รายงานการวิจัยฉบับสมบูรณ์

วิธีใหม่ในการสุ่มตัวอย่างแบบผสมสำหรับการจำแนกประเภทของชุดข้อมูลที่ไม่สมดุล

A New Hybrid Sampling Method for Imbalanced Datasets Classification

นางอนันตพร หรรษคุณาฒัย

งานวิจัยนี้ได้รับทุนสนับสนุนงานวิจัยจากเงินรายได้ ประจำปีงบประมาณ พ.ศ. 2560

คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

รายงานการวิจัยฉบับสมบูรณ์

วิธีใหม่ในการสุ่มตัวอย่างแบบผสมสำหรับการจำแนกประเภทของชุดข้อมูลที่ไม่สมดุล

A New Hybrid Sampling Method for Imbalanced Datasets Classification

นางอนันตพร หรรษคุณาฒัย

งานวิจัยนี้ได้รับทุนสนับสนุนงานวิจัยจากเงินรายได้ ประจำปีงบประมาณ พ.ศ. 2560

คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ชื่อโครงการ (ภาษาไทย) วิธีใหม่ในการสุ่มตัวอย่างแบบผสมสำหรับการจำแนกประเภทของชุดข้อมูลที่ไม่สมดุล

ชื่อโครงการ (ภาษาอังกฤษ) A New Hybrid Sampling Method for Imbalanced Datasets Classification

แหล่งเงิน เงินรายได้คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ประจำปีงบประมาณ 2560 จำนวนเงินที่ได้รับการสนับสนุน 45,000 บาท

ระยะเวลาทำการวิจัย 1 ปี ตั้งแต่ วันที่ 1 ตุลาคม พ.ศ. 2559 ถึง วันที่ 30 กันยายน พ.ศ. 2560

ชื่อ-สกุล หัวหน้าโครงการ

หัวหน้าโครงการวิจัย ผศ.ดร.อนันตพร หรรษคุณาฒัย สัดส่วน 100%

หน่วยงานต้นสังกัด ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

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

Research Title: A New Hybrid Sampling Method for Imbalanced Datasets Classification

Researcher: Asst.Prof.Dr. Anantaporn Hanskunatai

Faculty: Science **Department:** Computer Science

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

การวิจัยครั้งนี้ได้รับทุนสนับสนุนการวิจัยจากสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง จากเงินรายได้ ประจำปีงบประมาณ พ.ศ. 2560 นอกจากนี้การทำวิจัยครั้งนี้สำเร็จลุล่วงไปด้วยดีเพราะได้รับกำลังใจที่ดีจากครอบครัว และทรัพยากรทางด้านคอมพิวเตอร์ที่ใช้ประกอบการทำวิจัยของภาควิชาวิทยาการคอมพิวเตอร์ และขอขอบคุณทุนสนับสนุนการตีพิมพ์เผยแพร่ผลงานของคณะวิทยาศาสตร์ ที่ช่วยสนับสนุนการตีพิมพ์ผลงานทางวิชาการของงานวิจัยชิ้นนี้

นางอนันตพร หรรษคุณาฒัย

สารบัญ

สารบัญ (ต่อ)

สารบัญตาราง

สารบัญภาพ

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

การจำแนกประเภทนั้นมักพบกับปัญหาความไม่สมดุลของกลุ่มข้อมูล (imbalanced dataset) การนำข้อมูลที่มีปริมาณแตกต่างกันอย่างชัดเจนมาเรียนรู้ด้วยวิธีการจำแนกประเภท ย่อมได้โมเดลที่รู้จำรูปแบบของกลุ่มข้อมูลที่มีปริมาณมาก ทำให้ประสิทธิภาพของโมเดลลดต่ำลง ตัวอย่างปัญหาของความไม่สมดุลกันของข้อมูลในปัจจุบันที่มีอยู่คือข้อมูลทางด้านการแพทย์ เช่น การวิเคราะห์ความเสี่ยงต่อการเป็นมะเร็ง ซึ่งจำนวนผู้ป่วยที่เป็นมะเร็งกับผู้ป่วยทั่วไปที่ไม่ได้เป็นมะเร็งมีสัดส่วนไม่เท่ากันเป็นอย่างมาก หรือการวิเคราะห์การปลอมแปลงบัตรเครดิต ซึ่งเหตุการณ์ในการปลอมแปลงบัตรจะเกิดน้อยมากเมื่อเทียบกับเหตุการณ์ปกติ

ซึ่งวิธีการแก้ปัญหาความไม่สมดุลของกลุ่มข้อมูล สามารถทำได้ 3 แบบด้วยกัน คือ ระดับขั้นตอนวิธี (algorithm level) ระดับข้อมูล (data level) และ cost-sensitive ในงานวิจัยนี้จะกล่าวถึงการจัดการในระดับข้อมูลเป็นหลักโดยเทคนิคการสุ่มตัวอย่าง (sampling) ซึ่งแบ่งเป็น 3 เทคนิค คือ เทคนิคการลดจำนวนข้อมูลของกลุ่มข้อมูล (undersampling) เป็นวิธีการลดจำนวนข้อมูลของกลุ่มข้อมูลที่เป็นเสียงส่วนมาก (majority class) ให้น้อยลงจนมีปริมาณพอๆกับกลุ่มข้อมูลที่มีเสียงส่วนน้อย (minority class) เทคนิคการเพิ่มจำนวนข้อมูลของกลุ่มข้อมูล (oversampling) เป็นวิธีการเพิ่มจำนวนข้อมูลของกลุ่มข้อมูลที่เป็นเสียงส่วนน้อย (minority class) ให้เพิ่มขึ้นจนมีปริมาณพอๆกับกลุ่มข้อมูลที่มีเสียงส่วนมาก (majority class) และเทคนิคการสุ่มตัวอย่างแบบผสม (hybridsampling) เป็นการรวมเทคนิคการลดจำนวนข้อมูลของกลุ่มข้อมูล และเทคนิคการเพิ่มจำนวนข้อมูลของกลุ่มข้อมูลนำมาใช้ร่วมกัน งานวิจัยนี้จึงเป็นการพัฒนาขั้นตอนวิธีการสุ่มตัวอย่างแบบผสมเพื่อให้ได้วิธีที่มีประสิทธิภาพในการคัดเลือกและเพิ่มจำนวนตัวอย่างหรือข้อมูลที่สำคัญโดยมีเป้าหมายเพื่อให้ได้เซตของข้อมูลฝึกสอนที่เหมาะสม ซึ่งมีผลทำให้โมเดลที่สร้างได้มีประสิทธิภาพในการจำแนกประเภทข้อมูลได้มากยิ่งขึ้น

1.2 วัตถุประสงค์ของการวิจัย

พัฒนาขั้นตอนวิธีในการคัดเลือกและสุ่มตัวอย่างที่มีประสิทธิภาพเพื่อใช้เป็นข้อมูลฝึกสอน ในการสร้างโมเดลเพื่อแก้ปัญหาทางด้านการจำแนกประเภทข้อมูลในกรณีที่ชุดข้อมูลมีจำนวนตัวอย่างไม่สมดุลกันใจแต่ละคลาส

1.3 ขอบเขตของการวิจัย

1) ชุดข้อมูลที่ทำการวิจัยทั้งหมดจะเป็นชุดข้อมูลที่มีจำนวนคลาสเพียง 2 คลาสเท่านั้น
2) ทำการเปรียบเทียบประสิทธิภาพของวิธีใหม่ที่นำเสนอเทียบกับเทคนิคการสุ่มตัวอย่าง 3 อัลกอริทึมคือ SMOTE Tomek Links และ SMOTE+Tomek Links ด้วยโมเดลจำแนกประเภทที่สร้างจากเทคนิคของการเรียนรู้ของเครื่อง (machine learning)

1.4 วิธีดำเนินการวิจัย

1) กำหนดปัญหาและหาแนวทางแก้ไขปัญหา
2) ทบทวนวรรณกรรมที่เกี่ยวข้อง เพื่อศึกษาว่าปัจจุบันได้มีงานวิจัยเกี่ยวกับด้านนี้มากน้อยเพียงใดและหาจุดเด่นจุดด้อยของแต่ละงานวิจัย
3) พัฒนาอัลกอริทึมในการสุ่มข้อมูลแบบผสม
4) เก็บรวบรวมชุดข้อมูล โดยใช้ชุดข้อมูลที่เป็นมาตรฐานที่ใช้ในงานวิจัยทางด้านการเรียนรู้ของเครื่อง
5) เมื่อได้ชุดข้อมูลเรียบร้อยแล้วขั้นตอนต่อมาจะทำการจัดเตรียมข้อมูลแบ่งข้อมูลเป็นข้อมูลฝึกสอนและข้อมูลทดสอบเพื่อทดสอบประสิทธิภาพของวิธีที่พัฒนาขึ้น
6) ในส่วนของข้อมูลฝึกสอนจะมีการเตรียมข้อมูลโดยใช้อัลกอริทึมการสุ่มแบบผสมที่พัฒนาขึ้นใหม่เพื่อทำการคัดเลือกข้อมูลที่สำคัญและเพิ่มข้อมูลที่จำเป็นเพื่อให้ได้ข้อมูลฝึกสอนที่มีประสิทธิภาพ
7) สร้างโมเดลจำแนกประเภทด้วยเทคนิคการเรียนรู้ของเครื่องโดยใช้ข้อมูลฝึกสอนที่ได้จากขั้นตอนที่ 6 (จากอัลกอริทึมที่พัฒนาขึ้นใหม่) และทดสอบประสิทธิภาพของโมเดลโดยใช้ข้อมูลทดสอบที่แบ่งไว้ในขั้นตอนที่ 5
8) สร้างโมเดลจำแนกประเภทด้วยเทคนิคการเรียนรู้ของเครื่องโดยใช้ข้อมูลฝึกสอนที่ได้จากอัลกอริทึม SMOTE Tomek Links และ SMOTE+Tomek Links ตามลำดับ และทดสอบประสิทธิภาพของโมเดลโดยใช้ข้อมูลทดสอบที่แบ่งไว้ในขั้นตอนที่ 5
9) เปรียบเทียบประสิทธิภาพของโมเดลที่ได้จากการสุ่มตัวอย่างด้วยวิธีใหม่ที่พัฒนาขึ้นกับวิธี SMOTE Tomek Links และ SMOTE+Tomek Links พร้อมทั้งวิเคราะห์ผลการทดลองที่ได้

10) เขียนรูปเล่มรายงานการวิจัยฉบับสมบูรณ์

11) เขียนผลงานวิจัยเพื่อตีพิมพ์ในงานประชุมวิชาการระดับนานาชาติ

1.5 ประโยชน์ที่คาดว่าจะได้รับ

ขั้นตอนวิธีการสุ่มข้อมูลแบบผสมแบบใหม่สามารถแก้ปัญหาความไม่สมดุลของข้อมูลทำให้โมเดลจำแนกประเภทที่สร้างได้มีประสิทธิภาพเพิ่มมากขึ้น

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

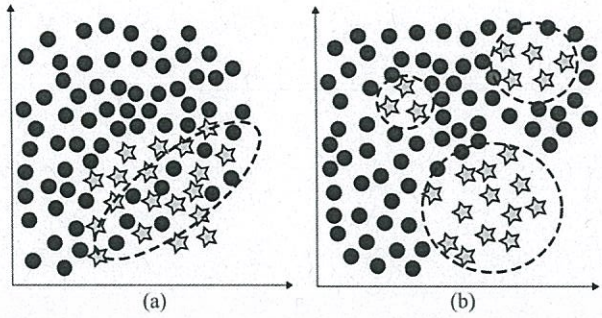
2.1 ความไม่สมดุลของข้อมูล

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 2.1 ภาพ (a) จะเห็นได้ว่ากลุ่มข้อมูลเสียงข้างน้อย(แทนด้วยสัญลักษณ์ดาว) แทรกอยู่ระหว่างกลุ่มข้อมูลเสียงข้างมาก (แทนด้วยสัญลักษณ์วงกลม) ซึ่งทำให้ยากที่จะจำแนกข้อมูลเสียงข้างน้อยออกจากกลุ่มข้อมูลเสียงข้างมาก ดังนั้นหากไม่มีการซ้อนทับกันระหว่างกลุ่มข้อมูล จะทำให้การจำแนกและการเรียนรู้ของเครื่องจักรง่ายขึ้น

3.) กลุ่มตัวอย่างมีการกระจายตัว (small distribution) คือการที่กลุ่มข้อมูลเสียงข้างน้อยจำนวนน้อย ๆ กระจายตัวกันออกไป จากภาพที่ 2.1 ภาพ (b) จะเห็นได้ว่า กลุ่มข้อมูลเสียงข้างน้อย (ตัวอย่างบวก แทนด้วยสัญลักษณ์ดาว)กระจายตัวกันออกไป ซึ่งถูกล้อมรอบด้วยกลุ่มข้อมูลเสียงข้างมาก (ตัวอย่างลบแทนด้วยสัญลักษณ์วงกลม) ทำให้ตัวอย่างบวกที่สนใจอาจถูกทำนายเป็นตัวอย่างลบ เนื่องจากทฤษฎีเพื่อนบ้านใกล้เคียงที่สุด k ตัวทำให้ตัวอย่างบวกที่มีเพียงไม่กี่ตัว ถูกทำนายเป็นตัวอย่างลบซึ่งมีปริมาณมากกว่า ด้วยเหตุนี้จึงทำให้มีค่าอัตราความผิดพลาด (error rate) ดังสมการที่ 2.1 ที่สูง และส่งผลต่อค่าความถูกต้องของโมเดลจำแนกประเภท



ภาพที่ 2.1 ตัวอย่างความไม่สมดุลของข้อมูล

ในงานวิจัยนี้พิจารณาชุดข้อมูลที่มี 2 กลุ่มข้อมูล คือ กลุ่มข้อมูลเสียงข้างน้อยหรือตัวอย่างบวก (positive class) และกลุ่มข้อมูลเสียงข้างมากหรือตัวอย่างลบ (negative class) และทั้งสองกลุ่มข้อมูลถูกนำไปพิจารณาเป็นค่าอัตราความไม่สมดุลของข้อมูลที่เรียกว่า *IR* (imbalanced rate) ซึ่งเป็นสัดส่วนระหว่างจำนวนข้อมูลในกลุ่มเสียงข้างน้อยและจำนวนข้อมูลในกลุ่มเสียงข้างมากดังสมการที่ 2.1 ค่าอัตราความไม่สมดุลนั้นบ่งบอกถึงระดับความไม่สมดุลของข้อมูล หากค่าที่ได้มีค่าเท่ากับ 1 หมายความว่า ข้อมูลดังกล่าวมีความสมดุลมากและจะมีความสมดุลของกลุ่มข้อมูลลดน้อยลงเมื่อค่าที่ได้มากกว่า 1

$$IR = \frac{\text{จำนวนเสียงข้างมาก}}{\text{จำนวนเสียงข้างน้อย}} \tag{2.1}$$

ค่าอัตราความไม่สมดุล เมื่อมีค่าที่มากแล้วนั้นจะส่งผลต่อโมเดลการจำแนกประเภท ทำให้โมเดลที่ได้ รู้จำรูปแบบของกลุ่มข้อมูลเสียงข้างมาก ทำให้ประสิทธิภาพของโมเดลลดต่ำลง ดังนั้นสามารถวัดประสิทธิภาพของโมเดลได้จาก Confusion Matrix ดังแสดงในตารางที่ 2.1 โดยแนวคอลัมน์คือ ตัวอย่างที่โมเดลทำนายได้และแนวนอนคือ ค่าของกลุ่มที่แท้จริงของตัวอย่าง ในตาราง Confusion Matrix ประกอบด้วยค่า TN FN TP และ FP ซึ่งนำไปหาค่าความถูกต้อง (accuracy rate) ดังสมการที่ 2.3 ของโมเดล

ตารางที่ 2.1 Confusion Matrix

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

$$\text{Error} = \frac{FP+FN}{TP+FN+FP+TN} \quad (2.2)$$

$$\text{Accuracy} = \frac{TP+TN}{TP+FN+FP+TN} \quad (2.3)$$

โดยที่ *TP* คือ จำนวนตัวอย่างที่โมเดลทายถูก และเป็นตัวอย่างบวก

TN คือ จำนวนตัวอย่างที่โมเดลทายถูก และเป็นตัวอย่างลบ

FP คือ จำนวนตัวอย่างที่โมเดลทายออกมาเป็นตัวอย่างบวก แต่แท้จริงแล้วเป็นตัวอย่างลบ

FN คือ จำนวนตัวอย่างที่โมเดลทายออกมาเป็นตัวอย่างลบ แต่แท้จริงแล้วเป็นตัวอย่างบวก

อย่างไรก็ตามค่าความถูกต้องสูงไม่ได้หมายถึงโมเดลนั้นมีประสิทธิภาพดีเสมอไป ดังนั้นจึงต้องใช้ค่าค้นคืน (recall) หรือ TP rate ซึ่งคำนวณได้ดังสมการที่ 2.4 ค่ารู้จำการทายตัวอย่างลบ (specificity) ดังสมการที่ 2.5 และค่าความแม่นยำ (precision) ดังสมการที่ 2.6 ช่วยในการพิจารณา

$$\text{recall} = \frac{TP}{TP+FN} \quad (2.4)$$

$$\text{specification} = \frac{TN}{FP+TN} \quad (2.5)$$

$$\text{precision} = \frac{TP}{FP+TP} \quad (2.6)$$

$$FP_{rate} = \frac{FP}{FP+TN} \quad (2.7)$$

ค่าเหล่านี้ถูกนำมารวมกันเป็นมาตรวัดหนึ่งที่มีชื่อว่า F-measure ซึ่งเป็นสัดส่วนระหว่างค่าความแม่นยำและค่าค้นคืนดังสมการที่ 2.8 โดยค่าที่ได้บ่งบอกว่าความแม่นยำและค่าค้นคืนมีสัดส่วนมากด้วยกันทั้งคู่หรือไม่ หากค่าที่ได้มีค่าที่สูงแสดงว่าโมเดลที่ได้นั้นมีประสิทธิภาพ และอีกมาตรวัดหนึ่งคือ AUC (the area under the ROC curve) ดังสมการที่ 2.9 ซึ่งค่าที่ได้บ่งบอกว่าโมเดลมีความสามารถเพียงใดในการรู้จำตัวอย่างบวก

$$\text{F-measure} = \frac{2* \text{Precision}* \text{recall}}{\text{Precision}+ \text{recall}} \quad (2.8)$$

$$AUC = \frac{1 + recall - FP_{rate}}{2} \quad (2.9)$$

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

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

2.2 เทคนิคการแก้ปัญหาในระดับข้อมูล (data level)

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

2.2.1 ขั้นตอนวิธี SMOTE

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

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

ภาพที่ 2.2 pseudo code แสดงวิธีการทำงานของขั้นตอนวิธี SMOTE

จากตัวอย่างภาพที่ 2.2 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

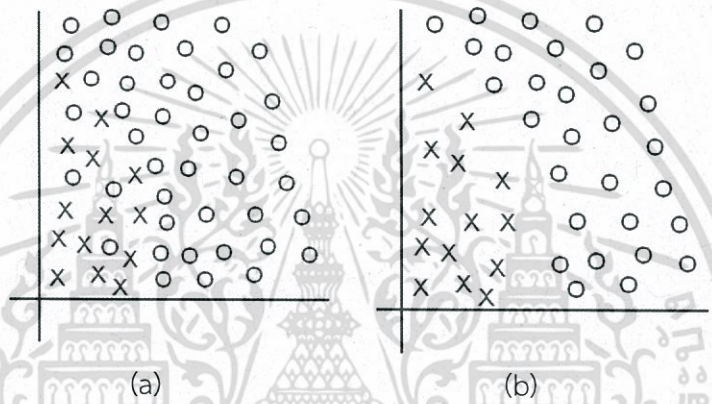
   Synthetic[0][1] = 6 + 0.5(-2) = 5

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

   ดังนั้นตัวอย่างใหม่จะถูกสร้างขึ้นเป็น(5,3.5)

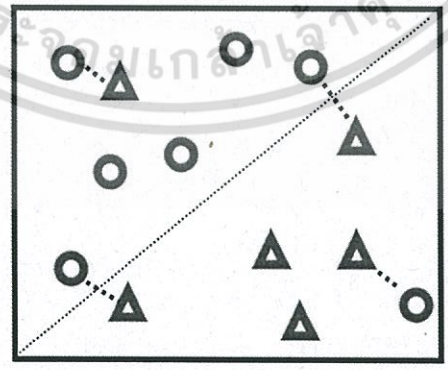
2.2.2 ขั้นตอนวิธี Tomek Links

ปัญหาความไม่สมดุลของข้อมูลสามารถแก้ปัญหาได้โดยใช้เทคนิคการลดจำนวนกลุ่มข้อมูลเสียงข้างมาก หนึ่งในนั้นคือเทคนิคที่มีชื่อว่า Tomek Links [2] ซึ่งเป็นการกำจัดกลุ่มข้อมูลเสียงข้างมากที่เกิดการซ้อนทับกันของกลุ่มข้อมูล (overlapping) จากภาพที่ 2.3 ภาพ (a) เป็นการจับคู่ระหว่างกลุ่มข้อมูลเสียงข้างน้อย (ตัวอย่างบวกแทนด้วยสัญลักษณ์ x) และกลุ่มข้อมูลเสียงข้างมาก (ตัวอย่างลบแทนด้วยสัญลักษณ์ o) โดยทำการเปรียบเทียบในแต่ละคู่เพื่อหาระยะทางที่ใกล้กันมากที่สุดระหว่างสองกลุ่มข้อมูล ซึ่งในแต่ละคู่ตัวอย่างที่ใกล้กันมากที่สุด เรียกว่า Tomek Links จากภาพที่ 2.3 ภาพ (b) จะทำการกำจัดตัวอย่างลบออกไปในแต่ละ Tomek Links



ภาพที่ 2.3 ตัวอย่างTomek Links

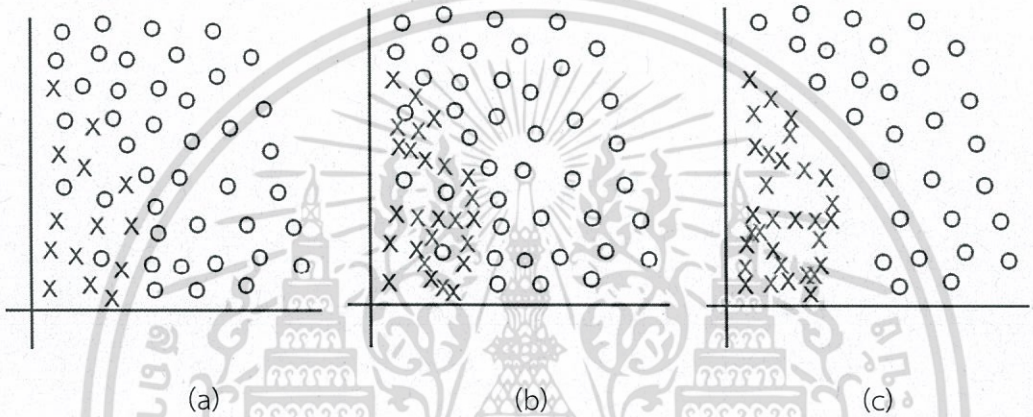
ตัวอย่างการหา Tomek Links กำหนดให้ E_i(วงกลม) และ E_j(สามเหลี่ยม) เป็นกลุ่มข้อมูลที่มีความแตกต่างกันและ (E_i,E_j) จะเป็น Tomek Links เมื่อระยะทางระหว่าง $d(E_i,E_k)$ หรือ $d(E_j,E_k)$ มีค่ามากกว่า $d(E_i,E_j)$ โดย E_k คือตัวอย่างใด ๆ ดังภาพที่ 2.4



ภาพที่ 2.4 ตัวอย่างการหา Tomek Links

2.2.3 ขั้นตอนวิธี SMOTE + Tomek Links

SMOTE + Tomek Links [3] คือเทคนิคหนึ่งในการผสมผสานระหว่างเทคนิคการเพิ่มจำนวนกลุ่มข้อมูลเสียงข้างน้อยและเทคนิคการลดจำนวนกลุ่มข้อมูลเสียงข้างมาก โดยเทคนิค SMOTE จะทำการเพิ่มจำนวนกลุ่มข้อมูลเสียงข้างน้อยให้มีจำนวนพอ ๆ กับจำนวนกลุ่มข้อมูลเสียงข้างมาก และใช้เทคนิค Tomek Links กำจัดกลุ่มข้อมูลที่เกิดการซ้อนทับกัน (overlapping) ของกลุ่มข้อมูล จากภาพที่ 2.5 ภาพ (a) คือกลุ่มข้อมูลที่เกิดความไม่สมดุลและเกิดการซ้อนทับกันของกลุ่มข้อมูลและใช้เทคนิค SMOTE ในการเพิ่มจำนวนตัวอย่างบวก ดังภาพที่ 2.5 ภาพ (b) และใช้เทคนิค Tomek Links ในการกำจัดตัวอย่างลบ ดังภาพที่ 2.5 ภาพ (c)



ภาพที่ 2.1 เทคนิค SMOTE + Tomek Links

2.3 เทคนิคการวัดประสิทธิภาพโมเดล

ในการสร้างโมเดลจำแนกประเภทข้อมูลได้นำเทคนิค *k*-fold Cross validation 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 ดังภาพที่ 2.6

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

ภาพที่ 2.6 ตัวอย่าง *k*-fold Cross validation เมื่อกำหนดค่า $k = 5$

2.4 งานวิจัยที่เกี่ยวข้อง

งานวิจัย Preprocessing of imbalanced breast cancer data using feature selection combined with over-sampling technique for classification [4] 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 [5] 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 [6] 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 %

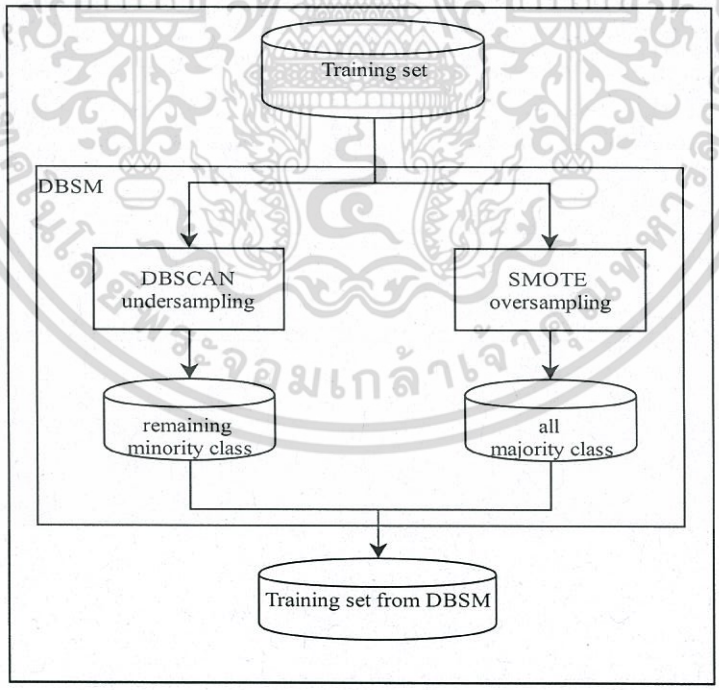
งานวิจัย Novel Algorithm for Imbalance Data Classification Based on Genetic Algorithm Improved SMOTE [7] นำเสนอการใช้เทคนิค SMOTE ร่วมกับ genetic algorithm (GA) เพื่อเพิ่มประสิทธิภาพในการแก้ปัญหาความไม่สมดุลของกลุ่มข้อมูลให้ดีขึ้น ซึ่งเทคนิค SMOTE เป็นหนึ่งในเทคนิคที่ได้รับความนิยมมากที่สุดในการแก้ปัญหาความไม่สมดุลของกลุ่มข้อมูล 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 ชุดข้อมูล เมื่อเทียบกับขั้นตอนวิธี 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



ภาพที่ 3.1 หลักการทำงานของ DBSM

จากภาพที่ 3.1 แสดงหลักการทำงานของขั้นตอนวิธี DBSM โดยเริ่มต้นชุดข้อมูลฝึกสอนจะถูกนำเข้าสู่กระบวนการสุ่มตัวอย่างด้วยขั้นตอนวิธีของ DBSM ซึ่งภายในจะประกอบไปด้วยสองเทคนิค คือ เทคนิคการลดจำนวนตัวอย่างของคลาสลบด้วยขั้นตอนวิธี DBSCAN undersampling และเทคนิคการเพิ่มจำนวนตัวอย่างบวกด้วยขั้นตอนวิธี SMOTE ดังนั้นผลลัพธ์สุดท้ายที่ได้จากขั้นตอนวิธีของ DBSM คือ จำนวนตัวอย่างที่คงเหลือของคลาสลบจากเทคนิค 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.

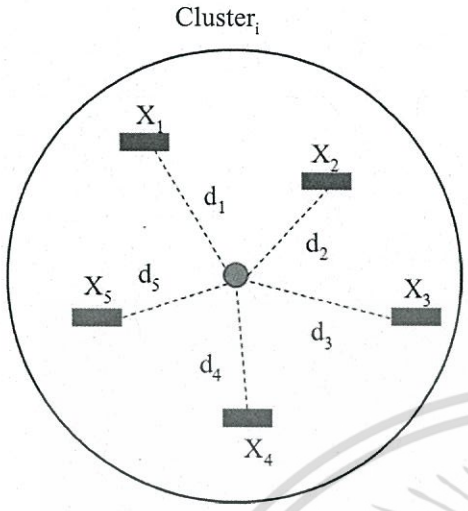
```
1.  [Cluster] = DBSCAN(S, ε, Minpts) // Cluster is a set of clusters
2.  For i = 1 to n // n is a number of clusters generated by DBSM.
3.      Let Di be a number of majority class instances in ith cluster
4.      If all members in Clusteri are majority class then
5.          Centroid = compute_centroid(Clusteri)
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
```

ภาพที่ 3.2 ขั้นตอนวิธีการทำงานของอัลกอริทึม DBSCAN undersampling

จากตัวอย่างภาพที่ 3.2 แสดงขั้นตอนวิธีของอัลกอริทึม DBSCAN undersampling โดยมีรายละเอียดดังนี้

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

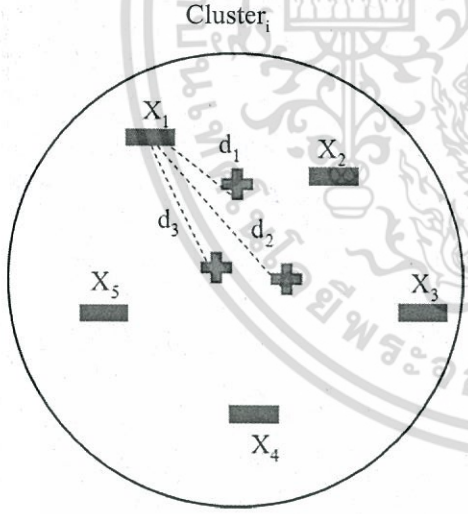
1. (บรรทัดที่ 1) นำชุดข้อมูลฝึกสอนผ่านขั้นตอนวิธี DBSCAN เพื่อจัดกลุ่มของกลุ่มข้อมูล
2. (บรรทัดที่ 2-19) การประมวลผลในแต่ละกลุ่มข้อมูล
3. (บรรทัดที่ 3) กำหนดให้ D_i คือจำนวนของกลุ่มข้อมูลเสียงส่วนมากในกลุ่มที่ i
4. (บรรทัดที่ 4) ถ้าจำนวนสมาชิกทั้งหมดในกลุ่มข้อมูลที่ i เป็นคลาสลบให้ทำงานในบรรทัดที่ 5
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



MI

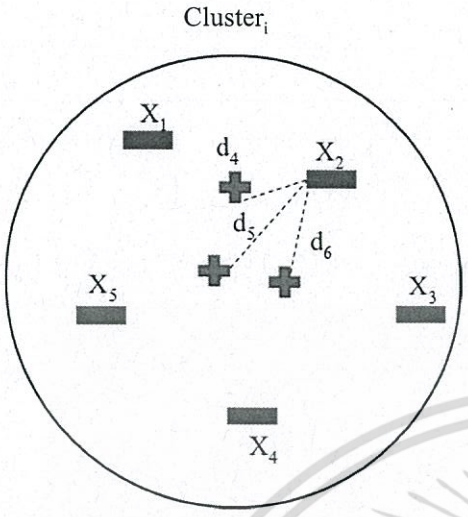
index	distance
X_1	d_1
X_2	d_2
X_3	d_3
X_4	d_4
X_5	d_5

ภาพที่ 3.3 การวัดระยะทางในกรณีที่ในกลุ่มข้อมูลมีสมาชิกเป็นคลาสลบทั้งหมด



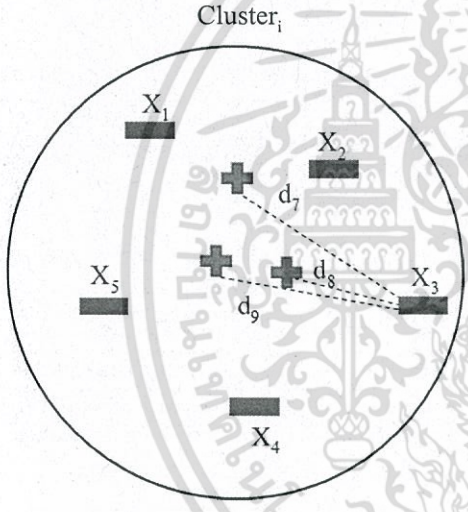
MI

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



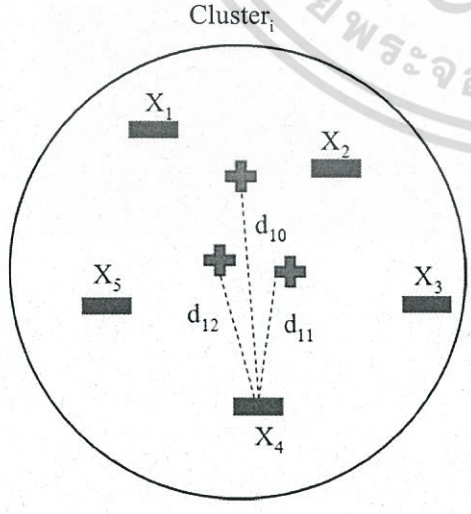
MI

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	



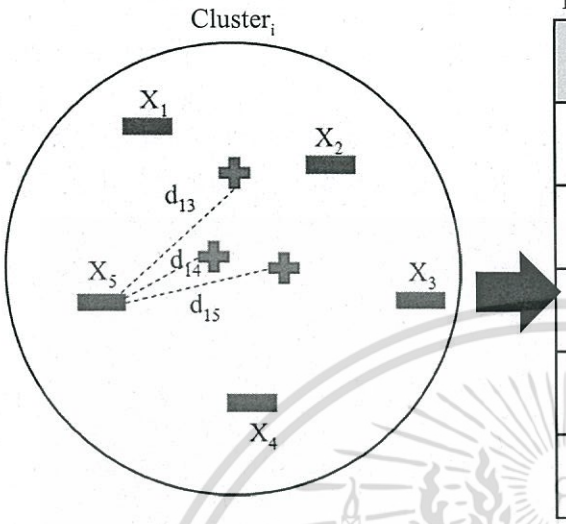
MI

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

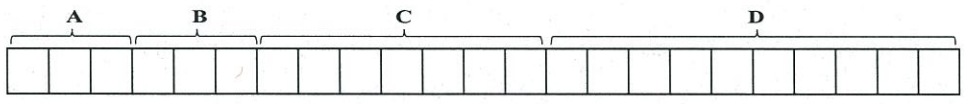
index	distance
X_1	findSmallestDistance(d_1, d_2, d_3)
X_2	findSmallestDistance(d_4, d_5, d_6)
X_3	findSmallestDistance(d_7, d_8, d_9)
X_4	findSmallestDistance(d_{10}, d_{11}, d_{12})
X_5	findSmallestDistance(d_{13}, d_{14}, d_{15})

ภาพที่ 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



ภาพที่ 3.5 การเข้ารหัสโครโมโซม

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

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

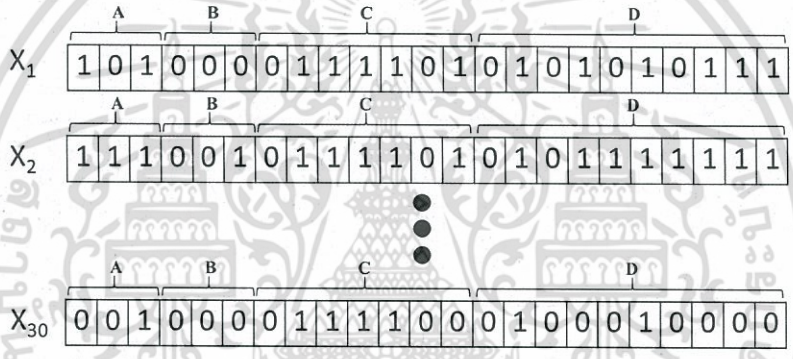
B แทนด้วย ปริมาณของการสุ่มตัวอย่างของ SMOTE ที่คิดเป็นเปอร์เซ็นต์ (*p*) โดยมีความยาวบิตสตริงเท่ากับ 3

C แทนด้วย จำนวนของจุดเพื่อนบ้านขั้นต่ำ (*Minpts*) โดยมีความยาวบิตสตริงเท่ากับ 7

D แทนด้วย ระยะทางของจุดเพื่อนบ้าน (*Eps*) โดยมีความยาวบิตสตริงเท่ากับ 10

ดังนั้นในหนึ่งโครโมโซมจะประกอบไปด้วยบิตสตริงทั้งหมด 23 บิต

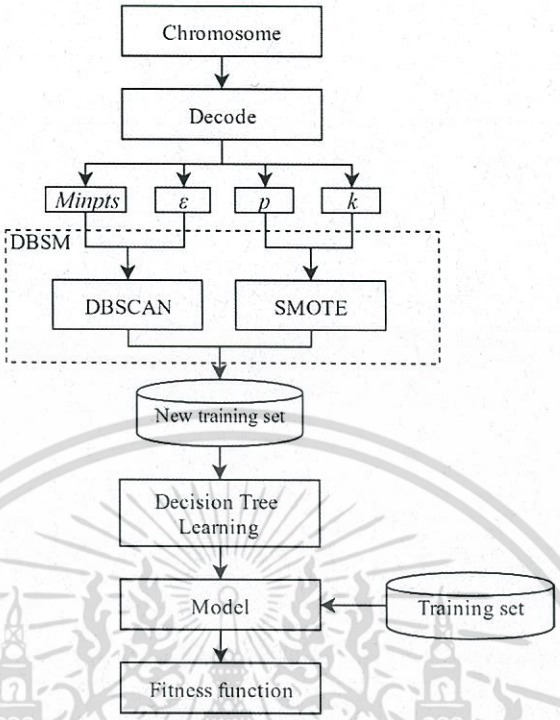
ขั้นตอนที่ 2 : สร้างประชากรในรุ่นแรกอย่างสุ่มให้มีจำนวนเท่ากับ N โดยขั้นตอนวิธี GADBSM ได้กำหนดจำนวนประชากรเท่ากับ 30 หรือ N=30 ดังนั้นประชากรในแต่ละรุ่นจะประกอบด้วยโครโมโซมทั้งหมด 30 โครโมโซม และแต่ละบิตในโครโมโซมจะถูกสุ่มด้วยเลขฐานสอง (0 หรือ 1) ดังภาพที่ 3.6



ภาพที่ 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 ซึ่งขั้นตอนการวัดประสิทธิภาพของประชากรสามารถแสดงได้ดังภาพที่ 3.7

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



ภาพที่ 3.7 ขั้นตอนการวัดประสิทธิภาพของประชากร

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) ตัวจำแนกแบบเบย์อย่างง่าย (NaiveBayes) และในส่วนสุดท้ายคือการเปรียบเทียบระหว่างเทคนิคการสุ่มตัวอย่างทั้ง 4 เทคนิค และเปรียบเทียบประสิทธิภาพของเทคนิค GADBSM ในแต่ละอัลกอริทึมการเรียนรู้

4.1 แหล่งที่มาและรายละเอียดของชุดข้อมูล

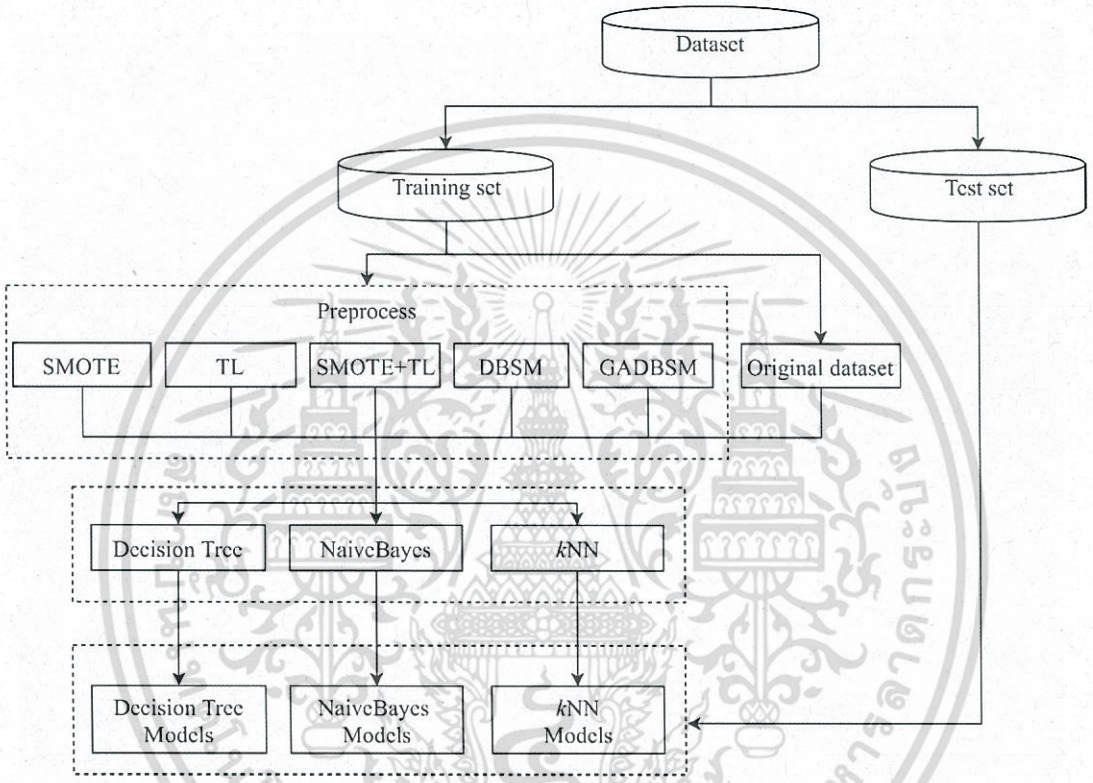
ในการทดลองนี้ได้นำข้อมูลมาจากเว็บไซต์ KEEL (www.keel.es) จำนวน 12 ชุดข้อมูล ชุดข้อมูลทั้งหมดที่นำมาทดลองมีค่าอัตราความไม่สมดุล (IR) อยู่ระหว่าง 1 ถึง 10 โดยรายละเอียดต่าง ๆ ของแต่ละชุดข้อมูลประกอบด้วย จำนวนคุณลักษณะ (attributes) จำนวนตัวอย่างทั้งหมดของแต่ละชุดข้อมูล(examples) และค่า IR แสดงได้ดังตารางที่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



ภาพที่ 4.1 ขั้นตอนการทดลอง

จากภาพที่ 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 ดังนั้นโมเดลทั้งหมดที่ใช้ในการเปรียบเทียบประสิทธิภาพของการเตรียมข้อมูลด้วยเทคนิคการ

สุ่มตัวอย่างแบบต่าง ๆ จะประกอบไปด้วยต้นไม้ตัดสินใจ ตัวจำแนกแบบเบย์อย่างง่าย และเพื่อนบ้านใกล้เคียงที่สุด k ตัว ทั้งหมดอย่างละ 6 โมเดล คือ โมเดลที่ผ่านเทคนิคการสุ่มตัวอย่างด้วยขั้นตอนวิธี SMOTE โมเดลที่ผ่านเทคนิคการสุ่มตัวอย่างด้วยขั้นตอนวิธี TL โมเดลที่ผ่านเทคนิคการสุ่มตัวอย่างด้วยขั้นตอนวิธี SMOTE+TL โมเดลที่ผ่านเทคนิคการสุ่มตัวอย่างด้วยขั้นตอนวิธี 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 มากที่สุดคือ เทคนิค 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 พบว่าเทคนิคการสุ่มตัวอย่าง 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
สรุปผลการวิจัยและข้อเสนอแนะ

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

บทที่ 6
สรุปผลผลิตที่ได้จากงานวิจัย

<u>การประชุมเชิงวิชาการระดับนานาชาติ</u> (International Conference)

1. Sanguanmak, Y., Hanskunatai, A., "**Auto-Tuning of parameters in hybrid sampling method for class imbalance problem**", 20th International Computer Science and Engineering Conference, 21 February 2017.

เอกสารอ้างอิง

[1] 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.

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

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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

ผลงานวิจัยที่ได้รับการตีพิมพ์เผยแพร่

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.

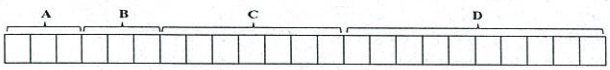


Fig. 1. The design of chromosome encoding.

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.

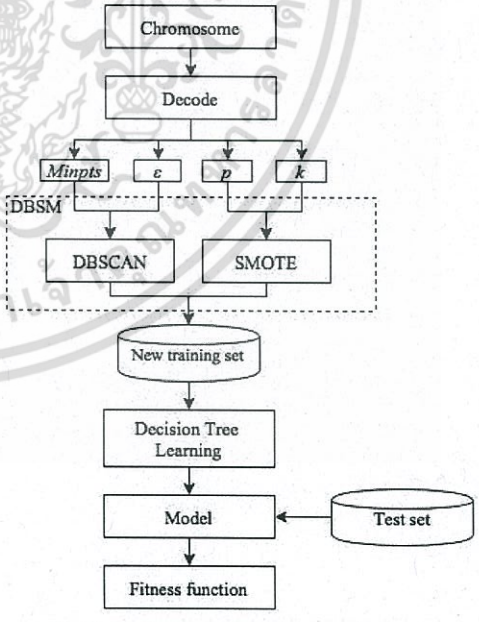


Fig. 2. The process of population evaluating.

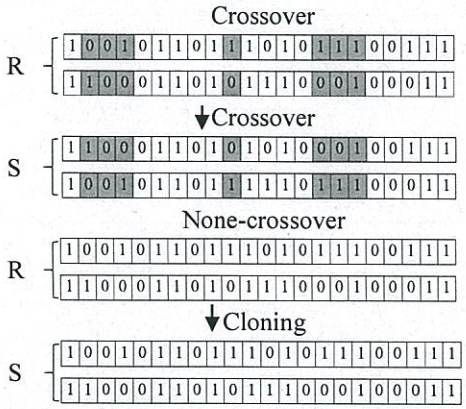


Fig. 3. Crossover and none- crossover operation.

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.

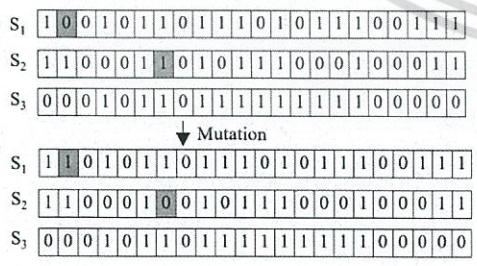


Fig. 4. Mutation operation.

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.

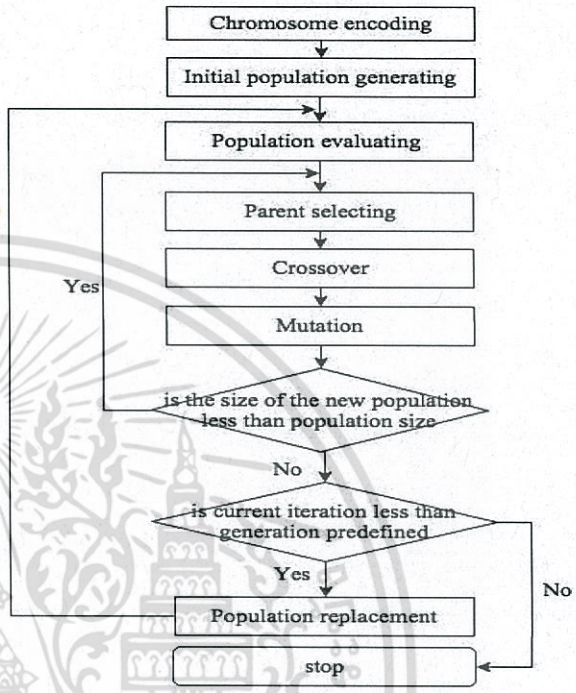


Fig. 5. The GADBSM algorithm.

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

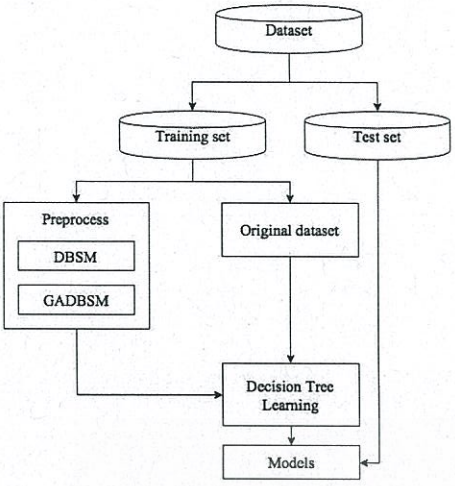


Fig. 6. The flow chart of experimental design.

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.

ประวัตินักวิจัย

ประวัติส่วนตัว

ชื่อ-สกุล นางอนันตพร หรรษคุณาฒัย

ตำแหน่งปัจจุบัน ผู้ช่วยศาสตราจารย์

ประวัติการศึกษา

ชื่อย่อปริญญา	สาขา	สถาบันที่จบ	ปีที่จบ
วศ.ด.	วิศวกรรมคอมพิวเตอร์	จุฬาลงกรณ์มหาวิทยาลัย	พ.ศ. 2551
วท.ม.	วิทยาการคณนา	จุฬาลงกรณ์มหาวิทยาลัย	พ.ศ. 2546
วท.บ.	วิทยาการคอมพิวเตอร์	มหาวิทยาลัยศิลปากร	พ.ศ. 2543

ผลงานวิจัยที่ตีพิมพ์เผยแพร่

1. Wongsirichot, T. and Hanskunatai, A., "**A Classification of Sleep Disorders with Optimal Features Using Machine Learning Techniques**", Journal of Health Research, Vol. 31, No.3, May-June, 2017, pp.209-217.
2. Sanguanmak, Y., Hanskunatai, A., "**Auto-Tuning of parameters in hybrid sampling method for class imbalance problem**", Proceeding of 2016 International Computer Science and Engineering Conference, IEEE, 2016.
3. Sanguanmak, Y., Hanskunatai, A., "**DBSM: The combination of DBSCAN and SMOTE for imbalanced data classification**", Proceeding of 13th International Joint Conference on Computer Science and Software Engineering, JCSSE 2016.
4. Wongsirichot, T. and Hanskunatai, A., "**A comparative investigation of PSG signal patterns to classify sleep disorders using machine learning techniques**", Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) vol. 9225, 2015, pp. 510–521.
5. Srisuan, J. and Hanskunatai, A., "**The ensemble of naïve bayes classifiers for hotel searching**", the 18th International Computer Science and Engineering Conference, KhonKaen, IEEE, Jan, 2015, pp. 168-173.

6. Srisuan, J. and Hunskunatai, A., "**An application of hotel searching based on opinion mining**", The 10th National Conference on Computing and Information Technology, Phuket, Thailand, 8th-9th May, 2014.
7. Hanskunatai, A., "**A new hybrid intelligent system for fast neural network training**", Lecture Notes in Computer Science, vol. 7952, Issue PART 2, 2013, pp. 331-340.
8. Srisawat, A., "**Discovery stock trading patterns: a case study of thai stock market**", International journal of Intelligent Information Processing (IJIIP), Vol. 3. No. 1, march 2012. pp. 119-127.