กับตัวอย่างที่ต้องการทำนาย

พิจารณาได้จากตัวอย่างนี้ โดยเป็นตัวอย่างเป็นการจำแนกประเภทของดอกไอริสซึ่งมีอยู่ 3 ประเภท คือ Iris-setosa Iris-versicolor และ Iris-verginica โดยแต่ละประเภทจะมีข้อมูลความกว้าง (width) และความยาว (length) ของกลีบดอก (petal) และกลีบเลี้ยง (sepal) แสดงได้ดังตารางที่ 2.4

ตารางที่ 2.4 ข้อมูลฝึกสอนของดอกไอริส

Example	Sepal Length	Sepal Width	Petal Length	Petal Width	Class
R_1	48	30	14	1	Iris-setosa
R_2	51	35	14	3	Iris-setosa
R_3	50	34	16	4	Iris-setosa
R_4	66	30	44	14	Iris-versicolor
R_5	67	31	47	15	Iris-versicolor
R_6	58	26	40	12	Iris-versicolor
R_7	77	26	69	23	Iris-verginica
R_8	77	30	61	23	Iris-verginica
R_9	67	30	52	23	Iris-verginica

สมมติกำหนดให้ $k = 3$ และกำหนดข้อมูลที่ต้องการทำนายมาเปรียบเทียบกับชุดข้อมูลตัวอย่างที่อยู่ตารางที่ 2.4 ข้อมูลที่ต้องการทำนายแสดงได้ดังตารางที่ 2.5

ตารางที่ 2.5 ข้อมูลทดสอบของดอกไอริส

Example	Sepal Length	Sepal Width	Petal Length	Petal Width	Class
x	56	37	13	4	???

ขั้นตอนวิธีการคำนวณ

คำนวณระยะห่างของข้อมูลที่ต้องการพิจารณา (x) กับกลุ่มข้อมูลตัวอย่างในแต่ละแถว (R) โดยใช้สมการที่ 2.9

ผลลัพธ์ที่ได้จากการคำนวณมีดังต่อไปนี้

$$distance(x,R_1) = \sqrt{(56-48)^2+(37-30)^2+(13-14)^2+(4-1)^2} = 11.09$$

$$distance(x,R_2) = \sqrt{(56-51)^2+(37-35)^2+(13-14)^2+(4-3)^2} = 5.57$$

$$distance(x,R_3) = \sqrt{(56-50)^2+(37-34)^2+(13-16)^2+(4-4)^2} = 7.35$$

$$distance(x,R_4) = \sqrt{(56-66)^2+(37-30)^2+(13-44)^2+(4-14)^2} = 34.79$$

$$distance(x,R_5) = \sqrt{(56-67)^2+(37-31)^2+(13-47)^2+(4-15)^2} = 37.87$$

$$distance(x,R_6) = \sqrt{(56-58)^2+(37-26)^2+(13-40)^2+(4-12)^2} = 30.30$$

$$distance(x,R_7) = \sqrt{(56-77)^2+(37-26)^2+(13-69)^2+(4-23)^2} = 63.71$$

$$distance(x,R_8) = \sqrt{(56-77)^2+(37-30)^2+(13-61)^2+(4-23)^2} = 56.17$$

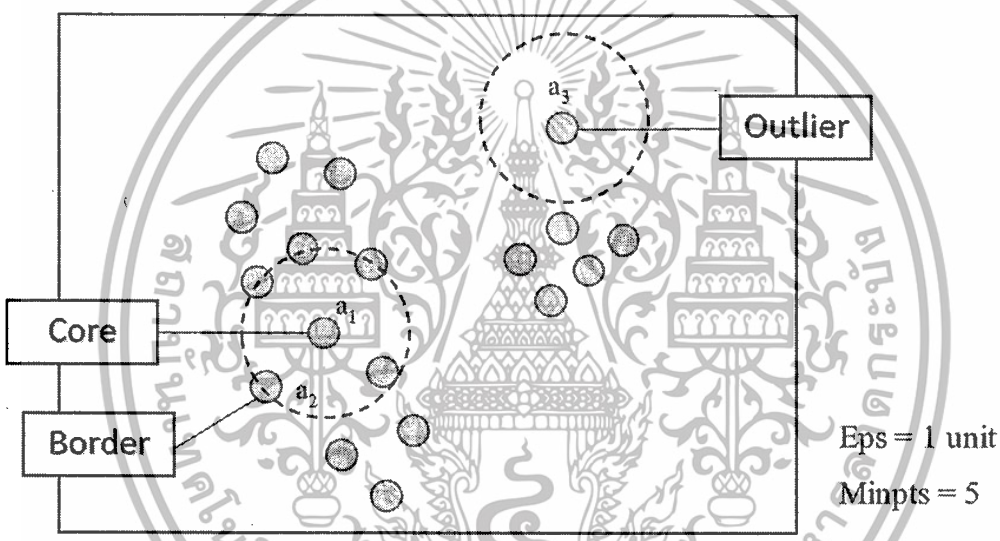
$$distance(x,R_9) = \sqrt{(56-67)^2+(37-30)^2+(13-52)^2+(4-23)^2} = 45.30$$

จากการคำนวณ ผลลัพธ์ที่ได้มีความใกล้เคียงกับกลุ่มข้อมูลตัวอย่าง R_1 R_2 และ R_3 ซึ่งเป็นดอกไอริสประเภท Iris-setosa ดังนั้นจึงสรุปได้ว่า ดอกไอริสชนิดนี้เป็นดอกไอริสประเภท Iris-setosa

2.5 ขั้นตอนวิธี DBSCAN

DBSCAN [4] เป็นขั้นตอนวิธีการจัดกลุ่มข้อมูลโดยใช้ความหนาแน่นของข้อมูลเป็นพื้นฐานที่สามารถตรวจหาค่าความผิดปกติ (outlier) ในข้อมูลได้ โดยต้องการพารามิเตอร์ 2 ตัวที่ผู้ใช้จะต้องกำหนดคือ Eps หมายถึงระยะทางของจุดเพื่อนบ้าน และ Minpts หมายถึงจำนวนขั้นต่ำของจำนวนเพื่อนบ้าน สำหรับจุดข้อมูลใด ๆ ถ้ามีจุดข้อมูลอีกจุดหนึ่งนั้นอยู่ในระยะของ Eps จะเรียกว่าเป็นจุดเพื่อนบ้านของจุดนั้น และถ้าจำนวนจุดเพื่อนบ้านของจุดนั้นมีค่ามากกว่าหรือเท่ากับค่า Minpts กลุ่มของจุดนี้จะถูกจัดขึ้นเป็นข้อมูลกลุ่มเดียวกัน โดย DBSCAN จะกำหนดจุดข้อมูลได้เป็น 3 ประเภท คือ

- จุดหลัก (core point) คือจุดที่มีจำนวนจุดเพื่อนบ้านอย่างน้อยสุดเท่ากับค่า Minpts ในระยะ Eps ที่กำหนด
- จุดขอบ (border point) คือ จุดที่ไม่ได้เป็นจุดหลักแต่เป็นจุดเพื่อนบ้านของจุดหลัก
- จุดผิดปกติ (outlier point) คือ จุดที่ไม่ได้เป็นจุดหลักและจุดขอบ

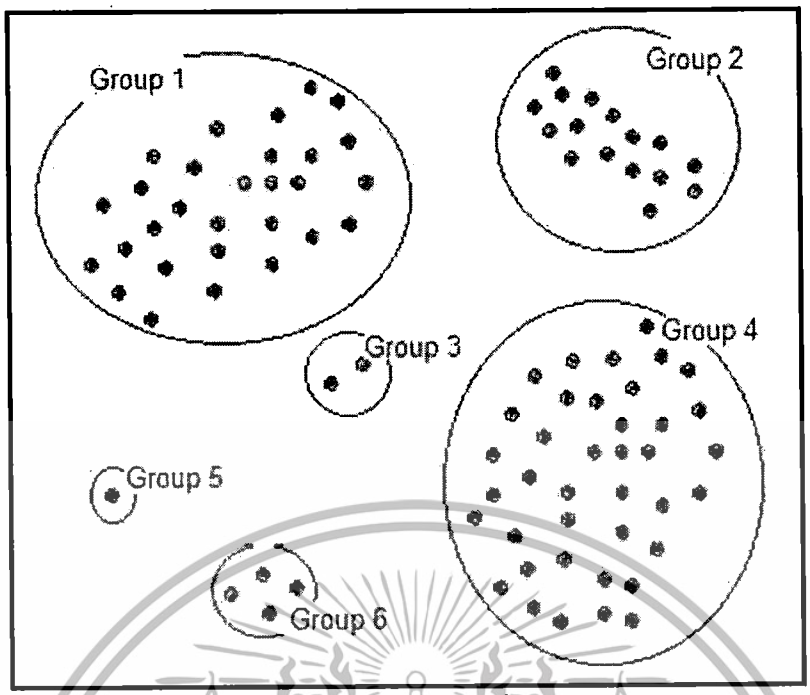


รูปที่ 2.5 การกำหนดจุดข้อมูลของขั้นตอนวิธี DBSCAN

จากรูป 2.5 จะเห็นได้ว่าจุด a1 ถูกกำหนดเป็นจุดหลักเพราะมีจุดเพื่อนบ้านที่อยู่ในระยะทาง Eps = 1 และมีค่า Minpts = 5 จุด a2 ถูกกำหนดเป็นจุดขอบเพราะไม่ได้ถูกกำหนดเป็นจุดหลักแต่เป็นแค่เพื่อนบ้านของจุดหลัก และจุด a3 ถูกกำหนดเป็นจุดผิดปกติเพราะไม่ได้ถูกกำหนดเป็นทั้งจุดหลักและจุดขอบ

ใน DBSCAN วิธีการจัดกลุ่มจะแตกต่างจากวิธีการจัดกลุ่มโดยทั่วไป DBSCAN สามารถค้นพบจุดผิดปกติที่ไม่ถูกจัดเข้ากับกลุ่มข้อมูลใด ๆ ได้ ในรูปที่ 2.6 แสดงให้เห็นถึงผลของการจัดกลุ่มข้อมูลโดยใช้ขั้นตอนวิธี DBSCAN ต้องการจำนวนขั้นต่ำของ Minpts ในกลุ่มเพื่อจะกำหนดเป็นกลุ่มข้อมูล ตัวอย่างเช่นถ้าค่า Minpts ถูกกำหนดเป็น 3 จะมีจำนวนกลุ่มข้อมูลที่ถูกสร้างทั้งหมด 4 กลุ่ม คือ กลุ่มที่ 1 กลุ่มที่ 2 กลุ่มที่ 4 และ กลุ่มที่ 6

ในทางตรงกันข้ามกลุ่มที่ 3 และกลุ่มที่ 5 จะถูกกำหนดเป็น ค่าผิดปกติเนื่องจากกลุ่มเหล่านี้ไม่มีจำนวนจุดเพื่อนบ้านที่เพียงพอในการกำหนดเป็นกลุ่มข้อมูล แต่ถ้ากำหนดให้ค่า Minpts เท่ากับ 5 กลุ่มที่ 3 กลุ่มที่ 5 และกลุ่มที่ 6 จะถูกกำหนดให้เป็นค่าผิดปกติ



รูปที่ 2.6 ตัวอย่างชุดข้อมูลสำหรับวิธีการตรวจจับข้อมูลที่มีความผิดปกติโดยใช้ DBSCAN

ขั้นตอนการทำงานของขั้นตอนวิธีของอัลกอริทึม DBSCAN แสดงดังรูปที่ 2.7

```
Inputs:
D: the dataset
Eps: the neighborhood distance
Minpts: the minimum number of points
Output:
Discovered outliers and clusters
Variables:
m, n: row and column values of D matrix, respectively
Dist: distance vector
class_no: indicates the clusters
Algorithm:
1. import the data-set into D
2. for i = 1 to m //row counter
3.      Dist = distance(i, D)
4.      neighbors= find(Dist =< Eps)
5.      neighbor_count = count(neighbors)
6.      core_neig=check_core_neighbor (neighbors)
7.      if (neighbor_count >=minpts)
8.              class(i) = class_no //clustered point
9.              while(more points near i)
10.             class(point) = class_no
11.             end while
12.             class_no += 1
13.     else if(neighbor_count<minpts & core_neig==True)
14.             class(i) = 0 //border point
15.     else if (neighbor_count<minpts)
16.             class(i) = -1 //outlier point
17.     end if
18. end for
19. return class
```

รูปที่ 2.7 ขั้นตอนวิธีของอัลกอริทึม DBSCAN

รูปที่ 2.7 แสดงขั้นตอนวิธีการทำงานของอัลกอริทึม DBSCAN ซึ่งมีรายละเอียดดังนี้
ตัวแปรเข้า : D คือชุดข้อมูล
Eps คือ ระยะทางของจุดเพื่อนบ้าน
Minpts คือ จำนวนของจุดเพื่อนบ้านขั้นต่ำ
ผลลัพธ์ : ค้นพบค่าผิดปกติและกลุ่มข้อมูล
ตัวแปร : m และ n คือ แถวและหลักของเมทริกซ์ข้อมูล
Dist คือ เวกเตอร์ระยะทาง
Class_no คือ หมายเลขของกลุ่มข้อมูล
ขั้นตอนวิธี :
1. นำเข้าชุดข้อมูลและการกำหนดค่า Eps และ Minpts ของผู้ใช้ในระหว่างขั้นตอนที่ 2 และ 16 จะมีการวนรอบทำซ้ำสำหรับจำนวนจุดในชุดข้อมูล
2. DBSCAN กำหนด D [i] เป็นจุดศูนย์กลาง
3. ทำการคำนวณระยะทางระหว่างจุดศูนย์กลาง D [i] และจุดที่เหลืออยู่
4. จุดใดที่มีระยะทางอยู่น้อยกว่าหรือเท่ากับ Eps จะได้รับการยอมรับเป็นจุดเพื่อนบ้านของจุดศูนย์กลาง
5. ทำการคำนวณจำนวนจุดเพื่อนบ้านของจุดศูนย์กลาง
6. ตรวจสอบหาจุดหลักใด ๆ ในรายการจุดเพื่อนบ้าน
7. ตรวจสอบเงื่อนไขแรก ถ้าจำนวนของจุดเพื่อนบ้านของจุด i มีค่ามากกว่าหรือเท่ากับ Minpts หรือไม่
8. ถ้าตรงตามเงื่อนไขจุด i จะถูกกำหนดหมายเลขของกลุ่มข้อมูล โดยที่หมายเลขของกลุ่มข้อมูลจะเริ่มต้นที่ 1
9. เป็นการตรวจสอบว่าถ้ายังมีจุดอื่นอยู่ใกล้เคียงกับ i อีกหรือไม่
10. ถ้าตรงตามเงื่อนไขจุดใกล้เคียงนั้นจะถูกกำหนดหมายเลขของกลุ่มข้อมูลเป็นหมายเลขเดียวกับ i
12. ทำการเพิ่มค่าหมายเลขของกลุ่มข้อมูล
13. ถ้าไม่ตรงกับเงื่อนไขแรกจะตรวจสอบเงื่อนไขที่ 2 ว่าจำนวนเพื่อนบ้านของจุด i น้อยกว่า Minpts และจุดนั้นเป็นเพื่อนบ้านของจุดหลักหรือไม่
14. ถ้าตรงตามเงื่อนไขที่ 2 หมายเลขของกลุ่มข้อมูลของจุด i จะเป็น 0 หมายถึง จุดนั้นเป็นจุดขอบ
15. ถ้าไม่ตรงกับสองเงื่อนไขที่ผ่านมาจะมาตรวจสอบกับเงื่อนไขที่ 3 ว่าจานวนเพื่อนบ้านของจุดนั้นน้อยกว่า Minpts หรือไม่
16. ถ้าตรงตามเงื่อนไขที่ 3 หมายเลขของกลุ่มข้อมูลของจุด i จะเป็น -1 หมายถึง จุดนั้นเป็นจุดผิดปกติ
19. คืนค่าของหมายเลขกลุ่มข้อมูลที่สร้างได้

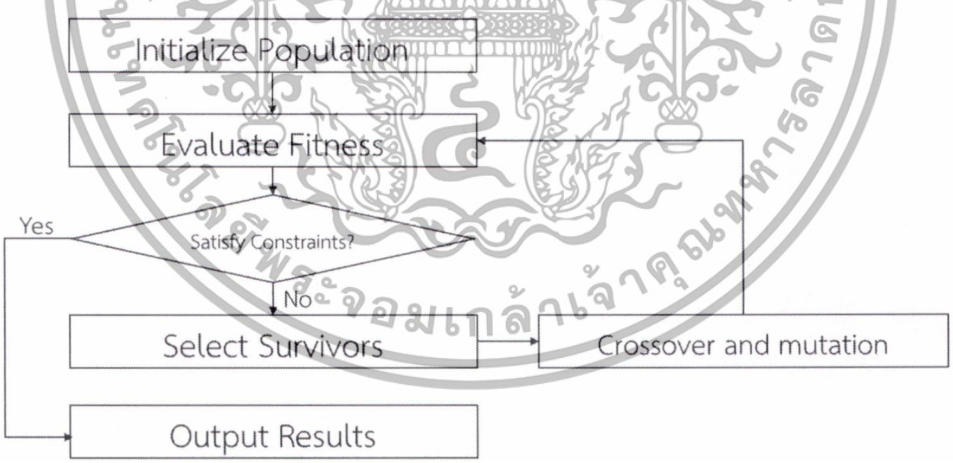
2.6 ขั้นตอนวิธีเชิงพันธุกรรม

ขั้นตอนวิธีเชิงพันธุกรรม (Genetic Algorithm : GA)[5] คิดค้นโดย John Holland ในช่วงปี 1970 โดยมีวัตถุประสงค์เพื่อให้คอมพิวเตอร์สามารถที่จะหาคำตอบหรือแก้ปัญหาใด ๆ ด้วยหลักการเดียวกับการคัดเลือกสิ่งมีชีวิตโดยธรรมชาติ โดยเริ่มต้นนั้นได้กำหนดสถานะหรือคำตอบให้อยู่ในรูปของบิตสตริง โดยที่แต่ละบิดสตริงจะเรียกว่า “โครโมโซม” ซึ่งประกอบด้วยยีน (แทนด้วยแต่ละบิตมีค่าเป็น 0 หรือ 1) แสดงดังรูปที่ 2.8

1	0	0	1	1	1	0	1	1	0

รูปที่ 2.8 โครโมโซมที่แทนด้วยบิตสตริง

หลักการทำงานของขั้นตอนวิธีเชิงพันธุกรรมประกอบไปด้วย 2 ขั้นตอนคือ การเข้ารหัส ซึ่งเป็นการแทนสถานะของคำตอบให้อยู่ในรูปบิดสตริงหรือโครโมโซม ส่วนขั้นตอนที่สองคือการประเมินค่าโครโมโซมแต่ละตัวว่าเข้าใกล้สู่คำตอบมากน้อยเพียงใดโดยการกำหนดฟังก์ชันฮิวริสติกตัวหนึ่งที่ชื่อว่า “fitness function” หรือฟังก์ชันความเหมาะโดยโครโมโซมที่มีค่าฟังก์ชันความเหมาะสูงก็จะมีโอกาสที่จะผลิตลูกได้มากกว่าโครโมโซมที่มีค่าฟังก์ชันความเหมาะต่ำ



รูปที่ 2.9 การทำงานของอัลกอริทึมเชิงพันธุกรรม

การทำงานของอัลกอริทึมเชิงพันธุกรรมมีขั้นตอนดังนี้

ขั้นตอนที่ 1: แทนคำตอบของปัญหาด้วยโครโมโซมที่มีจำนวนยีนคงที่ กำหนดจำนวนประชากรทั้งหมด N, ความน่าจะเป็นในการเกิดการไขว้เปลี่ยนพันธุกรรม (crossover rate) pc,และความน่าจะเป็นในการกลายพันธุ์ (mutation rate) p_m

ขั้นตอนที่ 2: กำหนดฟังก์ชันความเหมาะ (fitness function)

ขั้นตอนที่ 3: สร้างประชากรในรุ่นแรกอย่างสุ่มให้มีจำนวนเท่ากับ N ซึ่งจะได้ $x_1, x_2, ..., x_N$

ขั้นตอนที่ 4: คำนวณค่าฟังก์ชันความเหมาะของแต่ละโครโมโซม $f(x_1), f(x_2), ..., f(x_N)$

ขั้นตอนที่ 5: เลือกคู่ของโครโมโซมที่จะมาผสมพันธุ์กันเพื่อผลิตลูก โดยโครโมโซมพ่อแม่จะต้องถูกสุ่มขึ้นมาด้วยความน่าจะเป็นที่สอดคล้องกับค่าฟังก์ชันความเหมาะของแต่ละโครโมโซม

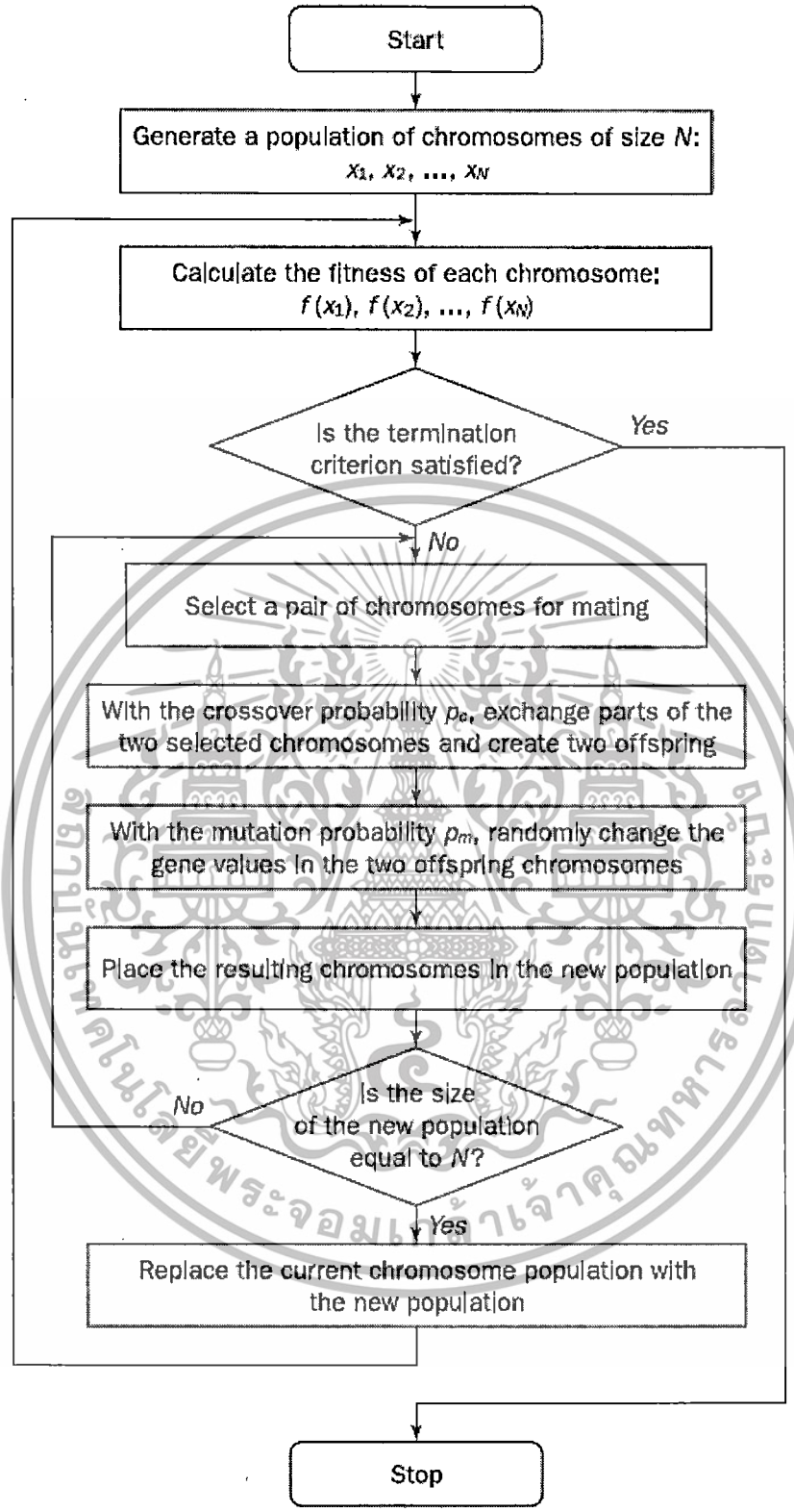
ขั้นตอนที่ 6: สร้างโครโมโซมของลูกจากโครโมโซมพ่อและแม่โดยการใช้ตัวดำเนินการเชิงพันธุกรรม ซึ่งมี 2 แบบคือการไขว้เปลี่ยน (crossover) และการกลายพันธุ์ (mutation)

ขั้นตอนที่ 7: นำโครโมโซมลูกที่ผลิตได้ไปใส่ในเซตประชากรรุ่นใหม่

ขั้นตอนที่ 8: ไปทำซ้ำในขั้นตอนที่ 5 เพื่อผลิตโครโมโซมลูกตัวใหม่จนกระทั่งได้จำนวนประชากรในรุ่นใหม่เท่ากับ N

ขั้นตอนที่ 9: แทนประชากรรุ่นเก่าด้วยประชากรรุ่นใหม่ซึ่งเป็นโครโมโซมลูกที่ผลิตได้ทั้งหมด

ขั้นตอนที่ 10: กลับไปทำในขั้นตอนที่ 4 ทำซ้ำจนกระทั่งเงื่อนไขในการวนซ้ำเป็นจริง

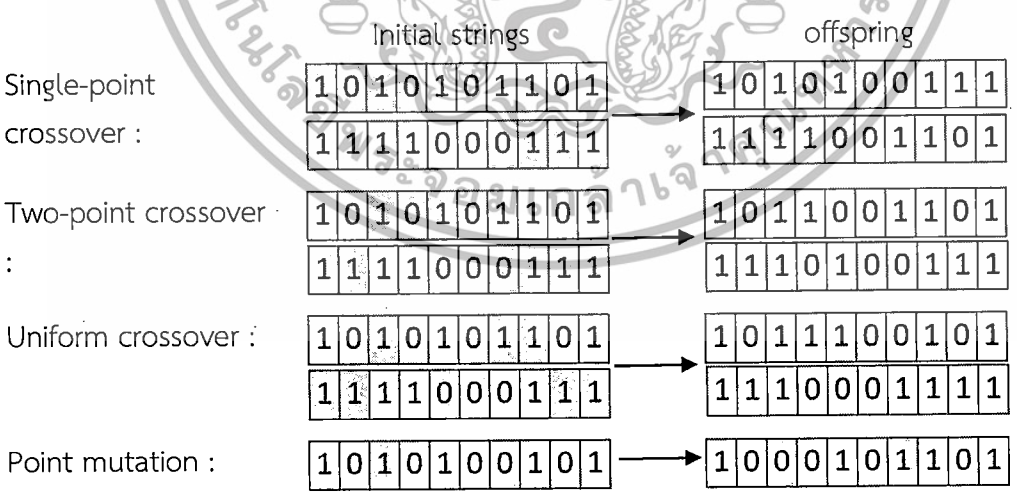


รูปที่ 2.10 การทำงานของขั้นตอนวิธีเชิงพันธุกรรมโดยละเอียด

การทำงานของขั้นตอนวิธีเชิงพันธุกรรม (GA) มีการทำงานเป็นแบบวนซ้ำเป็นรอบรอบแต่ละรอบการทำงานเรียกว่า generation โดยปกติแล้วจำนวนรอบของการทำงานสำหรับ GA อยู่ที่ประมาณ 50 ถึง 100 หรือมากกว่า 500 รอบซึ่งขึ้นอยู่กับความซับซ้อนของปัญหาส่วน generation ทั้งหมดเรียกว่า run ในทางปฏิบัติกระบวนการ GA จะสิ้นสุด (terminate) เมื่อได้มีการทำงานครบรอบตามจำนวน generation ที่ระบุไว้และเลือกโครโมโซมที่ให้ค่าฟังก์ชันความเหมาะที่ดีที่สุดในบรรดาประชากรทั้งหมดมาเป็นคำตอบ ถ้ายังไม่มีคำตอบที่น่าพอใจกระบวนการทำงานของ GA จะเริ่มต้นทำงานใหม่อีกครั้ง

ตัวดำเนินการทางพันธุกรรม (Genetic Operators)

ตัวดำเนินการทางพันธุกรรมมีหน้าที่ผลิตโครโมโซมลูกที่ได้จากโครโมโซมพ่อหรือแม่ ซึ่งจะเกิดขึ้นหลังจากที่มีการคัดเลือกโครโมโซมที่เหมาะสมในการผสมพันธุ์แล้ว ประกอบด้วย 2 ตัวดำเนินการคือ การไขว้เปลี่ยนพันธุกรรม (crossover) และการกลายพันธุ์ (mutation) สำหรับการไขว้เปลี่ยนพันธุกรรม โครโมโซมพ่อและโครโมโซมแม่จะผลิตโครโมโซมลูกจำนวน 2 โครโมโซม (offspring) โดยทำการคัดลอกบางส่วนของบิตสตริงจากส่วนของโครโมโซมพ่อและแม่ ซึ่งโดยทั่วไปความน่าจะเป็นในการเกิดการไขว้เปลี่ยนพันธุกรรม อยู่ในช่วงระหว่าง 0.6 ถึง 1.0 และชนิดของการไขว้เปลี่ยนพันธุกรรมมี 3 แบบ คือ Single-point crossover Two-point crossover และ Uniform crossover สำหรับการกลายพันธุ์ การสร้างโครโมโซมลูกหนึ่งโครโมโซมจากโครโมโซมแม่หนึ่งตัว จะสร้างโดยสุ่มตำแหน่งของยีนหนึ่งตำแหน่งและทำการเปลี่ยนค่าของยีนในตำแหน่งนั้น ซึ่งโดยทั่วไปความน่าจะเป็นในการเกิดการกลายพันธุ์จะต่ำมากประมาณ 0.001 ถึง 0.01 และมักจะเกิดการการกลายพันธุ์หลังจากการทำการไขว้เปลี่ยนพันธุกรรมแล้ว



รูปที่ 2.11 การทำการไขว้เปลี่ยนพันธุกรรมและการกลายพันธุ์

ตัวอย่างการหาค่าที่มากที่สุดของฟังก์ชัน $15x - x^2$ โดยที่ x มีค่าอยู่ระหว่าง 0 ถึง 15

สมมติให้ x มีค่าเป็นเลขจำนวนเต็มเพื่อให้ง่ายต่อการเข้ารหัสเป็นบิตสตริงและกำหนดความยาวของบิตสตริงให้มีค่าเท่ากับ 4 โดยทำการแปลงค่า x ให้เป็นเลขฐานสองเมื่อแปลงแล้วจะได้บิตสตริงที่แตกต่างกัน 16 ตัวซึ่งมีค่าเลขฐานสิบตั้งแต่ 0 ถึง 15 ดังตารางที่ 2.6

ตารางที่ 2.6 การเข้ารหัสบิตสตริง

Integer	Binary code	Integer	Binary code	Integer	Binary code
1	0 0 0 1	6	0 1 1 0	11	1 0 1 1
2	0 0 1 0	7	0 1 1 1	12	1 1 0 0
3	0 0 1 1	8	1 0 0 0	13	1 1 0 1
4	0 1 0 0	9	1 0 0 1	14	1 1 1 0
5	0 1 0 1	10	1 0 1 0	15	1 1 1 1

ขั้นตอนต่อไปเป็นการกำหนดค่าพารามิเตอร์ต่าง ๆ ที่ใช้ในการทำงานของกระบวนการ GA ดังนี้

- ขนาดของจำนวนประชากร n = 6
- ความน่าจะเป็นในการเกิด crossover เท่ากับ 0.7 (p_c=0.7)
- ความน่าจะเป็นในการเกิด mutation เท่ากับ 0.001 (p_m=0.001)
- กำหนดให้ฟังก์ชันความเหมาะสมมีค่า $f(x) = 15x - x^2$

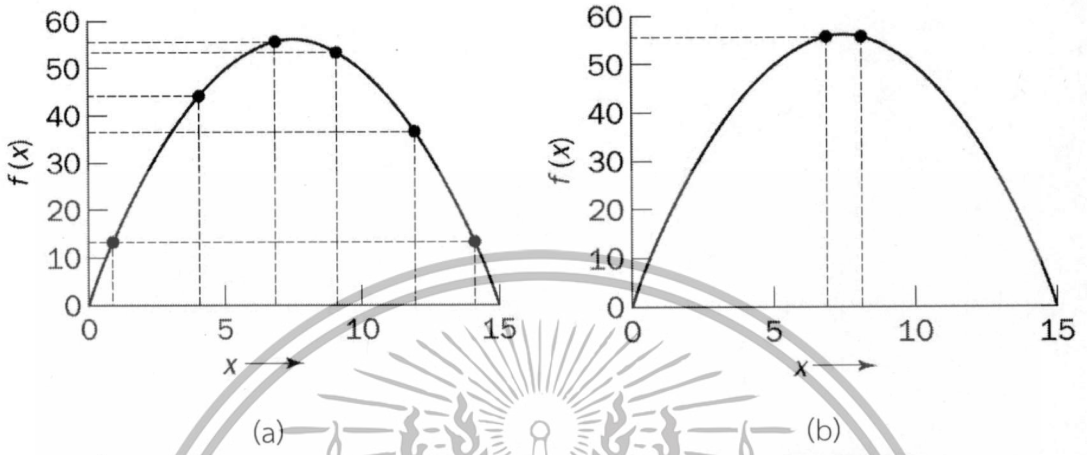
ตารางที่ 2.7 ค่าฟังก์ชั่นความเหมาะของแต่ละโครโมโซม

Chromosome label	Chromosome string	Decoded integer	Chromosome fitness	Fitness ratio, %
X1	1 1 0 0	12	36	16.5
X2	0 1 0 0	4	44	20.2
X3	0 0 0 1	1	14	6.4
X4	1 1 1 0	14	14	6.4
X5	0 1 1 1	7	56	25.7
X6	1 0 0 1	9	54	24.8

จากตารางที่ 2.7 แสดงขั้นตอนการสุ่มประชากรให้มีจำนวนเท่ากับ 6 จะได้โครโมโซม x1 ถึง x6 หลังจากนั้นมีการคำนวณค่าฟังก์ชันความเหมาะสมของประชากรแต่ละตัว และทำการหาค่า fitness ratio จากสูตรในสมการที่ 2.10

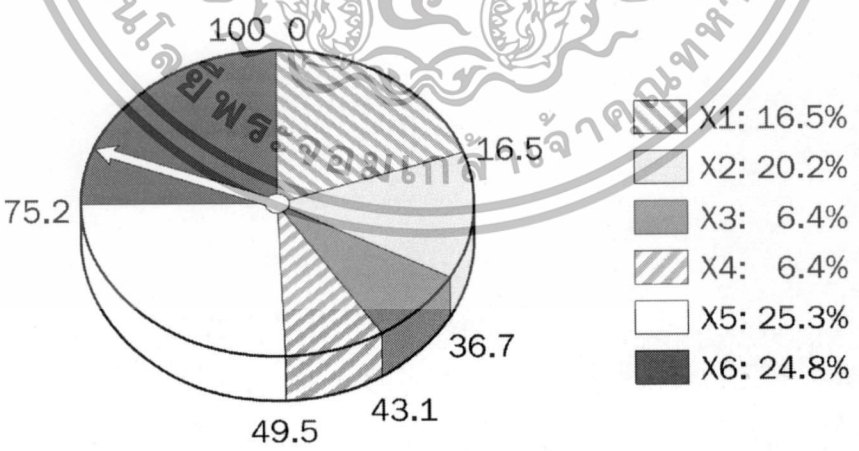
$$\text{Fitness ratio } (X_i) = \frac{f(x_i)}{\sum_{i=1}^{N} f(x_i)} \qquad (2.10)$$

รูปที่ 2.12 แสดงตำแหน่งการพล็อตจุดระหว่างค่า X (ที่แทนแต่ละโครโมโซม) และค่าความชันความเหมาะสม รูปที่ 2.12 ภาพ (a) แสดงตำแหน่งของโครโมโซม 6 ตัวที่ถูกสุ่มมาในรอบแรก ส่วนรูปที่ 2.12 ภาพ (b) แสดงโครโมโซมที่ได้ในรอบสุดท้ายของกระบวนการ GA พบว่าเมื่อผ่านไปหลาย generation หน้าตาของโครโมโซมที่ได้จากลู่เข้าสู่ 2 ค่า คือ 7 และ 8



รูปที่ 2.12 การพล็อตจุดระหว่างค่า X ในแต่ละโครโมโซมกับค่าฟังก์ชันความเหมาะ

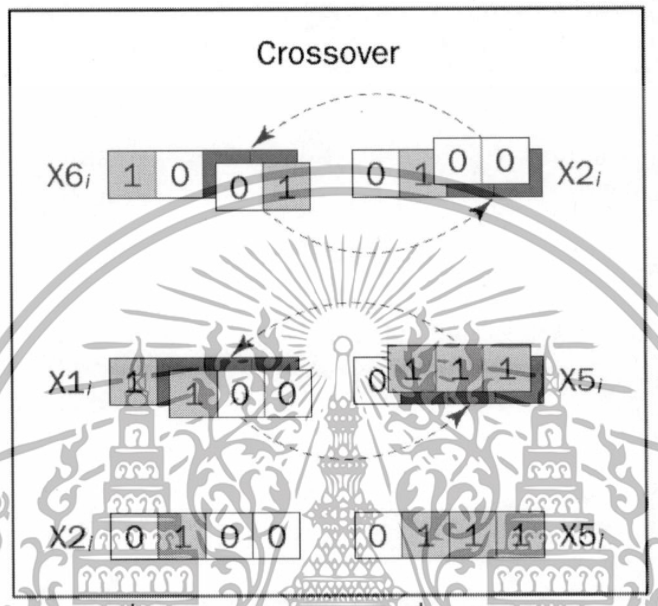
เมื่อทำการสร้างโครโมโซมแบบสุ่มและคำนวณค่าฟังก์ชันความเหมาะสมและค่า fitness ratio ของแต่ละโครโมโซมแล้ว ขั้นตอนต่อไปเป็นการสุ่มเลือกโครโมโซมเพื่อจะทำการผสมพันธุ์ ซึ่งโดยทั่วไปใช้เทคนิคที่เรียกว่า "roulette wheel selection" ดังรูปที่ 2.13 จากรูปจะพบว่าโครโมโซมที่มีค่า fitness ratio สูงมีโอกาสที่จะถูกเลือกสูงกว่าโครโมโซมที่มีค่า fitness ratio ต่ำ



รูปที่ 2.13 การสุ่มแบบการหมุนวงล้อ

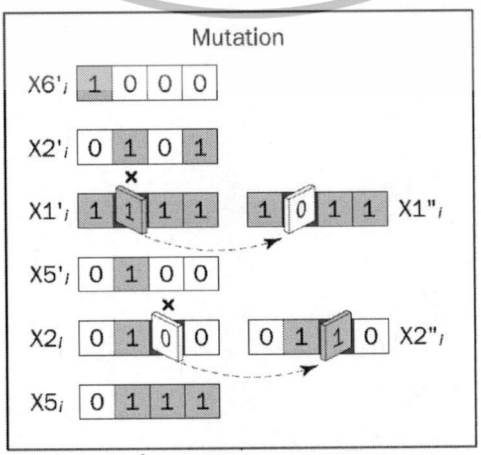
เริ่มแรกประชากรมีจำนวนโครโมโซม 6 ตัวและในแต่ละรอบของการสร้างประชากรใหม่จะต้องมีจำนวนประชากรคงที่ ดังนั้นต้องทำการหมุนวงล้อ (roulette wheel) ทั้งหมด 6 รอบ ซึ่งแต่ละรอบจะได้โครโมโซมหนึ่งตัวมาเป็นโครโมโซมพ่อหรือแม่เมื่อทำการหมุนวงล้อไป 2 รอบจะได้คู่ของ

โครโมโซมที่จะผสมพันธุ์กันและเข้าสู่ขั้นตอนการไขว้เปลี่ยนพันธุกรรม (crossover) ในแต่ละรอบของการทำ cross over โปรแกรมจะสุ่มตำแหน่ง crossover mask และทำการแลกเปลี่ยนยีนของโครโมโซมพ่อและแม่เพื่อสร้างโครโมโซมลูกสองโครโมโซม ถ้าคู่ของโครโมโซมพ่อแม่ไม่ได้ทำการ crossover ก็จะทำการคัดลอกโครโมโซมทั้งหมดของพ่อและแม่ให้เป็นโครโมโซมลูกสองตัว (cloning) ดังรูปที่ 2.14 ซึ่งมีอยู่ด้วยกัน 3 คู่ คือ ผู้แรก X6 กับ X2 คู่ที่สอง คือ X1 กับ X5 และคู่ที่สาม คือ X2 กับ X5 ซึ่งในคู่สุดท้ายจะไม่มีการทำ crossover ดังนั้นจึงทำการ cloning โครโมโซมพ่อและแม่

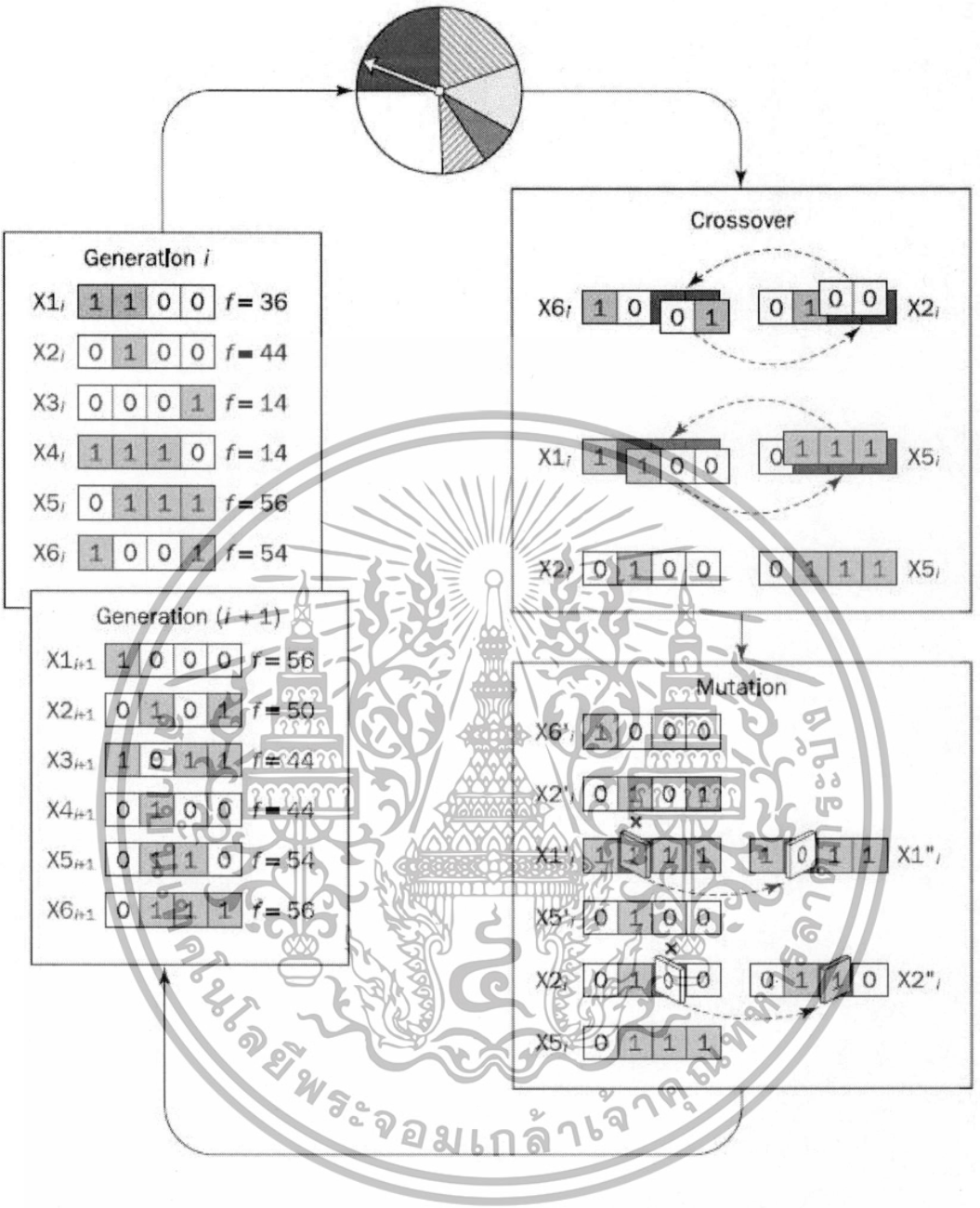


รูปที่ 2.14 การทำการไขว้เปลี่ยนพันธุกรรม

เมื่อสิ้นสุดกระบวนการ crossover จะได้ประชากรตัวใหม่ 6 ตัว ดังรูปที่ 2.15 (โครโมโซมในรูปด้านซ้าย) จากนั้นเข้าสู่กระบวนการ mutation ด้วยการสุ่มตามค่าความน่าจะเป็นของการ mutation ตามที่กำหนดไว้ในตอนต้น จากรูปข้างล่างพบว่า โครโมโซม x1' และ x2 เกิดการ mutation และแต่ละครั้งในขั้นตอนการกลายพันธุ์จะทำการสุ่มตำแหน่งยีนเพียงหนึ่งตำแหน่งเพื่อทำการเปลี่ยนค่าของยีน



รูปที่ 2.15 การกลายพันธุ์



รูปที่ 2.16 กระบวนการทำงานของขั้นตอนวิธีเชิงพันธุกรรม

รูปที่ 2.16 แสดงการทำงานของอัลกอริทึม GA ที่มีการทำงานแบบวนซ้ำไปทีละ generation ซึ่งมีขั้นตอนการทำงานเป็นลำดับ ดังนี้

- คำนวณค่าฟังก์ชันความเหมาะสมของประชากรแต่ละตัว
- ตั้งการปรับโคโมโซมในการทำ crossover ด้วยเทคนิค roulette wheel selection
- ทำการ crossover
- ทำการ mutation จากผลผลิตที่ได้จากขั้นตอน crossover
- แทนเซตของประชากรรุ่นเก่าด้วยประชากรรุ่นใหม่

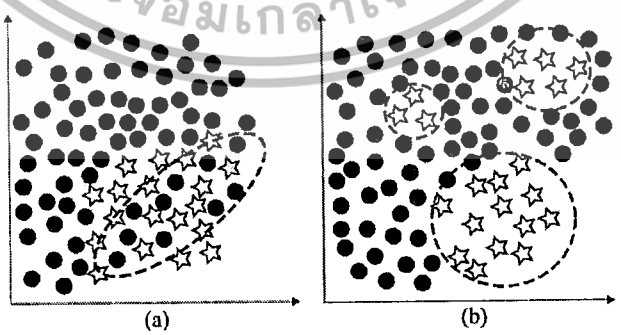
2.7 ความไม่สมดุลของข้อมูล

ปัญหาความไม่สมดุลของข้อมูล (class imbalance) เป็นปัญหาหนึ่งที่ส่งผลต่อประสิทธิภาพในการเรียนรู้ของเครื่องจักรซึ่งเกิดจากความไม่สมดุลของข้อมูลในข้อมูลฝึกสอน โดยที่จำนวนตัวอย่างของกลุ่มข้อมูลกลุ่มหนึ่ง (กลุ่มข้อมูลเสียงส่วนน้อย) มีจำนวนน้อยกว่าตัวอย่างของกลุ่มข้อมูลอีกกลุ่มหนึ่ง(กลุ่มข้อมูลเสียงส่วนมาก) ซึ่งปัญหาที่ยากต่อการจำแนกประเภท ตัวอย่างของข้อมูลที่มีปัญหาความไม่สมดุลของข้อมูลมี 3 กรณีดังนี้

1.) กลุ่มตัวอย่างขนาดเล็ก (small sample size) เป็นปัญหาหนึ่งของความไม่สมดุลของข้อมูล โดยที่กลุ่มข้อมูลเสียงข้างน้อยมีจำนวนน้อยมาก เมื่อเทียบกับกลุ่มข้อมูลเสียงข้างมากที่มีปริมาณมาก จึงส่งผลให้ยากต่อการจำแนกประเภท

2.) กลุ่มตัวอย่างซ้อนทับกัน (overlapping) คือการที่กลุ่มข้อมูลเกิดการซ้อนทับกันระหว่างกลุ่มข้อมูลเสียงข้างน้อยและกลุ่มข้อมูลเสียงข้างมาก จากรูป 2.17 ภาพ (a) จะเห็นได้ว่ากลุ่มข้อมูลเสียงข้างน้อย(แทนด้วยสัญลักษณ์ดาว) แทรกอยู่ระหว่างกลุ่มข้อมูลเสียงข้างมาก (แทนด้วยสัญลักษณ์วงกลม) ซึ่งทำให้ยากที่จะจำแนกข้อมูลเสียงข้างน้อยออกจากกลุ่มข้อมูลเสียงข้างมาก ดังนั้นหากไม่มีการซ้อนทับกันระหว่างกลุ่มข้อมูล จะทำให้การจำแนกและการเรียนรู้ของเครื่องจักรง่ายขึ้น

3.) กลุ่มตัวอย่างมีการกระจายตัว (small distribution) คือการที่กลุ่มข้อมูลเสียงข้างน้อยจำนวนน้อย ๆ กระจายตัวกันออกไป จากรูปที่ 2.17 ภาพ (b) จะเห็นได้ว่า กลุ่มข้อมูลเสียงข้างน้อย(ตัวอย่างบวก แทนด้วยสัญลักษณ์ดาว)กระจายตัวกันออกไป ซึ่งถูกล้อมรอบด้วยกลุ่มข้อมูลเสียงข้างมาก(ตัวอย่างลบแทนด้วยสัญลักษณ์วงกลม) ทำให้ตัวอย่างบวกที่สนใจอาจถูกทำนายเป็นตัวอย่างลบเนื่องจากทฤษฎีเพื่อนบ้านใกล้เคียงที่สุด k ตัวทำให้ตัวอย่างบวกที่มีเพียงไม่กี่ตัว ถูกทำนายเป็นตัวอย่างลบซึ่งมีปริมาณมากกว่า ด้วยเหตุนี้จึงทำให้มีค่าอัตราความผิดพลาด (error rate) ดังสมการที่ 2.12 ที่สูง และส่งผลต่อค่าความถูกต้องของโมเดลจำแนกประเภท



รูปที่ 2.17 ตัวอย่างความไม่สมดุลของข้อมูล

ในงานวิจัยนี้พิจารณาชุดข้อมูลที่มี 2 กลุ่มข้อมูล คือ กลุ่มข้อมูลเสียงข้างน้อยหรือตัวอย่างบวก (positive class) และกลุ่มข้อมูลเสียงข้างมากหรือตัวอย่างลบ (negative class) และทั้งสองกลุ่มข้อมูลถูกนำไปพิจารณาเป็นค่าอัตราความไม่สมดุลของข้อมูลที่เรียกว่า *IR* (imbalanced rate) ซึ่งเป็นสัดส่วนระหว่างจำนวนข้อมูลในกลุ่มเสียงข้างน้อยและจำนวนข้อมูลในกลุ่มเสียงข้างมากดังสมการที่ 2.11 ค่าอัตราความไม่สมดุลนั้นบ่งบอกถึงระดับความไม่สมดุลของข้อมูล หากค่าที่ได้มีค่าเท่ากับ 1 หมายความว่า ข้อมูลดังกล่าวมีความสมดุลมากและจะมีความสมดุลของกลุ่มข้อมูลลดน้อยลงเมื่อค่าที่ได้มากกว่า 1

$$IR = \frac{\text{จำนวนเสียงข้างมาก}}{\text{จำนวนเสียงข้างน้อย}} \tag{2.11}$$

ค่าอัตราความไม่สมดุล เมื่อมีค่าที่มากแล้วนั้นจะส่งผลต่อโมเดลการจำแนกประเภท ทำให้โมเดลที่ได้ รู้จำรูปแบบของกลุ่มข้อมูลเสียงข้างมาก ทำให้ประสิทธิภาพของโมเดลลดต่ำลง ดังนั้นสามารถวัดประสิทธิภาพของโมเดลได้จาก Confusion Matrix ดังแสดงในตารางที่ 2.8 โดยแนวคอลัมน์คือ ตัวอย่างที่โมเดลทำนายได้และแนวนอนคือ ค่าของกลุ่มที่แท้จริงของตัวอย่าง ในตาราง Confusion Matrix ประกอบด้วยค่า TN FN TP และ FP ซึ่งนำไปหาค่าความถูกต้อง (accuracy rate) ดังสมการที่ 2.13 ของโมเดล

ตารางที่ 2.8 Confusion Matrix

	Positive Prediction	Negative Prediction
Positive Class	True Positive (TP)	False Negative (FN)
Negative Class	False Positive (FP)	True Negative (TN)

$$Error = \frac{FP+FN}{TP+FN+FP+TN} \tag{2.12}$$

$$Accuracy = \frac{TP+TN}{TP+FN+FP+TN} \tag{2.13}$$

โดยที่ *TP* คือ จำนวนตัวอย่างที่โมเดลทายถูก และเป็นตัวอย่างบวก

TN คือ จำนวนตัวอย่างที่โมเดลทายถูก และเป็นตัวอย่างลบ

FP คือ จำนวนตัวอย่างที่โมเดลทายออกมาเป็นตัวอย่างบวก แต่แท้จริงแล้วเป็นตัวอย่างลบ

FN คือ จำนวนตัวอย่างที่โมเดลทายออกมาเป็นตัวอย่างลบ แต่แท้จริงแล้วเป็นตัวอย่างบวก

อย่างไรก็ตามค่าความถูกต้องสูงไม่ได้หมายถึงโมเดลนั้นมีประสิทธิภาพดีเสมอไป ดังนั้นจึงต้องใช้ค่าค้นคืน (recall) หรือ TP rate ซึ่งคำนวณได้ดังสมการที่ 2.14 ค่ารู้จำการทายตัวอย่างลบ (specificity) ดังสมการที่ 2.15 และค่าความแม่นยำ (precision) ดังสมการที่ 2.16 ช่วยในการพิจารณา

$$recall = \frac{TP}{TP+FN} \tag{2.14}$$

$$specification = \frac{TN}{FP+TN} \tag{2.15}$$

$$precision = \frac{TP}{FP+TP} \tag{2.16}$$

$$FP_{rate} = \frac{FP}{FP+TN} \tag{2.17}$$

ค่าเหล่านี้ถูกนำมารวมกันเป็นมาตรวัดหนึ่งที่มีชื่อว่า F-measure ซึ่งเป็นสัดส่วนระหว่างค่าความแม่นยำและค่าค้นคืนดังสมการที่ 2.18 โดยค่าที่ได้บ่งบอกว่าความแม่นยำและค่าค้นคืนมีสัดส่วนมากด้วยกันทั้งคู่หรือไม่ หากค่าที่ได้มีค่าที่สูงแสดงว่าโมเดลที่ได้นั้นมีประสิทธิภาพ และอีกมาตรวัดหนึ่งคือ AUC (the area under the ROC curve) ดังสมการที่ 2.19 ซึ่งค่าที่ได้บ่งบอกว่าโมเดลมีความสามารถเพียงใดในการรู้จำตัวอย่างบวก

$$F\text{-measure} = \frac{2^* Precision^* recall}{Precision + recall} \tag{2.18}$$

$$AUC = \frac{1 + recall - FP_{rate}}{2} \tag{2.19}$$

ในปัจจุบันมีเทคนิคมากมายที่ใช้ในการแก้ปัญหาความไม่สมดุลของข้อมูล ซึ่งสามารถจัดกลุ่มได้ 3 แบบ คือการแก้ไขปัญหาในระดับขั้นตอนวิธี (algorithm level) เป็นวิธีการปรับปรุงหรือดัดแปลงวิธีที่มีอยู่เพื่อแก้ปัญหาความไม่สมดุล การแก้ปัญหาในระดับข้อมูล (data level) เป็นการทำให้กลุ่มข้อมูลเกิดความสมดุลโดยใช้การสุ่มตัวอย่าง และการใช้เทคนิค cost-sensitive ซึ่งเป็นวิธีที่อยู่

ระหว่างการการแก้ไขปัญหาในระดับขั้นตอนวิธีและแก้ปัญหาในระดับข้อมูลซึ่งเป็นการรวมทั้งการการดัดแปลงวิธีที่มีอยู่และทำให้กลุ่มข้อมูลเกิดความสมดุล ทำให้การจำแนกกลุ่มข้อมูลเสียงข้างน้อยได้ดีขึ้นและได้โมเดลที่มีประสิทธิภาพมากขึ้น

ในบทนี้จะกล่าวถึงการจัดการในระดับข้อมูลเป็นหลักโดยเทคนิคการสุ่มตัวอย่าง (sampling) ซึ่งแบ่งเป็น 3 เทคนิค คือ เทคนิคการลดจำนวนข้อมูลของกลุ่มข้อมูล (undersampling) เป็นวิธีการลดจำนวนข้อมูลของกลุ่มข้อมูลที่เป็นเสียงข้างมาก (majority class) ให้น้อยลงจนมีปริมาณพอ ๆ กับกลุ่มข้อมูลที่มีเสียงข้างน้อย (minority class) เทคนิคการเพิ่มจำนวนข้อมูลของกลุ่มข้อมูล (oversampling) เป็นวิธีการเพิ่มจำนวนข้อมูลของกลุ่มข้อมูลที่เป็นเสียงข้างน้อยให้เพิ่มขึ้นจนมีปริมาณพอ ๆ กับกลุ่มข้อมูลที่มีเสียงข้างมากและเทคนิคผสมผสาน (hybridsampling) เป็นการรวมเทคนิคการลดจำนวนข้อมูลของกลุ่มข้อมูลและเทคนิคการเพิ่มจำนวนข้อมูลของกลุ่มข้อมูลนำมาใช้ร่วมกัน

2.8 เทคนิคการแก้ปัญหาในระดับข้อมูล (data level)

การแก้ปัญหาในระดับข้อมูลเป็นการทำให้กลุ่มข้อมูลเกิดความสมดุลโดยใช้การสุ่มตัวอย่าง (sampling) ซึ่งแบ่งเป็น 3 เทคนิค คือ เทคนิคการเพิ่มจำนวนข้อมูลของกลุ่มข้อมูล (oversampling) เทคนิคการลดจำนวนข้อมูลของกลุ่มข้อมูล (undersampling) และเทคนิคผสมผสาน (hybridsampling)

2.8.1 ขั้นตอนวิธี SMOTE

SMOTE [6] เป็นวิธีการเพิ่มกลุ่มข้อมูลที่เป็นเสียงส่วนน้อยขึ้นมาใหม่โดยการสุ่มเปรียบเทียบจากเพื่อนบ้านใกล้เคียงที่สุด k ตัวในกลุ่มข้อมูลที่เป็นเสียงส่วนน้อย วิธีนี้ถูกนำไปใช้ในการปรับปรุงโมเดลให้มีประสิทธิภาพในการทำนายตัวอย่างให้มีค่าความถูกต้อง (accuracy) เพิ่มมากขึ้นในกรณีที่มีกลุ่มข้อมูลที่เป็นเสียงส่วนน้อย

Algorithm *SMOTE*(T, N, k)
Input: Number of minority class samples T; Amount of SMOTE N%; Number of nearest neighbors k
Output: (N/100) * T synthetic minority class samples
(* *If N is less than 100%, randomize the minority class samples as only a random percent of them will be SMOTEd.* *)
if $N < 100$
  then Randomize the T minority class samples
    $T = (N/100) * T$
    $N = 100$
endif
$N = (int)(N/100)$ (* *The amount of SMOTE is assumed to be in integral multiples of 100.* *)
k = Number of nearest neighbors
numattrs = Number of attributes
Sample[ ][ ]: array for original minority class samples
newindex: keeps a count of number of synthetic samples generated, initialized to 0
Synthetic[ ][ ]: array for synthetic samples
(* *Compute k nearest neighbors for each minority class sample only.* *)
for $i \leftarrow 1$ **to** T
    Compute k nearest neighbors for i, and save the indices in the *nnarray*
    Populate(N, i, *nnarray*)
endfor

Populate(N, i, *nnarray*) (* *Function to generate the synthetic samples.* *)
while $N \neq 0$
    Choose a random number between 1 and k, call it nn. This step chooses one of the k nearest neighbors of i.
    for $attr \leftarrow 1$ **to** *numattrs*
      Compute: $dif = Sample[nnarray[nn]][attr] - Sample[i][attr]$
      Compute: gap = random number between 0 and 1
      $Synthetic[newindex][attr] = Sample[i][attr] + gap * dif$
    endfor
    newindex++
    $N = N - 1$
endwhile
return (* *End of Populate.* *)
End of Pseudo-Code.

รูปที่ 2.18 pseudo code แสดงวิธีการทำงานของขั้นตอนวิธี SMOTE

จากตัวอย่างรูปที่ 2.18 แสดง pseudo code ของขั้นตอนวิธี SMOTE โดยมีรายละเอียดดังนี้

ตัวแปรเข้า : T คือ จำนวนตัวอย่างของกลุ่มข้อมูลที่มีเสียงส่วนน้อย (minority class)

N คือ ปริมาณของ SMOTE คิดเป็นเปอร์เซ็นต์

k คือ จำนวนของเพื่อนบ้านใกล้เคียงที่สุด k ตัว (nearest neighbors)

ผลลัพธ์ : (N/100)*T คือ จำนวนตัวอย่างจากกลุ่มข้อมูลที่มีเสียงส่วนน้อยที่ได้จากการสังเคราะห์ด้วยขั้นตอนวิธี SMOTE

การทำงานของขั้นตอนวิธี SMOTE

1. (บรรทัดที่ 2-6) เงื่อนไขตรวจสอบค่า *N* จะสุ่มตัวอย่างของกลุ่มข้อมูลที่มีเสียงส่วนน้อยเมื่อมีค่าน้อยกว่า 100 โดยกำหนดให้

   T เก็บค่าจำนวนตัวอย่างของกลุ่มข้อมูลสังเคราะห์ที่คิดจากปริมาณของ SMOTE

   N เก็บค่าเท่ากับ 100 เมื่อจบการงานเงื่อนไขตรวจสอบ
2. (บรรทัดที่ 7) คำนวณค่า *N*/100 โดยกำหนดให้เก็บค่าเป็นประเภทข้อมูล integer เท่านั้น
3. (บรรทัดที่ 8-12) กำหนดค่าตัวแปรดังนี้

   k คือ ตัวแปรเก็บจำนวนของเพื่อนบ้านใกล้เคียงที่สุด k ตัว

   numattrs คือ ตัวแปรเก็บจำนวนคุณลักษณะ

   Sample คือ อาเรย์ 2 มิติ เก็บตัวอย่างดั้งเดิมของกลุ่มข้อมูลที่มีเสียงส่วนน้อย

   newindex คือ ตัวแปรใช้ในการเก็บลำดับของตัวอย่างที่สร้าง มีค่าเริ่มต้นเป็น 0

   Synthetic คือ อาเรย์ 2 มิติ เก็บตัวอย่างที่สังเคราะห์จากขั้นตอนวิธี SMOTE
4. (บรรทัดที่ 13-16) การวนรอบทำซ้ำกำหนดตัวแปร *i* = 1 ทำงานวนซ้ำจำนวน *i* ถึง *T* รอบ โดยจะคำนวณเพื่อนบ้านด้วยค่า *i* และเก็บตำแหน่งเพื่อนบ้านที่ได้ในอาเรย์ *nnarray* และส่งค่า *N*, *i*, ตัวชี้อาเรย์ *nnarray* ไปยังฟังก์ชัน *Populate*
5. (บรรทัดที่ 17-27) ฟังก์ชัน*Populate* มีการทำงานเพื่อสร้างตัวอย่างสังเคราะห์มีการทำงานแบบวนรอบทำซ้ำโดยตรวจสอบค่า *N* มีค่าไม่เท่ากับศูนย์จริงหรือไม่ โดยจะสมมติค่า *N* มีค่าไม่เท่ากับศูนย์เป็นจริงเพื่อแสดงตัวอย่างการคำนวณค่า (บรรทัดที่ 21) *gap* คือ จริงที่ได้จากการสุ่มจำนวนระหว่าง 0 ถึง 1 พิจารณาตัวอย่าง (6,4) และ (4,3) เป็นเพื่อนบ้านใกล้เคียงที่สุด k ตัว โดยกำหนดให้

   (6,4) คือ ตัวอย่างที่สนใจเก็บอยู่ในอาเรย์ *Sample*[1][1] = 6 และ *Sample*[1][2]= 4

   (4,3) คือ หนึ่งในเพื่อนบ้านใกล้เคียงที่สุด k ตัวเก็บอยู่ในอาเรย์*Sample*[2][1] = 4 และ *Sample*[2][2] = 3

   0.5 คือ ค่า *gap* ที่ได้จากการสุ่ม

   Sample[1][1] = 6, *Sample*[2][1] = 4 จะได้ *dif*= 4-6 = -2

   Sample[1][2]= 4, *Sample*[2][2] = 3 จะได้ *dif*=3-4 = -1(บรรทัดที่ 20)

   จากบรรทัดที่ 22จะทำการสังเคราะห์ตัวอย่างใหม่

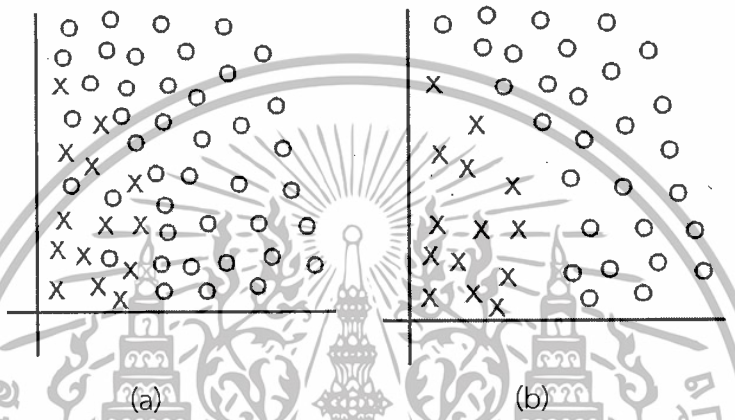
   Synthetic[0][1] = 6 + 0.5(-2) = 5

   Synthetic[0][2] = 4 + 0.5(-1) = 3.5

   ดังนั้นตัวอย่างใหม่จะถูกสร้างขึ้นเป็น(5,3.5)

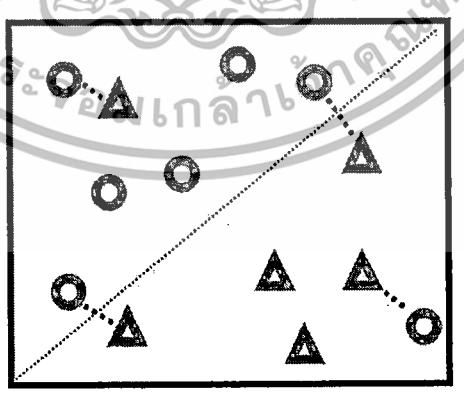
2.8.2 ขั้นตอนวิธี Tomek Links

ปัญหาความไม่สมดุลของข้อมูลสามารถแก้ปัญหาได้โดยใช้เทคนิคการลดจำนวนกลุ่มข้อมูลเสียงข้างมาก หนึ่งในนั้นคือเทคนิคที่มีชื่อว่า Tomek Links [7] ซึ่งเป็นการกำจัดกลุ่มข้อมูลเสียงข้างมากที่เกิดการซ้อนทับกันของกลุ่มข้อมูล (overlapping) จากรูปที่ 2.19 ภาพ (a) เป็นการจับคู่ระหว่างกลุ่มข้อมูลเสียงข้างน้อย (ตัวอย่างบวกแทนด้วยสัญลักษณ์ x) และกลุ่มข้อมูลเสียงข้างมาก (ตัวอย่างลบแทนด้วยสัญลักษณ์ o) โดยทำการเปรียบเทียบในแต่ละคู่เพื่อหาระยะทางที่ใกล้กันมากที่สุดระหว่างสองกลุ่มข้อมูล ซึ่งในแต่ละคู่ตัวอย่างที่ใกล้กันมากที่สุด เรียกว่า Tomek Links จากรูป 2.19 ภาพ (b) จะทำการกำจัดตัวอย่างลบออกไปในแต่ละ Tomek Links



รูปที่ 2.19 ตัวอย่างTomek Links

ตัวอย่างการหา Tomek Links กำหนดให้ E_i(วงกลม) และ E_j(สามเหลี่ยม) เป็นกลุ่มข้อมูลที่มีความแตกต่างกันและ (E_i,E_j) จะเป็น Tomek Links เมื่อระยะทางระหว่าง $d(E_i,E_k)$ หรือ $d(E_j,E_k)$ มีค่ามากกว่า $d(E_i,E_j)$ โดย E_k คือตัวอย่างใด ๆ ดังรูปที่ 2.20



รูปที่ 2.20 ตัวอย่างการหา Tomek Links

2.8.3 ขั้นตอนวิธี SMOTE + Tomek Links

SMOTE + Tomek Links [8] คือเทคนิคหนึ่งในการผสมผสานระหว่างเทคนิคการเพิ่มจำนวนกลุ่มข้อมูลเสียงข้างน้อยและเทคนิคการลดจำนวนกลุ่มข้อมูลเสียงข้างมาก โดยเทคนิค SMOTE จะทำการเพิ่มจำนวนกลุ่มข้อมูลเสียงข้างน้อยให้มีจำนวนพอ ๆ กับจำนวนกลุ่มข้อมูลเสียงข้างมาก และใช้เทคนิค Tomek Links กำจัดกลุ่มข้อมูลที่เกิดการซ้อนทับกัน (overlapping) ของกลุ่มข้อมูล จากรูปที่ 2.21 ภาพ (a) คือกลุ่มข้อมูลที่เกิดความไม่สมดุลและเกิดการซ้อนทับกันของกลุ่มข้อมูลและใช้เทคนิค SMOTE ในการเพิ่มจำนวนตัวอย่างบวก ดังรูปที่ 2.22 ภาพ (b) และใช้เทคนิค Tomek Links ในการกำจัดตัวอย่างลบ ดังรูปที่ 2.22 ภาพ (c)

รูปที่ 2.21 เทคนิค SMOTE + Tomek Links

2.9 เทคนิคการวัดประสิทธิภาพโมเดล

ในการสร้างโมเดลจำแนกประเภทข้อมูลได้นำเทคนิค *k*-fold Crossvalidation มาใช้ในการแบ่งชุดข้อมูลซึ่งเป็นวิธีการแบ่งชุดข้อมูลออกเป็น k ชุด $\{D_1, D_2, \ldots, D_k\}$ แต่ละชุดจะถูกแบ่งด้วยขนาดที่เท่ากัน ชุดข้อมูลฝึกสอนและชุดข้อมูลทดสอบจะถูกนำมาใช้จำนวน k ครั้ง โดยครั้งแรกกำหนด D_1 ให้เป็นชุดข้อมูลทดสอบและชุดข้อมูลที่เหลือคือชุดข้อมูลฝึกสอน $\{D_2, D_3, \ldots, D_k\}$ และครั้งถัดไปกำหนด D_2ให้เป็นชุดข้อมูลทดสอบและชุดข้อมูลที่เหลือคือชุดข้อมูลฝึกสอน $\{D_1, D_3, \ldots, D_k\}$ ทำจนกระทั่งครบ k ครั้งและในการทดลองนี้ได้กำหนดค่า $k = 5$ เพื่อวัดประสิทธิภาพของโมเดล ดังรูปที่ 2.22

fold 1	test	train	train	train	train
fold 2	train	test	train	train	train
fold 3	train	train	test	train	train
fold 4	train	train	train	test	train
fold 5	train	train	train	train	test

รูปที่ 2. 22 ตัวอย่าง *k*-fold Crossvalidation เมื่อกำหนดค่า $k = 5$

2.10 งานวิจัยที่เกี่ยวข้อง

งานวิจัย Preprocessing of imbalanced breast cancer data using feature selection combined with over-sampling technique for classification [9] เป็นการศึกษาการแก้ปัญหาความไม่สมดุลของข้อมูลที่เกิดขึ้นในวงการแพทย์ที่เกี่ยวกับผู้ป่วยมะเร็งเต้านม โดยรวมวิธีการคัดเลือกคุณลักษณะ (feature selection) และการสุ่มตัวอย่าง (oversampling) ซึ่งมีชื่อเรียกว่า FOT วิธีดังกล่าวเป็นการทำความสะอาดข้อมูลก่อนทำการสร้างโมเดลจำแนกประเภท โดยทำการกำจัดคุณลักษณะที่ไม่จำเป็นออกไปและนำข้อมูลที่เหลือผ่านกระบวนการ oversampling และสร้างโมเดลจำแนกประเภทด้วยโมเดล 3 โมเดล คือ Decision tree BayesNets และ OneR จากผลการทดลองปรากฏว่าโมเดลทั้ง 3 โมเดลที่มีการใช้เทคนิค FOT มีค่า F-measure สูงขึ้นกว่าโมเดลที่ไม่ได้ใช้เทคนิค FOT

งานวิจัย A novel evolutionary preprocessing method based on over-sampling and under-sampling for imbalanced datasets [10] เป็นการศึกษาการสุ่มตัวอย่างใหม่ของกลุ่มข้อมูลโดยใช้เทคนิค SMOTE ซึ่งหากใช้เทคนิค SMOTE เพียงวิธีเดียวอาจนำไปสู่ประสิทธิภาพที่ลดต่ำลง ดังนั้นจึงใช้เทคนิคที่ชื่อว่า CHC ร่วมในการสุ่มตัวอย่าง ซึ่งจะมีประสิทธิภาพที่มากกว่าการใช้ SMOTE เพียงวิธีเดียว ดังนั้นงานวิจัยนี้จึงรวมวิธีของ SMOTE เข้ากับ CHC และเปรียบเทียบวิธีต่าง ๆ อีก 5 วิธี คือ RUS, TL, ROS, SMOTE, และ SMOTE+TL โดยใช้โมเดล Decision tree C4.5 ในการจำแนกประเภท จากการทดลองพบว่า ประสิทธิภาพของ oversampling (SMOTE,ROS) และ hybrid (SMOTE+TL,SMOTE+CHC) มีประสิทธิภาพมากกว่าวิธีundersampling (TL,RUS) อย่างไรก็ตามการเพิ่มจำนวนตัวอย่างอาจนำไปสู่ประสิทธิภาพที่ต่ำลงดังนั้นจึงใช้ค่าอัตราการเพิ่มตัวอย่าง (over-sampling rate) ในการพิจารณาเทคนิคการเพิ่มจำนวนตัวอย่างซึ่งหากมีค่าที่สูงจะส่งผลให้ประสิทธิภาพลดต่ำลง ผลปรากฏว่า SMOTE+CHC ให้ค่าอัตราการเพิ่มตัวอย่างในอัตราที่ต่ำมากเมื่อเทียบกับเทคนิคการเพิ่มจำนวนตัวอย่างอื่น ๆ

งานวิจัย Improving emotion classification in imbalanced YouTube dataset using SMOTE algorithm [11] เป็นการนำเทคนิคทางดาต้าไมน์นิ่งมาประยุกต์ใช้กับแอปพลิเคชั่นในปัจจุบันอย่าง Youtube ซึ่งเป็นสื่อมัลติมีเดียที่ประกอบไปด้วยสื่อหลากหลายประเภท เช่น เพลง โฆษณา หรือตัวอย่างภาพยนตร์ โดยการจำแนกประเภทของสื่อมัลติมีเดียจากข้อความที่แสดงความคิดเห็น (comment) ซึ่งแบ่งออกเป็น 9 ประเภทด้วยกัน คือ anger, disgust, fear, happiness, sadness, surprise, emotion, related และ unrelated และใช้วิธีการเรียนรู้ของเครื่องจักร (machine learning) ในการจำแนกประเภท ประกอบไปด้วย decision tree, naïve bayes และ Support Vector Machine และใช้เทคนิค SMOTE ในการเพิ่มจำนวนตัวอย่างของเสียงส่วนน้อยเพื่อแก้ปัญหาความไม่สมดุลของข้อมูล ผลการทดลองพบว่าการใช้ SMOTE สามารถเพิ่มประสิทธิภาพของโมเดลได้ ซึ่งสามารถเพิ่มได้ถึง 16.9 %

งานวิจัย Novel Algorithm for Imbalance Data Classification Based on Genetic Algorithm Improved SMOTE [12] นำเสนอการใช้เทคนิค SMOTE ร่วมกับ genetic algorithm

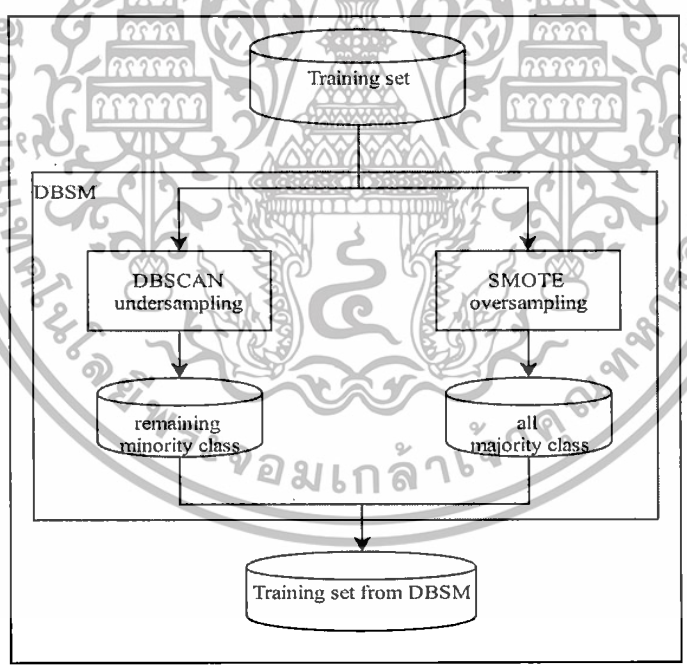
(GA) เพื่อเพิ่มประสิทธิภาพในการแก้ปัญหาความไม่สมดุลของกลุ่มข้อมูลให้ดีขึ้น ซึ่งเทคนิค SMOTE เป็นหนึ่งในเทคนิคที่ได้รับความนิยมมากที่สุดในการแก้ปัญหาความไม่สมดุลของกลุ่มข้อมูล แต่เนื่องจากเทคนิค SMOTE ใช้อัตราการสุ่มตัวอย่าง (percent of synthetic instance) เพียงค่าเดียวกับทุก ๆ ตัวอย่าง (minority class sample) ซึ่งแต่ละตัวอย่างนั้นควรจะมีค่าจำเพาะในการสุ่มตัวอย่างเป็นของตัวเอง ดังนั้นงานวิจัยชิ้นนี้จึงได้นำเทคนิค GA ซึ่งเป็นเทคนิคทางปัญญาประดิษฐ์อย่างหนึ่งที่ใช้ในการค้นหา การเพิ่มประสิทธิภาพ และการเรียนรู้ด้วยการเลียนแบบทฤษฎีการวิวัฒนาการทางธรรมชาติมาใช้ร่วมกับเทคนิค SMOTE เพื่อหาอัตราการสุ่มตัวอย่างที่เหมาะสมในแต่ละตัวอย่าง โดยงานวิจัยชิ้นนี้ใช้ชื่อย่อว่า GASMOTE Algorithms จากการทดลองเทคนิค GASMOTE กับชุดข้อมูลที่เกิดปัญหาความไม่สมดุลทั้งหมดสิบชุดข้อมูล พบว่า GASMOTE สามารถเพิ่มประสิทธิภาพของโมเดลได้ดีที่สุดในทุก ๆ ชุดข้อมูลเมื่อเทียบกับขั้นตอนวิธี C4.5, SMOTE+C4.5 และ Borderline-SMOTE+C4.5 เมื่อวัดประสิทธิภาพของโมเดลด้วยค่า F-measure นอกจากนี้ GASMOTE สามารถเพิ่มประสิทธิภาพของโมเดลได้มากขึ้นถึง 5.9 เปอร์เซ็นต์ เมื่อเทียบกับเทคนิค SMOTE

บทที่ 3
ขั้นตอนวิธี GADBSM

ในบทนี้จะอธิบายถึงขั้นตอนวิธี DBSM ซึ่งเป็นเทคนิคใหม่ที่พัฒนาขึ้นเพื่อแก้ปัญหาความไม่สมดุลของข้อมูล และการนำขั้นตอนวิธีทางพันธุกรรม (Genetic algorithm) มาใช้ร่วมกับขั้นตอนวิธี DBSM หรือเรียกว่า GADBSM โดยในส่วนของเทคนิค DBSM จะประกอบไปด้วยขั้นตอนการลดจำนวนตัวอย่างของกลุ่มข้อมูลด้วยขั้นตอนวิธี DBSCAN undersampling และเทคนิคการเพิ่มจำนวนตัวอย่างของกลุ่มข้อมูลด้วยขั้นตอนวิธี SMOTE สำหรับขั้นตอนวิธีเชิงพันธุกรรมจะกล่าวถึงการนำมาใช้ปรับพารามิเตอร์ในขั้นตอนวิธี DBSM

3.1 ขั้นตอนวิธี DBSM

DBSM คือขั้นตอนวิธีใหม่ในการสุ่มตัวอย่างแบบผสมผสานโดยการนำเทคนิค SMOTE ซึ่งเป็นหนึ่งในเทคนิคการแก้ปัญหาความไม่สมดุลแบบเพิ่มจำนวนตัวอย่างของกลุ่มข้อมูล ร่วมกับการประยุกต์ใช้ขั้นตอนวิธี DBSCAN ซึ่งเป็นการจัดกลุ่มข้อมูลโดยใช้ความหนาแน่น โดยจะถูกใช้เป็นตัวแทนของเทคนิคการลดจำนวนตัวอย่างของกลุ่มข้อมูล



รูปที่ 3.1 หลักการทำงานของ DBSM

จากรูปที่ 3.1 แสดงหลักการทำงานของขั้นตอนวิธี DBSM โดยเริ่มต้นชุดข้อมูลฝึกสอนจะถูกนำเข้าสู่กระบวนการสุ่มตัวอย่างด้วยขั้นตอนวิธีของ DBSM ซึ่งภายในจะประกอบไปด้วยสองเทคนิค คือ เทคนิคการลดจำนวนตัวอย่างของคลาสสลบด้วยขั้นตอนวิธี DBSCAN undersampling และเทคนิคการเพิ่มจำนวนตัวอย่างบวกด้วยขั้นตอนวิธี SMOTE ดังนั้นผลลัพธ์สุดท้ายที่ได้จากขั้นตอนวิธี

ของ DBSM คือจำนวนตัวอย่างที่คงเหลือของคลาสลบจากเทคนิค DBSCAN undersampling และจำนวนตัวอย่างทั้งหมดของคลาสบวกหลังจากผ่านขั้นตอนวิธี SMOTE

สำหรับเทคนิคการลดจำนวนตัวอย่างของกลุ่มข้อมูล (DBSCAN undersampling) ในขั้นตอนวิธี DBSM สามารถแบ่งออกเป็นขั้นตอนหลักได้ 3 ขั้นตอน คือ ขั้นตอนการจัดกลุ่ม ขั้นตอนการวัดระยะทาง และขั้นตอนการลดจำนวนตัวอย่าง ดังนี้

1.) ขั้นตอนการจัดกลุ่ม จำนวนตัวอย่างทั้งหมดในชุดข้อมูลฝึกสอนจะถูกจัดกลุ่มด้วยขั้นตอนวิธี DBSCAN หลังจากผ่านขั้นตอนนี้จะได้รูปแบบของกลุ่มที่แตกต่างกัน 3 รูปแบบ คือ กลุ่มที่มีสมาชิกทั้งหมดมีคลาสเป็นเสียงส่วนน้อย (คลาสบวก) กลุ่มที่มีสมาชิกทั้งหมดเป็นคลาสเสียงส่วนมาก (คลาสลบ) และกลุ่มสุดท้ายคือกลุ่มที่มีสมาชิกเป็นคลาสบวกและลบ ซึ่งในงานวิจัยนี้เลือกพิจารณาเฉพาะกลุ่มข้อมูลที่มีสมาชิกเป็นทั้งหมดเป็นคลาสลบ และกลุ่มข้อมูลที่มีสมาชิกเป็นทั้งคลาสลบและคลาสบวก

2.) ขั้นตอนการวัดระยะทาง ในขั้นตอนนี้จะแบ่งการวัดระยะทางออกเป็น 2 ประเภทตามเงื่อนไขของรูปแบบของกลุ่มข้อมูล คือ กลุ่มข้อมูลที่มีรูปแบบสมาชิกในกลุ่มเป็นคลาสลบเพียงอย่างเดียวจะทำการหาจุดศูนย์กลางของกลุ่มข้อมูล และทำการวัดระยะทางจากทุกตัวอย่างในกลุ่มข้อมูลเทียบกับจุดศูนย์กลางของกลุ่มข้อมูล เพื่อหาตัวอย่างที่ใกล้กับจุดศูนย์กลางมากที่สุดตามลำดับ และในส่วนของกลุ่มข้อมูลที่มีสมาชิกเป็นทั้งคลาสบวกและคลาสลบจะทำการวัดระยะทางระหว่างคลาสลบเทียบกับคลาสบวกเพื่อหาตัวอย่างของคลาสลบที่ใกล้กับคลาสบวกที่สุด

3.) ขั้นตอนการลดจำนวนตัวอย่าง ในขั้นตอนนี้จะทำการลบจำนวนตัวอย่างคลาสลบที่อยู่ใกล้กับตัวอย่างที่เป็นคลาสบวกออก 50 เปอร์เซ็นต์ของจำนวนตัวอย่างที่เป็นคลาสลบในกลุ่มข้อมูลนั้น ๆ สำหรับกลุ่มข้อมูลที่มีรูปแบบสมาชิกในกลุ่มเป็นคลาสลบเพียงอย่างเดียวจะทำการลบตัวอย่างที่ใกล้กับจุดศูนย์กลางของกลุ่มข้อมูลออก 50 เปอร์เซ็นต์เช่นกัน

Algorithm: DBSCAN Undersampling

Input: S: All training set with minority class and majority class, **ε**: Epsilon, Minpts: Minpoints

Output: D: remaining majority class instances.

```
1.  [Cluster] = DBSCAN(S, ε, Minpts) // Cluster is a set of clusters
2.  For i = 1 to n // n is a number of clusters generated by DBSM.
3.      Let Di be a number of majority class instances in ith cluster
4.      If all members in Clusteri are majority class then
5.          Centroid = compute_centroid(Clusteri)
6.          For j = 1 to m // m is a number of clusteri's members
7.              MI = calDistance(Centroid, j)
8.          End for
9.      Else if member in Clusteri are minority and majority class instances then
10.         For j = 1 to m1 // m1 is majority class instances
11.             For k = 1 to m2 // m2 is minority class instances
12.                 MI = findSmallestDistance(j, k)
13.             End for
14.         End for
15.     End if
16.     MI = sortingDistance(MI)
17.     Di = removeSmallestDistance(MI, 50%)
18.     return D = ∪ Di
19. End for
```

รูปที่ 3.2 ขั้นตอนวิธีการทำงานของอัลกอริทึม DBSCAN undersampling

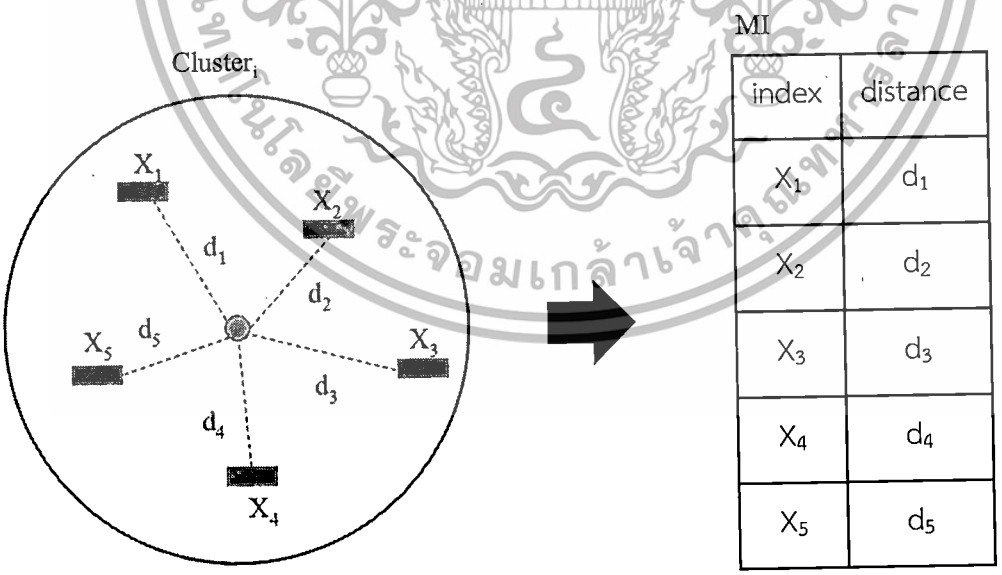
จากตัวอย่างรูปที่ 3.2 แสดงขั้นตอนวิธีของอัลกอริทึม DBSCAN undersampling โดยมีรายละเอียดดังนี้

ตัวแปรเข้า : S คือชุดข้อมูลฝึกสอนซึ่งประกอบไปด้วยกลุ่มข้อมูลเสียงส่วนมาก (คลาสลบ) และกลุ่มข้อมูลเสียงส่วนน้อย (คลาสบวก)
Eps คือ ระยะทางของจุดเพื่อนบ้าน
Minpts คือ จำนวนของจุดเพื่อนบ้านขั้นต่ำ

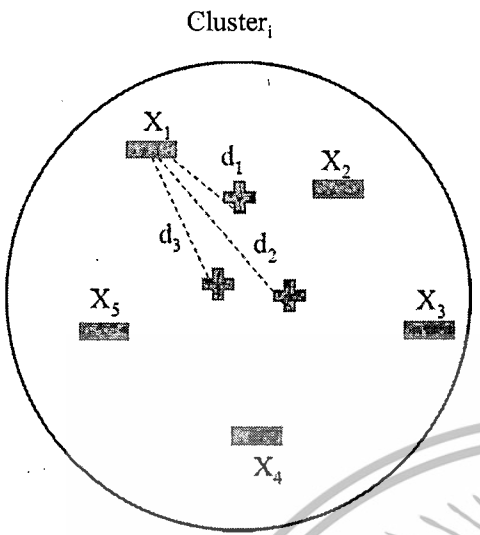
ผลลัพธ์ : จำนวนตัวอย่างหลังจากทำการกำจัดกลุ่มข้อมูลเสียงส่วนมากบางส่วน

ขั้นตอนวิธี :

1. (บรรทัดที่ 1) นำชุดข้อมูลฝึกสอนผ่านขั้นตอนวิธี DBSCAN เพื่อจัดกลุ่มของกลุ่มข้อมูล
2. (บรรทัดที่ 2-19) การประมวลผลในแต่ละกลุ่มข้อมูล
3. (บรรทัดที่ 3) กำหนดให้ D_i คือจำนวนของกลุ่มข้อมูลเสียงส่วนมากในกลุ่มที่ i
4. (บรรทัดที่ 4) ถ้าจำนวนสมาชิกทั้งหมดในกลุ่มข้อมูลที่ i เป็นคลาสลบให้ทำงานในบรรทัดที่ 5
5. (บรรทัดที่ 5) หาจุดศูนย์กลางของกลุ่มข้อมูลที่ i
6. (บรรทัดที่ 6-8) วัดระยะทางระหว่างคลาสลบทั้งหมดกับจุดศูนย์กลาง และเก็บระยะทางของแต่ละตัวอย่างลงอาร์เรย์ MI [index, distance] แสดงดังรูปที่ 3.3
7. (บรรทัดที่ 9) ถ้าจำนวนสมาชิกทั้งหมดในกลุ่มข้อมูลที่ i ประกอบด้วยคลาสลบและคลาสบวกให้ทำงานในบรรทัดที่ 10
8. (บรรทัดที่ 10-14) วัดระยะทางระหว่างคลาสลบกับคลาสบวก และเก็บระยะทางที่น้อยที่สุดของคลาสลบลงอาร์เรย์ MI [index, distance] แสดงดังรูปที่ 3.4
9. (บรรทัดที่ 16) เรียงลำดับ distance ในอาร์เรย์ MI โดยเรียงจากค่าน้อยไปมาก
10. (บรรทัดที่ 17) เลือกจำนวนตัวอย่างในคลาสลบ 50 เปอร์เซ็นต์จาก MI โดยคิดจากค่า distance น้อยที่สุด และทำการกำจัดตัวอย่างเหล่านั้นออกจาก D_i
11. (บรรทัดที่ 18) คืนค่าจำนวนตัวอย่างของคลาสลบหลังจากผ่านขั้นตอนการกำจัดตัวอย่าง

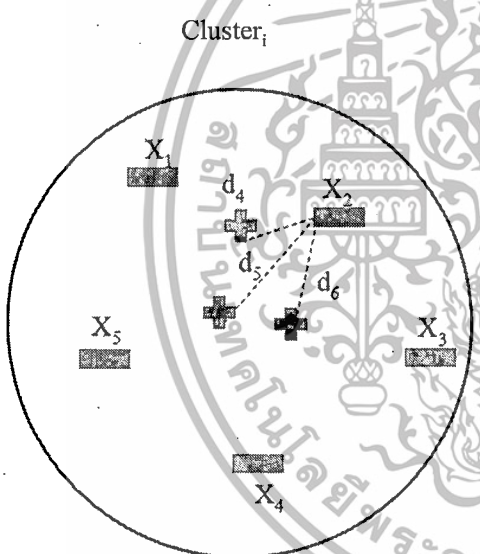


รูปที่ 3.3 การวัดระยะทางในกรณีที่ในกลุ่มข้อมูลมีสมาชิกเป็นคลาสลบทั้งหมด



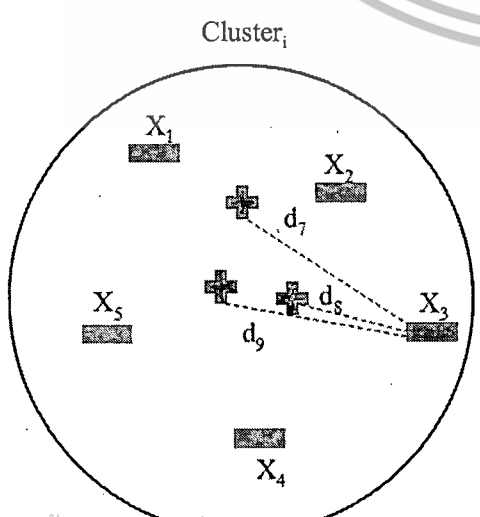
MI

index	distance
X_1	findSmallestDistance(d_1, d_2, d_3)
X_2	
X_3	
X_4	
X_5	



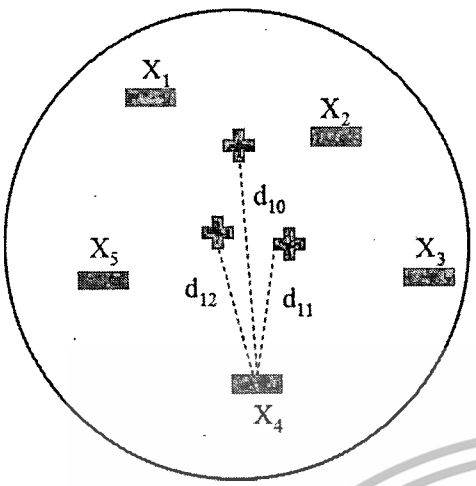
MI

index	Distance
X_1	findSmallestDistance(d_1, d_2, d_3)
X_2	findSmallestDistance(d_4, d_5, d_6)
X_3	
X_4	
X_5	



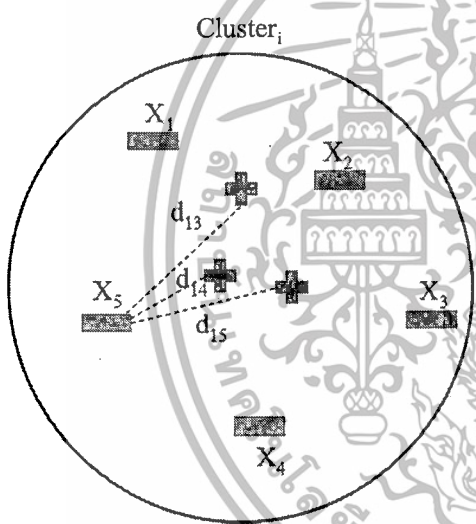
MI

index	distance
X_1	findSmallestDistance(d_1, d_2, d_3)
X_2	findSmallestDistance(d_4, d_5, d_6)
X_3	findSmallestDistance(d_7, d_8, d_9)
X_4	
X_5	



MI

index	distance
X_1	findSmallestDistance(d_1, d_2, d_3)
X_2	findSmallestDistance(d_4, d_5, d_6)
X_3	findSmallestDistance(d_7, d_8, d_9)
X_4	findSmallestDistance(d_{10}, d_{11}, d_{12})
X_5	



MI

index	distance
X_1	findSmallestDistance(d_1, d_2, d_3)
X_2	findSmallestDistance(d_4, d_5, d_6)
X_3	findSmallestDistance(d_7, d_8, d_9)
X_4	findSmallestDistance(d_{10}, d_{11}, d_{12})
X_5	findSmallestDistance(d_{13}, d_{14}, d_{15})

รูปที่ 3.4 การวัดระยะทางในกรณีที่ในกลุ่มข้อมูลมีสมาชิกเป็นคลาสลบและคลาสบวก

3.2 ขั้นตอนวิธี GADBSM

GADBSM คือ การนำขั้นตอนวิธีเชิงพันธุกรรมมาประยุกต์ใช้ร่วมกับขั้นตอนวิธี DBSM โดยขั้นตอนวิธีเชิงพันธุกรรมจะถูกนำมาใช้ในการปรับค่าพารามิเตอร์ในขั้นตอนวิธี DBSM เนื่องจากพารามิเตอร์ที่ต้องใช้ในขั้นตอนวิธี DBSM นั้นมีจำนวนมาก ซึ่งประกอบไปด้วยพารามิเตอร์จำนวน 4 ค่า คือ ระยะทางของจุดเพื่อนบ้าน จำนวนของจุดเพื่อนบ้านขั้นต่ำ ปริมาณของ SMOTE คิดเป็นเปอร์เซ็นต์ และจำนวนของเพื่อนบ้านใกล้เคียงที่สุด k ตัว นอกจากนี้ในแต่ละชุดข้อมูลฝึกสอนนั้นจะมีการกำหนดค่าพารามิเตอร์ที่แตกต่างกันออกไป เนื่องจากข้อมูลแต่ละชุดมีการกระจายตัวของข้อมูลที่ต่างกันทำให้การกำหนดพารามิเตอร์ด้วยมือ (manual) เป็นเรื่องที่ยาก ดังนั้นขั้นตอนวิธีเชิงพันธุกรรมจึงถูกนำมาใช้ในการอำนวยความสะดวกให้กับผู้ใช้ โดยมีขั้นตอนและรายละเอียดดังนี้

ขั้นตอนที่ 1 : แทนคำตอบของปัญหาด้วยโครโมโซมที่มีจำนวนยีนคงที่

รูปที่ 3.5 การเข้ารหัสโครโมโซม

จากรูปที่ 3.5 แสดงการออกแบบโครโมโซมโดยแทนคำตอบของปัญหาด้วยบิตสตริง ซึ่งประกอบด้วยตัวแปรจำนวน 4 ตัว คือ A B C และ D

โดยที่ A แทนด้วย จำนวนของเพื่อนบ้านใกล้เคียงที่สุด k ตัว (*k*) โดยมีความยาวบิตสตริงเท่ากับ 3

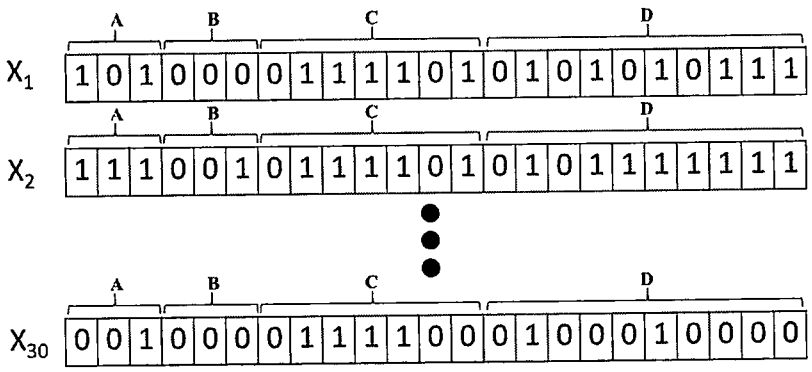
B แทนด้วย ปริมาณของการสุ่มตัวอย่างของ SMOTE ที่คิดเป็นเปอร์เซ็นต์ (*p*) โดยมีความยาวบิตสตริงเท่ากับ 3

C แทนด้วย จำนวนของจุดเพื่อนบ้านขั้นต่ำ (*Minpts*) โดยมีความยาวบิตสตริงเท่ากับ 7

D แทนด้วย ระยะทางของจุดเพื่อนบ้าน (*Eps*) โดยมีความยาวบิตสตริงเท่ากับ 10

ดังนั้นในหนึ่งโครโมโซมจะประกอบไปด้วยบิตสตริงทั้งหมด 23 บิต

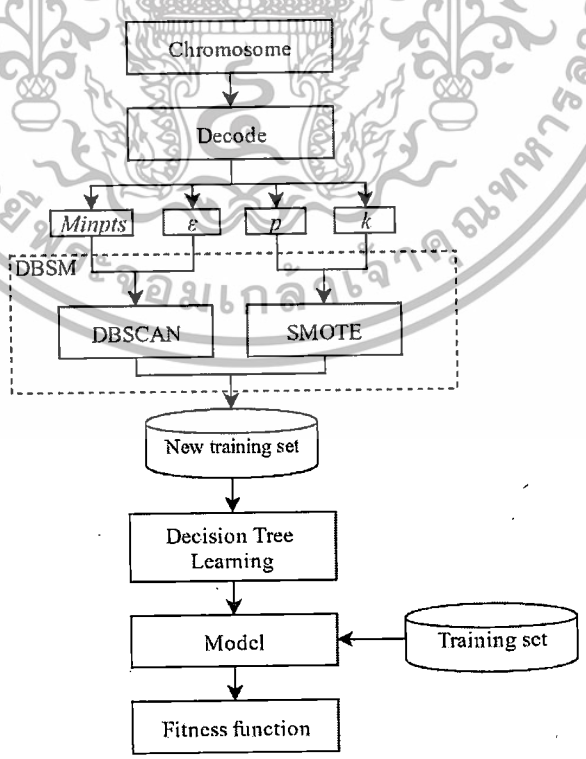
ขั้นตอนที่ 2 : สร้างประชากรในรุ่นแรกอย่างสุ่มให้มีจำนวนเท่ากับ N โดยขั้นตอนวิธี GADBSM ได้กำหนดจำนวนประชากรเท่ากับ 30 หรือ N=30 ดังนั้นประชากรในแต่ละรุ่นจะประกอบด้วยโครโมโซมทั้งหมด 30 โครโมโซม และแต่ละบิตในโครโมโซมจะถูกสุ่มด้วยเลขฐานสอง (0 หรือ 1) ดังรูปที่ 3.6



รูปที่ 3.6 การสร้างประชากรในรุ่นแรกอย่างสุ่มให้มีจำนวน N = 30

ขั้นตอนที่ 3 : วัดประสิทธิภาพของประชากรหรือคำนวณค่าฟังก์ชันความเหมาะของแต่ละโครโมโซม โดยตัวแปร A B C และ D ในแต่ละโครโมโซมจะถูกถอดรหัส (decode) ให้กลายเป็นเลขฐานสิบ จากนั้นค่าที่ได้หลังจากการถอดรหัสจะถูกใช้เป็นค่าของพารามิเตอร์ในขั้นตอนวิธี DBSM ที่ได้กล่าวมาแล้วในหัวข้อ 3.1 และชุดข้อมูลฝึกสอนที่ได้จากขั้นตอนวิธี DBMS จะถูกนำไปสร้างโมเดลจำแนกประเภทและวัดประสิทธิภาพของโมเดลด้วยชุดข้อมูลฝึกสอน ซึ่งมาตรวัดที่ใช้ในการวัดประสิทธิภาพของโมเดล คือ F-measure และ AUC ดังนั้นค่าฟังก์ชันความเหมาะสมของแต่ละโครโมโซมจะถูกกำหนดโดยสมการที่ 3.1 ซึ่งขั้นตอนการวัดประสิทธิภาพของประชากรสามารถแสดงได้ดังรูปที่ 3.7

$$f(x) = \text{F-measure}(x) + \text{AUC}(x) \tag{3.1}$$



รูปที่ 3.7 แสดงขั้นตอนการวัดประสิทธิภาพของประชากร

ขั้นตอนที่ 4 : เลือกคู่ของโครโมโซมที่จะมาผสมพันธุ์กันเพื่อผลิตลูก โดยโครโมโซมพ่อแม่จะถูกสุ่มขึ้นมาด้วยความน่าจะเป็นที่สอดคล้องกับค่าฟังก์ชันความเหมาะสมของแต่ละโครโมโซม ซึ่งการสุ่มเลือกโครโมโซมจะใช้เทคนิค roulette wheel selection ดังนั้นโครโมโซมที่มีค่าฟังก์ชันความเหมาะสมสูงมีจะโอกาสที่ถูกเลือกสูงกว่าโครโมโซมที่มีค่าค่าฟังก์ชันความเหมาะสมต่ำ

ขั้นตอนที่ 5 : สร้างโครโมโซมของลูกจากโครโมโซมพ่อและแม่โดยการใช้ตัวดำเนินการการไขว้เปลี่ยนและการกลายพันธุ์ โดยเลือกใช้ตัวดำเนินการการไขว้เปลี่ยนชนิด Uniform crossover ซึ่งกำหนดให้ $p_c = 0.7$ และ $p_m = 0.01$

ขั้นตอนที่ 6 : แทนประชากรรุ่นเก่าด้วยประชากรรุ่นใหม่ซึ่งเป็นโครโมโซมลูกที่ผลิตได้ทั้งหมด และกลับไปทำซ้ำในขั้นตอนที่ 3 จนกระทั่งเงื่อนไขในการวนซ้ำเป็นจริง

บทที่ 4
การออกแบบการทดลองและผลการทดลอง

ในบทนี้จะกล่าวถึงแหล่งที่มาและรายละเอียดของชุดข้อมูล การออกแบบการทดลอง และผลการทดลองของชุดข้อมูลที่ไม่ผ่านเทคนิคการสุ่มตัวอย่างและชุดข้อมูลที่ผ่านเทคนิคการสุ่มตัวอย่างทั้ง 4 เทคนิค คือ SMOTE Tomek Links SMOTE + Tomek Links และ DBSM โดยใช้อัลกอริทึมการเรียนรู้ต้นไม้ตัดสินใจ (classification and regression trees) เพื่อนบ้านใกล้เคียงที่สุด k ตัว (*k*NN) ตัวจำแนกแบบเบย์อย่าง่าย (NaiveBayes) และในส่วนสุดท้ายคือการเปรียบเทียบระหว่างเทคนิคการสุ่มตัวอย่างทั้ง 4 เทคนิค และเปรียบเทียบประสิทธิภาพของเทคนิค GADBSM ในแต่ละอัลกอริทึมการเรียนรู้

4.1 แหล่งที่มาและรายละเอียดของชุดข้อมูล

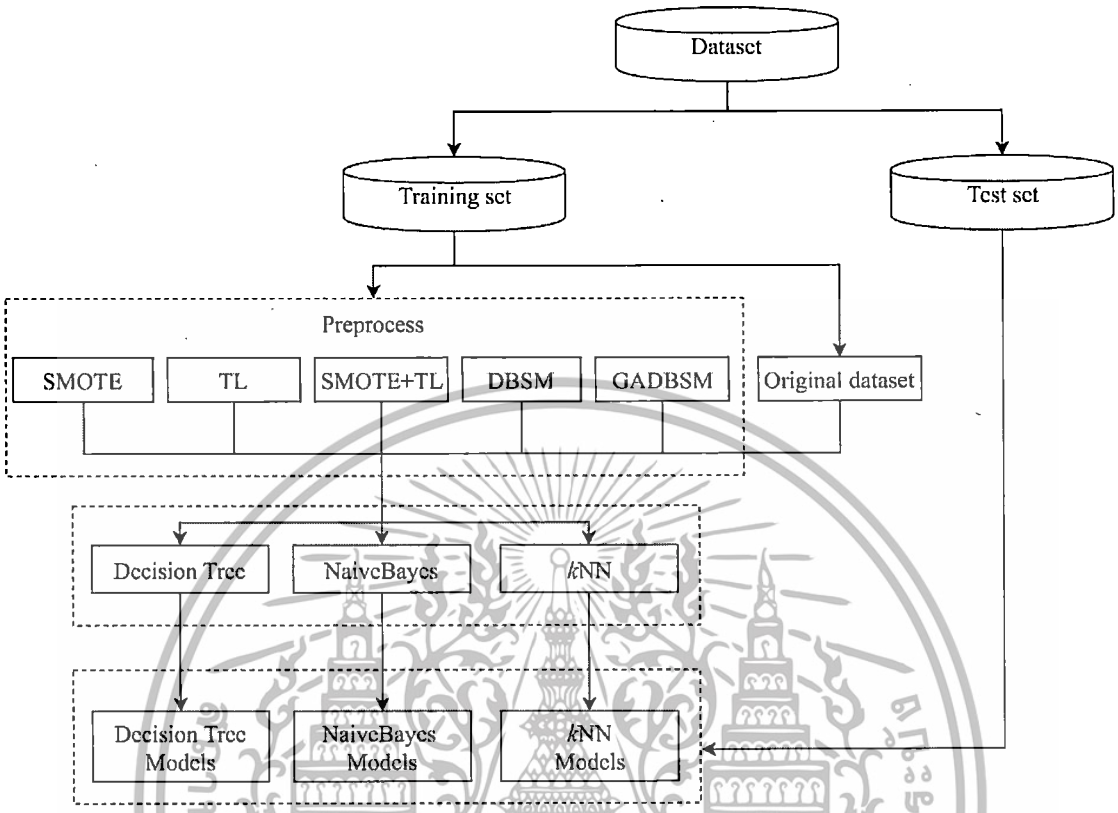
ในการทดลองนี้ได้นำข้อมูลมาจากเว็บไซต์ KEEL (www.keel.es) จำนวน 12 ชุดข้อมูล ชุดข้อมูลทั้งหมดที่นำมาทดลองมีค่าอัตราความไม่สมดุล (IR) อยู่ระหว่าง 1 ถึง 10 โดยรายละเอียดต่าง ๆ ของแต่ละชุดข้อมูลประกอบด้วย จำนวนคุณลักษณะ (attributes) จำนวนตัวอย่างทั้งหมดของแต่ละชุดข้อมูล(examples) และค่า IR แสดงได้ดังตารางที่ 4.1

ตารางที่ 4.1 รายละเอียดของชุดข้อมูล

Dataset Name	Attributes (R/I/N)	Examples	IR
glass1	9 (9/0/0)	214	1.82
wiscosin	9 (0/9/0)	683	1.86
glass0	9 (9/0/0)	214	2.06
yeast1	8 (8/0/0)	1484	2.46
haberman	3 (0/3/0)	306	2.78
vehicle2	18 (0/18/0)	846	2.88
vehicle1	18 (0/18/0)	846	2.9
new-thyroid1	5 (4/1/0)	215	5.14
new-thyroid2	5 (4/1/0)	215	5.14
ecoli2	7 (7/0/0)	336	5.46
glass6	9 (9/0/0)	214	6.38
yeast3	8 (8/0/0)	1484	8.1

จากตารางที่ 4.1 ในคอลัมน์ Attributes (R/I/N) R คือคุณลักษณะข้อมูลประเภทจำนวนจริง (real/continuous) I คือคุณลักษณะประเภทจำนวนเต็ม (integer) และ N คือคุณลักษณะข้อมูลประเภทคำหรือข้อความ (nominal/categorical)

4.2 การออกแบบการทดลอง



รูปที่ 4.1 ขั้นตอนการทดลอง

จากรูปที่ 4.1 แสดงขั้นตอนการทดลองโดยเริ่มต้นจะทำการแบ่งชุดข้อมูลโดยใช้เทคนิค *5*-fold Crossvalidation ซึ่งจะถูกแบ่งเป็นชุดข้อมูลฝึกสอนและชุดข้อมูลทดสอบ สำหรับชุดข้อมูลฝึกสอนนั้นใช้สำหรับสร้างโมเดลจำแนกประเภท ในการทดลองได้ใช้ขั้นตอนวิธี CART ซึ่งเป็นเทคนิคต้นไม้ตัดสินใจประเภทหนึ่ง ตัวจำแนกแบบเบย์อย่างง่าย และเพื่อนบ้านใกล้เคียงที่สุด k ตัวมาทำการสร้างโมเดลจำแนกประเภท และชุดข้อมูลทดสอบนั้นใช้สำหรับวัดประสิทธิภาพของโมเดลจำแนกประเภท ในส่วนของการเตรียมข้อมูลได้ใช้เทคนิคการสุ่มตัวอย่าง 3 เทคนิค คือเทคนิคการสุ่มตัวอย่างแบบลดจำนวนตัวอย่าง เทคนิคการสุ่มตัวอย่างแบบเพิ่มจำนวนตัวอย่าง และเทคนิคการสุ่มตัวอย่างแบบผสมผสาน ในงานวิจัยนี้เลือกใช้เทคนิค Tomek Links SMOTE และ SMOTE+Tomek Links ตามลำดับ และได้ทำการเปรียบเทียบประสิทธิภาพกับเทคนิคการสุ่มตัวอย่างด้วยขั้นตอนวิธี DBSM และ GADBSM ที่ได้พัฒนาขึ้น ดังนั้นโมเดลทั้งหมดที่ใช้ในการเปรียบเทียบประสิทธิภาพของการเตรียมข้อมูลด้วยเทคนิคการสุ่มตัวอย่างแบบต่าง ๆ จะประกอบไปด้วยต้นไม้ตัดสินใจ ตัวจำแนกแบบเบย์อย่างง่าย และเพื่อนบ้านใกล้เคียงที่สุด k ตัว ทั้งหมดอย่างละ 6 โมเดล คือ โมเดลที่ผ่านเทคนิคการสุ่มตัวอย่างด้วยขั้นตอนวิธี SMOTE โมเดลที่ผ่านเทคนิคการสุ่มตัวอย่างด้วยขั้นตอนวิธี TL โมเดลที่ผ่านเทคนิคการสุ่มตัวอย่างด้วยขั้นตอนวิธี SMOTE+TL โมเดลที่ผ่านเทคนิคการสุ่มตัวอย่างด้วย

ขั้นตอนวิธี DBSM โมเดลที่ผ่านเทคนิคการสุ่มตัวอย่างด้วยขั้นตอนวิธี GADBSM และโมเดลที่ไม่ผ่านเทคนิคการสุ่มตัวอย่างใด ๆ (original)

ในการทดลองได้มีการกำหนดพารามิเตอร์ต่าง ๆ ดังนี้

-เทคนิค SMOTE มีการกำหนดค่าพารามิเตอร์สองตัวคือ จำนวนเปอร์เซ็นต์ของการเพิ่มจำนวนตัวอย่างบวกและจำนวนเพื่อนบ้านใกล้เคียงที่สุด k ตัว ซึ่งกำหนดค่าพารามิเตอร์เป็น 100 เปอร์เซ็นต์และ 5 ตามลำดับ

-เทคนิค DBSM มีการกำหนดค่าพารามิเตอร์ของ Minpoints เท่ากับ 5 และ epsilon อยู่ในช่วง [0.001-1] ตามลำดับ

-เทคนิค GADBSM มีการกำหนดจำนวนประชากร รอบในการทำซ้ำ ความน่าจะเป็นในการเกิด crossover และ mutation เท่ากับ 100 500 0.7 และ 0.1 ตามลำดับ

-อัลกอริทึมการเรียนรู้เพื่อนบ้านใกล้เคียงที่สุด k ตัวมีการกำหนดค่าพารามิเตอร์ของ *k* เท่ากับ 3 และ 5

4.3 ผลการทดลอง

ตารางที่ 4.2 เปรียบเทียบค่า AUC ระหว่างชุดข้อมูลที่ไม่ผ่านเทคนิคการสุ่มตัวอย่างและชุดข้อมูลที่ผ่านเทคนิคการสุ่มตัวอย่างโดยใช้อัลกอริทึมการเรียนรู้ต้นไม้ตัดสินใจ

Dataset Name	Original	SMOTE	TL	SM+TL	DBSM	GADBSM
glass1	0.660	0.805	0.747	**0.888**	0.776	0.785
wiscosin	0.944	0.750	**0.963**	0.761	0.947	0.948
glass0	0.784	0.695	0.778	0.705	0.788	**0.799**
yeast1	0.669	0.848	0.680	**0.859**	0.651	0.653
haberman	0.532	0.589	**0.604**	0.584	0.555	0.555
vehicle2	0.941	0.935	**0.950**	0.932	0.944	0.944
vehicle1	0.652	**0.923**	0.678	0.920	0.705	0.705
new-thyroid1	0.909	0.698	0.909	0.704	0.949	**0.952**
new-thyroid2	0.932	0.943	0.904	0.946	0.963	**0.966**
ecoli2	0.810	0.937	0.873	**0.948**	0.883	0.841
glass6	0.848	0.664	0.850	0.672	0.860	**0.892**
yeast3	0.830	0.827	0.875	**0.877**	0.859	0.865
Average	0.793	0.801	0.818	0.816	0.823	**0.825**

จากตารางที่ 4.2 แสดงการเปรียบเทียบค่า AUC ระหว่างชุดข้อมูลที่ไม่ผ่านเทคนิคการสุ่มตัวอย่าง (original) และชุดข้อมูลที่ผ่านเทคนิคการสุ่มตัวอย่าง (SMOTE, TL, SM+TL, DBSM, GADBSM) โดยใช้อัลกอริทึมการเรียนรู้ต้นไม้ตัดสินใจ พบว่าค่าเฉลี่ยของ AUC เพิ่มขึ้นเมื่อใช้เทคนิคการสุ่มตัวอย่างและเทคนิคที่ให้ค่าเฉลี่ยของ AUC มากที่สุดคือเทคนิค GADBSM DBSM TL SM+TL และ SMOTE ตามลำดับ

ตารางที่ 4.3 เปรียบเทียบค่า F-measure ระหว่างชุดข้อมูลที่ไม่ผ่านเทคนิคการสุ่มตัวอย่างและชุดข้อมูลที่ผ่านเทคนิคการสุ่มตัวอย่างโดยใช้อัลกอริทึมการเรียนรู้ต้นไม้ตัดสินใจ

Dataset Name	Original	SMOTE	TL	SM+TL	DBSM	GADBSM
glass1	0.540	0.659	0.683	**0.810**	0.715	0.724
wiscosin	0.925	0.666	**0.945**	0.678	0.929	0.931
glass0	0.704	0.609	0.696	0.637	0.710	**0.719**
yeast1	0.528	**0.734**	0.551	0.728	0.509	0.511
haberman	0.287	0.397	**0.439**	0.411	0.378	0.378
vehicle2	0.909	0.901	0.909	0.885	0.915	**0.915**
vehicle1	0.482	**0.897**	0.519	0.884	0.557	0.559
new-thyroid1	0.836	0.550	0.833	0.555	0.916	**0.929**
new-thyroid2	0.886	0.909	0.819	0.907	0.930	**0.943**
ecoli2	0.700	0.918	0.771	**0.930**	0.752	0.754
glass6	0.734	0.524	0.746	0.541	0.770	**0.781**
yeast3	0.703	0.689	0.720	**0.744**	0.737	0.742
Average	0.686	0.705	0.719	0.726	0.735	**0.740**

จากตารางที่ 4.3 แสดงการเปรียบเทียบค่า F-measure ระหว่างชุดข้อมูลที่ไม่ผ่านเทคนิคการสุ่มตัวอย่าง (original) และชุดข้อมูลที่ผ่านเทคนิคการสุ่มตัวอย่าง (SMOTE, TL, SM+TL, DBSM, GADBSM) โดยใช้อัลกอริทึมการเรียนรู้ต้นไม้ตัดสินใจ พบว่าค่าเฉลี่ยของ F-measure เพิ่มขึ้นเมื่อใช้เทคนิคการสุ่มตัวอย่างและเทคนิคที่ให้ค่าเฉลี่ยของ F-measure มากที่สุดคือเทคนิค GADBSM DBSM SM+TL TL และ SMOTE ตามลำดับ

ตารางที่ 4.4 เปรียบเทียบค่า AUC ระหว่างชุดข้อมูลที่ไม่ผ่านเทคนิคการสุ่มตัวอย่างและชุดข้อมูลที่ผ่านเทคนิคการสุ่มตัวอย่างโดยใช้อัลกอริทึมการเรียนรู้เพื่อนบ้านใกล้เคียงที่สุด k ตัว (*k*=3)

Dataset Name	Original	SMOTE	TL	SM+TL	DBSM	GADBSM
glass1	0.749	0.929	0.765	**0.938**	0.768	0.771
wiscosin	0.969	0.825	**0.975**	0.819	0.970	0.973
glass0	0.803	0.793	**0.825**	0.783	0.802	0.804
yeast1	0.645	**0.855**	0.690	0.852	0.655	0.664
haberman	0.546	0.560	0.562	**0.591**	0.547	0.555
vehicle2	0.950	**0.977**	0.947	0.977	0.949	0.951
vehicle1	0.656	0.966	0.691	**0.994**	0.680	0.686
new-thyroid1	0.966	0.701	**0.980**	0.713	0.975	0.977
new-thyroid2	0.937	0.948	0.966	0.944	0.992	**0.992**
ecoli2	0.936	0.976	0.947	**0.978**	0.922	0.923
glass6	0.838	0.657	0.838	0.692	0.885	**0.885**
yeast3	0.830	0.856	0.858	**0.878**	0.871	0.871
Average	0.819	0.837	0.837	**0.847**	0.835	0.838

จากตารางที่ 4.4 แสดงการเปรียบเทียบค่า AUC ระหว่างชุดข้อมูลที่ไม่ผ่านเทคนิคการสุ่มตัวอย่าง (original) และชุดข้อมูลที่ผ่านเทคนิคการสุ่มตัวอย่าง (SMOTE, TL, SM+TL, DBSM, GADBSM) โดยใช้อัลกอริทึมการเรียนรู้เพื่อนบ้านใกล้เคียงที่สุด k ตัว (*k*=3) พบว่าค่าเฉลี่ยของ AUC เพิ่มขึ้นเมื่อใช้เทคนิคการสุ่มตัวอย่างและเทคนิคที่ให้ค่าเฉลี่ยของ AUC มากที่สุดคือเทคนิค SM+TL GADBSM TL SMOTE และ DBSM ตามลำดับ

ตารางที่ 4.5 เปรียบเทียบค่า F-measure ระหว่างชุดข้อมูลที่ไม่ผ่านเทคนิคการสุ่มตัวอย่างและชุดข้อมูลที่ผ่านเทคนิคการสุ่มตัวอย่างโดยใช้อัลกอริทึมการเรียนรู้เพื่อนบ้านใกล้เคียงที่สุด k ตัว (*k*=3)

Dataset Name	Original	SMOTE	TL	SM+TL	DBSM	GADBSM
glass1	0.668	**0.860**	0.698	0.843	0.700	0.705
wiscosin	0.957	0.749	**0.963**	0.736	0.950	0.959
glass0	0.728	0.731	**0.748**	0.720	0.713	0.719
yeast1	0.485	**0.802**	0.560	0.788	0.526	0.530
haberman	0.301	0.368	0.371	**0.421**	0.377	0.378
vehicle2	0.920	**0.945**	0.907	0.945	0.888	0.893
vehicle1	0.486	0.943	0.537	**0.973**	0.521	0.530
new-thyroid1	0.943	0.551	**0.958**	0.559	0.894	0.945
new-thyroid2	0.910	0.898	0.943	0.885	0.960	**0.960**
ecoli2	0.894	0.963	0.883	**0.965**	0.804	0.836
glass6	0.780	0.517	0.780	0.565	0.828	**0.828**
yeast3	0.721	0.717	**0.744**	0.736	0.709	0.726
Average	0.733	0.753	0.758	**0.761**	0.739	0.751

จากตารางที่ 4.5 แสดงการเปรียบเทียบค่า F-measure ระหว่างชุดข้อมูลที่ไม่ผ่านเทคนิคการสุ่มตัวอย่าง (original) และชุดข้อมูลที่ผ่านเทคนิคการสุ่มตัวอย่าง (SMOTE, TL, SM+TL, DBSM, GADBSM) โดยใช้อัลกอริทึมการเรียนรู้เพื่อนบ้านใกล้เคียงที่สุด k ตัว (*k*=3) พบว่าค่าเฉลี่ยของ F-measure เพิ่มขึ้นเมื่อใช้เทคนิคการสุ่มตัวอย่างและเทคนิคที่ให้ค่าเฉลี่ยของ F-measure มากที่สุดคือเทคนิค SM+TL TL SMOTE GADBSM และ DBSM ตามลำดับ

ตารางที่ 4.6 เปรียบเทียบค่า AUC ระหว่างชุดข้อมูลที่ไม่ผ่านเทคนิคการสุ่มตัวอย่างและชุดข้อมูลที่ผ่านเทคนิคการสุ่มตัวอย่างโดยใช้อัลกอริทึมการเรียนรู้ตัวจำแนกแบบเบย์อย่างง่าย

Dataset Name	Original	SMOTE	TL	SM+TL	DBSM	GADBSM
glass1	**0.717**	0.697	0.669	0.662	0.676	0.688
wiscosin	0.965	0.970	0.964	0.968	0.973	**0.978**
glass0	**0.731**	0.724	0.697	0.700	0.714	0.721
yeast1	0.525	0.567	0.566	**0.613**	0.609	0.611
haberman	0.415	0.411	0.507	0.442	0.617	**0.642**
vehicle2	0.839	0.855	0.837	0.856	0.871	**0.874**
vehicle1	0.667	0.676	0.671	0.670	0.679	**0.680**
new-thyroid1	**0.994**	0.989	0.994	0.989	0.989	0.992
new-thyroid2	**1.000**	0.994	1.000	0.994	0.994	0.997
ecoli2	0.822	0.856	0.859	0.854	0.911	**0.912**
glass6	0.860	0.830	**0.877**	0.860	0.802	0.808
yeast3	0.544	0.818	0.601	0.842	0.822	**0.871**
Average	0.757	0.782	0.770	0.788	0.805	**0.814**

จากตารางที่ 4.6 แสดงการเปรียบเทียบค่า AUC ระหว่างชุดข้อมูลที่ไม่ผ่านเทคนิคการสุ่มตัวอย่าง (original) และชุดข้อมูลที่ผ่านเทคนิคการสุ่มตัวอย่าง (SMOTE, TL, SM+TL, DBSM, GADBSM) โดยใช้อัลกอริทึมการเรียนรู้จำแนกแบบเบย์อย่างง่าย พบว่าค่าเฉลี่ยของ AUC เพิ่มขึ้นเมื่อใช้เทคนิคการสุ่มตัวอย่างและเทคนิคที่ให้ค่าเฉลี่ยของ AUC มากที่สุดคือเทคนิค GADBSM DBSM SM+TL SMOTE และ TL ตามลำดับ

ตารางที่ 4.7 เปรียบเทียบค่า F-measure ระหว่างชุดข้อมูลที่ไม่ผ่านเทคนิคการสุ่มตัวอย่างและชุดข้อมูลที่ผ่านเทคนิคการสุ่มตัวอย่างโดยใช้อัลกอริทึมการเรียนรู้ตัวจำแนกแบบเบย์อย่างง่าย

Dataset Name	Original	SMOTE	TL	SM+TL	DBSM	GADBSM
glass1	**0.645**	0.632	0.601	0.601	0.613	0.623
wiscosin	0.954	0.957	0.952	0.953	0.956	**0.963**
glass0	**0.644**	0.637	0.611	0.614	0.627	0.632
yeast1	0.169	0.391	0.357	0.478	0.492	**0.496**
haberman	0.193	0.335	0.293	0.357	0.475	**0.479**
vehicle2	0.760	0.782	0.752	0.778	0.791	**0.799**
vehicle1	0.505	0.517	0.510	0.510	0.520	**0.520**
new-thyroid1	**0.973**	0.950	0.973	0.950	0.950	0.962
new-thyroid2	**1.000**	0.975	1.000	0.975	0.975	0.987
ecoli2	0.747	0.777	0.797	0.771	0.810	**0.842**
glass6	0.758	0.732	**0.776**	0.767	0.671	0.689
yeast3	0.150	0.658	0.295	0.686	0.698	**0.727**
Average	0.625	0.695	0.660	0.703	0.715	**0.727**

จากตารางที่ 4.7 แสดงการเปรียบเทียบค่า F-measure ระหว่างชุดข้อมูลที่ไม่ผ่านเทคนิคการสุ่มตัวอย่าง (original) และชุดข้อมูลที่ผ่านเทคนิคการสุ่มตัวอย่าง (SMOTE, TL, SM+TL, DBSM, GADBSM) โดยใช้อัลกอริทึมการเรียนรู้ตัวจำแนกแบบเบย์อย่างง่าย พบว่าค่าเฉลี่ยของ F-measure เพิ่มขึ้นเมื่อใช้เทคนิคการสุ่มตัวอย่างและเทคนิคที่ให้ค่าเฉลี่ยของ F-measure มากที่สุดคือเทคนิค GADBSM DBSM SM+TL SMOTE และ TL ตามลำดับ

ตารางที่ 4.8 เปรียบเทียบค่าเฉลี่ยของค่า AUC ในเทคนิคการสุ่มตัวอย่างกับแต่ละอัลกอริทึมการเรียนรู้

	Original	SMOTE	TL	SM+TL	DBSM	GADBSM
Decision tree	0.793	0.801	0.818	0.816	0.823	**0.825**
k-Nearest Neighbor	0.819	0.837	0.837	**0.847**	0.835	0.838
NaiveBayes	0.757	0.782	0.770	0.788	0.805	**0.814**
Average	0.789	0.807	0.808	0.817	0.821	**0.826**

ตารางที่ 4.9 เปรียบเทียบค่าเฉลี่ยของค่า F-measure ในเทคนิคการสุ่มตัวอย่างกับแต่ละอัลกอริทึมการเรียนรู้

	Original	SMOTE	TL	SM+TL	DBSM	GADBSM
Decision tree	0.686	0.705	0.719	0.726	0.735	**0.740**
k-Nearest Neighbor	0.733	0.753	0.758	**0.761**	0.739	0.751
NaiveBayes	0.625	0.695	0.660	0.703	0.715	**0.727**
Average	0.681	0.718	0.712	0.730	0.730	**0.739**

จากตารางที่ 4.8 และ4.9 แสดงการเปรียบเทียบค่าเฉลี่ยของค่า AUC และ F-measure ในแต่ละเทคนิคการสุ่มตัวอย่างกับแต่ละอัลกอริทึมการเรียนรู้ พบว่าเทคนิคที่ให้ค่าเฉลี่ยสูงสุดทั้งค่า F-measure และ AUC คือ เทคนิค GADBSM

ตารางที่ 4.10 เปรียบเทียบเปอร์เซ็นความคืบหน้าสำหรับค่าเฉลี่ยของ AUC ในเทคนิค GADBSM กับเทคนิคการสุ่มตัวอย่างอื่นๆ ในแต่ละอัลกอริทึมการเรียนรู้

	Original	SMOTE	TL	SM+TL	DBSM	Average
Decision tree	4.14%	3.06%	0.93%	1.13%	0.24%	1.90%
k-Nearest Neighbor	2.34%	0.12%	0.10%	-1.04%	0.38%	0.38%
NaiveBayes	7.63%	4.10%	5.74%	3.39%	1.20%	**4.41%**

ตารางที่ 4.11 เปรียบเทียบเปอร์เซ็นความคืบหน้าสำหรับค่าเฉลี่ยของ F-measure ในเทคนิค GADBSM กับเทคนิคการสุ่มตัวอย่างอื่นๆ ในแต่ละอัลกอริทึมการเรียนรู้

	Original	SMOTE	TL	SM+TL	DBSM	Average
Decision tree	7.90%	5.10%	2.95%	2.02%	0.76%	3.75%
k-Nearest Neighbor	2.48%	-0.36%	-0.90%	-1.38%	1.57%	0.28%
NaiveBayes	16.30%	4.53%	10.12%	3.32%	1.65%	**7.18%**

จากตารางที่ 4.10 และ 4.11 แสดงการเปรียบเทียบเปอร์เซ็นความคืบหน้าสำหรับค่าเฉลี่ยของ F-measure ในเทคนิค GADBSM กับเทคนิคการสุ่มตัวอย่างอื่นๆ ในแต่ละอัลกอริทึมการเรียนรู้ พบว่า เทคนิคการสุ่มตัวอย่าง GADBSM สามารถเพิ่มประสิทธิภาพการทำนายของโมเดลได้สูงสุดในอัลกอริทึมตัวจำแนกแบบเบย์อย่างง่าย ต้นไม้ตัดสินใจ และเพื่อนบ้านใกล้เคียงที่สุด k ตัว ตามลำดับด้วยมาตรวัด AUC และสำหรับมาตรวัด F-measure เทคนิค GADBSM สามารถเพิ่มประสิทธิภาพการทำนายของโมเดลได้สูงสุดในอัลกอริทึมตัวจำแนกแบบเบย์อย่างง่าย ต้นไม้ตัดสินใจ และเพื่อนบ้านใกล้เคียงที่สุด k ตัว ตามลำดับ

เนื่องจากอัลกอริทึมการเรียนรู้เพื่อนบ้านใกล้เคียงที่สุด k ตัว เป็นอัลกอริทึมจำแนกประเภทข้อมูลแบบที่ไม่มีการสร้างโมเดลเพื่อนำไปใช้ในการทำนายเหมือนกับอัลกอริทึมอื่นๆ ซึ่งการจำแนกประเภทข้อมูลของเพื่อนบ้านใกล้เคียงที่สุด k ตัว จะจำแนกข้อมูลโดยอ้างอิงจากข้อมูลที่ใกล้ที่สุดจำนวน k ตัว เมื่อพิจารณาเทคนิค GADBSM มีความเป็นไปได้ที่จะลบประเภทข้อมูลส่วนมากออกเป็นจำนวนมากเกินไป ทำให้มีแนวโน้มที่จะทำนายข้อมูลผิดพลาด ดังนั้นเทคนิค GADBSM จึงไม่สามารถเพิ่มประสิทธิภาพได้สูงที่สุดเมื่อเทียบกับเทคนิคการสุ่มตัวอย่าง SMOTE TL และ SMOTE + TL

บทที่ 5
สรุปผลการทดลองและข้อเสนอแนะ

5.1 สรุปผลการทดลอง

งานวิจัยนี้มีจุดมุ่งหมายเพื่อพัฒนาอัลกอริทึมการสุ่มตัวอย่างแบบผสมวิธีใหม่ที่มีประสิทธิภาพในการแก้ปัญหาความไม่สมดุลของกลุ่มข้อมูล โดยใช้ชื่อว่าอัลกอริทึม DBSM และ GADBSM ซึ่งเทคนิค DBSM เป็นเทคนิคการผสมระหว่างเทคนิคการเพิ่มจำนวนตัวอย่างด้วยอัลกอริทึม SMOTE และเทคนิคการลดจำนวนตัวอย่างด้วยอัลกอริทึม DBSCAN แต่เนื่องจากอัลกอริทึม DBSM เป็นเทคนิคที่ประกอบด้วยอลักอริทึม SMOTE และ DBSCAN จึงทำให้จำนวนพามิเตอร์มีจำนวนมาก ซึ่งการกำหนดพารามิเตอร์ด้วยมือ (manual) จึงเป็นเรื่องที่ค่อนข้างลำบากสำหรับผู้ใช้ในการเลือกค่าพารามิเตอร์ที่ทำให้โมเดลมีประสิทธิภาพในการคัดเลือกตัวอย่างที่ดีหรือลดจำนวนตัวอย่างที่ไม่จำเป็นในการสร้างชุดข้อมูลฝึกสอน ดังนั้นขั้นตอนวิธีเชิงพันธุกรรมจึงถูกนำมาประยุกต์ใช้ในการแก้ปัญหาในส่วนนี้ ซึ่งใช้ชื่อว่าอัลกอริทึม GADBSM

สำหรับขั้นตอนการดำเนินงานวิจัยเริ่มต้นจากการศึกษาพฤติกรรมความไม่สมดุลของข้อมูล (class imbalanced) และอัลกอริทึมการเรียนรู้ ได้แก่ ต้นไม้ตัดสินใจ (classification and regression trees) เพื่อนบ้านใกล้เคียงที่สุด k ตัว และตัวจำแนกแบบเบย์อย่างง่าย โดยในงานวิจัยนี้ได้เปรียบเทียบเทคนิค DBSM และ GADBSM กับเทคนิคการสุ่มตัวอย่างอื่นๆ จำนวน 3 เทคนิค ได้แก่ อัลกอริทึม Tomek Links เป็นเทคนิคการลดจำนวนข้อมูลของกลุ่มข้อมูล อัลกอริทึม SMOTE เป็นเทคนิคการเพิ่มจำนวนข้อมูลของกลุ่มข้อมูล และอัลกอริทึม SMOTE + Tomek Links ซึ่งเป็นเทคนิคผสมผสานระหว่างเทคนิคการลกจำนวนตัวอย่างและเทคนิคการเพิ่มจำนวนตัวอย่าง โดยชุดข้อมูลที่ใช้ในการทดลองนำมาจากเว็บไซต์ KEEL จำนวน 12 ชุดข้อมูล ซึ่งแต่ละชุดข้อมูลมีค่าอัตราความไม่สมดุล (IR) อยู่ระหว่าง 1 ถึง 10

จากผลการทดลองที่นำชุดข้อมูลที่มาผ่านเทคนิคการสุ่มตัวอย่าง พบว่าทั้ง 5 เทคนิคสามารถลดปัญหาความไม่สมดุลของข้อมูลลงได้ในทุก ๆ อัลกอริทึมการเรียนรู้ โดยที่เทคนิค GADBSM สามารถเพิ่มประสิทธิภาพการทำนายของโมเดลได้สูงที่สุดเมื่อใช้กับอัลกอริทึมการเรียนรู้ตัวจำแนกแบบเบย์อย่างง่ายและต้นไม้ตัดสินใจ ตามลำดับ ซึ่งสามารถเพิ่มประสิทธิภาพในอัลกอริทึมการเรียนรู้ตัวจำแนกแบบเบย์อย่างง่ายได้ถึง 7.18% และ 4.41% ด้วยมาตรวัด F-measure และ AUC ตามลำดับ

5.2 ปัญหาและข้อเสนอแนะ

- เนื่องจากเทคนิค GADBSM เป็นการประยุกต์ใช้ขั้นตอนวิธีเชิงพันธุกรรม ส่งผลให้ขั้นตอนการปรับพารามิเตอร์ใช้เวลาในการหาคำตอบค่อนข้างนาน ดังนั้นการปรับปรุงขั้นตอนวิธีให้มีการทำงานแบบขนานจะช่วยให้ขั้นตอนในการปรับพารามิเตอร์ใช้เวลาลดลง

เอกสารอ้างอิง

[1] ปองภพ ขัมภรัตน์ และศิรินญา จันทร์แดง. 2555. “กรณีศึกษาทัศนคติที่มีต่อโรงแรม (Opinion Classification based on Naïve Bayes Classifier: a Case Study of Hotel Review)." โครงงานพิเศษวิทยาศาสตรบัณฑิต สาขาวิทยาการคอมพิวเตอร์, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. หน้า 6-7.

[2] Daniel T. Larose “Discovering Knowledge in Data: An Introduction to Data Mining.” หน้า 110-113.

[3] ยศธร สงวนมาก กรวรรณ สุวรรณเบญจพล และวิภาวี พัฒนกิจวิบูลย์ “การศึกษาเชิงทดลองของปัญหาความไม่สมดุลของกลุ่มข้อมูลในการทำเหมืองข้อมูล” โครงงานปัญหาพิเศษวิทยาศาสตรบัณฑิต สาขาวิทยาการคอมพิวเตอร์, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

[4] ธนพร ภัทรลิขิตสกุล และน้ำทิพย์ ศิลป์สุวรรณ์ “การเพิ่มประสิทธิภาพการจำแนกประเภทของโครงข่ายประสาทเทียมโดยใช้การจัดกลุ่มแบบ DBSCAN (Enhancing Classification Performance of Artificial Neural Network by Using DBSCAN Clustering).” หน้า 3-8.

[5] อนันตพร หรรษคุณาฒัย “ขั้นตอนวิธีเชิงพันธุกรรม” หนังสือเรียนวิชา AI สาขาวิทยาการคอมพิวเตอร์, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

[6] Nitesh, V. Chawla. Kevin, W. Bowyer. Lawrence, O. Hall. and Philip, W. Kegelmeyer. “SMOTE: Synthetic Minority Over-sampling Technique.” Journal of Artificial Intelligence Research 16. pp. 329 –330. 2002.

[7] Tomek, I., “Two modifications of CNN,” IEEE Trans. Systems, Man and Cybernetics, vol. SMC-6, pp. 769-772, Nov. 1976.

[8] G. Batista, R. C. Prati, and M. C. Monard, “A study of the behavior of several methods for balancing machine learning training data,” ACM SIGKDD Explor. Newslett., vol. 6, no. 1, pp. 20–29, 2004.

[9] Janjira Jojan and Anongnart Srivihok. “Preprocessing Of Imbalanced Breast Cancer Data Using Feature Selection Combined With Over-Sampling Technique For Classification.” International Conference on Advanced Computer Science and Information Systems 2013. Bangkok: Kasetsart University. pp. 407-412. Sept. 2013.

[10] Ginny, Y. Wong. Frank, H.F. Leung. Sai-Ho Ling. “A Novel Evolutionary Preprocessing Method Based On Over-Sampling and Under-Sampling for Imbalanced Datasets.” IECON 2013 - 39th Annual Conference of the IEEE. Hong Kong: Hong Kong Polytechnic University. pp. 2354 – 2359. 2013.

[11] P. Sarakit, T. Theeramunkong, and C. Haruechaiyasak, "Improving emotion classification in imbalanced YouTube dataset using SMOTE algorithm," 2015 2nd International Conference on Advanced Informatics: Concepts, Theory and Applications (ICAICTA), 2015.

[12] K. Jiang, J. Lu, and K. Xia, "A Novel Algorithm for Imbalance Data Classification Based on Genetic Algorithm Improved SMOTE," Arabian Journal for Science and Engineering, Volume 41, pp 3255–3266., 201

ภาคผนวก

2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE)

DBSM: The Combination of DBSCAN and SMOTE for Imbalanced Data Classification

Yotsathon Sanguanmak
Department of Computer Science, King Mongkut's Institute of Technology Ladkrabang, Ladkrabang, Bangkok, 10520, Thailand

Anantaporn Hanskunatai
Department of Computer Science, King Mongkut's Institute of Technology Ladkrabang, Ladkrabang, Bangkok, 10520, Thailand

***Abstract*—Many applications in the real world encounter the class imbalance problem. This problem affects the performance of the model prediction. Nowadays, resampling technique is a popular technique to handle the class imbalance problem such as oversampling, undersampling, and hybridsampling. Thus, this paper proposes a new hybrid resampling technique to deal with the class imbalance problem, called DBSM. The concept of DBSM is to use DBSCAN algorithm for undersampling and apply SMOTE technique for oversampling. The experimental results of the DBSM algorithm are compared with an original datasets and other sampling techniques, which are SMOTE, Tomek Links, SMOTE+Tomek Links and DBSCAN. The results show that the DBSM can improve the predictive performance of the classifiers. In addition, it yields the best in the average of AUC, F-measure, and accuracy.**

Keywords- imbalanced dataset; hybridsampling; SMOTE; DBSCAN

I. INTRODUCTION

In recent years, the study of data mining has been focusing on imbalanced dataset. The class imbalanced problem occurs when representative samples of one class is that less than the other classes, the class with less number is called "minority class" and other classes are called "majority class". This problem affects model efficiently in classification. For example, cancer diagnosis that has number of cancer patients far less than others. Thus the model has a very high error rate and the model might be predicted incorrectly from cancer patients to normal. As a result, cancer patients haven't been treated correctly. Another example, oil spill detection can be raised from natural or accidental cause. As a consequence, lives in the ocean are dead due to oxygen being depleted. For the credit risk, it is important to detect "bad" arise from borrowers failing payment which has less than a normal case. Furthermore, the real world imbalanced problem are weather forecast dataset, spam-mail dataset, earthquake dataset. Until now, the class imbalance problem can be solved by three levels. First, algorithm level is a modification of classifier technique than can enhance a performance of a model. Second, data level is rebalancing the class distribution technique in the dataset. Finally, cost-sensitive is both of the incorporate algorithm level and data level.

One of the most popular techniques for solving the imbalanced dataset is resampling technique which is in the data level category. Resampling can be categorized into three groups. First, decreasing majority classes to approximately equivalent minority class is called undersampling technique. Second, increasing minority classes to approximately equivalent majority class by creating synthetic instances is called oversampling technique. The last, combining both of decreasing majority class and increasing minority class together is called hybridsampling. The aim of this paper is to propose a new hybridsampling technique by using both of the SMOTE technique and the DBSCAN algorithm to enhance a predictive performance of imbalanced data classification.

II. RELATED WORKS

The model prediction is more bias toward the normal class than the extraordinary attentiveness class due to a number of extraordinary attentiveness class (minority class) less than the normal class (majority class). This problem called the class imbalance problem is one of the ten challenging problems in data mining [1]. Thus, a lot of researchers propose many techniques to deal with the class imbalance problem. M. Galar et al [2] presents the solution of imbalance problem by dividing into three levels, algorithm level, data level, and cost sensitive.

For undersampling technique, random under sampling (RUS) is random elimination of some of majority class instances in order to rebalance classes distribution in dataset. In RUS, it is a nonheuristic method for rebalancing the classes by randomly discard example from majority class instances in dataset. This approach achieves rebalancing classes distribution and faster learning. However, the drawback of RUS may cause loss in valuable information from randomly eliminate majority class that may be important pattern to a classifier. Hence, a heuristic method such as Tomek Links algorithm [3] may be a better way. The main idea of Tomek Links is searching a minimum distance between minority class instance and majority class instance then removing majority class only or removing noise and both classes depend on undersampling method or cleaning method.

For oversampling technique, the random over sampling (ROS) is an algorithm for increasing size of minority class instance in order to rebalance class distribution in a dataset. In

978-1-5090-2033-1/16/$31.00 ©2016 IEEE

978-1-5090-2033-1/16/$31.00 ©2016 IEEE

ROS, it is a nonheuristic method for rebalancing the classes by randomly duplicating minority class instances. Nevertheless, the drawback of ROS may cause overfitting. Therefore, a heuristic method such as SMOTE algorithm [4] may be a smarter approach. The concept of SMOTE is to create new synthetic instances throughout on the line of minority class instances and the size of synthetic instances depends on percentage chosen by a user.

SMOTE + Tomek Links [5] is a hybridsampling technique. The main idea of SMOTE + Tomek Links is to increase minority class instances by using SMOTE algorithm to create new synthetic instances and decrease majority class instances by using Tomek Links algorithm to eliminate some of majority class instances.

There are many papers that applied SMOTE technique for real world applications. J. Jojan and A. Srivihok [6] combine feature selection technique with SMOTE, called "FOT". For the FOT approach, it uses feature selection technique for removing irrelevant attributes and applies SMOTE for rebalancing data on the breast cancer dataset.

Moreover, P. Sarakit et al [7] improve the classification performance of emotion datasets in YouTube by using SMOTE technique. R. I. Rashu et al [8] enhance the performance of model to predict final grade of student by using resampling techniques: SMOTE, RUS, and ROS. The experimental results show that the SMOTE yields the highest accuracy. Furthermore, T. M. Padmaja et al [9] use the combination of SMOTE and other techniques for fraud detection. For another application, G. Batista et al [10] use SMOTE + Tomek Links to deal with class imbalance of annotation of protein in bioinformatics.

III. THE DBSM ALGORITHM

This paper proposes a new hybridsampling algorithm called DBSM. The concept of the DBSM algorithm is using DBSCAN algorithm [11] for undersampling and applying SMOTE technique for oversampling. These two methods (DBSCAN and SMOTE) are separately run by themselves using the same training set. Finally, the outputs of both techniques are combined to form a new training dataset.

The flowchart of DBSM algorithm is shown in Fig. 1. The process of DBSM consists of two parts: undersampling, and oversampling. For the undersampling part, DBSCAN was applied to create clusters from all training set. Then 50% of negative instances were eliminated from each cluster. The outputs of the undersampling technique are only negative instances. For the oversampling part, synthetic instances of positive class were added into the training set using SMOTE. There are two parameters of the SMOTE algorithm which are amount of synthetic samples and the number of nearest neighbors. In this work the first parameter was set to 100% and the second parameter was set to 5. Therefore, the final output of the DBSM algorithm is the new training set that consists of only negative instances from the undersampling part and all positive instances from the oversampling part.

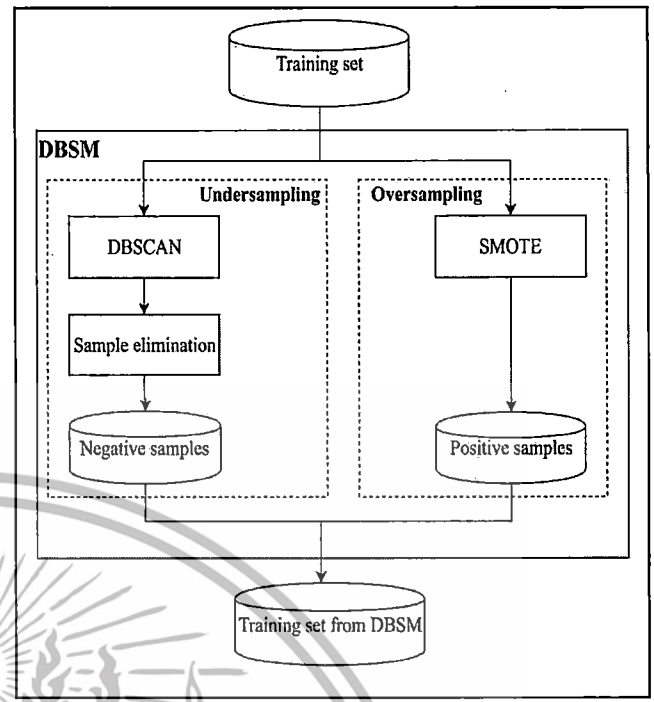


Figure 1. Flowchart of DBSM algorithm.

Algorithm: Undersampling
Input: S: All training set with positive instances and negative instances, ε: Epsilon, Minpts: Minpoints
Output: D: remaining negative instances.

```
1.  [Cluster] = DBSCAN(S, ε, Minpts) // Cluster is a set of clusters
2.  For i = 1 to n // n is a number of clusters generated by DBSM.
3.    Let Di be a number of negative instances in ith cluster
4.    If all members in Clusteri are negative instance then
5.      Centroid = compute_centroid(Clusteri)
6.      For j = 1 to m // m is a number of clusteri's members
7.        MI = calDistance(Centroid, j)
8.      End for
9.    Else if member in Clusteri are positive and negative instances then
10.     For j = 1 to m1 // m1 is negative instances
11.       For k = 1 to m2 // m2 is positive instances
12.         MI = findSmallestDistance(j, k)
13.       End for
14.     End for
15.   End if
16.   MI = sortingDistance(MI)
17.   Di = removeSmallestDistance(MI, 50%)
18.   return D = ∪ Di
19. End for
```

Figure 2. Pseudo code of undersampling technique.

Fig. 2 shows the pseudo code of undersampling algorithm. The algorithm consists of three phases: clustering, distance measuring, and instance removing. In the first phase (line 1), the algorithm finds all clusters from all training set by using the DBSCAN algorithm. In the second step (line 2-15), there are two conditions of distance calculation. If all members of a cluster are negative, then find a centroid of the cluster and computes distances between all negative instances and the centroid (line 4-8). On the other hand, if there are positive and negative instances in the same cluster, then calculate distances between negative instance and overall positive instances and then assign the smallest distance into MI (line 9-15). In the last phase (line 16-17), sorting all distances in an ascending order and removing top 50 percentage of negative instances in the rank. The outputs of the undersampling are the negative instances in the remaining of instances in each cluster (line 18).

IV. Experiment and Results

A. Datasets

In the experiment, twelve datasets from KEEL [12] were used to evaluate the classification models. These datasets are two-class classification problem. Table I shows the characteristic of all datasets used in the experiment. Attribute (R/I/N) is type of dataset which are real, integer, and nominal. Examples represent the number of instances. Imbalance ratio (IR) is a number of positive instances compare with negative instances which is calculated by (1).

$$\text{imbalance ratio} = \frac{\text{the number of positive instances}}{\text{the number of negative instances}} \quad (1)$$

B. Experimental Design

This section explains about our experimental design (shown in Fig. 3). First, the datasets are splited to the training set and the test set by using *k*-fold cross validation technique. In this experiment, *k* was set to 5. Next step is data preprocessing which are divided into five cases separately according to five sampling algorithms which are SMOTE, Tomek Links (TL), SMOTE+Tomek Links (SM+TL), DBSCAN, and DBSM. In the experiment, DBSCAN is considered as an undersampling method. The process of DBSCAN for undersampling is shown in Fig. 2. Thus in each fold there are six training datasets generated from five different sampling methods including an original training data. After that, each training dataset is used to construct a decision tree model. Finally, there are six decision tree models in each fold, and these models are evaluated for classification performance by using the same test set.

C. Experimental Results

This section presents the experimental results. The area under the ROC curve [13] (called AUC) and F-measure are used to evaluate the predictive performance of various sampling method including the DBSM algorithm.

AUC is a measure to evaluate performance of the single model based on True Positive rate (TPrate) and False Positive rate (FPrate). TPrate is a percentage of positive instance correctly predicted and FPrate is a percentage of negative instance incorrectly predicted. Therefore, the perfectly model is high percentage of positive correctly predicted and low percentage of negative incorrectly predicted. The formula of AUC is shown in (2).

TABLE I. Dataset Characteristics

Dataset Name	Attributes (R/I/N)		Examples	IR
glass1	9	(9/0/0)	214	1.82
wiscosin	9	(0/9/0)	683	1.86
glass0	9	(9/0/0)	214	2.06
yeast1	8	(8/0/0)	1484	2.46
haberman	3	(0/3/0)	306	2.78
vehicle2	18	(0/18/0)	846	2.88
vehicle1	18	(0/18/0)	846	2.9
new-thyroid1	5	(4/1/0)	215	5.14
new-thyroid2	5	(4/1/0)	215	5.14
ecoli2	7	(7/0/0)	336	5.46
glass6	9	(9/0/0)	214	6.38
yeast3	8	(8/0/0)	1484	8.1

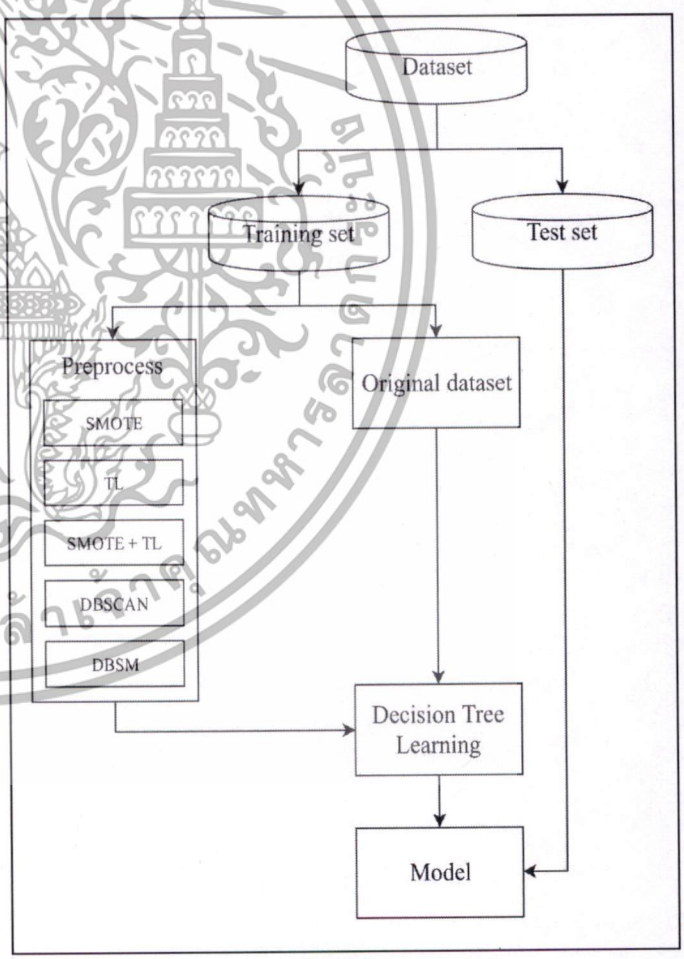


Figure 3. Design of the experiment.

$$AUC = \frac{1+TP_{rate}-FP_{rate}}{2} \quad (2)$$

F-measure [14] is a measurement for evaluating a model based on precision and recall where recall is TP rate described previously. Precision is a proportion of the positive instances that correctly classified as positive instances and classified by a model as positive class (calculated by 3). Finally, F-measure is calculated by (4).

$$precision = \frac{TP}{TP + FP} \quad (3)$$

$$F\text{-measure} = \frac{2 * Recall * Precision}{Recall + Precision} \quad (4)$$

Table II shows the AUC of the original datasets compared with five resampling techniques (SMOTE, TL, SM+TL, DBSCAN and DBSM) using decision tree classifiers on twelve imbalanced datasets. The experimental results show that the average of AUC for all resampling techniques are better than the original dataset, and the highest performance (an average of AUC) is DBSM. In addition, the DBSM has the best predictive performance on nine datasets. In summary, the DBSM technique enhances the predictive performance of positive instances when compare with SMOTE+TL, TL, SMOTE, original and DBSCAN upto 8.50% 7.84% 8.60% 17.70% and 18.75% respectively (display in Table V).

Moreover, Table III shows the F-measure of the original compared with five resampling techniques. The DBSM has the best performance and improve the performance when compared with SMOTE+TL, TL, SMOTE, DBSCAN and original upto 13.85% 16.23% 24.42% 38.10% and 62.62% respectively (show in Table VI).

Table IV shows the percentage of the accuracy of the original datasets compared with five resampling techniques. The DBSM also yields the best average in accuracy.

TABLE II. THE COMPARISON OF AUC BETWEEN THE ORIGINAL DATASETS AND ALL DATASETS USING RESAMPLING TECHNIQUES

Dataset Name	Original	SMOTE	TL	SM+TL	DBSCAN	DBSM
glass1	0.6605	0.758	0.7476	0.7661	0.7295	**0.7774**
wiscosin	0.9441	0.9452	**0.9628**	0.9628	0.9537	0.9528
glass0	0.7838	0.7881	0.7779	0.7603	0.7804	**0.7882**
yeast1	0.6688	0.666	0.6805	0.6924	0.6599	**0.6989**
haberman	0.532	0.5734	0.6044	0.6403	0.5244	**0.6227**
vehicle2	0.9415	0.9469	**0.9495**	0.9444	0.9418	0.948
vehicle1	0.6523	0.6883	0.6785	0.6729	0.7104	**0.7301**
new-thyroid1	0.9091	**0.9373**	0.9091	0.9202	0.8986	0.9338
new-thyroid2	0.9317	0.929	0.9036	0.9290	0.9277	**0.9744**
ecoli2	0.8097	0.8555	0.8735	0.8817	0.8666	**0.894**
glass6	0.8477	0.8423	0.8505	0.8977	0.8505	**0.9032**
yeast3	0.8297	0.8547	0.8755	0.8764	0.8618	**0.8755**
Average	0.7926	0.8154	0.8178	0.8287	0.8088	**0.8416**

TABLE III. THE COMPARISON OF F-MEASURE BETWEEN THE ORIGINAL DATASETS AND ALL DATASETS USING RESAMPLING TECHNIQUES

Dataset Name	Original	SMOTE	TL	SM+TL	DBSCAN	DBSM
glass1	0.5401	0.6876	0.6834	0.6993	0.6612	**0.7102**
wiscosin	0.9248	0.9268	0.9451	**0.9451**	0.9332	0.9315
glass0	0.7037	**0.7117**	0.6962	0.6789	0.6986	0.7117
yeast1	0.528	0.5265	0.5506	0.5645	0.5276	**0.5732**
haberman	0.2873	0.3755	0.4389	**0.4854**	0.3383	0.4672
vehicle2	0.909	0.9158	0.9094	0.9102	0.9091	**0.9194**
vehicle1	0.482	0.5318	0.5191	0.5108	0.5597	**0.5816**
new-thyroid1	0.8364	0.9113	0.8327	0.8831	0.8515	**0.9139**
new-thyroid2	0.8857	0.8768	0.819	0.8768	0.8877	**0.9519**
ecoli2	0.7	0.7363	0.7706	**0.7876**	0.759	0.7679
glass6	0.7345	0.7154	0.746	0.7993	0.7466	**0.8296**
yeast3	0.7034	0.7401	0.7201	**0.7531**	0.7244	0.7503
Average	0.6862	0.7213	0.7192	0.7412	0.7164	**0.759**

TABLE IV. THE COMPARISON OF ACCURACY BETWEEN THE ORIGINAL DATASETS AND ALL DATASETS USING RESAMPLING TECHNIQUES

Dataset Name	Original	SMOTE	TL	SM+TL	DBSCAN	DBSM
glass1	70.13%	77.11%	74.33%	76.63%	71.69%	**76.65%**
wiscosin	94.73%	94.88%	96.05%	**96.05%**	95.20%	95.08%
glass0	**81.75%**	80.35%	78.53%	76.67%	81.28%	80.35%
yeast1	**73.65%**	71.29%	70.01%	69.74%	66.98%	68.80%
haberman	66.01%	**66.32%**	65.00%	65.00%	58.36%	63.82%
vehicle2	95.27%	95.63%	95.15%	95.27%	95.27%	**95.84%**
vehicle1	74.11%	**74.58%**	72.45%	71.51%	72.79%	73.69%
new-thyroid1	94.42%	97.21%	94.42%	96.28%	95.35%	**97.40%**
new-thyroid2	96.28%	95.81%	93.49%	95.81%	96.37%	**98.42%**
ecoli2	91.67%	91.37%	92.86%	**93.16%**	92.44%	91.91%
glass6	93.01%	92.08%	93.47%	94.40%	93.48%	**95.34%**
yeast3	93.67%	94.27%	93.20%	**94.34%**	93.61%	94.21%
Average	85.39%	85.91%	84.91%	85.40%	84.40%	**85.96%**

TABLE V. THE IMPROVEMENT OF THE DBSM ALGORITHM IN AUC

Dataset Name	Original	SMOTE	TL	SM+TL	DBSCAN
glass1	**17.70%**	2.56%	3.99%	1.47%	6.57%
wiscosin	0.92%	0.80%	-1.04%	-1.04%	-0.09%
glass0	0.56%	0.01%	1.32%	3.67%	1.00%
yeast1	4.50%	4.94%	2.70%	0.95%	5.91%
haberman	17.05%	**8.60%**	3.03%	-2.75%	**18.75%**
vehicle2	0.69%	0.12%	-0.16%	0.38%	0.66%
vehicle1	**11.93%**	**6.07%**	**7.61%**	**8.50%**	2.77%
new-thyroid1	2.72%	-0.37%	2.72%	1.47%	3.92%
new-thyroid2	4.58%	4.89%	**7.84%**	4.89%	5.03%
ecoli2	**10.41%**	4.50%	2.35%	1.40%	3.16%
glass6	6.55%	7.23%	6.20%	0.61%	6.20%
yeast3	5.52%	2.43%	0.00%	-0.11%	1.59%

TABLE VI. THE IMPROVEMENT OF THE DBSM ALGORITHM IN F-MEASURE

Dataset Name	Original	SMOTE	TL	SM+TL	DBSCAN
glass1	31.49%	3.29%	3.92%	1.55%	7.41%
wiscosin	0.72%	0.51%	-1.44%	-1.44%	-0.18%
glass0	1.14%	0.00%	2.23%	4.83%	1.88%
yeast1	8.56%	8.87%	4.10%	1.55%	8.64%
haberman	**62.62%**	**24.42%**	6.45%	-3.75%	**38.10%**
vehicle2	1.14%	0.39%	1.10%	1.01%	1.13%
vehicle1	20.66%	9.36%	12.04%	**13.85%**	3.91%
new-thyroid1	9.27%	0.29%	9.75%	3.49%	7.33%
new-thyroid2	7.47%	8.57%	**16.23%**	8.57%	7.23%
ecoli2	9.70%	4.29%	-0.35%	-2.50%	1.17%
glass6	12.95%	15.96%	11.21%	3.79%	11.12%
yeast3	6.67%	1.38%	4.19%	-0.37%	3.58%

V. CONCLUSION

This paper presents a new hybrid resampling technique for imbalanced data classification. In the experiment, twelve imbalanced datasets were rebalanced by five resampling techniques, which are SMOTE, TL, SMOTE+TL, DBSCAN, and DBSM. The models were constructed by decision tree algorithm. For experimental results, the DBSM algorithm successfully accomplish to handle the imbalance problem. An average of accuracy, AUC, and F-measure of the DBSM outperforms other resampling techniques. The effectiveness of the DBSM algorithm improves the performance of F-measure upto 62.62% when compared with the original datasets.

REFERENCES

[1] Q. Yang and X.Wu, "10 Challenging problems in data mining research," Int. J. Inform. Technol. Decision Making, vol. 5, no. 4, pp. 597–604, 2006.

[2] M. Galar, A. Fern´andez, E. Barrenechea, H. Bustince, and F. Herrera, "A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches," IEEE Trans. Systems, Man, and Cybernetics, Part C: Applications and Reviews, vol. 42, no. 4, pp. 463–484, 2012.

[3] Tomek, I., "Two modifications of CNN," IEEE Trans. Systems, Man and Cybernetics, vol. SMC-6, pp. 769-772, Nov. 1976.

[4] N. V. Chawla, L. O. Hall, K.W. Bowyer, and W. P. Kegelmeyer, "SMOTE: Synthetic minority oversampling technique," J. Artif. Intell. Res., vol. 16, pp. 321–357, 2002.

[5] G. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," ACM SIGKDD Explor. Newslett., vol. 6, no. 1, pp. 20–29, Jun. 2004.

[6] J. Jojan and A. Srivihok, "Preprocessing of imbalanced breast cancer data using feature selection combined with over-sampling technique for classification," 2013 International Conference on Advanced Computer Science and Information Systems (ICACSIS), pp. 407 - 412, 2013.

[7] P. Sarakit, T. Theeramunkong, and C. Haruechaiyasak, "Improving emotion classification in imbalanced YouTube dataset using SMOTE algorithm," 2015 2nd International Conference on Advanced Informatics: Concepts, Theory and Applications (ICAICTA), 2015.

[8] R. I. Rashu, N. Haq, and R. M. Rahman, "Data mining approaches to predict final grade by overcoming class imbalance problem," 2014 17th International Conference on Computer and Information Technology (ICCIT), pp. 14-19, 2014.

[9] T. M. Padmaja, N. Dhulipalla, R. S. Bapi, and P. R. Krishna, "Unbalanced data classification using extreme outlier elimination and sampling techniques for fraud detection," ADCOM 2007. International Conference on Advanced Computing and Communications, pp. 511-516, 2007.

[10] G. Batista, A. Bazan, and M. Monard, "Balancing training data for automated annotation of keywords: a case study," in Proceedings of the Second Brazilian Workshop on Bioinformatics, pp. 35–43, 2003.

[11] M. Ester, H. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," Second International Conference on Knowledge Discovery and Data Mining, pp. 226-231, 1996.

[12] J. Alcal´a-Fdez, A. Fern´andez, J. Luengo, J. Derrac, S. Garc´ıa, L. S´anchez, and F. Herrera, "KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework," J. Mult. Valued Logic Soft Comput., vol. 17, no. 2–3, pp. 255–287, 2011.

[13] J. Huang and C. X. Ling, "Using AUC and accuracy in evaluating learning algorithms", IEEE Trans. Knowl. Data Eng., vol. 17, no. 3, pp. 299-310, 2005.

[14] M. Buckland and F. Gey, "The relationship between recall and precision" Journal of the American Society for Information Science, vol. 45(1), pp. 12-19, 1994.

14 - 17 December 2016, Chiang Mai, Thailand

The 20th International Computer Science & Engineering Conference 2016

"Smart Ubiquitous Computing & Knowledge"

Auto-Tuning of Parameters in Hybrid Sampling Method for Class Imbalance Problem

Yotsathon Sanguanmak
Department of Computer Science
King Mongkut's Institute of Technology Ladkrabang
Ladkrabang, Bangkok, 10520, Thailand
58605075@kmitl.ac.th

Anantaporn Hanskunatai
Department of Computer Science
King Mongkut's Institute of Technology Ladkrabang
Ladkrabang, Bangkok, 10520, Thailand
anantaporn.ha@kmitl.ac.th

***Abstract*—the class imbalance is a major problem in machine learning. This problem affects the performance of a model prediction. The DBSM algorithm, a hybrid-sampling technique, was developed to deal with the class imbalance for two-class classification problem. Although the DBSM algorithm is the effective solution, there are too many parameters for tuning in the algorithm. Thus, this paper proposes an automatic parameter tuning for the DBSM algorithm by using a genetic algorithm (GA), called GADBSM. The experimental results of GADBSM are compared with the DBSM algorithm. The results show that the GADBSM can enhance the classification performance of the DBSM algorithm. Moreover, the GADBSM provides the best in F-measure and AUC in all datasets.**

Keywords—genetic algorithm; imbalance dataset problem; hybrid-sampling; SMOTE; DBSCAN

I. INTRODUCTION

Data mining is a procedure of applying machine learning or clustering technique to find relation and pattern in data. The data mining is used to benefit many fields, such as business, medical, natural disaster. For the business, data mining is applied for advertisement planning, finding appropriate promotion for customer, offering cross-sells and up-sells, transportation route planning, an air traffic controller. In medical, it is used to analyze heart disease. In natural disaster, it is used to predict earthquake. However, there are some problems that impact the performance of machine learning such as high-feature classification problem, complex knowledge problem, sequential and time series [1]. One of the important problems which has been most challenging to handle is class imbalance problem.

The class imbalance problem is a number of classes with high difference, in other words a number of attentiveness class less than inattentiveness class. This problem affects performance of model prediction since the model tends to predict the class with larger number of sample, for example, an application for detection of earthquake. There are two types of vibration, a slight shaking or normal level and a violent shaking or critical level. That happens numerous times per day. Although there is frequently happen, most of those vibrations are shaking slightly or on normal level. Thus, the model may predict the critical level to the normal level. In this case, it will damage too much the life and living. Furthermore, a class imbalance problem can occur in many applications, such as fraudulent of telephone calls [2] or credit card detection [3], oil spill [4], spam mail [5] and network intrusions detection [6]. Since the class imbalance problem is important and interesting, the researchers propose many approaches to deal with this issue. They can be divided into three groups, data level, algorithm level, and cost-sensitive. First, data level is one of a data preprocess techniques by using re-sampling technique to rebalance class distribution. Second, an algorithm level is a modification of existing algorithm that improves a performance of positive class instance recognition. Last, a cost-sensitive is a combination between a data level and an algorithm level.

The DBSM algorithm is a hybrid-sampling technique for solving class imbalance problem presented in our previous work [7]. The concept of DBSM algorithm is the combination of DBSCAN under-sampling technique and SMOTE technique. However, the drawback of this technique is a parameter setting since there are many parameters for tuning in the algorithm. These parameter are radius of cluster (ε), the minimum of neighbor in cluster (*Minpts*), a number of k-nearest neighbor, and a percentage of synthetic instance. For example, DBSCAN is applied to eliminate samples of majority class in order to decrease the number of majority classes into minority class region. Since each dataset has a different density, determination of ε is different. If a dataset has a low density, the ε should not be set to a low value due to DBSCAN [8] cannot discover any cluster. On the other hand, if a dataset has a high density, the ε should not be set to a high value because DBSCAN will create too few clusters. Besides, SMOTE [9] is used to increase minority class instances by using nearest neighbor and percentage of synthetic instances. If these parameters have too high value, a decision boundary may not be correct because synthetic instances are generated into majority class area. If these parameters have too low value, the number of minority class instances is not enough to improve the performance of decision boundary. Because of these reasons, the ε and *Minpts* should be determined in a proper value corresponding to a density in each dataset. Moreover, nearest neighbor and percentage of synthetic instances should be determined in a proper value corresponding to a class distribution between majority class and minority class. Therefore, this paper proposes an automatic parameter tuning method based on GA for the DBSM algorithm.

978-1-5090-4420-7/16/$31.00 ©2016 IEEE

II. AUTOMATIC PARAMETER TUNING OF DBSM

Genetic algorithm (GA) [10] is a technique for dealing with an optimization problem that simulating natural evolution process. It can find good parameters without being advised what to learn and adjust. Normally, GA is applied for tuning the parameter in various applications such as C. Huang and C. Wang [11] used a GA approach to optimize the parameters (c, γ) of support vector machine (SVM) algorithm, Z. Lanlan et al [12] proposed the SVM algorithm based on GA to automatically search parameters optimization (c, γ, ε) for material fatigue life prediction, M. Bashiri and A. Geranmayeh [13] applied GA for finding parameters optimization of an artificial neural network (ANN) which are the percentage of training data, the number of neurons in the first layer and the number of neurons in the second layer, and K. Jiang et al [14] proposed a novel technique base on SMOTE algorithm and applies the GA to find optimal sampling rate for the rockburst prediction. Therefore, GA is applied for automatic parameter tuning in the DBSM algorithm and called GADBSM. For the GADBSM method, there are 7 steps which are chromosome encoding, initial population generating, population evaluating, parent selecting, crossover, mutation, and population replacement.

step1: chromosome encoding is a representation of chromosome as a bit string. A problem state is defined by a chromosome. Each chromosome contains a number of genes and each gene is represented by a bit 0 or 1. Since the DBSM algorithm requires four parameters which are the number of *k*-nearest neighbors (*k*), percentage of synthetic instances (*p*), a minimum of neighbors in a cluster (*Minpts*), and a radius of a cluster (*ε*), thus each chromosome consists of four parts represented by variable A, B, C and D as shown in Fig.1.

Table I shows the detail of a chromosome encoding. Variables "A" and "B" are used for the SMOTE algorithm and variables "C" and "D" are run by the DBSCAN algorithm. The variable "A" represents the number of k-nearest neighbors (*k*). Since the number of k-nearest neighbors equals to 3 providing a good performance in our previous work [7], the range of the variable "A" is varied between 1 to 8. Thus bit length of this parameter is set to three. The variable "B" is a parameter of a percentage of synthetic instances (*p*). The range of this value is between 100% to 800% because the highest imbalance ratio from all datasets used in the experiment is equals to 8.1. For this reason, this parameter is represented by three bits. The variable "C" represents a minimum of neighbors in a cluster (*Minpts*) created by DBSCAN. If *Minpts* is set too high, many majority class instances (negative instances) will be considered as noises thus these negative instances will not be removed from a dataset by DBSCAN under-sampling algorithm [7]. Consequently, a range of *Minpts* is set between 1 to 100 and represented by seven bits. For the last variable "D", this parameter indicates a radius of a cluster (*ε*) for the DBSCAN algorithm. In the experiment, this parameter is set as a real value between [0, 1]. Hence there are ten bits length of variable "D". Therefore, a total bits of the encoded chromosome are twenty-three.

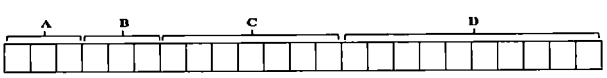


Fig. 1. The design of chromosome encoding.

TABLE I. THE DETAIL OF CHROMOSOME ENCODING

Variable	Parameters	Value
A	the number of k-nearest neighbors (*k*)	1-8
B	a percentage of synthetic instances (*p*)	100% -800%
C	a minimum of neighbors in a cluster (*Minpts*)	1-100
D	a radius of a cluster (*ε*)	[0-1]

Step2: initial population generating is a process of sampling a set of initial chromosome for beginning the genetic algorithm. In this experiment, initial populations is set to 30. Thus, each generation consists of 30 chromosomes. For generating a chromosome, each bit is represented by a random number, 0 or 1.

Step3: population evaluating is a method of measuring a performance of a chromosome. After a set of populations is generated, each chromosome is decoded by converting bit string into actual values of all parameters *k*, *p*, *Minpts*, and *ε* respectively. These parameter's values are passed through the SMOTE and DBSCAN algorithms in order to generate a training set. After that, a new training set is used to construct a model by using decision tree (DT) algorithm. Finally, the model is evaluate a classification performance with F-measure and AUC [7] by a test set. Thus the fitness function of a chromosome is illustrated in (1). Fig.2 shows an overall processes of the population evaluation step.

$$f(x) = \text{F-measure}(x) + \text{AUC}(x) \quad (1)$$

Where F-measure(x) and AUC(x) are a F-measure and AUC values respectively of the DT model learned from the training set with parameter setting from the chromosome x.

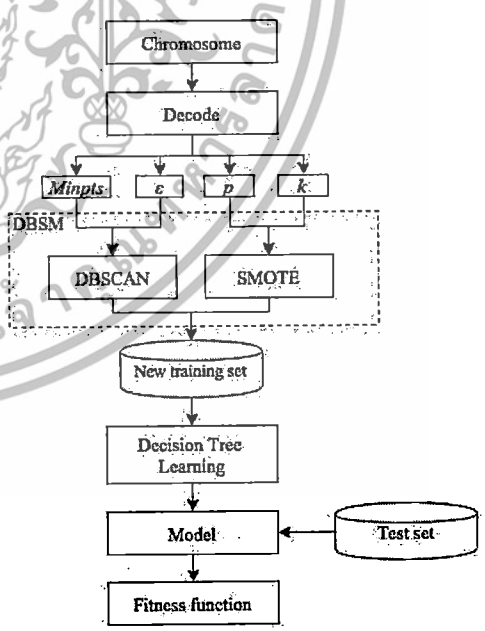


Fig. 2. The process of population evaluating.

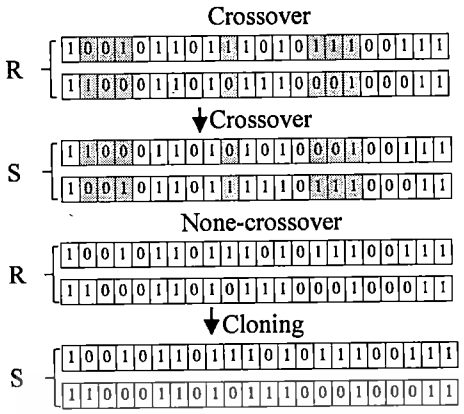


Fig. 3. Crossover and none- crossover operation.

Step4: parent selecting is a process of selecting survivor chromosomes for breeding. The higher value of a chromosome's fitness function, the more chance to be selected as a parent chromosome. In this experiment, a roulette wheel technique [15] is used for selecting a chromosome.

Step5: crossover is a genetic operator for creating new offsprings by swapping some genes of parent chromosomes. If a crossover occurs, crossover points in parent chromosomes will be randomly selected for positions and then genes in parent chromosomes are swapped at crossover points. At the end, there are two new offsprings. If a crossover does not occur, two new offsprings are generated by cloning parent chromosomes. In this experiment, crossover rate is set to 0.7 and a uniform crossover is applied to generate a crossover mask. An example of crossover operation is illustrated in Fig.3. From Fig.3, R is a set of parent chromosomes. S is a set of offspring chromosomes. The highlighted bits are random position masks of the parent chromosomes.

Step6: mutation is another genetic operation which randomly changes a gene value of an offspring chromosome. If a mutation occurs, a value of a selected bit is changed. The mutation bit is selected by random a position in a chromosome. In this experiment, the mutation rate is set to 0.01. As shown in Fig.4, a mutation operation will occur in chromosome S1 and S2 whereas chromosome S3 will not mutate. The highlighted bit is a position of gene mutation.

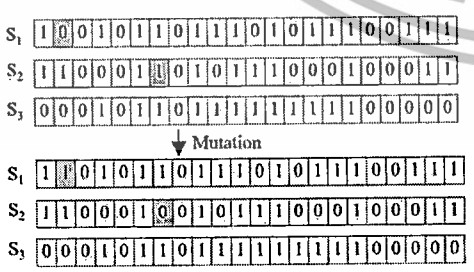


Fig. 4. Mutation operation.

Step7: population replacement is a process of replacing a set of old-generation populations as a set of new populations (or chromosomes). If a current iteration is less than the number of generation specified by user, the offspring chromosomes become a set of new populations and then proceed to step 3 and repeats the procedures until the end paradigm. In this experiment, the number of generations is set to 100. According to previous description, the automatic tuning parameter of DBSM (or GADBSM) is shown in Fig. 5.

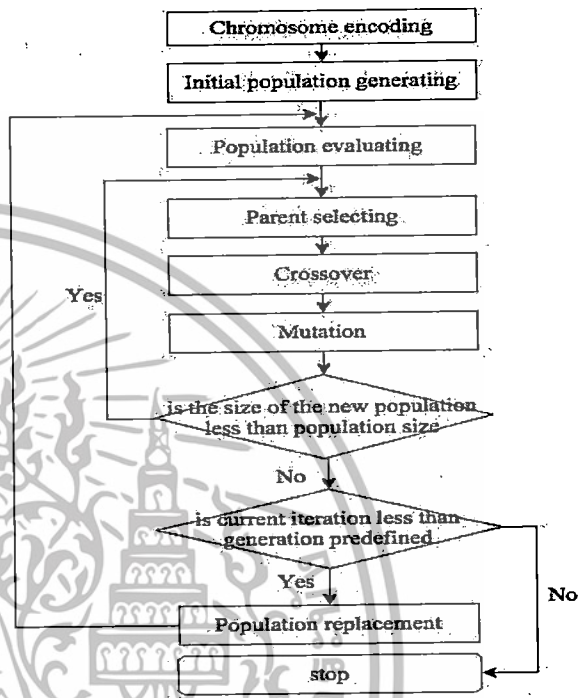


Fig. 5. The GADBSM algorithm.

III. Experimental Design and Result

A. Dataset

TABLE II. Characteristic of Datasets

Dataset Name	Attributes (R/I/N)	Examples	IR
glass1	9 (9/0/0)	214	1.8
wiscosin	9 (0/9/0)	683	1.9
glass0	9 (9/0/0)	214	2.1
yeast1	8 (8/0/0)	1484	2.5
haberman	3 (0/3/0)	306	2.8
vehicle2	18 (0/18/0)	846	2.9
vehicle1	18 (0/18/0)	846	2.9
new-thyroid1	5 (4/1/0)	215	5.1
new-thyroid2	5 (4/1/0)	215	5.1
ecoli2	7 (7/0/0)	336	5.5
glass6	9 (9/0/0)	214	6.4
yeast3	8 (8/0/0)	1484	8.1

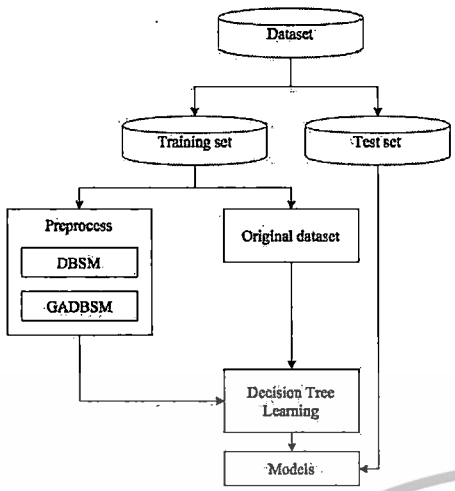


Fig. 6. The flow chart of experimental design.

There are 12 datasets downloaded from KEEL [16] used in the experiment. These datasets are two-class classification problem. Table II shows the characteristic datasets of which attribute (R/I/N) means the number of real, integer, and nominal attribute, respectively. Column examples means the number of samples in dataset. Column IR is an imbalance rate in a dataset which is a proportion of negative (majority) classes divided by the number of positive (minority) classes.

B. Experimental Design

There are four steps in the experiment which are data generating, data preprocessing, model construction, and model evaluation. In the first step, *k*-fold crossvalidation is used to split a dataset into a training set and a test set. The training set is used to construct a model and the test set is used to evaluate the model in the last step. In this step, *k* is set to 5. For data preprocessing, there are two techniques, which are DBSM algorithm and GADBSM algorithm for rebalancing the class distribution in a dataset. These techniques are a hybrid-sampling technique that both decreases the number of majority classes and increases the number of minority classes. In the third step, model construction, a decision tree algorithm is applied to build a classifier to evaluate the performance of both resampling techniques and none-resampling technique (original dataset). Lastly for the evaluation model, the model is evaluated the performance by the test set. In this experiment, F-measure, accuracy, and AUC [7] are used to evaluate the classification performance. Therefore in the last step, there are three models in each fold, which are DBSM model, GADBSM model, and original model. These steps can be represented by the flow chart shown in Fig.6.

C. Experimental result

In the experiment, three performance measurements, accuracy, F-measure, and AUC are used to evaluate preprocess techniques. The classification performance of the DBSM, GADBSM and original dataset are compared in Table III, IV and VI.

Table III shows the percentage of accuracy for each technique where the highest accuracy is highlighted in bold. From this table, for the original technique, there is only one dataset providing the highest accuracy and there are 11 datasets of GADBSM that yield the best accuracy.

Table IV shows the F-measure results for each technique where the best F-measure is highlighted in bold. From this table, the GADBSM provides the best F-measure in every datasets when compared with the original and DBSM technique. Moreover, Table V shows the improvement of GADBSM algorithm in F-measure. The GADBSM can improve the performance of DBSM and original upto 12.58% and 83.07 % respectively.

Table VI shows the AUC results for each technique where the best AUC is highlighted in bold. From this table, the GADBSM also outperforms the other techniques on AUC. Moreover, Table VII shows the improvement of GADBSM algorithm in AUC. The GADBSM can improve the performance of DBSM and original upto 9.39% and 28.03% respectively.

TABLE III. THE ACCURACY OF THE ORIGINAL AND RESAMPLING TECHNIQUES

Dataset Name	IR	Original	DBSM	GADBSM
glass1	1.8	70.13%	76.65%	**83.19%**
wiscosin	1.9	94.73%	95.08%	**95.90%**
glass0	2.1	81.75%	80.35%	**82.19%**
yeast1	2.5	**73.65%**	68.80%	69.47%
haberman	2.8	66.01%	63.82%	**67.63%**
vehicle2	2.9	95.27%	95.84%	**96.93%**
vehicle1	2.9	74.11%	73.69%	**74.70%**
new-thyroid1	5.1	94.42%	97.40%	**97.21%**
new-thyroid2	5.1	96.28%	98.42%	**99.07%**
ecoli2	5.5	91.67%	91.91%	**92.85%**
glass6	6.4	93.01%	95.34%	**95.81%**
yeast3	8.1	93.67%	94.21%	**94.27%**
Average		85.39%	85.96%	**87.44%**

TABLE IV. THE COMPARISON ON F-MEASURE OF EACH TECHNIQUE

Dataset Name	IR	Original	DBSM	GADBSM
glass1	1.8	0.5401	0.7102	**0.7743**
wiscosin	1.9	0.9248	0.9315	**0.9426**
glass0	2.1	0.7037	0.7117	**0.7625**
yeast1	2.5	0.5280	0.5732	**0.5762**
haberman	2.8	0.2873	0.4672	**0.5260**
vehicle2	2.9	0.9090	0.9194	**0.9419**
vehicle1	2.9	0.4820	0.5816	**0.6000**
new-thyroid1	5.1	0.8364	0.9139	**0.9181**
new-thyroid2	5.1	0.8857	0.9519	**0.9713**
ecoli2	5.5	0.7000	0.7679	**0.7937**
glass6	6.4	0.7345	0.8296	**0.8590**
yeast3	8.1	0.7034	0.7503	**0.7639**
Average		0.6862	0.7590	**0.7858**

TABLE V. THE IMPROVEMENT OF GADBSM ALGORITHM IN F-MEASURE

Dataset Name	Original	DBSM
glass1	43.36%	9.03%
wiscosin	1.92%	1.19%
glass0	8.35%	7.13%
yeast1	9.13%	0.52%
haberman	**83.07%**	**12.58%**
vehicle2	3.61%	2.44%
vehicle1	24.49%	3.17%
new-thyroid1	9.77%	0.46%
new-thyroid2	9.66%	2.04%
ecoli2	13.38%	3.36%
glass6	16.95%	3.54%
yeast3	8.60%	1.82%

TABLE VI. THE COMPARISON OF AUC BETWEEN GADBSM, DBSM AND ORIGINAL

Dataset Name	IR	Original	DBSM	GADBSM
glass1	1.8	0.6605	0.7774	**0.8294**
wiscosin	1.9	0.9441	0.9528	**0.9598**
glass0	2.1	0.7838	0.7882	**0.8307**
yeast1	2.5	0.6688	0.6989	**0.7016**
haberman	2.8	0.5320	0.6227	**0.6811**
vehicle2	2.9	0.9415	0.9480	**0.9674**
vehicle1	2.9	0.6523	0.7301	**0.7454**
new-thyroid1	5.1	0.9091	0.9338	**0.9603**
new-thyroid2	5.1	0.9317	0.9744	**0.9829**
ecoli2	5.5	0.8097	0.8940	**0.9038**
glass6	6.4	0.8477	0.9032	**0.9338**
yeast3	8.1	0.8297	0.8755	**0.8954**
Average		0.7926	0.8416	**0.8660**

TABLE VII. THE IMPROVEMENT OF GADBSM ALGORITHM IN AUC

Dataset Name	Original	DBSM
glass1	25.57%	6.69%
wiscosin	1.66%	0.73%
glass0	5.98%	5.39%
yeast1	4.90%	0.38%
haberman	**28.03%**	**9.39%**
vehicle2	2.76%	2.05%
vehicle1	14.27%	2.09%
new-thyroid1	5.63%	2.84%
new-thyroid2	5.50%	0.88%
ecoli2	11.62%	1.09%
glass6	10.15%	3.39%
yeast3	7.92%	2.27%

IV. CONCLUSION

This paper applies the genetic algorithm (GA) for an automatic parameter tuning in the DBSM algorithm, called GADBSM. In the experiment, 12 datasets were rebalanced by two techniques, which are DBSM and GADBSM. The decision tree algorithm was used to construct a model for performance evaluation between two resampling techniques and an original dataset. The experimental results point out the GADBSM is the best algorithm when compared with DBSM and original. In addition the GADBSM can improve the performance of DBSM algorithm upto 12.58% and 9.39% on F-measure and AUC respectively. In summary, the GADBSM is more convenient in parameter setting of the DBSM algorithm and increase the performance of the DBSM algorithm.

REFERENCES

[1] Q. Yang and X.Wu, "10 Challenging problems in data mining research," Int. J. Inform. Technol. Decision Making, vol. 5, no. 4, pp. 597–604, 2006.

[2] T. Fawcett, and F. Provost, "Adaptive fraud detection", Data Mining and Knowledge Discovery, pp. 291–316, 1997

[3] P. K. Chan and, S. J. Stolfo, "Toward scalable learning with non-uniform class and cost distributions: a case study in credit card fraud detection," in Proc. 4th International Conference Knowledge Discovery Data Mining (KDD-98), pp. 164–168, 1998.

[4] M. Kubat, R. Holte, and S. Matwin, "Machine learning for the detection of oil spills in satellite radar images", pp. 195–215,1998.

[5] J. Alqatawna, H. Faris, K. Jaradat, M. Al-Zewairi, and O. Adwan"Improving knowledge based spam detection methods: the effect of malicious related features in imbalance Data distribution" International journal Communications Network and System Sciences, 2015.

[6] D. A Cieslak, N. V. Chawla, and A. Striegel "Combating imbalance in network intrusion datasets", GrC, 2006.

[7] Y. Sanguanmak, and A. Hanskunatai, "DBSM: the combination of DBSCAN and SMOTE for imbalanced data classification" International Joint Conference on Computer Science and Software Engineering (JCSSE), 2016.

[8] M. Ester, H. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," Second International Conference on Knowledge Discovery and Data Mining, pp. 226-231, 1996.

[9] N. V. Chawla, L. O. Hall, K.W. Bowyer, and W. P. Kegelmeyer, "SMOTE: Synthetic minority oversampling technique," J. Artif. Intell. Res., vol. 16, pp. 321–357, 2002.

[10] m. negnevitsky, "Artificial intelligence: a guide to intelligent systems", 2rd Edition, pp 222-232, 2005.

[11] C. Huang and C. Wang, "A GA-based Feature Selection and Parameters Optimization for Support Vector Machines," Expert Systems with Applications, pp. 231–240, 2006.

[12] Z. Lanlan, L. Juyang, Z. Qilin, and W. Yudong, "Using Genetic Algorithm to Optimize Parameters of Support Vector Machine and Its Application in Material Fatigue Life Prediction," Advances in Natural Science, Vol. 8, pp. 21-26, 2015.

[13] M. Bashiri and A. Geranmayeh, "Tuning The Parameters of An Artificial Neural Network using Central Composite Design and Genetic Algorithm," Scientia Iranica, pp.1600–1608, 2011.

[14] K. Jiang, J. Lu, and K. Xia, "A Novel Algorithm for Imbalance Data Classification Based on Genetic Algorithm Improved SMOTE," Arabian Journal for Science and Engineering, Volume 41, pp 3255–3266. 2016.

[15] Goldberg,and D.E., "Genetic algorithms in search, optimisation and machine learning", 1989.

[16] [12] J. Alcal´a-Fdez, A. Fern´andez, J. Luengo, J. Derrac, S. Garc´ıa, L. S´anchez, and F. Herrera, "KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework," J. Mult. Valued Logic Soft Comput., vol. 17, no. 2–3, pp. 255–287, 2011.

ประวัติผู้เขียน

ชื่อ	นาย ยศธร สงวนมาก
วัน เดือน ปีเกิด	3 สิงหาคม 2535
ที่อยู่ปัจจุบัน	72/10 ต.บางละมุง อ.บางละมุง จ.ชลบุรี 20150
ประวัติการศึกษา	วิทยาศาสตรบัณฑิต สาขาวิทยาการคอมพิวเตอร์
	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง