การปรับปรุงวิธีการสุ่มตัวอย่างใหม่สำหรับข้อมูลไม่สมดุล ด้วยเทคนิคแบบผสม DB2SM

ภาณุภณ จิระอัมพร

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมข้อมูลขนาดใหญ่
วิทยาลัยนวัตกรรมด้านเทคโนโลยีและวิศวกรรมศาสตร์
มหาวิทยาลัยธุรกิจบัณฑิตย์
พ.ศ. 2564

AN IMPROVEMENT OF OVERSAMPLING FOR IMBALANCED DATA BY COMBINED TECHNIQUE: DB2SM

PHARNUPHON JIRAAMPHORN

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Engineering

Department of Big Data Engineering,

College of Innovative Technology and Engineering,

Dhurakij Pundit University

2020

ใบรับรองงานวิทยานิพนธ์

วิทยาลัยนวัตกรรมด้านเทคโนโลยีและวิศวกรรมศาสตร์ มหาวิทยาลัยธุรกิจบัณฑิตย์

ปริญญา วิศวกรรมศาสตรมหาบัณฑิต

หัวข้อวิทยานิพนธ์ การปรับปรุงวิธีการสุ่มตัวอย่างใหม่สำหรับข้อมูลไม่สมดุลด้วยเทคนิคแบบผสม DB2SM

เสนอโดย ภาณุภณ จิระอัมพร

สาขาวิชา วิศวกรรมข้อมูลขนาดใหญ่

อาจารย์ที่ปรึกษาวิทยานิพนธ์ ดร.เอกสิทธิ์ พัชรวงศ์ศักดา

ได้พิจารณาเห็นชอบโดยคณะกรรมการสอบวิทยานิพนธ์แล้ว

.....ประธานกรรมการ

(ดร.สรรพฤทธิ์ มฤคทัต)

.....กรรมการและอาจารย์ที่ปรึกษาหลัก

(ดร.เอกสิทธิ์ พัชรวงศ์ศักดา)

.....กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.ดวงใจ จิตคงชื่น)

.....กรรมการ

(ดร.ธนภัทร ฆังคะจิตร)

วิทยาลัยนวัตกรรมด้านเทคโนโลยีและวิศวกรรมศาสตร์รับรองแล้ว

.....

(ดร.ชัยพร เขมะภาตะพันธ์)

คณบดีวิทยาลัยนวัตกรรมด้านเทคโนโลยีและวิศวกรรมศาสตร์

วันที่ 31 เดือน กรกฎาคม พ.ศ. 2564

หัวข้อวิทยานิพนธ์	การปรับปรุงวิธีการสุ่มตัวอย่างใหม่สำหรับข้อมูลไม่สมดุลด้วยเทคนิคแบบผสม DB2SM
ชื่อผู้เขียน	ภาณุภณ จิระอัมพร
อาจารย์ที่ปรึกษา	ดร. เอกสิทธิ์ พัชรวงศ์ศักดา
สาขาวิชา	วิศวกรรมข้อมูลขนาดใหญ่
ปีการศึกษา	2563

บทคัดย่อ

การจำแนกข้อมูลหรือการแบ่งกลุ่มข้อมูลจัดเป็นภารกิจที่สำคัญในกระบวนการเรียนรู้ของเครื่องจักร ไม่เพียงแต่การเลือกใช้อัลกอริธึมในการเรียนรู้ที่ดี ข้อมูลที่ดีสำหรับใช้ในการฝึกก็มีส่วนสำคัญและถือว่าเป็นปัจจัยหลักที่ส่งผลต่อประสิทธิภาพของโมเดลที่สร้างขึ้น การเผชิญหน้ากับปัญหาข้อมูลไม่สมดุลเป็นสิ่งที่หลีกเลี่ยงไม่ได้ มีวิธีการมากมายได้ถูกคิดค้นขึ้นเพื่อจัดการและแก้ไขปัญหานี้ ทั้งแบบที่เป็นเทคนิคเชิงเดี่ยว เช่น ROS, SMOTE, Tomek-Link หรือเทคนิคแบบผสมที่เกิดจากการนำเทคนิคเชิงเดี่ยวมาใช้งานร่วมกับเทคนิคทางสถิติอื่นในการประมวลผล เช่น SMOTEBoost, Over-Bagging, Under -Bagging, IIVotes เพื่อให้ได้ชุดข้อมูลที่ดีเพียงพอสำหรับการเรียนรู้ งานวิจัยนี้นำเสนอวิธีการเพื่อปรับปรุงการสุ่มตัวอย่างใหม่ของกลุ่มตัวอย่างที่มีลักษณะของข้อมูลไม่สมดุล โดยใช้เทคนิคแบบผสมที่เรียกว่า DB2SM โดยประสานการทำงานระหว่างการแบ่งกลุ่มข้อมูลด้วยเทคนิค DBSCAN และการสังเคราะห์ตัวอย่างใหม่ด้วยเทคนิค SMOTE ที่สามารถปรับใช้งานได้ง่าย และใช้เวลาในการประมวลผลไม่มากนัก เพื่อค้นหาบริเวณที่ดีที่สุดของ Minority Class ที่เหมาะสมสำหรับการสร้างตัวอย่างใหม่ จากผลการทดลองกับชุดข้อมูลไม่สมดุลของ UCI หลายชุด และเปรียบเทียบประสิทธิภาพของโมเดลที่ได้จากใช้ชุดข้อมูลฝึกที่ได้จากเทคนิคอื่น ได้แก่ SMOTE, DBCS และ DBSM ซึ่งเป็นเทคนิคที่มีลักษณะการทำงานที่คล้ายกันพบว่า ชุดข้อมูลฝึกใหม่ที่ได้จากเทคนิค DB2SM ส่งผลต่อประสิทธิภาพของโมเดลที่สร้างขึ้นอย่างมีนัยสำคัญ

Thesis Title	An Improvement of Oversampling for Imbalanced Data by Combined Technique: DB2SM
Author	Pharnuphon Jiraamphorn
Thesis Advisor	Dr. Eakasit Pacharawongsakda
Department	Big Data Engineering
Academic Year	2020

ABSTRACT

Data classification and clustering are importance tasks in machine learning. Only the optimal classifier was insufficient to gain the best result because of the impact from provided train data. We inevitably have to face an imblance problems. Resampling is the most used technique to handle an imbalanced data as single technique such as ROS, SMOTE, Tomek-links or ensemble technique such as SMOTEBoost, Over-Bagging, Under -Bagging, IIVotes. In this paper we suggest an improvement of combined technique to oversampling called DB2SM. We assume to generate synthetic instances in the best area or cluster of minority class will yield high quality instances. The process locate the area by using DBSCAN technique and applying SMOTE for upsampling data. After completion experimented with UCI Imbalanced datasets and compared with SMOTE, DBCS and DBSM. The result showed the significant model's performance from using the new train data from proposed methodology.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงด้วยการให้คำแนะนำช่วยเหลือของอาจารย์ ดร. เอกสิทธิ์ พัชรวงศ์ศักดา ที่กรุณาให้ข้อคิดเห็น ตรวจสอบ ปรับปรุง แก้ไขในกระบวนงานวิจัยมาโดยตลอด ผู้วิจัยจึงของกราบขอบพระคุณไว้ ณ โอกาสนี้

ผู้วิจัยขอกราบขอบพระคุณอาจารย์ทุกท่านที่สละเวลาเพื่อมาเป็นกรรมการในการสอบวิทยานิพนธ์ และให้คำแนะนำในการปรับปรุงแก้ไขที่เป็นประโยชน์ต่อการวิจัย รวมถึงขอขอบคุณ คุณกุลธิดา รอดบุญ และเจ้าหน้าที่บัณฑิตวิทยาลัยทุกท่านที่ให้ความช่วยเหลือ ประสานงาน และอำนวยความสะดวก ในระหว่างการศึกษาที่ผ่านมาตลอดจนทำวิทยานิพนธ์สำเร็จสมบูรณ์

ภาณุภณ จิระอัมพร

สารบัญ

สารบัญ(ต่อ)

สารบัญตาราง

ตารางที่ **หน้า**

สารบัญภาพ

ภาพที่ **หน้า**

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของปัญหา

การจัดการกับข้อมูลไม่สมดุล (Imbalanced Data) เป็นสิ่งที่ท้าทายอย่างมากในการศึกษาด้านการเรียนรู้ของเครื่องจักร (Machine Learning) เพราะในความเป็นจริงแล้ว ข้อมูลที่เราสนใจหรือต้องการศึกษาส่วนใหญ่มักจะเป็นกลุ่มข้อมูลส่วนน้อย (Minority) ซึ่งถูกพบในปริมาณที่น้อยกว่าเมื่อเทียบกับปริมาณของข้อมูลตัวอย่างที่เหลือทั้งหมดหรือข้อมูลส่วนมาก (Majority) เช่น การค้นหาการทุจริตทางการเงิน (Financial Fraud Detection) การวิเคราะห์การโจมตีทางเครือข่าย (Network Intrusions Analysis) การวินิจฉัยโรค (Disease Diagnosis) เป็นต้น วิธีในการรับมือกับปัญหาความข้อมูลไม่สมดุลนี้ ถูกคิดค้นพัฒนามาอย่างต่อเนื่อง โดยส่วนใหญ่แบ่งออกเป็น 2 ระดับ ได้แก่ ระดับ Data Level คือการประมวลผลข้อมูลตัวอย่างที่นำมาใช้งาน เพื่อพิจารณาว่าตัวอย่างใดควรจะถูกนำออก และเมื่อใดควรจะสร้างตัวอย่างเทียมเพิ่มเข้ามา เพื่อให้จำนวนตัวอย่างของกลุ่ม Majority และ Minority มีจำนวนที่ใกล้เคียงหรือเท่ากัน วิธีการนี้ถูกเรียกว่า การสุ่มตัวอย่างใหม่ (Resampling) เทคนิคการสุ่มตัวอย่างใหม่ที่รู้จักกันอย่างแพร่หลาย เช่น SMOTE, ROS, ROSE, Tomek-link เป็นต้น วิธีการแก้ไขปัญหาข้อมูลไม่สมดุลในระดับถัดมา คือระดับ Algorithm Level ซึ่งเป็นการใช้เครื่องมือทางสถิติหรืออัลกอริธึมทางการเรียนรู้ของเครื่องจักรเข้ามาช่วยในการจำแนกความแตกต่างระหว่าง Majority และ Minority เพื่อป้องกันไม่ให้เกิดความเอนเอียงของการจำแนกไปทางกลุ่มหนึ่งกลุ่มใดมากเกินไป ตัวอย่างของอัลกอริธึมที่ถูกนำมาใช้อย่างแพร่หลาย ได้แก่ อัลกอริธึมพวก Clustering หรือ Support Vector Machine (SVM) หรือ One Class Learning เป็นต้น และในระดับ Algorithm Level นี้ยังแตกออกเป็นแนวทางต่างๆ อีกหลายแนวทาง เช่น การทำ Feature Selection เพื่อช่วยปรับปรุงประสิทธิภาพในการเรียนรู้ของ Classifier ให้ดีขึ้น การทำ Cost Sensitive Learning โดยการปรับเพิ่มค่าความสำคัญให้กับสมาชิกของกลุ่ม Minority ที่มีจำนวนสมาชิกน้อยกว่า ในขณะที่ปรับลดค่าความสำคัญของสมาชิกของกลุ่ม Majority ที่มีจำนวนมากกว่าให้มีบทบาทน้อยลง เพื่อลดความเหลื่อมล้ำระหว่างสองกลุ่มให้ได้มากที่สุด ตัวอย่างอัลกอริธึมที่นำมาใช้งาน เช่น อัลกอริธึมที่ปรับปรุงมาจาก AdaBoost Learning Framwork เป็นต้น นอกจากนี้ ยังมีการใช้เทคนิค Ensemble Learning โดยการนำอัลกอริธึมประเภทตัว Classifier ต่างๆ มาทำงาน

ร่วมกับเทคนิคอื่นมากกว่าหนึ่งเทคนิคเพื่อให้ได้ผลลัพธ์ที่ดีขึ้น เช่น เทคนิค SMOTEBoost เทคนิค Over-Bagging เทคนิค Under -Bagging ซึ่งแนวทางนี้เป็นที่มาของการแก้ปัญหาในลักษณะที่เรียกว่า เทคนิคแบบผสม (Combined Technique หรือ Hybrid Technique) นั่นเอง

การแก้ปัญหาข้อมูลไม่สมดุลแต่ละวิธีมีข้อดีและข้อด้อยที่แตกต่างกันไป เช่น เทคนิค SMOTE จำเป็นต้องคำนึงถึงคุณภาพของตัวอย่างที่สร้างขึ้นมาใหม่ ซึ่งขึ้นอยู่กับกลุ่มข้อมูลตัวอย่างที่นำมาใช้อ้างอิง ถ้าหากในกลุ่มตัวอย่างประกอบด้วยข้อมูลอยู่ อาจส่งผลต่อความน่าเชื่อถือของข้อมูลที่สังเคราะห์ขึ้นตามไปด้วย หรือการใช้เทคนิค Ensemble Learning ถึงแม้จะช่วยให้ได้ค่าความแม่นยำที่ดีขึ้น แต่ต้องแลกมาด้วยขั้นตอน ที่ยุ่งยากซับซ้อน และใช้เวลาในการประมวลผลที่ยาวนานขึ้น (Ali, 2019)

จุดมุ่งหมายของงานวิจัยนี้ คือการนำเสนอแนวทางการปรับปรุงเทคนิคแบบผสมสำหรับการทำการสุ่มตัวอย่างใหม่ เรียกว่า DB2SM เพื่อจะแก้ไขจุดบกพร่อง และเพิ่มประสิทธิภาพของการสุ่มตัวอย่างใหม่ในข้อมูลไม่สมดุล โดยใช้เทคนิคการแบ่งกลุ่มแบบ DBSCAN ร่วมกับเทคนิคการทำ Oversampling แบบ SMOTE ซึ่งสามารถปรับใช้งานได้ง่าย

1.2 วัตถุประสงค์ของการวิจัย

1. ออกแบบเทคนิคแบบผสมสำหรับปรับปรุงการสุ่มตัวอย่างใหม่ เพื่อแก้ไขปัญหาข้อมูลไม่สมดุล

2. เปรียบเทียบประสิทธิภาพของโมเดลที่สร้างจากการใช้ผลลัพธ์ของเทคนิคที่ออกแบบขึ้น กับโมเดลที่สร้างจากการใช้ผลลัพธ์ของเทคนิคอื่นที่มีการทำงานในลักษณะคล้ายกัน โดยเปรียบเทียบจากค่า Accuracy, AUC และ F-measure

1.3 สมมติฐานการวิจัย

1. บริเวณที่มีตัวอย่างในกลุ่ม (Class) เดียวกันจับกลุ่มกันอยู่มากที่สุด หรือคลัสเตอร์ที่ใหญ่ที่สุด จะเป็นบริเวณที่เหมาะสมสำหรับการสังเคราะห์ข้อมูลตัวอย่างเทียม โดยวัดจากประสิทธิภาพของโมเดลที่ใช้ข้อมูลตัวอย่างเดิมร่วมกับข้อมูลที่สังเคราะห์ขึ้นมาใหม่ในการเรียนรู้

1.4 ขอบเขตงานวิจัย

1. ข้อมูลที่ใช้ในการทดลองเบื้องต้นเป็นชุดข้อมูลไม่สมดุลจาก UCI Machine Learning Repository และข้อมูลที่ใช้ในการทดลองเพิ่มเติมเป็นชุดข้อมูลไม่สมดุลจาก KEEL (Knowledge Extraction based on Evolutionary Learning) - Dataset Repository

2. ข้อมูลที่ใช้ในการทดลองไม่มีข้อมูลที่สูญหาย (Non-Missing Values)

3. ข้อมูลที่ใช้ในการทดลองแต่ละชุดจะประกอบด้วยสมาชิกเพียง 2 Class เท่านั้น

4. เปรียบเทียบประสิทธิภาพของโมเดลที่เรียนรู้โดยการใช้ข้อมูลที่ผ่านการปรับปรุงด้วยเทคนิคที่พัฒนาขึ้น กับโมเดลที่ใช้ข้อมูลที่ผ่านการปรับปรุงด้วยเทคนิคอื่นที่มีลักษณะการทำงานคล้ายคลึงกัน 3 เทคนิค ได้แก่ เทคนิค SMOTE เทคนิค DBCS และเทคนิค DBSM โดยการเปรียบเทียบค่า Accuracy, AUC และ F-measure

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้เทคนิคการทำ Oversampling ที่สามารถแก้ไขปัญหาข้อมูลไม่สมดุล

2. สามารถเพิ่มประสิทธิภาพของโมเดลที่เรียนรู้โดยการใช้ข้อมูลที่ผ่านการปรับปรุงด้วยเทคนิคที่พัฒนาขึ้น โดยวัดจากค่า Accuracy, AUC และ F-measure

บทที่ 2
แนวคิด ทฤษฎีและผลงานวิจัยที่เกี่ยวข้อง

ข้อมูลไม่สมดุล (Imbalanced Data) เป็นปรากฏการณ์ที่เกิดขึ้นกับการในการทำงานด้านการประมวลผลข้อมูล (Data Processing) อย่างหลีกเลี่ยงได้ยาก ทั้งนี้ ลักษณะของปัญหาที่เกิดขึ้น มีรูปแบบที่หลากหลาย และมีรายละเอียดที่แตกต่างกันออกไป การปรับปรุงข้อมูลหรือการแก้ไขปัญหาให้ได้ผล จำเป็นต้องอาศัยแนวทางที่เหมาะสมแตกต่างกันออกไป เพื่อให้บรรลุวัตถุประสงค์ของการวิจัย จำเป็นจะต้องมีความรู้ความเข้าใจพื้นฐานที่เกี่ยวข้องกับปัญหา รวมถึงเครื่องมือ และวิธีการที่จะนำมาใช้ประเมินความสำเร็จหรือประสิทธิภาพของผลลัพธ์ที่ได้จากการปรับใช้แนวทางหรือวิธีการที่ออกแบบขึ้น ดังนั้น ในบทนี้จึงเป็นการรวบรวมแนวคิด ทฤษฎี และผลงานวิจัยที่เกี่ยวข้องที่สอดคล้องกับวัตถุประสงค์ และอยู่ภายใต้ขอบเขตของงานวิจัย ประกอบด้วยหัวข้อ ดังนี้

2.1 แนวคิดข้อมูลไม่สมดุล
2.2 แนวทางการแก้ปัญหาข้อมูลไม่สมดุล
2.3 ระยะทางแบบยูคลิ
2.4 แนวคิด SMOTE
2.5 แนวคิด DBSCAN
2.6 ผลงานวิจัยที่เกี่ยวข้อง

2.1 แนวคิดข้อมูลไม่สมดุล

ในการเก็บรวบรวมข้อมูลสำหรับการนำมาประมวลผลนั้น โดยปกติจะพบว่าข้อมูลจะถูกแบ่งออกเป็น 3 กลุ่ม ได้แก่ กลุ่มแรกคือข้อมูลโดยปกติหรือข้อมูลส่วนใหญ่ของกลุ่มตัวอย่างนั้นเป็นข้อมูลที่เกิดขึ้นจากเหตุการณ์ในสภาวะปกติ เช่น ข้อมูลการทำธุรกรรมการเงินผ่านธนาคาร หรือข้อมูล packet ที่ส่งผ่านทางระบบเครือข่าย เราเรียกข้อมูลในกลุ่มนี้ว่าข้อมูลส่วนใหญ่ หรือ Majority กลุ่มที่สองคือข้อมูลที่เราให้ความสนใจหรือสังเกต โดยทั่วไปมักจะเกิดขึ้นจากเหตุการณ์ที่ไม่ปกติ หรือมีความถี่ในการเกิดขึ้นน้อยครั้ง เช่น ข้อมูลการทุจริตในการทำธุรกรรมการเงิน หรือข้อมูลการโจมตีทางระบบเครือข่าย เราเรียกข้อมูลในกลุ่มนี้ว่า ข้อมูลส่วนน้อย หรือ Minority และ

กลุ่มที่สามคือข้อมูลรบกวน (Noise หรือ Outlier) ซึ่งส่วนใหญ่จะเป็นข้อมูลที่มีความผิดพลาด ไม่ครบถ้วนสมบูรณ์ หรือมีค่าของข้อมูลที่ผิดปกติแปลกแยกออกไป บางครั้งอาจรวมถึงข้อมูลที่กำกวม (Ambiguous) ด้วยเช่นกัน ปรากฏการณ์ที่จำนวนตัวอย่าง Mojority มีมากกว่า Minority ในอัตราส่วนที่แตกต่างกันมากนี้ ถูกเรียกว่า ข้อมูลไม่สมดุล (Imbalanced Data) โดยสามารถแบ่งลักษณะของความไม่สมดุลออกเป็น 3 ลักษณะ ดังนี้

1. ข้อมูลมีจำนวนน้อยมาก หมายถึงจำนวนตัวอย่างข้อมูลในกลุ่ม Minority มีจำนวนน้อยกว่าจำนวนตัวอย่างข้อมูลในกลุ่ม Majority เป็นอย่างมาก ซึ่งถือเป็นเรื่องปกติที่พบในการเก็บรวบรวมข้อมูลโดยทั่วไป

2. ข้อมูลมีการซ้อนทับกัน (Class Overlapping) โดยปกติข้อมูลที่ดีจะต้องมีค่าของตัวอย่างในแต่ละกลุ่มที่มีลักษณะที่แตกต่างกัน และสามารถแบ่งแยกได้อย่างชัดเจน แต่ก็มีบางครั้งที่ข้อมูลตัวอย่างของแต่ละกลุ่มบางตัว อาจจะมีค่าที่ใกล้เคียงหรือเหมือนกัน ซึ่งลักษณะเช่นนี้ เรียกว่าข้อมูลกำกวม (Abigious Data) ซึ่งเมื่อนำไปประมวลผล ข้อมูลประเภทนี้มักจะถูกคัดออก และส่งผลให้จำนวนตัวอย่างของ Minority ยิ่งมีจำนวนลดน้อยลงไปอีก

3. ข้อมูลกระจายตัวมากเกินไป (Scattered Data) หรือแปลกแยกออกไป (Isolated Data) คือการที่ข้อมูลตัวอย่างของกลุ่ม Minority กระจัดกระจายออกเป็นกลุ่มเล็กกลุ่มน้อยอยู่ท่ามกลางกลุ่มตัวอย่าง Majority ส่งผลให้ความสามารถในการทำนายของโมเดลที่สร้างขึ้น มีโอกาสที่จะทำนายเป็น Majority มากขึ้น ส่วนข้อมูลที่แยกตัวออกไป (Isolated) จากกลุ่มของข้อมูลตัวอย่างส่วนใหญ่ หรืออยู่ห่างจากบริเวณที่มีกลุ่มข้อมูลเดียวกันอยู่กันอย่างความหนาแน่นนั้น ก็ทำให้มีโอกาสที่จะถูกโมเดลมองว่าเป็นข้อมูล Noise หรือ Outlier ได้

จากลักษณะความไม่สมดุลของข้อมูลที่กล่าวมานั้น ไม่ว่าจะเกิดขึ้นในรูปแบบใดก็ตาม เมื่อนำข้อมูลนั้นมาประมวลผลหรือทำการวิเคราะห์ ก็อาจจะทำให้เกิดความเอนเอียง (Biased) ไปตามกลุ่มของ Majority ที่มีจำนวนมากกว่าได้ ตัวอย่างเช่น การสร้างโมเดลเพื่อทำนายหรือค้นหาความผิดปกติของการรับ-ส่งข้อมูลในระบบเครือข่าย โดยเรียนรู้จากข้อมูลที่มีจำนวนตัวอย่างของ Majority มากกว่า Minority ในอัตราส่วนที่แตกต่างกันเป็นอย่างมาก ก็จะส่งผลให้โมเดลมีโอกาสในการทำนายข้อมูลเป็นเหตุการณ์ปกติมากกว่า เพราะโมเดลรู้จักข้อมูลจาก Majority มากกว่าข้อมูล Minority ที่เราสนใจนั่นเอง

เราสามารถวัดอัตราความไม่สมดุลของข้อมูล (Imbalance Ratio หรือ IR) ได้จากการคำนวณในสมการนี้

$$IR = \frac{number\ of\ Majority}{number\ of\ Minority} \qquad (1)$$

เมื่ออัตราความไม่สมดุลของข้อมูลมีค่าเท่ากับ 1 แสดงว่าจำนวนตัวอย่างของข้อมูลทั้งสองกลุ่มมีจำนวนเท่ากัน และถ้าหากอัตราความไม่สมดุลของข้อมูลมีค่ามากขึ้น แสดงถึงอัตราความแตกต่างของความไม่สมดุลในข้อมูลที่มากขึ้น และอาจส่งผลต่อประสิทธิภาพของโมเดลที่ได้จากการใช้ข้อมูลชุดนั้นด้วย

นอกจากค่าของอัตราความไม่สมดุลของข้อมูลแล้วยังมีค่าหนึ่งที่น่าสนใจ นั่นคือค่า Lack of Information for the minority class (LI) หมายถึงค่าความต่างของจำนวนตัวอย่างระหว่างกลุ่ม *positive* และ *negative* คำนวณได้จากสมการด้านล่าง ค่า LI ถูกใช้สำหรับกำหนดจำนวนในการทำสุ่มตัวอย่างใหม่ (Resampling) เพื่อให้จำนวนตัวอย่างของทั้งสองกลุ่มมีจำนวนที่ใกล้เคียงหรือเท่ากัน

$$LI = number\ of\ Majority - number\ of\ Minority \qquad (2)$$

2.2 แนวทางการแก้ปัญหาข้อมูลไม่สมดุล

การจัดการกับปัญหาข้อมูลไม่สมดุลได้ถูกค้นคว้าพัฒนาวิธีการอย่างต่อเนื่อง ตั้งแต่แก้ไขที่วิธีการเก็บรวบรวมข้อมูลเพิ่มเติม การหาวิธีสังเคราะห์ข้อมูลขึ้นมาใหม่ ตลอดไปจนถึงการปรับปรุงอัลกอริธึมที่นำมาใช้ในการประมวลผลข้อมูลที่ไม่สมดุลนี้ ในปัจจุบันสามารถแบ่งแนวทางในการแก้ไขปัญหาออกเป็น 2 ระดับ ดังนี้

2.2.1 Data Level การเกิดปัญหาข้อมูลไม่สมดุล ในระดับ Data level ส่วนใหญ่มักจะไม่สามารถแก้ไขได้ด้วยวิธีการเก็บข้อมูลตัวอย่างเพิ่ม อันเนื่องมาจากธรรมชาติของกลุ่มตัวอย่างนั้น เช่น ข้อมูลการจราจรในระบบเครือข่ายที่มี Data Packet เกิดขึ้นตลอดเวลาจำนวนมาก ในขณะที่ข้อมูลที่เกิดจากการถูกโจมตีด้วยไวรัสหรือมัลแวร์จะมีจำนวนที่น้อยกว่ามากๆ ถึงแม้จะทำการเก็บข้อมูลที่ผิดปกติหรือ Minority มาเพิ่ม ข้อมูล Majority ก็จะมากขึ้นตามมาด้วย วิธีการแก้ปัญหาที่นิยมใช้กันมากที่สุดคือ การสุ่มตัวอย่างใหม่ (Resampling)

การสุ่มตัวอย่างใหม่คือการตัดสินใจว่าข้อมูลตัวอย่างใดจะถูกนำออกจากกลุ่มของ Majority และเมื่อใดควรจะสร้างหรือเพิ่มข้อมูลใหม่เข้าไปในกลุ่มของ Minority โดยมีวัตถุประสงค์คือ ปรับจำนวนสมาชิกของทั้งสองกลุ่มให้มีจำนวนใกล้เคียงหรือเท่ากัน

2.2.1.1 Oversampling เป็นวิธีการเพิ่มจำนวนสมาชิกในกลุ่มของ Minority ขึ้นมาให้มีจำนวนใกล้เคียงกับ Majority วิธีที่ง่ายที่สุดคือการสุ่มจากตัวอย่างเดิม (Random Oversampling หรือ ROS) แต่การสุ่มนี้ทำให้มีโอกาสได้ตัวอย่างเดิมซ้ำกันเป็นจำนวนมากได้ และนำไปสู่การเกิด Over-fitting ซึ่งทำให้ผลการจำแนกของโมเดลที่ได้จากข้อมูลนี้มีความลำเอียงสูง (Biased) ต่อมาได้มีการเสนอวิธีการที่เรียกว่า SMOTE (Synthetic Minority Oversampling Technique) ซึ่งเป็นวิธีการสังเคราะห์หรือสร้างสมาชิกใหม่ขึ้นมาในกลุ่ม โดยอ้างอิงจากสมาชิกและกลุ่มของสมาชิกที่อยู่ใกล้เคียงกัน (Nearest Neighbors) เพื่อป้องกันการเกิด Over-fitting จากการเรียนรู้ตัวอย่างซ้ำๆ ต่อมาได้มีการปรับปรุงวิธีการ SMOTE ให้มีความเหมาะสมกับกลุ่มตัวอย่างที่มีลักษณะที่หลากหลายออกไป เช่น Borderline-SMOTE เพื่อแก้ปัญหาจุดบอดและเพิ่มคุณภาพในการสร้างตัวอย่างให้ดีขึ้น

2.2.1.2 Undersampling คือการเลือกสมาชิกในกลุ่มของ Majority ที่มีจำนวนมากกว่าออกไปบางส่วน เพื่อให้เหลือสมาชิกใกล้เคียงกับ Minority วิธีการที่ง่ายที่สุดคือการสุ่มออก (Random Undersampling) แต่มีโอกาสที่จะเลือกข้อมูลตัวอย่างที่มีศักยภาพออกไป ในทางปฏิบัติควรพยายามหลีกเลี่ยงการใช้วิธีนี้ ตัวอย่างอัลกอริธึม Undersampling ที่นิยม เช่น Tomek Links ซึ่งทำงานโดยการตัดตัวอย่าง Majoritiy ที่อยู่ใกล้กับ Minority ออก เพื่อลดความสับสนระหว่างข้อมูลทั้งสองกลุ่ม

2.2.1.3. Mixed Sampling การสุ่มตัวอย่างใหม่ทั้งสองวิธีข้างต้น มีความเหมาะสมในการใช้งานที่แตกต่างกัน จึงมีการศึกษาในการนำทั้งสองวิธีมาทำงานร่วมกัน เช่น SMOTE-Tomek Links โดยการใช้ Tomek Links ในการตัดตัวอย่าง Majoritiy ที่อยู่ใกล้กับ Minority ในบริเวณที่มีการทับซ้อนกันออกไป เพื่อให้มองเห็นขอบเขตระหว่างทั้งสองกลุ่มได้ชัดเจนยิ่งขึ้น ก่อนที่จะทำการ Oversampling ด้วย SMOTE ซึ่งในบางครั้งการทำ Mixed sampling อาจให้ผลลัพธ์ที่ดีขึ้นกว่าการใช้การสุ่มตัวอย่างเพียงวิธีเดียว

2.2.2 Algorithm Level ในบางสถานการณ์การแก้ปัญหาข้อมูลไม่สมดุลในระดับ Data Level อาจให้ผลลัพธ์ที่ไม่ดีพอต่อการนำไปใช้งาน หรือมีข้อจำกัดในการสุ่มตัวอย่างใหม่ จึงได้มีการนำวิธีการทางสถิติเข้ามาช่วยในการจัดการกับข้อมูล เพื่อช่วยให้สามารถนำข้อมูลที่ปรับแต่งแล้วนั้นไปประมวลผลต่อได้ โดยมีวิธีที่นิยมใช้ในปัจจุบัน ได้แก่

2.2.2.1 Cost-sensitive Algorithm เป็นการเพิ่มตัวแปรมูลค่า (Cost) หรือน้ำหนัก (Weight) ให้กับตัวอย่าง โดยให้ความสำคัญแก่กลุ่ม Minority มากขึ้น ในขณะที่ลดความสำคัญของ Majority ลง ก่อนที่จะนำข้อมูลไปประมวลผลด้วยอัลกอริธึมพวก Classification เช่น Decision tree หรือ Neural network เพื่อปรับค่าน้ำหนักให้ได้ผลลัพธ์ที่ดีที่สุด ก่อนที่จะนำข้อมูลที่ได้รับการปรับปรุงค่าน้ำหนักแล้วนั้นไปใช้งาน

2.2.2.2 One-class Learning Algorithm โดยปกติโมเดลที่ดีควรจะต้องเรียนรู้จากข้อมูลทั้งสองกลุ่มอย่างเพียงพอ แต่ถ้าหากจำนวนสมาชิกของกลุ่ม Minority มีจำนวนน้อยมากเกินไป และไม่สามารถปรับปรุงข้อมูลด้วยวิธีอื่นได้ การใช้เทคนิค One-class Learning น่าจะเป็นทางออกที่น่าสนใจ เทคนิคนี้คือการให้โมเดลเรียนรู้ตัวอย่างจากกลุ่มใดกลุ่มหนึ่งเพียงกลุ่มเดียว เพื่อให้โมเดลมีความสามารถในการจำแนกตัวอย่างที่มีลักษณะเหมือนหรือคล้ายคลึงกันกับกลุ่มที่เรียนรู้นั้นเป็นประเภทเดียวกัน และอนุมานว่าตัวอย่างอื่นที่เหลือเป็นประเภทที่ตรงกันข้าม

2.2.2.3 Feature Selection Algorithm การประมวลผลข้อมูลคือการนำตัวแปร (Variable) หรือ Attribute หรือ Feature ที่ประกอบกันเป็นข้อมูลนั้น มาวิเคราะห์เปรียเทียบกันในเชิงสถิติ ไม่เฉพาะเจาะจงว่าข้อมูลนั้นจะเป็นข้อมูลไม่สมดุลหรือไม่ หากข้อมูลมีลักษณะที่เรียกว่า High-dimensional Data ยิ่งทำให้การประมวลผลมีความซับซ้อนมากขึ้น ซึ่งในทางปฏิบัติแล้ว ตัวแปรแต่ละตัวนั้นจะให้ผลต่อการเรียนรู้ที่เหมือนหรือแตกต่างกันได้ จึงเป็นที่มาของการทำ Feature Selection เพื่อพิจารณาว่า ตัวแปรใดที่ส่งผลต่อการเรียนรู้ของโมเดลอย่างแท้จริง หรือตัวแปรใดสามารถตัดออกจากการนำมาประมวลผลได้

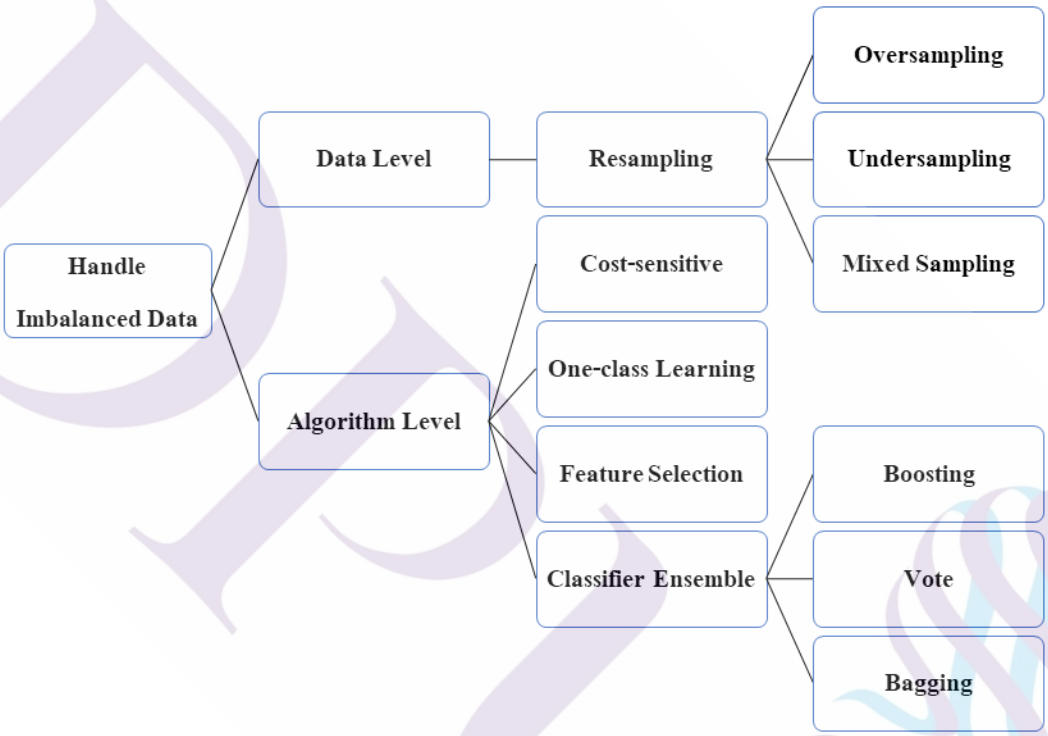
การนำวิธีนี้มาปรับใช้ในการแก้ปัญหาข้อมูลไม่สมดุลมักจะถูกใช้เมื่อพบว่าการกระจายตัวของ Majority และ Minority ไม่สามารถทำการแบ่งแยกออกจากกันได้ หรือหาขอบเขตที่จะจำแนกได้ยาก จึงแก้ไขโดยหันมาพิจารณาจากการกระจายตัวของข้อมูลตามค่าของแต่ละตัวแปรที่ละตัวแทน แล้วเลือกตัวแปรที่ส่งผลในการจำแนกได้ดีมาใช้

2.2.2.4. Classifier Ensemble Algorithm เป็นวิธีการที่ถูกนำมาใช้ในขั้นตอนของการสร้างโมเดลที่จะใช้จำแนกข้อมูลไม่สมดุล เพื่อให้โมเดลมีประสิทธิภาพในการจำแนกที่ดีขึ้น การใช้ Classifier Ensemble มักจะถูกใช้ร่วมกันกับวิธีการอื่นด้วย เช่น การทำ Cost-sensitive หรือ Feature Selection มาก่อน เพื่อชดเชยเวลาที่ต้องใช้จากการทำงานของ Classifier ที่เพิ่มขึ้น เราสามารถแบ่งการทำ Classifier Ensemble เป็น 3 รูปแบบ ได้แก่

1. การทำ Boosting โดยปกติที่ Classifier ที่ใช้เวลาในการประมวลผลน้อย (Low Computational Time) มักจะให้ผลลัพธ์ในการจำแนกที่ไม่สูงมากนัก ในทางกลับกัน Classifier ที่มีความซับซ้อนมากสามารถให้ผลลพัธ์ที่ดีกว่า แต่ต้องแลกด้วยเวลาในการประมวลผล (Computational Time) ที่สูงขึ้น การทำ Classifier Ensemble คือทางออกสำหรับปัญหานี้ โดยการนำ Weak Classifier มากกว่าหนึ่งตัวมาช่วยกันประมวลผล เพื่อให้ได้ผลลัพธ์ที่ใกล้เคียงหรือดีกว่าการใช้ Classifier ที่มีความซับซ้อน และใช้เวลาในการประมวลผลที่ไม่สูงมาก ตัวอย่างของวิธีการนี้ เช่น SMOTEBoost หรือ RUSBoost เป็นต้น

2. การทำ Vote Ensemble โดยการใช่ Classifier หลายตัวเพื่อในการจำแนกข้อมูลชุดเดียวกัน จากนั้นจึงดูที่ผลการจำแนกของแต่ละ Classifier เพื่อทำการ Vote ว่าตัวอย่างนั้ถูกจัดอยู่ในกลุ่มใดมากที่สุด

3. การทำ Bagging คือ การแบ่งข้อมูลออกเป็นหลายชุด แล้วนำไปผ่าน Classifier ชนิดเดียวกัน จากนั้นจึงใช้ผลลัพธ์จากการจำแนกของแต่ละโมเดลเพื่อ Vote ว่าแต่ละตัวอย่างถูกจัดอยู่ในกลุ่มใดมากที่สุด วิธี Bagging ยังสามารถนำไปใช้ช่วยในการสุ่มตัวอย่างใหม่ เช่น SMOTEBagging และ Under Bagging ได้อีกด้วย



ภาพที่ 2.1 แสดงแนวทางในการแก้ไขปัญหาข้อมูลไม่สมดุล

2.3 ระยะทางแบบยูคลิด

ระยะทางแบบยูคลิด (Euclidean Distance) คือระยะห่างระหว่างจุดสองจุดในแนวเส้นตรง ซึ่งคำนวณได้จากสมการด้านล่าง

$$d_{Euclidean}(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad (3)$$

เมื่อ $i = 1, 2, 3, \ldots, n$ หมายถึงจำนวนมิติของตัวแปร x และ y

2.4 แนวคิด SMOTE

SMOTE หรือ Synthetic Minority Over-sampling TEchnique เป็นเทคนิคการสุ่มตัวอย่างใหม่ นำเสนอโดย Chawla, N.V., Bowyer, K.W., Hall, L.O., & Kegelmeyer, W.P. (2002) เพื่อแก้ปัญหาข้อมูลไม่สมดุล โดยการสังเคราะห์ตัวอย่างเทียมขึ้นในระหว่างกลุ่มของตัวอย่างเดิม ที่อยู่ใกล้กันจำนวน k ตัว เรียกว่า The k-Nearest Neighbors ซึ่งแตกต่างจากการสุ่มแบบ Random Oversampling (ROS) ที่ใช้วิธีการทำซ้ำสมาชิกเดิมบางตัวขึ้นมาเท่านั้น ผลดีของวิธีนี้คือ ช่วยลดการเกิด Over-fitting ตัวอย่างใหม่ที่สร้างขึ้น และเพิ่มความสามารถของโมเดลอย่างมีนัยสำคัญ

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 Psudo Code ของเทคนิค SMOTE

ที่มา: Chawla, N.V., Bowyer, K.W., Hall, L.O., & Kegelmeyer, W.P. (2002).

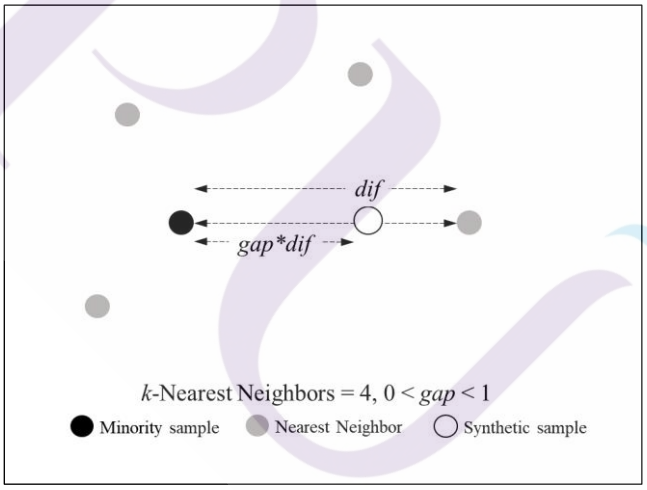
จาก Psudo-code ด้านบนพอสรุปขั้นตอนในการทำงานของ SMOTE ได้ ดังนี้

ขั้นที่ 1 Input ประกอบด้วย T = จำนวนสมาชิกของ Minority, N = เปอร์เซนต์ของจำนวนสมาชิกที่จะสร้างขึ้นเทียบกับสมาชิกเดิม และ k = จำนวนสมาชิกที่จะอ้างอิงในการกำหนดกลุ่มเพื่อสร้างสมาชิกใหม่ ส่วน Output คือสมาชิกใหม่ที่สร้างขึ้นจำนวน (N/100)*T ตัวอย่าง

ขั้นที่ 2 ถ้าค่า $N < 100$ หมายถึงไม่ต้องการใช้สมาชิกเดิมทุกตัวในการทำ SMOTE ดังนั้น จึงทำการสุ่มเลือกสมาชิกเดิมขึ้นมาจำนวน (N/100)*T ตัวอย่าง

ขั้นที่ 3 สำหรับทุกๆ สมาชิกที่ได้จากขั้นที่ 2 ให้ทำการคำนวณหาเพื่อนบ้านจำนวน k ตัวที่อยู่ใกล้ แล้วเก็บข้อมูลที่ได้ไว้ในตัวแปร *nnarray*

ขั้นที่ 4 สำหรับทุกๆ กลุ่มของเพื่อนบ้านที่เก็บไว้ใน *nnarray* ให้สุ่มเลือกเพื่อนบ้านขึ้นมาหนึ่งตัวอย่าง แล้วคำนวณหาระยะห่างระหว่างสมาชิกและเพื่อนบ้านที่ถูกเลือก ซึ่งหมายถึงค่าความต่างของแต่ละ Attribute สมาชิกทั้งสองตัว โดยทั่วไปมักนิยมใช้ Euclidean Distance ในการคำนวณ จากนั้นเก็บผลลัพธ์ไว้ในตัวแปร *dif* แล้วทำการสุ่มค่า *gap* ที่มีค่าระหว่าง 0 ถึง 1 เพื่อนำค่า *gap* ไปคำนวณตามสมการในบรรทัดที่ 22 เพื่อเป็นการสร้างสมาชิกใหม่ที่มีตำแหน่งอยู่ระหว่างสมาชิกเดิมกับเพื่อนบ้านที่สุ่มเลือกขึ้นมานั่นเอง เมื่อทำจนครบทุกกลุ่มใน *nnarray* แล้ว ผลลัพธ์คือสมาชิกใหม่จำนวน (N/100)*T ตัวอย่าง



ภาพที่ 2.3 แสดงวิธีการสร้างตัวอย่างเทียมด้วยเทคนิค SMOTE

2.5 แนวคิด DBSCAN

Ester, M., Kriegel, H., Sander, J., & Xu, X. (1996) ได้นำเสนอแนวคิด DBSCAN หรือ Density Based Spatial Clustering of Applications with Noise เพื่อใช้แบ่งกลุ่มข้อมูลออกเป็นคลัส

เตอร์ย่อย โดยพิจารณาจากความหนาแน่นของสมาชิกที่อยู่ใกล้เคียงกับจุดที่สนใจ (Core Point) แล้วขยายวงออกไปผ่านจุดเชื่อมต่อ (Border Point) ภายใต้เงื่อนไขของจำนวนสมาชิกขั้นต่ำ (Minimum Number of Points) ที่อยู่ใกล้เคียงในระยะเท่ากับค่าคงที่ค่าหนึ่ง (Epsilon) วิธีนี้มีข้อสังเกตว่าตัวอย่างที่ไม่สามารถจัดเข้ากลุ่ม มีความน่าจะเป็นว่าอาจเป็นข้อมูลรบกวนได้

```
DBSCAN (SetOfPoints, Eps, MinPts)

// SetOfPoints is UNCLASSIFIED
  ClusterId := nextId(NOISE);
  FOR i FROM 1 TO SetOfPoints.size DO
    Point := SetOfPoints.get(i);
    IF Point.ClId = UNCLASSIFIED THEN
      IF ExpandCluster(SetOfPoints, Point,
            ClusterId, Eps, MinPts) THEN
        ClusterId := nextId(ClusterId)
      END IF
    END IF
  END FOR
END; // DBSCAN
ExpandCluster(SetOfPoints, Point, ClId, Eps,
               MinPts) : Boolean;
  seeds:=SetOfPoints.regionQuery(Point,Eps);
  IF seeds.size<MinPts THEN // no core point
    SetOfPoint.changeClId(Point,NOISE);
    RETURN False;
  ELSE   // all points in seeds are density-
         // reachable from Point
    SetOfPoints.changeClIds(seeds,ClId);
    seeds.delete(Point);
    WHILE seeds <> Empty DO
      currentP := seeds.first();
      result := SetOfPoints.regionQuery(currentP,
                                        Eps);
      IF result.size >= MinPts THEN
        FOR i FROM 1 TO result.size DO
          resultP := result.get(i);
          IF resultP.ClId
              IN {UNCLASSIFIED, NOISE} THEN
            IF resultP.ClId = UNCLASSIFIED THEN
              seeds.append(resultP);
            END IF;
            SetOfPoints.changeClId(resultP,ClId);
          END IF; // UNCLASSIFIED or NOISE
        END FOR;
      END IF; // result.size >= MinPts
      seeds.delete(currentP);
    END WHILE; // seeds <> Empty
    RETURN True;
  END IF
END; // ExpandCluster
```

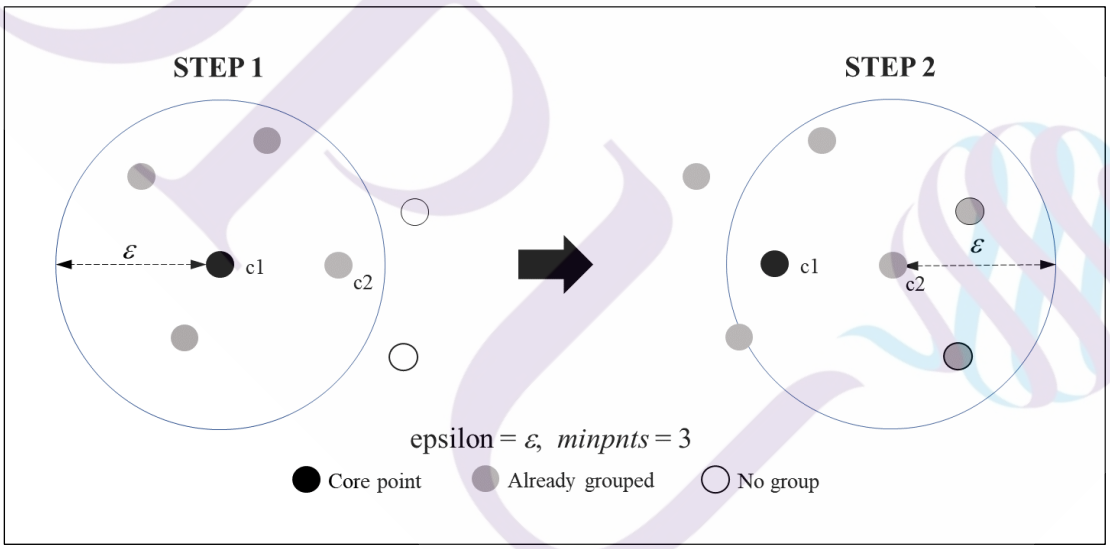
ภาพที่ 2.4 Psudo Code ของเทคนิค DBSCAN

ที่มา: Ester, M., Kriegel, H., Sander, J., & Xu, X. (1996)

จาก Psudo-code พบว่า DBSCAN ประกอบด้วยการทำงาน 2 ส่วน คือ

ฟังก์ชั่น DBSCAN() ทำหน้าที่กำหนดกลุ่มให้กับตัวอย่างหรือสมาชิกที่เป็น Core Point โดยเลือกสมาชิกตัวแรกแบบสุ่มขึ้นมาก่อนเพื่อทำการกำหนดกลุ่มเริ่มต้น แล้วเรียกฟังก์ชั่น ExpandCluster() ขึ้นมาใช้งาน

ฟังก์ชั่น ExpandCluster() ทำหน้าที่หาสมาชิกในบริเวณใกล้เคียง ภายใต้เงื่อนไขของอัลกอริธึม นั้นคือ สมาชิกที่จะอยู่ในกลุ่มเดียวกันกับ Core Point จะต้องอยู่ใกล้กับ Core Point น้อยกว่าหรือเท่ากับค่า Epsilon และมีจำนวนสมาชิกขั้นต่ำอย่างน้อยเท่ากับค่า Minimum Number of Points เมื่อพบสมาชิกในพื้นที่ตามเงื่อนไขแล้ว อัลกอริธึมจะขยับ Core Point ไปยังสมาชิกตัวถัดไปในกลุ่ม เพื่อขยายพื้นที่ของกลุ่มออกไปอีกโดยใช้วิธีการเดิม และหยุดทำงานเมื่อสมาชิกทุกตัวได้รับการกำหนดกลุ่มจนครบแล้ว ข้อดีของ DBSCAN คือ เป็นวิธีการแบบ Unsupervised Learning ที่ใช้กับข้อมูลที่เป็น Unlabeled ได้ มีความสามารถในการแบ่งกลุ่มหรือบริเวณของกลุ่มได้อย่างอิสระ โดยไม่จำเป็นต้องเป็นพื้นที่รูปวงกลมภายใต้ขนาดรัศมีที่คงที่เสมอไป



ภาพที่ 2.5 แสดงการทำงานของเทคนิค DBSCAN

2.6 ผลงานวิจัยที่เกี่ยวข้อง

จากการศึกษาพบว่า ประสิทธิภาพของโมเดลเมื่อใช้ข้อมูลที่ได้จากการใช้เทคนิคการสุ่มตัวอย่างใหม่แบบผสมส่วนใหญ่ให้ความแม่นยำที่ใกล้เคียงหรือดีกว่าการใช้เทคนิคใดเทคนิคหนึ่งเพียงอย่างเดียว ด้วยวัตถุประสงค์ของงานวิจัยนี้คือ เพื่อปรับปรุงเทคนิคการสุ่มตัวอย่างใหม่ โดยการ

ใช้เทคนิค DBSCAN ร่วมกับเทคนิค SMOTE และเปรียบเทียบกับการใช้เทคนิคแบบผสมอื่นที่มีการทำงานในลักษณะที่คล้ายคลึงกัน จึงได้ศึกษางานวิจัยที่เกี่ยวข้องบนแนวทางการใช้เทคนิคดังกล่าว โดยสรุปได้ ดังนี้

Sanguanmak, Y., & Hanskunatai, A. (2016) ได้นำเสนอเทคนิคการสุ่มตัวอย่างใหม่แบบผสมที่เรียกว่า DBSM ซึ่งใช้เทคนิค DBSCAN ร่วมกับเทคนิค SMOTE ในการทำ Resampling โดยแบ่ง Train Data ออกเป็นคลัสเตอร์ย่อยๆ ด้วยเทคนิค DBSCAN แล้วดูว่า ถ้าสมาชิกในคลัสเตอร์นั้นเป็น Majority ทั้งหมด จำทำการคำนวณหาระยะทางระหว่างทุกๆ สมาชิกเทียบกับจุดศูนย์กลางของคลัสเตอร์ (Centroid) แต่ถ้าในคลัสเตอร์นั้น ประกอบด้วยสมาชิกทั้งสองกลุ่มก็จะทำการหาระยะระหว่างทุกๆ Majority และ Minority แล้วทำการกำจัดสมาชิก Majority ออกไปครึ่งหนึ่งของสมาชิก Majority ทั้งหมดโดยดูจากระยะห่างที่สั้นที่สุดก่อน ซึ่งเป็นการ Downsampling สมาชิก Majority จากนั้นจึงทำ Oversampling ให้กับสมาชิก Minority ทั้งหมด โดยใช้วิธีการ SMOTE จนได้ข้อมูลที่มีความสมดุล ผู้วิจัยได้ทำการทดลองกับชุดข้อมูล glass0, glass1, glass6, wiscosin, yeast1, yeast3, Haberman, vehicle1, vehicle2, new-thyroid1, new-thyroid2, ecoli2 โดยเปรียบเทียบกับเทคนิค SMOTE, TOMEK LINKS)TLและ (SMOTE+TOMEK LINKS เพื่อวัดประสิทธิภาพ พบว่าเทคนิคที่นำเสนอให้ Accuracy ที่ดีขึ้นในหลายชุดข้อมูล แต่ค่า AUC และ F-measure ยังไม่สูงนัก

Verma, M.K., Xaxa, M.K., & Verma, S. (2017) นำเสนอวิธีการที่เรียกว่า DBCS โดยแบ่งข้อมูลตัวอย่างออกเป็นคลัสเตอร์ย่อยๆ โดยใช้เทคนิค DBSCAN จากนั้นจึงพิจารณาอัตราส่วนของสมาชิกในแต่ละคลัสเตอร์ว่า ถ้าหากมีสมาชิกเป็นคลาสหนึ่งคลาสใดมากกว่าร้อยละ 50 ให้ทำ Downsampling สมาชิกคลาสนั้นออกภายใต้เงื่อนไขระยะห่างระหว่างสมาชิกเท่ากับค่าคงที่ค่าหนึ่งเรียกว่า Epsilon และถ้าหากสมาชิกในคลัสเตอร์เป็นคลาสหนึ่งคลาสใดน้อยกว่าร้อยละ 50 ก็ให้ทำ Oversampling สมาชิกเพิ่มขึ้นมาในระหว่างสมาชิกของคลาสเดียวกัน ทั้งนี้ การดำเนินการ Resampling ทั้งหมดจะต้องไม่กระทบต่อขอบเขตของคลัสเตอร์ โดยทำการทดลองกับชุดข้อมูล bands, chess, crx, german, hepatitis, heart, housevotes, ionosphere, pima, sonar และ wdbc วัดประสิทธิภาพโดยเปรียบเทียบกับการใช้เทคนิค SMOTE, ROS และ RUS ผลการทดลองพบว่า DBCS ให้ค่า AUC ที่ดีขึ้นเกือบทุกชุดข้อมูลที่นำมาทดสอบ แสดงให้เห็นถึงประสิทธิภาพในการจำแนกที่ดีขึ้นของโมเดลที่ได้จากการใช้ข้อมูลที่ปรับสมดุลแล้ว

Netirungroj, N., & Pacharawongsakda, E. (2018) เสนอเทคนิคการสุ่มตัวอย่างใหม่ ที่เรียกว่า TwO-levels of Positive Resampling Framework)TOPโดย (ใช้วิธีการสร้างขอบเขตรอบข้อมูลที่มีสมาชิก Minority เป็นศูนย์กลางขึ้น 2 ชั้น ให้มีระยะห่างเป็นค่าคงที่ 2 ค่าคือ ε_{inner}, ε_{outer}

จากนั้นสร้างสมาชิก Minorityขึ้นใน Inner Area และลบสมาชิกของ Majority ที่อยู่ใน Outer Area ออก จนได้จำนวนสมาชิกทั้งสองคลาสที่สมดุลกัน ทำการทดลองกับชุดข้อมูล ecoli2, glass0, glass1, glass6, haberman, liver- disorders, new- thyroid1, new-thyroid2, page- blocks1, page-blocks3, page-blocks4, pima, vehicle1, vehicle2, wisconsin, yeast1 และ yeast3 วัดผลด้วยค่า F1, GM, AUROC และ AUPRC เปรียบเทียบกับการใช้เทคนิค Baseline-SMOTE, ROS, SMOTE, RSLS, TOP+V, TOP+ROS, TOP+SMOTE, TOP+RSLS, OVUN, ROSE และ DBSM ผลการทดลองพบว่า เมื่อใช้ TOP ร่วมกับวิธีการอื่นสามารถให้ค่า F1 ที่ดีขึ้นในหลายชุดข้อมูลที่ทดสอบ

Xiaolong, X., Wen, C., & Yenfei, S. (2019) เสนอวิธีการ Oversampling กลุ่มข้อมูล Minority โดยวิธีที่เรียกว่า DSMOTE โดยการใช้ DBSCAN แบ่งข้อมูลออกเป็นสามกลุ่มได้แก่ Core Samples คือพวกที่เกาะกลุ่มกันอย่างชัดเจน Borderline Samples คือพวกที่มีสมาชิก Majority ปะปนอยู่ด้วย และ Noise Samples คือพวกที่อยู่ห่างออกไปหรืออยู่โดดเดี่ยวท่ามกลางกลุ่มของ Majority จากนั้นทำการลบ Noise Samples ทิ้ง และทำ Oversampling ในอีกสองกลุ่มที่เหลือด้วยวิธี SMOTE โดยทดลองกับชุดข้อมูล pima, breast-w, vehicle และ ecoli วัดผลด้วยค่า Precision, Recall และ F-value เปรียบเทียบกับการใช้เทคนิค SMOTE และ Borderline-SMOTE ผลการทดลองพบว่าโมเดลที่สร้างด้วยข้อมูลที่ได้จาก DSMOTE มีประสิทธิภาพที่ดีขึ้นกว่าเทคนิคที่นำมาเปรียบเทียบ

ในงานวิจัยก่อนหน้าพยายามแก้ไขปัญหาข้อมูลไม่สมดุลโดยมุ่งเน้นในส่วนที่เกิดการซ้อนทับของตัวอย่างกลุ่ม Majority และ Minority โดยใช้วิธีการสุ่มออกหรือการจงใจเลือกสมาชิกของ Majority ออกไปตามจำนวนที่ต้องการ ซึ่งอาจทำให้อาจสูญเสียข้อมูลที่สำคัญ และคุณลักษณะ (Characteristics) ของข้อมูลไปอย่างไม่ตั้งใจ อีกด้านหนึ่งคือ การพยายามสร้าง (Generate) สมาชิก Minority เพิ่มเข้าไป เพื่อให้เกิดความสมดุลระหว่างทั้งสองคลาส โดยตามหลักการของ SMOTE นั้น การสร้างสมาชิกใหม่ขึ้นมาจำเป็นต้องอาศัยหรืออ้างอิงกับสมาชิก Minority เดิม และถ้าหากในกลุ่มตัวอย่างที่ใช้ทำ Oversampling มีตัวอย่างที่เป็นข้อมูลรบกวนปะปนอยู่ อาจจะทำตัวอย่างใหม่ที่สร้างขึ้นมีคุณภาพไม่ดีเท่าที่ควร การใช้เทคนิคการแบ่งกลุ่ม (Clustering) เข้ามาช่วยนั้น มักจะเลือกทำการสุ่มตัวอย่างใหม่กับคลัสเตอร์ที่มีขนาดใหญ่ที่พบ โดยมองข้ามคลัสเตอร์ที่มีขนาดรองลงมา ซึ่งในความเป็นจริงอาจมีความเป็นไปได้ว่าคลัสเตอร์ที่มีขนาดเล็กกว่านั้น อาจจะเป็นบริเวณที่ Minority รวมตัวกันชัดเจนมากกว่า และเหมาะสมสำหรับการใช้ในการสุ่มตัวอย่างใหม่มากกว่าก็ได้

บทที่ 3
ระเบียบวิธีวิจัย

ในบทนี้จะกล่าวถึงขั้นตอนการทำงานเทคนิค DB2SM ที่ออกแบบขึ้นเพื่อใช้ปรับปรุงประสิทธิภาพการทำ Oversampling บน Imbalanced Data ที่สามารถปรับใช้งานได้ง่าย และใช้เวลาในการประมวลผลที่ไม่สูงมากนัก โดยมีรายละเอียดที่กล่าวถึง ดังนี้

3.1 ขั้นตอนวิธีของ DB2SM

3.2 การออกแบบการทดลอง

3.3 ข้อมูลที่ใช้ในการทดลอง

3.4 การวัดประสิทธิภาพ

3.5 เครื่องมือที่ใช้ในการวิจัย

3.1 ขั้นตอนวิธีของ DB2SM

DB2SM เป็นเทคนิคการสุ่มตัวอย่างใหม่แบบผสม (Combined Technique) สำหรับทำ Oversampling เพื่อเพิ่มจำนวนตัวอย่างของกลุ่มส่วนน้อย (Minority) โดยอาศัยการทำงานร่วมกันระหว่างเทคนิค DBSCAN กับเทคนิค SMOTE โดยมีจุดมุ่งหมายคือค้นหาบริเวณที่สมาชิก Minority เกาะกลุ่มกันอยู่มากที่สุด และแยกตัวอย่างที่มีโอกาสเป็นข้อมูลรบกวนออกไป แล้วทำการสังเคราะห์สมาชิกใหม่ขึ้นในพื้นที่นั้น จนได้จำนวนสมาชิกของ Minority และ Majority ใกล้เคียงหรือเท่ากัน แล้วรวมตัวอย่างทั้งหมดเป็นชุดข้อมูลใหม่สำหรับนำไปใช้งาน เทคนิค DB2SM ประกอบด้วยขั้นตอนการทำงาน ดังนี้

ขั้นที่ 1 ใช้เทคนิค DBSCAN เพื่อแบ่งชุดข้อมูลตัวอย่างออกเป็นคลัสเตอร์ย่อย ด้วยพารามิเตอร์ ε = Epsilon และ *minpnts* = Minimum Number of Point ผลลัพธ์ที่ได้จะประกอบไปด้วยกลุ่มของคลัสเตอร์ 3 แบบ ได้แก่ คลัสเตอร์ผสม (Mixed Cluster) คือคลัสเตอร์ที่ประกอบด้วยสมาชิก Minority และสมาชิก Majority อยู่รวมกัน คลัสเตอร์ของ Minority ล้วน และคลัสเตอร์ของ Majority ล้วน

ขั้นที่ 2 สำหรับทุกๆ คลัสเตอร์ผสม ให้คำนวณค่า Euclidean Distance ระหว่างสมาชิก Minority และสมาชิก Majority ทุกตัวภายในคลัสเตอร์นั้น

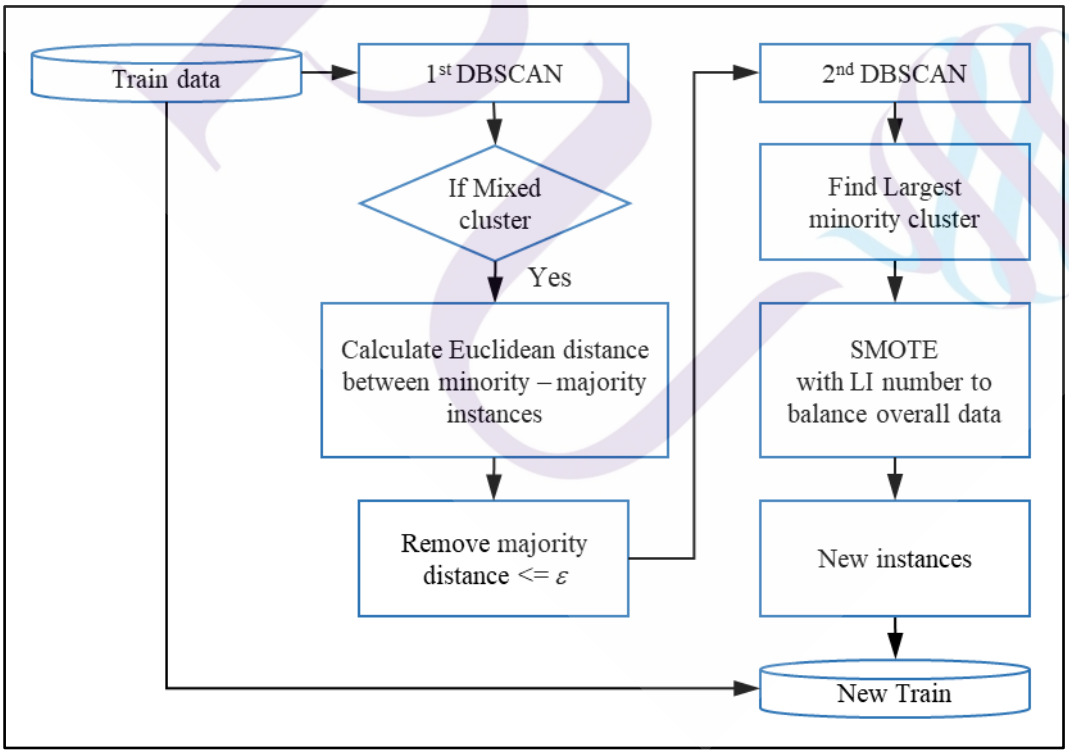
ขั้นที่ 3 ลบสมาชิก Majority ที่มี Euclidean Distance <= ε ออกจากคลัสเตอร์และเก็บข้อมูลที่ลบไว้ในตัวแปร *delInstance* เพื่อใช้ในการเปรียบเทียบในภายหลัง

ขั้นที่ 4 รวมสมาชิกที่เหลือทั้งหมดที่ได้จากขั้นที่ 2-3 เพื่อทำ DBSCAN อีกรอบ โดยใช้ค่าพารามิเตอร์เดียวกันกับขั้นที่ 1 ผลลัพธ์ที่ได้ในขั้นตอนนี้จะเหลือกลุ่มของคลัสเตอร์ที่เป็น Minority ล้วน หรือคลัสเตอร์ที่เป็น Majority ล้วน และตัวอย่างที่อยู่โดดเดี่ยวเท่านั้น

ขั้นที่ 5 เลือกคลัสเตอร์ Minority ล้วน ที่มีจำนวนสมาชิกมากที่สุดจากกลุ่มคลัสเตอร์ทั้งหมดที่ได้จากขั้นที่ 1 และ 4 แล้วทำ SMOTE ในคลัสเตอร์นั้น เพื่อสร้างสมาชิก Minority ขึ้นใหม่ตามจำนวนที่เท่ากับค่า *LI* ในสมาการที่ (2)

ขั้นที่ 6 นำตัวอย่างใหม่ที่ได้มาเปรียบเทียบกับตัวอย่างใน *delInstance* เพื่อกำจัดตัวอย่างใหม่ที่ซ้ำกันออก โดยเมื่อเสร็จสิ้นขั้นตอนนี้ จะมีจำนวนสมาชิก Minority รวมน้อยกว่าหรือเท่ากันกับจำนวนสมาชิก Majority รวม เพื่อเป็นการป้องกันโอกาสที่จะเกิด Over-fitting จากสมาชิก Minority ที่ได้มาด้วย

ขั้นที่ 7 นำตัวอย่างใหม่ที่เหลือจากขั้นที่ 6 รวมกับข้อมูลเดิมทั้งหมด เพื่อเป็นชุดข้อมูลตัวอย่างใหม่ที่มีความสมดุลระหว่างสองคลาส สำหรับนำไปใช้งาน



ภาพที่ 3.1 แสดงขั้นตอนการทำงานของเทคนิค DB2SM

จากขั้นตอนการทำงานข้างต้นสามารถนำมาสร้าง Psudo-code ได้ดังนี้

Algorithm: DB2SM

Input: *T*: Train dataset contains positive & negative,
ε: Epsilon,
minpnts: Minimum Number of Point (parameter for DBSCAN)
LI: Lack of Information for the minority class (LI)

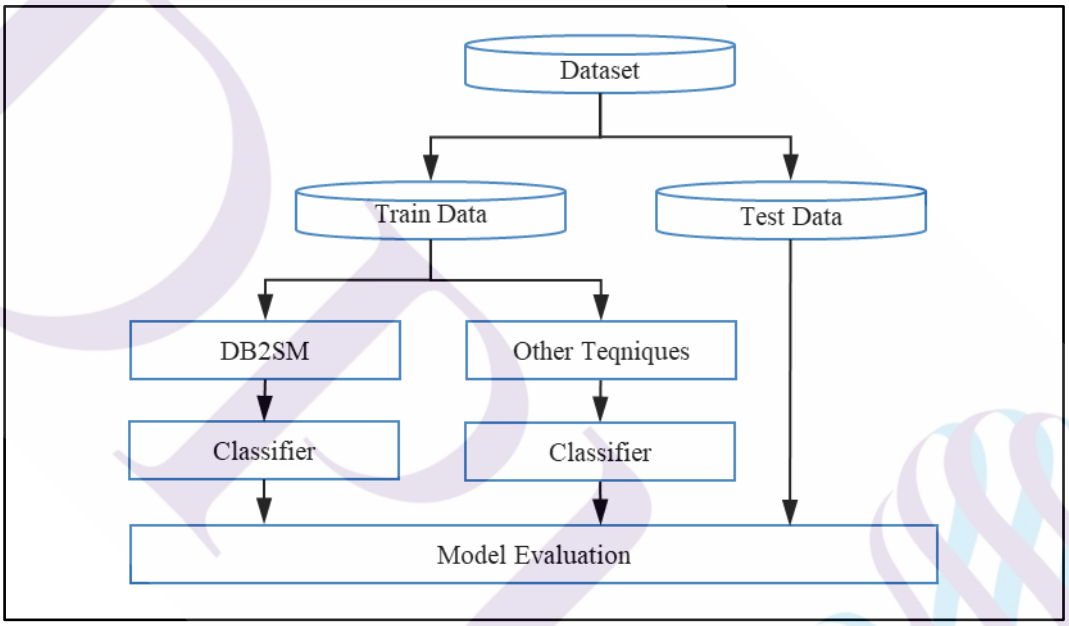
Output: *New_T*: New Balanced train data

```
1.  C = DBSCAN(T, ε, minpnts) // C: set of cluster
2.  for i = 1 to a // a: number of cluster in C do
3.    if Ci is mixed cluster then // contains Majority & Minority
4.      for j = 1 to b do // b: number of Minority in Ci
5.        for k = 1 to c do // c: number of Majority in Ci
6.          djk = calEuclideanDistance(pj, nk) // pj: Minority, nk: Majority
7.          if djk <= ε then
8.            moveInstance(nk, delInstance) // delInstance: empty dataset
9.          end if
10.       end for
11.     end for
12.     appendInstance(D, Ci) // D: empty dataset
13.   end if
14.   deleteCluster(C, Ci)
15. end for // now C contains only pure class clusters
16. E = DBSCAN(D, ε, minpnts)
17. C = C + E
18. Lp = findLargest_MinorityCluster(S)
19. newInstance = SMOTE(Lp, LI) // newInstance: dataset
20. newInstance = removeDuplicate(newInstance, delInstance)
21. New_T = T + new_instance
```

ภาพที่ 3.2 Psudo Code ของเทคนิค DB2SM

3.2 การออกแบบการทดลอง

เพื่อให้เห็นถึงประสิทธิภาพของผลลัพธ์ที่ได้จากการปรับปรุงด้วยเทคนิคการสุ่มตัวอย่างใหม่ที่นำเสนอ จึงออกแบบการทดลองโดยนำชุดข้อมูลที่ผ่านการปรับให้สมดุลแล้ว ไปเป็นข้อมูล Train Data สำหรับสร้างโมเดลการเรียนรู้ จากนั้นจึงวัดประสิทธิภาพของโมเดลที่ได้ และเปรียบเทียบกับโมเดลที่ใช้ Train Data ที่ปรับสมดุลด้วยเทคนิคอื่นที่มีลักษณะการทำงานคล้ายคลึงกัน 3 เทคนิค ได้แก่ SMOTE, DBCS และ DBSM รวมถึงการใช้ข้อมูลต้นฉบับที่ไม่มีการปรับแก้ โดยใช้ Classifier และชุดข้อมูลตัวอย่างชุดเดียวกันตลอดการทดลอง ดังแสดงในแผนภาพ



ภาพที่ 3.3 แสดงขั้นตอนในการทดลอง

ในการทดลองแต่ละชุดข้อมูล จะแบ่งข้อมูลออกเป็น Train Data: Test Data ในอัตราส่วน 70: 30 แล้วส่งข้อมูล Train Data เข้าสู่กระบวนการปรับปรุงข้อมูลแต่ละเทคนิค และนำ Train Data ใหม่ที่ได้จากแต่ละเทคนิคไปสร้างโมเดล โดยใช้ตัวจำแนกแบบ Decision Tree โดยกำหนดพารามิเตอร์เหมือนกัน คือ Criterion แบบ gain_ratio, Maximal depth= 10, Confidence= 0.1, Minimal gain= 0.01, Minimal leaf size=2, Minimal size for split= 4 และ Number of Prepruning alternatives= 3 จากนั้น ทำการวัดประสิทธิภาพของแต่ละโมเดลด้วย Test Data เดียวกัน และเปรียบเทียบค่า Accuracy, AUC และ F-measure ที่ได้

3.3 ข้อมูลที่ใช้ในการทดลอง

ข้อมูลที่ใช้ในการทดลองเป็นข้อมูล Imbalanced Dataset จาก UCI Machine Learning Repository (Dua, D. & Graff, C., 2019) ประกอบด้วยชุดข้อมูล ecoli1, ecoli2, ecoli3, glass0, glass1, glass6, haberman, iris0, new-thyroid1, new-thyroid2, page-blocks0, pima, segment0, vehicle0, vehicle1, vehicle2, vehicle3, wiscosin, yeast1 และ yeast3 ทุกชุดข้อมูลจะมีสมาชิกเพียง 2 Class คือ *positive* และ *negative* เท่านั้น และไม่มีข้อมูลที่สูญหาย (Non-Missing Values) คุณลักษณะของชุดข้อมูลแสดงในตาราง

ตารางที่ 3.1 แสดงรายละเอียดของข้อมูลที่ใช้ในการทดลอง

Dataset	Attribute	Example	Majority	Minority	IR	LI
ecoli1	8	336	259	77	3.36	182
ecoli2	8	336	284	52	5.46	232
ecoli3	8	336	301	35	8.60	266
glass0	10	214	144	70	2.06	74
glass1	10	214	138	76	1.82	62
glass6	10	214	185	29	6.38	156
harberman	4	306	225	81	2.78	144
iris0	5	150	100	50	2.00	50
new-thyroid1	6	215	180	35	5.14	145
new-thyroid2	6	215	180	35	5.14	145
page-blocks0	11	5472	4913	559	8.79	4354
pima	9	768	500	268	1.87	232
segment0	20	2308	1979	329	6.02	1650
vehicle0	19	846	647	199	3.25	448
vehicle1	19	846	629	217	2.90	412
vehicle2	19	846	628	218	2.88	410
vehicle3	19	846	634	212	2.99	422
wiscosin	10	683	444	239	1.86	205
yeast1	9	1484	1055	429	2.46	626
yeast3	9	1484	1321	163	8.10	1158

3.4 การวัดประสิทธิภาพ

ในการทดลองนี้มีจุดมุ่งหมายที่จะสร้างสมาชิกของ Monority ขึ้นมาใหม่ เพื่อแก้ปัญหา Imbalanced Data ของชุดข้อมูล แต่การวัดคุณภาพของชุดข้อมูลใหม่ที่ได้จากที่ออกแบบขึ้นนั้นยังไม่สามารถกระทำได้โดยตรง จึงต้องประเมินจากประสิทธิภาพของโมเดลที่ได้จากการเรียนรู้ด้วยชุดข้อมูลใหม่ที่สร้างขึ้นแทน โดยอาศัยเครื่องมือทางสถิติเป็นตัววัด ดังนี้

3.4.1 Confusion Matrix คือตารางแสดงผลการจำแนกด้วยโมเดลที่สร้างขึ้น โดยเปรียบเทียบกับค่ากลุ่มที่แท้จริงของข้อมูล ในรูปแบบของตารางเมตริกซ์ เมื่อ Predicted หมายถึงค่าที่โมเดลทำนาย และ Actual หมายถึงค่ากลุ่มที่แท้จริงของข้อมูล โดยมีรายละเอียด ดังนี้

TP (True Positive) = จำนวนตัวอย่างที่เป็น *positive* และถูกทำนายเป็น *positive*

FP (False Positive) = จำนวนตัวอย่างที่เป็น *negative* แต่ถูกทำนายเป็น *positive*

FN (True Negative) = จำนวนตัวอย่างที่เป็น *positive* แต่ถูกทำนายเป็น *negative*

TN (True Negative) = จำนวนตัวอย่างที่เป็น *negative* และถูกทำนายเป็น *negative*

ตารางที่ 3.2 Confusion matrix

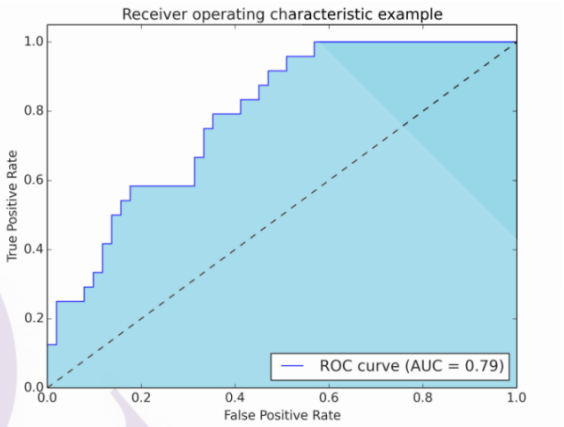
	Predicted as *positive*	Predicted as *negative*
Actually *positive*	TP	FN
Actually *negative*	FP	TN

3.4.2 Accuracy คือค่าความแม่นยำในการจำแนกโดยรวมของตัวโมเดล คำนวณจากค่าของ Confusion Matrix ตามสมการด้านล่าง โดยที่ค่า *Accuracy* ยิ่งมีค่าเข้าใกล้ 1 แสดงว่าโมเดลมีประสิทธิภาพในการทำนายที่แม่นยำขึ้น

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

3.4.3 AUC (Area Under the Curve) คือการพิจารณาพื้นที่ภายใต้กราฟ ROC (Receiver Operating Characteristic) ซึ่งได้จากการนำผลความแม่นยำในการทำนายของโมเดลมาสร้างเป็น

กราฟเส้นโค้งโดยมีจุดเริ่มต้นที่ (0,0) และสิ้นสุดที่จุด (1,1) โดยที่จุดแบ่งเท่ากับ 0 หมายถึงโมเดลทำนายออกมาเป็น *negative* ทั้งหมด เท่ากับค่าของ TP, FP เป็น 0 ทั้งหมด ในทางตรงกันข้าม ที่จุดแบ่งเท่ากับ 1 หมายถึงโมเดลทำนายออกมาเป็น *positive* ทั้งหมด หมายถึงโมเดลสามารถทำนาย *positive* ได้ถูกต้องทั้งหมด



ภาพที่ 3.4 ตัวอย่างกราฟ ROC (Receiver Operating Characteristic)

ที่มา: เว็บไซต์ https://qastack.in.th/stats/132777

จากตัวอย่างกราฟ ROC ด้านบนจะเห็นว่าเราสามารถพิจารณาจากกราฟคร่าวๆ โดยการดูปริมาณพื้นที่ใต้กราฟ จะสามารถเข้าใจได้ว่าถ้าค่าของ TP_{rate} บนกราฟเข้าใกล้ค่า 1 ได้ เร็ว โมเดลก็ยิ่งมีความแม่นยำในการทำนายได้ดีขึ้นตามไปด้วย นั่นหมายถึงพื้นที่ใต้กราฟหรือ AUC ยิ่งมีค่ามากขึ้นและเข้าใกล้ค่า 1 นั่นเอง การคำนวณค่าของ TP_{rate} ค่า FP_{rate} และ AUC สามารถคำนวณได้จากสมการด้านล่างนี้

$$TP_{rate} = \frac{TP}{(TP + FN)} \tag{5}$$

$$FP_{rate} = \frac{FP}{(FP + TN)} \tag{6}$$

$$AUC = \frac{1 + TP_{rate} + FP_{rate}}{2} \tag{7}$$

3.4.4 Precision เป็นการประเมินความแม่นยำในการทำนายของโมเดลที่สามารถทำนายข้อมูลที่เป็น *positive* ได้ถูกต้องเทียบกับจำนวนตัวอย่างที่ถูกทำนายเป็น *positive* ทั้งหมด ดังสมการด้านล่าง

$$Precision = \frac{TP}{(TP + FP)} \quad (8)$$

3.4.5 Recall เป็นการประเมินความแม่นยำในการทำนายของโมเดลที่สามารถทำนายข้อมูลที่เป็น *positive* ได้ถูกต้องเทียบกับจำนวนตัวอย่างที่เป็น *positive* ทั้งหมด ดังสมการด้านล่าง

$$Recall = \frac{TP}{(TP + FN)} \quad (9)$$

3.4.6 F-measure เป็นการประเมินความแม่นยำในการทำนายของโมเดล โดยพิจารณาจากผลเฉลี่ยของ *Precision* และ *Recall* ดังสมการที่ด้านล่าง ค่า *F-measure* ยิ่งมีค่าเข้าใกล้ 1 แสดงว่าโมเดลมีประสิทธิภาพในการทำนายที่แม่นยำขึ้น

$$F - measure = \frac{2 * Recall * Precision}{Recall + Precision} \quad (10)$$

3.5 เครื่องมือที่ใช้ในการวิจัย

เครื่องมือที่ใช้ในการวิจัยประกอบด้วย

1. เครื่องคอมพิวเตอร์ส่วนบุคคล 1 เครื่อง
2. ซอฟต์แวร์ RapidMiner Studio เวอร์ชั่น 9.8

บทที่ 4
ผลการศึกษา

ในส่วนของผลการศึกษา ผู้วิจัยได้ทำการทดลองกับชุดข้อมูล Imbalanced Dataset ของ UCI Machine Learning Repository และประเมินคุณภาพของการสุ่มตัวอย่างใหม่ที่พัฒนาขึ้นโดยพิจารณาจากประสิทธิภาพของโมเดลที่เรียนรู้จากข้อมูลที่ผ่านการปรับสมดุลด้วยเทคนิคที่นำเสนอ และนำผลลัพธ์มาเปรียบเทียบกับโมเดลที่ได้จากการใช้ข้อมูลที่ผ่านการปรับสมดุลด้วยเทคนิคอื่นที่มีลักษณะการทำงานที่คล้ายคลึงกันเพื่อดูพัฒนาการ นอกจากนี้ยังทำการทดลองเพิ่มเติมกับชุดข้อมูลของ KEEL (Knowledge Extraction based on Evolutionary Learning) ที่มีค่าอัตราความไม่สมดุลของข้อมูล มากกว่า 9 เพื่อศึกษาความเป็นไปได้ในการนำเทคนิคที่พัฒนาขึ้นไปประยุกต์ใช้งานกับข้อมูลที่มีความหลากหลายมากยิ่งขึ้น และทำการทดลองด้วยการขยายผลการสุ่มตัวอย่างใหม่ โดยออกแบบวิธีการที่เรียกว่า DB2SM + Extended SMOTE หรือ DB2SMx เพื่อทดสอบสมมติฐานที่ตั้งไว้ ผลการทดลองออกเป็น 3 ส่วน ดังนี้

4.1 ผลการทดลองเบื้องต้น

4.2 ผลการทดลองกับชุดข้อมูลเพิ่มเติม

4.3 ผลการทดลองขยายผลการสุ่มตัวอย่างใหม่

4.1 ผลการทดลองเบื้องต้น

ในส่วนนี้เป็นผลการทดลองกับชุดข้อมูลของ UCI Machine Learning Repository ได้แก่ ชุดข้อมูล ecoli1, ecoli2, ecoli3, glass0, glass1, glass6, haberman, iris0, new-thyroid1, new-thyroid2, page-blocks0, pima, segment0, vehicle0, vehicle1, vehicle2, vehicle3, wiscosin, yeast1 และ yeast3 ซึ่งแต่ละชุดข้อมูลประกอบด้วยสมาชิกเพียง 2 Class คือ *positive* และ *negative* เท่านั้น และไม่มีข้อมูลที่สูญหาย (Non-Missing Values) ในทุกชุดข้อมูล ตามรายละเอียดในบทที่ 3 จากนั้นทำการประเมินคุณภาพของข้อมูลที่ปรับสมดุลแล้ว โดยวัดจากประสิทธิภาพของโมเดลการเรียนรู้ผ่านอัลกอริธึม Decision Tree ที่ใช้งานชุดข้อมูลใหม่ในการฝึก ได้แก่ ค่า Accuracy ค่า AUC และค่า F-measure พร้อมเปรียบเทียบกับเทคนิคอื่นที่มีลักษณะการทำงานที่คล้ายคลึงกัน 3 เทคนิค ได้แก่

เทคนิค SMOTE เทคนิค DBCS เทคนิค DBSM และการใช้ข้อมูลต้นฉบับที่ไม่ผ่านการปรับสมดุล ผลการทดลองแสดงในตารางต่อไปนี้

ตารางที่ 4.1 เปรียบเทียบค่า Accuracy ของโมเดลในการทดลอง

Dataset	Original	SMOTE	DBCS	DBSM	DB2SM
ecoli1	**0.9109**	0.8911	0.8812	0.8911	0.8317
ecoli2	0.9208	0.9307	0.9406	0.9307	**0.9703**
ecoli3	0.8812	0.8614	0.8911	0.7822	**0.9307**
glass0	**0.8438**	0.7500	0.8125	0.7812	**0.8438**
glass1	0.7188	0.6250	0.6875	0.5938	**0.7344**
glass6	0.9844	**1.0000**	0.9688	0.9844	**1.0000**
haberman	**0.7065**	0.6196	0.6522	0.6087	0.6739
iris0	1.0000	1.0000	1.0000	1.0000	1.0000
new-thyroid1	**0.9688**	0.9531	**0.9688**	0.9531	0.9531
new-thyroid2	0.9219	0.9531	0.9531	0.9531	**0.9688**
page-blocks0	0.9604	0.9683	0.9287	0.9562	**0.9714**
pima	0.6957	0.6696	0.5957	0.6696	**0.7000**
segment0	**0.9928**	0.9884	0.9855	0.9884	0.9913
vehicle0	0.9055	0.8031	0.7205	0.8031	**0.9252**
vehicle1	0.7323	0.3858	0.3937	0.3858	**0.7677**
vehicle2	0.8976	0.8858	0.9134	0.8858	**0.9409**
vehicle3	**0.7598**	0.3819	0.4213	0.3819	0.7047
wiscosin	0.9463	0.9512	**0.9659**	0.9610	0.9512
yeast1	0.7326	0.7348	0.4629	0.7348	**0.7551**
yeast3	**0.9618**	0.9528	0.9169	0.9483	**0.9618**

ตารางที่ 4.2 เปรียบเทียบค่า AUC ของโมเดลในการทดลอง

Dataset	Original	SMOTE	DBCS	DBSM	DB2SM
ecoli1	0.8640	**0.9340**	0.8960	0.8590	0.8270
ecoli2	0.8410	0.8590	0.8910	0.9160	**0.9330**
ecoli3	0.7380	0.8130	0.8840	0.8250	**0.8510**
glass0	0.8390	0.8110	0.7950	0.7820	**0.8890**
glass1	0.6760	0.6320	0.6930	0.6540	**0.7180**
glass6	0.8890	**1.0000**	0.9640	0.9820	**1.0000**
haberman	0.5200	0.5520	**0.5980**	0.5470	0.5710
iris0	0.5000	0.5000	0.5000	0.5000	0.5000
new-thyroid1	0.9750	0.8450	**0.9810**	0.8450	0.8910
new-thyroid2	0.8440	0.9270	0.8850	0.9270	**0.9730**
page-blocks0	0.8230	**0.9460**	0.9330	0.9420	0.9140
pima	0.7040	**0.7460**	0.7360	**0.7460**	0.7340
segment0	**0.9720**	0.9690	0.9680	0.9690	0.9710
vehicle0	0.9170	0.8630	0.8050	0.8630	**0.9390**
vehicle1	0.5000	0.6050	0.5790	0.6050	**0.6370**
vehicle2	0.8870	0.9070	0.8850	0.9070	**0.9180**
vehicle3	**0.6530**	0.5960	0.6400	0.5960	0.6460
wiscosin	0.9280	0.9470	**0.9630**	0.9540	0.9380
yeast1	0.6950	**0.7610**	0.6180	**0.7610**	0.7030
yeast3	0.9050	0.9430	0.8830	**0.9450**	0.9250

ตารางที่ 4.3 เปรียบเทียบค่า F-measure ของโมเดลในการทดลอง

Dataset	Original	SMOTE	DBCS	DBSM	DB2SM
ecoli1	**0.8000**	0.7843	0.7500	0.7660	0.7018
ecoli2	0.7778	0.8205	0.8500	0.8108	**0.9231**
ecoli3	0.2500	0.4615	**0.5600**	0.3889	0.3636
glass0	**0.7619**	0.6190	0.7273	0.6667	0.7368
glass1	0.5909	0.5556	0.6154	0.5517	**0.6667**
glass6	0.9412	**1.0000**	0.9000	0.9474	**1.0000**
haberman	0.4000	0.4068	0.4286	0.4000	**0.4444**
iris0	1.0000	1.0000	1.0000	1.0000	1.0000
new-thyroid1	0.9000	0.8421	**0.9091**	0.8421	0.8421
new-thyroid2	0.7619	0.8696	0.8696	0.8696	**0.9167**
page-blocks0	0.7735	0.8452	0.7221	0.8125	**0.8554**
pima	0.4068	0.6346	0.6235	0.6346	**0.6634**
segment0	**0.9701**	0.9529	0.9419	0.9529	0.9643
vehicle0	0.8065	0.7024	0.6502	0.7024	**0.8652**
vehicle1	**0.8110**	0.4730	0.4797	0.4730	0.3059
vehicle2	0.7937	0.7914	0.8333	0.7914	**0.8855**
vehicle3	0.2469	0.4530	**0.4806**	0.4530	0.4681
wiscosin	0.9209	0.9265	**0.9510**	0.9429	0.9254
yeast1	0.4848	0.6289	0.5286	**0.6312**	0.5856
yeast3	0.8172	0.7879	0.6783	0.7723	**0.8350**

ตารางที่ 4.4 แสดงค่าพารามิเตอร์ในการทดลอง และประสิทธิภาพของโมเดล

Dataset	ε	*minpts*	LI	SMOTE Cluster size	Minority Scale (%)	Accuracy	AUC	F-measure
ecoli1	0.50	16	182	56	**72.73**	0.8317	0.8270	0.7018
ecoli2	0.15	14	232	32	**61.54**	**0.9703**	**0.9330**	**0.9231**
ecoli3	0.10	1	266	4	11.43	**0.9307**	**0.8510**	0.3636
glass0	0.50	5	74	28	40.00	**0.8438**	**0.8890**	0.7368
glass1	0.50	3	62	29	38.16	**0.7344**	**0.7180**	**0.6667**
glass6	0.35	2	156	16	**55.17**	**1.0000**	**1.0000**	**1.0000**
harberman	0.10	2	144	55	**67.90**	0.6739	0.5710	**0.4444**
iris0	0.20	2	50	19	38.00	1.0000	0.5000	1.0000
new-thyroid1	0.10	11	145	25	**71.43**	0.9531	0.8910	0.8421
new-thyroid2	0.10	16	145	24	**68.57**	**0.9688**	**0.9730**	**0.9167**
page-blocks0	0.10	8	4354	391	**69.95**	**0.9714**	0.9140	**0.8554**
pima	0.10	14	232	182	**67.91**	**0.7000**	0.7340	**0.6634**
segment0	0.85	5	1650	245	**74.47**	0.9913	0.9710	0.9643
vehicle0	0.10	7	448	132	**66.33**	**0.9252**	**0.9390**	**0.8652**
vehicle1	0.10	2	412	146	**67.28**	**0.7677**	**0.6370**	0.3059
vehicle2	0.10	13	410	156	**71.56**	**0.9409**	**0.9180**	**0.8855**
vehicle3	0.11	11	422	142	**66.98**	0.7047	0.6460	0.4681
wiscosin	0.10	17	205	171	**71.55**	0.9512	0.9380	0.9254
yeast1	0.05	1	626	5	1.17	**0.7551**	0.7030	0.5856
yeast3	0.10	7	1158	61	37.42	**0.9618**	0.9250	**0.8350**

จากผลการทดลองตามตารางข้างต้น ค่าที่แสดงเป็นตัวอักษรแบบหนา แสดงถึงประสิทธิภาพสูงสุดในแต่ละการทดลองของแต่ละชุดข้อมูลตัวอย่าง พบว่าการใช้เทคนิคที่ออกแบบสามารถค้นหากลุ่มของตัวอย่างหรือคลัสเตอร์ของ Minority ที่มีขนาดใหญ่กว่าร้อยละ 50 ของจำนวนสมาชิก Minority ทั้งหมดได้ในหลายชุดข้อมูลที่นำมาทดลอง โดยดูได้จากค่าของ Minority Scale ซึ่งเป็นอัตราส่วนระหว่างจำนวนสมาชิกของคลัสเตอร์ของ Minority ที่ใหญ่ที่สุดที่ค้นพบเทียบกับจำนวนสมาชิก Minority ทั้งหมดในชุดข้อมูลนั้น และประสิทธิภาพของโมเดลที่เรียนรู้จากการใช้ข้อมูลตัวอย่างที่ผ่านการปรับสมดุลด้วยแล้วนั้น สามารถให้ประสิทธิภาพการจำแนกที่ดีขึ้นในหลายชุดข้อมูล ได้แก่ ecoli2, ecoli3, glass0, glass1, glass6, new-thyroid2, page-blocks0, pima, vehicle0, vehicle1, vehicle2, yeast1 และ yeast3 โดยส่วนใหญ่มีค่าประสิทธิภาพที่สอดคล้องกันระหว่างค่า Accuracy และค่า F-measure ที่แปรผันตามไปในทิศทางเดียวกัน

4.2 ผลการทดลองกับชุดข้อมูลเพิ่มเติม

จากผลการทดลองเบื้องต้น ชุดข้อมูลตัวอย่างที่ถูกนำมาใช้มีค่าอัตราความไม่สมดุลของข้อมูล (Imbalance Ratio หรือ IR) อยู่ระหว่าง 0.77 ถึง 8.79 ซึ่งเป็นค่าที่แสดงถึงความแตกต่างของจำนวนตัวอย่างของ Majority และ Minority ในชุดข้อมูลในระดับที่ไม่สูงมากนัก แต่ถ้าหากค่าอัตราความไม่สมดุลของข้อมูลยิ่งมีค่าสูงมาก แสดงว่าความแตกต่างระหว่างจำนวนสมาชิกตัวอย่างของ Majority มีจำนวนมากกว่าจำนวนสมาชิกตัวอย่างของ Minority มากขึ้นด้วย ซึ่งเป็นลักษณะของข้อมูลที่มีความไม่สมดุลสูง และเมื่อนำข้อมูลนั้นไปใช้งาน ก็มีโอกาสทำให้โมเดลที่ได้มีความเอนเอียงสูงขึ้นตามไปด้วย

ด้วยเหตุนี้ ผู้วิจัยจึงทำการทดลองเพิ่มเติมเพื่อศึกษาความเป็นไปได้ที่จะนำเทคนิคที่พัฒนาขึ้นไปปรับใช้ในการแก้ปัญหากับกลุ่มข้อมูลที่มีความไม่สมดุลสูง โดยทำการทดลองกับชุดข้อมูลไม่สมดุลของ KEEL (Alcalá-Fdez, J., Fernandez, A., Luengo, J., Derrac, J., García, S., Sánchez, L., & Herrera, F., 2011). ได้แก่ ecoli-0-1-4-7_vs_2-3-5-6, ecoli-0-3-4-7_vs_5-6, ecoli-0-6-7_vs_3-5, glass-0-1-5_vs_2, glass-0-1-6_vs_2, glass-0-6_vs_5, glass2, glass4, glass5, page-blocks-1-3_vs_4, poker-8_vs_6, poker-8-9_vs_5, poker-8-9_vs_6, poker-9_vs_7, shuttle-6_vs_2-3 , shuttle-c0 - vs-c4 , shuttle-c2 - vs-c4 , vowel0 , winequality-red-3 _vs_5 , winequality-red-4 , winequality-red-8_vs_6, yeast-0-5-6-7-9_vs_4, yeast-2_vs_4, yeast5 และ yeast6 ซึ่งชุดข้อมูลเหล่านี้มีค่าอัตราความไม่สมดุลของข้อมูลสูงกว่า 9 และประกอบด้วยสมาชิกเพียง 2 Class คือ *positive* และ *negative* และไม่มีข้อมูลที่สูญหาย (Non-Missing Values) ในทุกชุดข้อมูล รายละเอียดของชุดข้อมูลที่นำมาใช้ในการทดลอง และผลการทดลองดังแสดงในตารางต่อไปนี้

ตารางที่ 4.5 แสดงรายละเอียดของข้อมูลที่ใช้ในการทดลองเพิ่มเติม

Dataset	Attribute	Example	Majority	Minority	IR	LI
ecoli-0-1-4-7_vs_2-3-5-6	8	336	307	29	10.59	278
ecoli-0-3-4-7_vs_5-6	8	257	232	25	9.28	207
ecoli-0-6-7_vs_3-5	8	222	200	22	9.09	178
glass-0-1-5_vs_2	10	172	155	17	9.12	138
glass-0-1-6_vs_2	10	192	175	17	10.29	158
glass-0-6_vs_5	10	108	99	9	11.00	90
glass2	10	214	197	17	11.59	180
glass4	10	214	201	13	15.46	188
glass5	10	214	205	9	22.78	196
page-blocks-1-3_vs_4	11	472	444	28	15.86	416
poker-8_vs_6	11	1477	1460	17	85.88	1443
poker-8-9_vs_5	11	2075	2050	25	82.00	2025
poker-8-9_vs_6	11	1485	1460	25	58.40	1435
poker-9_vs_7	11	244	236	8	29.50	228
shuttle-6_vs_2-3	10	230	220	10	22.00	210
shuttle-c0-vs-c4	10	1829	1706	123	13.87	1583
shuttle-c2-vs-c4	10	129	123	6	20.50	117
vowel0	14	988	898	90	9.98	808
winequality-red-3_vs_5	12	691	681	10	68.10	671
winequality-red-4	12	1599	1546	53	29.17	1493
winequality-red-8_vs_6	12	656	638	18	35.44	620
yeast-0-5-6-7-9_vs_4	9	528	477	51	9.35	426
yeast-2_vs_4	9	514	463	51	9.08	412
yeast5	9	1484	1440	44	32.73	1396
yeast6	9	1484	1449	35	41.40	1414

ตารางที่ 4.6 เปรียบเทียบค่า Accuracy ของโมเดล เมื่อทดลองกับข้อมูลที่ค่า IR สูงกว่า 9

Dataset	Original	SMOTE	DBCS	DBSM	DB2SM
ecoli-0-1-4-7_vs_2-3-5-6	**0.9307**	0.9109	0.7426	0.9010	0.9208
ecoli-0-3-4-7_vs_5-6	**0.9351**	0.9091	0.8831	0.8961	**0.9351**
ecoli-0-6-7_vs_3-5	0.9403	0.8507	**0.9701**	0.8507	0.9552
glass-0-1-5_vs_2	**0.9038**	0.7500	NA	0.7500	0.8462
glass-0-1-6_vs_2	0.8966	0.7069	NA	0.6724	**0.9310**
glass-0-6_vs_5	1.0000	1.0000	1.0000	1.0000	1.0000
glass2	**0.9219**	0.8125	NA	0.8125	**0.9219**
glass4	**0.9844**	0.9688	NA	0.9688	0.9531
glass5	0.9531	0.9531	0.9531	0.9531	0.9531
page-blocks-1-3_vs_4	0.9789	**1.0000**	**1.0000**	**1.0000**	0.9930
poker-8_vs_6	**0.9910**	0.7223	0.6117	0.7223	**0.9910**
poker-8-9_vs_5	**0.9871**	0.5804	0.6141	0.5788	**0.9871**
poker-8-9_vs_6	**0.9820**	0.7326	0.5910	0.7326	**0.9820**
poker-9_vs_7	**0.9726**	0.9315	0.9589	0.9452	0.9178
shuttle-6_vs_2-3	1.0000	1.0000	1.0000	1.0000	1.0000
shuttle-c0-vs-c4	1.0000	1.0000	1.0000	1.0000	1.0000
shuttle-c2-vs-c4	1.0000	1.0000	1.0000	1.0000	1.0000
vowel0	0.9764	0.9831	NA	**0.9865**	0.9831
winequality-red-3_vs_5	**0.9903**	0.9469	0.9372	0.9372	0.9758
winequality-red-4	**0.9708**	0.6479	0.5875	0.6479	0.6946
winequality-red-8_vs_6	**0.9695**	0.8782	0.8579	0.8731	0.9543
yeast-0-5-6-7-9_vs_4	**0.9430**	0.8987	0.7848	0.9114	0.9051
yeast-2_vs_4	**0.9610**	0.9481	**0.9610**	0.9481	**0.9610**
yeast5	**0.9843**	0.9798	0.9258	0.9640	**0.9843**
yeast6	**0.9753**	0.9416	0.8135	0.9326	**0.9753**

ตารางที่ 4.7 เปรียบเทียบค่า AUC ของโมเดล เมื่อทดลองกับข้อมูลที่ค่า IR สูงกว่า 9

Dataset	Original	SMOTE	DBCS	DBSM	DB2SM
ecoli-0-1-4-7_vs_2-3-5-6	0.7390	0.7800	0.6990	0.7990	**0.8410**
ecoli-0-3-4-7_vs_5-6	0.7440	0.8850	**0.8910**	0.8790	0.5000
ecoli-0-6-7_vs_3-5	0.9370	**0.9840**	0.9680	**0.9840**	0.5000
glass-0-1-5_vs_2	**0.6720**	0.5940	NA	0.5940	0.6400
glass-0-1-6_vs_2	0.5380	0.5680	NA	0.5490	**0.7530**
glass-0-6_vs_5	1.0000	1.0000	1.0000	1.0000	1.0000
glass2	0.8460	0.6240	NA	0.6240	**0.8580**
glass4	0.6860	0.5000	NA	0.6670	**0.9510**
glass5	0.0000	0.0000	0.0000	0.0000	**0.5000**
page-blocks-1-3_vs_4	0.9780	**1.0000**	0.5000	**1.0000**	0.9930
poker-8_vs_6	0.5000	**0.5650**	0.5270	**0.5650**	0.5000
poker-8-9_vs_5	0.5000	0.4610	**0.5160**	0.4630	0.5000
poker-8-9_vs_6	0.5000	**0.6580**	0.4530	**0.6580**	0.5000
poker-9_vs_7	0.6900	0.6450	**0.8190**	0.6520	0.4690
shuttle-6_vs_2-3	0.5000	0.5000	0.5000	0.5000	0.5000
shuttle-c0-vs-c4	0.5000	0.5000	0.5000	0.5000	0.5000
shuttle-c2-vs-c4	0.5000	0.5000	0.5000	0.5000	0.5000
vowel0	0.9300	0.9480	NA	**0.9640**	0.5000
winequality-red-3_vs_5	0.5000	**0.7050**	0.6390	0.6990	0.4410
winequality-red-4	0.5000	**0.6220**	0.5960	**0.6220**	0.5550
winequality-red-8_vs_6	**0.7810**	0.4340	0.7210	0.5350	0.6700
yeast-0-5-6-7-9_vs_4	0.6970	0.8570	0.7690	**0.8590**	0.6900
yeast-2_vs_4	**0.9080**	0.8640	0.8740	0.8640	0.8770
yeast5	0.8750	0.9170	0.5000	**0.9810**	0.8040
yeast6	0.7390	**0.9630**	0.6310	0.9490	0.5530

ตารางที่ 4.8 เปรียบเทียบค่า F-measure ของโมเดล เมื่อทดลองกับข้อมูลที่ค่า IR สูงกว่า 9

Dataset	Original	SMOTE	DBCS	DBSM	DB2SM
ecoli-0-1-4-7_vs_2-3-5-6	**0.6667**	0.6400	0.3810	0.6429	0.6364
ecoli-0-3-4-7_vs_5-6	0.6154	**0.6667**	0.6087	0.6364	0.6154
ecoli-0-6-7_vs_3-5	0.6667	0.4444	**0.8000**	0.4444	0.7273
glass-0-1-5_vs_2	**0.4444**	0.2353	NA	0.2353	0.3333
glass-0-1-6_vs_2	NAN	0.1905	NA	0.1739	**0.6000**
glass-0-6_vs_5	1.0000	1.0000	1.0000	1.0000	1.0000
glass2	0.5455	0.2500	NA	0.2500	**0.6154**
glass4	**0.8000**	0.5000	NA	0.5000	0.6667
glass5	NA	NA	NA	NA	NA
page-blocks-1-3_vs_4	0.8235	**1.0000**	**1.0000**	**1.0000**	0.9333
poker-8_vs_6	NA	**0.0315**	0.0227	**0.0315**	NA
poker-8-9_vs_5	NA	0.0297	**0.0323**	0.0296	NA
poker-8-9_vs_6	NA	**0.0630**	0.0215	**0.0630**	NA
poker-9_vs_7	0.5000	0.2857	**0.5714**	0.3333	NA
shuttle-6_vs_2-3	1.0000	1.0000	1.0000	1.0000	1.0000
shuttle-c0-vs-c4	1.0000	1.0000	1.0000	1.0000	1.0000
shuttle-c2-vs-c4	1.0000	1.0000	1.0000	1.0000	1.0000
vowel0	0.8727	0.9180	NA	**0.9355**	0.9153
winequality-red-3_vs_5	NA	NA	NA	NA	NA
winequality-red-4	NA	**0.0963**	0.0833	**0.0963**	NA
winequality-red-8_vs_6	0.5000	**0.0769**	0.1765	0.7410	0.3077
yeast-0-5-6-7-9_vs_4	**0.6667**	0.4286	0.3200	0.5625	0.5161
yeast-2_vs_4	0.7500	0.7143	0.7500	0.7143	**0.7692**
yeast5	0.6957	**0.7273**	0.4407	0.6190	0.6957
yeast6	**0.4211**	0.4091	0.1263	0.3750	**0.4211**

ตารางที่ 4.9 แสดงค่าพารามิเตอร์ในการทดลอง และประสิทธิภาพของโมเดล เมื่อทดลองกับข้อมูลที่ค่า IR สูงกว่า 9

Dataset	ε	*minpts*	LI	SMOTE Cluster size	Minority Scale (%)	Accuracy	AUC	F-measure
ecoli-0-1-4-7_vs_2-3-5-6	0.10	11	278	17	**58.62**	0.9208	**0.8410**	0.6364
ecoli-0-3-4-7_vs_5-6	0.10	8	207	17	**68.00**	**0.9351**	0.5000	0.6154
ecoli-0-6-7_vs_3-5	0.10	9	178	18	**81.82**	0.9552	0.5000	0.7273
glass-0-1-5_vs_2	0.35	2	138	7	41.18	0.8462	0.6400	0.3333
glass-0-1-6_vs_2	0.30	2	158	9	**52.94**	**0.9310**	**0.7530**	**0.6000**
glass-0-6_vs_5	0.10	2	90	8	**88.89**	1.0000	1.0000	1.0000
glass2	0.50	2	180	5	29.41	**0.9219**	**0.8580**	**0.6154**
glass4	0.90	2	188	3	23.08	0.9531	**0.9510**	0.6667
glass5	0.10	2	196	9	**100.00**	0.9531	**0.5000**	NA
page-blocks-1-3_vs_4	0.10	6	416	21	**75.00**	0.9930	0.9930	0.9333
poker-8_vs_6	0.10	6	1443	13	**76.47**	**0.9910**	0.5000	NA
poker-8-9_vs_5	0.10	3	2025	17	**68.00**	**0.9871**	0.5000	NA
poker-8-9_vs_6	0.10	14	1435	17	**68.00**	**0.9820**	0.5000	NA
poker-9_vs_7	0.10	2	228	5	**62.50**	0.9178	0.4690	NA
shuttle-6_vs_2-3	0.10	6	210	8	**80.00**	1.0000	0.5000	1.0000
shuttle-c0-vs-c4	0.10	4	1583	80	**65.04**	1.0000	0.5000	1.0000
shuttle-c2-vs-c4	0.10	4	117	5	**83.33**	1.0000	0.5000	1.0000
vowel0	0.75	4	808	32	35.56	0.9831	0.5000	0.9153
winequality-red-3_vs_5	0.10	7	671	8	**80.00**	0.9758	0.4410	NA
winequality-red-4	0.10	4	1493	39	**73.58**	0.6946	0.5550	NA
winequality-red-8_vs_6	0.10	2	620	13	**72.22**	0.9543	0.6700	0.3077
yeast-0-5-6-7-9_vs_4	0.30	5	426	35	**68.63**	0.9051	0.6900	0.5161
yeast-2_vs_4	0.25	13	412	32	**62.75**	**0.9610**	0.8770	**0.7692**
yeast5	0.15	16	1396	31	**70.45**	**0.9843**	0.8040	0.6957
yeast6	0.10	13	1414	25	**71.43**	**0.9753**	0.5530	**0.4211**

จากผลการทดลองตามตารางข้างต้น ค่าผลการทดลองที่แสดงเป็นตัวอักษรแบบหนา แสดงถึงค่าสูงสุดในแต่ละการทดลองของแต่ละชุดข้อมูลตัวอย่าง จากค่า Minority Scale พบว่าเทคนิคที่ออกแบบสามารถตรวจพบคลัสเตอร์ของ Minority ขนาดใหญ่ที่มีจำนวนสมาชิกมากกว่าร้อยละ 50 ของจำนวนสมาชิก Minority ทั้งหมดได้ในเกือบทุกชุดข้อมูลที่นำมาทดลอง และโมเดลที่เรียนรู้จากข้อมูลตัวอย่างที่ผ่านการปรับสมดุลแล้วนั้น สามารถให้ประสิทธิภาพการจำแนกที่ดีขึ้นในบางชุดข้อมูลเท่านั้น โดยให้ค่า Accuracy ที่ดีในชุดข้อมูลตัวอย่าง ecoli-0-3-4-7_vs_5-6, glass-0-1-6_vs_2, glass2, poker-8_vs_6, poker-8-9_vs_5, poker-8-9_vs_6, yeast-2_vs_4, yeast5 และ yeast6 แต่ให้ค่า F-measure ที่ต่ำกว่าค่อนข้างมาก แล้วไม่สามารถคำนวณค่าได้ในหลายๆ ชุดข้อมูล

4.3 ผลการทดลองขยายผลการสุ่มตัวอย่างใหม่

ภายใต้สมมติฐานการวิจัยว่าบริเวณที่มี Positive Instance จับกลุ่มกันอยู่มากที่สุดหรือคลัสเตอร์ที่ใหญ่ที่สุด จะเป็นบริเวณที่เหมาะสมสำหรับการสร้างข้อมูลตัวอย่างใหม่นั้น ในเทคนิคที่พัฒนาขึ้นจึงมุ่งเน้นการสร้างตัวอย่างใหม่ขึ้นในเฉพาะคลัสเตอร์ที่ใหญ่ที่สุดที่ค้นพบเพียงคลัสเตอร์เดียวเท่านั้น โดยตั้งข้อสังเกตุว่าคลัสเตอร์ที่มีขนาดเล็กกว่าหรือตัวอย่างที่อยู่โดดเดี่ยวดออกไปภายหลังจากการแบ่งคลัสเตอร์ทั้งสองรอบแล้ว มีความเป็นไปได้ที่จะเป็นข้อมูลรบกวน(Noise หรือ Outlier) ดังนั้น การทดลองในส่วนนี้ ผู้วิจัยจึงได้ออกแบบวิธีการสังเคราะห์ตัวอย่างใหม่ ที่เรียกว่า DB2SM + Extended SMOTE หรือ DB2SMx เพื่อทดสอบสมมติฐานดังกล่าว โดยมีหลักการทำงานคือ กระจายการสังเคราะห์ตัวอย่างใหม่ด้วยเทคนิค SMOTE ไปยังคลัสเตอร์อื่นๆ ตามสัดส่วนของจำนวนสมาชิก Minority ที่อยู่ในแต่ละคลัสเตอร์ จนได้ผลรวมของสมาชิก Minority เท่ากันหรือใกล้เคียงกันกับจำนวนสมาชิก Majority จากนั้นจึงนำข้อมูลตัวอย่างที่ได้ปรับสมดุลแล้ว ไปใช้ในการสร้างโมเดลเพื่อวัดประสิทธิภาพด้วยวิธีการเดียวกันกับการทดลองเบื้องต้น โดยใช้ชุดของข้อมูลตัวอย่างเดียวกัน แล้วนำผลลัพธ์ที่ได้มาเปรียบเทียบ ดังแสดงในตารางต่อไปนี้

ตารางที่ 4.10 เปรียบเทียบประสิทธิภาพของโมเดล ระหว่างการใช้เทคนิคการสุ่มตัวอย่างใหม่ DB2SM และ DB2SMx

Dataset	Parameters		Accuracy		AUC		F-measure	
	ε	*minpts*	DB2SM	DB2SMx	DB2SM	DB2SMx	DB2SM	DB2SMx
ecoli1	0.50	16	0.8317	**0.8911**	0.8275	**0.9320**	0.7018	**0.7843**
ecoli2	0.15	14	**0.9703**	0.8416	**0.9327**	0.8580	**0.9231**	0.6522
ecoli3	0.10	1	**0.9307**	0.8713	**0.8508**	0.8150	0.3636	**0.4800**
glass0	0.50	5	**0.8438**	0.8125	0.8886	**0.8890**	**0.7368**	0.7143
glass1	0.50	3	**0.7344**	0.6562	**0.7179**	0.6190	**0.6667**	0.5600
glass6	0.35	2	**1.0000**	0.9219	**1.0000**	0.9640	**1.0000**	0.7826
harberman	0.10	2	**0.6739**	0.5435	**0.5711**	0.5470	**0.4444**	0.3824
iris0	0.20	2	1.0000	1.0000	0.5000	0.5000	1.0000	1.0000
new-thyroid1	0.10	11	0.9531	0.9531	0.8907	**0.9330**	0.8421	**0.8571**
new-thyroid2	0.10	16	0.9688	**0.9844**	0.9734	**1.0000**	0.9167	**0.9524**
page-blocks0	0.10	8	**0.9714**	0.9671	0.9142	**0.9450**	**0.8554**	0.8439
pima	0.10	14	**0.7000**	0.6478	**0.7425**	0.7230	**0.6634**	0.6553
segment0	0.85	5	**0.9913**	0.9855	0.9711	**0.9750**	**0.9643**	0.9419
vehicle0	0.10	7	**0.9252**	0.8386	**0.9394**	0.8980	**0.8652**	0.7453
vehicle1	0.10	2	**0.7677**	0.4094	**0.6374**	0.5860	0.3059	**0.4828**
vehicle2	0.10	13	**0.9409**	0.8780	**0.9185**	0.8750	**0.8855**	0.7770
vehicle3	0.11	11	**0.7047**	0.3819	**0.6458**	0.5510	**0.4681**	0.4530
wiscosin	0.10	17	**0.9512**	0.9268	**0.9376**	0.8920	**0.9254**	0.8889
yeast1	0.05	1	**0.7551**	0.7124	0.7026	**0.7300**	0.5856	**0.6121**
yeast3	0.10	7	**0.9618**	0.9483	**0.9248**	0.9210	**0.8350**	0.7810

จากผลการทดลองพบว่า DB2SMx สามารถเพิ่ม Accuracy ได้ดีกว่า DB2SM ในข้อมูลบางชุดข้อมูลตัวอย่างเท่านั้น ได้แก่ ecoli1 และ new-thyroid2 แต่ก็สามารถเพิ่ม F-measure ได้ในหลายชุดข้อมูล ได้แก่ ecoli1, ecoli3, new-thyroid1, new-thyroid2, vehicle1 และ yeast1

บทที่ 5
บทสรุปและข้อเสนอแนะ

งานวิจัยนี้มีวัตถุประสงค์ในการปรับปรุงวิธีการสุ่มตัวอย่างใหม่สำหรับข้อมูลไม่สมดุล เพื่อที่จะแก้ไขจุดบกพร่องของชุดข้อมูลตัวอย่างระหว่างจำนวนสมาชิกของกลุ่มตัวอย่าง Majority และ Minority ให้มีสัดส่วนอยู่ในปริมาณที่เท่ากันหรือใกล้เคียงกัน และเหมาะสำหรับการนำไปใช้ในการเรียนรู้ของเครื่องจักร โดยคาดว่าชุดข้อมูลตัวอย่างที่ปรับปรุงขึ้นจะส่งผลให้โมเดลการเรียนรู้มีประสิทธิภาพในการจำแนกที่ดีขึ้น และลดโอกาสที่จะการเกิดความเอนเอียงในการจำแนกลง

5.1 สรุปผลการศึกษา

เพื่อให้สอดคล้องกับวัตถุประสงค์การวิจัยที่ตั้งไว้ จึงแบ่งสรุปผลการศึกษาออกเป็น 2 ส่วน ดังนี้

1. การออกแบบเทคนิคการสุ่มตัวอย่างใหม่ งานวิจัยนี้ได้นำเสนอเทคนิคการสุ่มตัวอย่างใหม่แบบผสม (Combined Technique) สำหรับแก้ไขปัญหาข้อมูลไม่สมดุล เรียกว่า เทคนิค DB2SM ภายใต้แนวคิดหลักสองประการคือ หลีกเลี่ยงการใช้เทคนิคการลดจำนวนตัวอย่าง (Undersampling) เพื่อลดจำนวนสมาชิกของกลุ่ม Majority ออก ซึ่งอาจจะมีความเป็นไปได้ที่จะทำให้เกิดการสูญเสียข้อมูลที่สำคัญหรือคุณลักษณะของข้อมูล (Data Characteristics) ไปโดยไม่ตั้งใจ อีกประการหนึ่งคือการค้นหาบริเวณที่เหมาะสมที่สุดสำหรับการสร้างตัวอย่างเพิ่ม (Oversampling) ในกลุ่มของ Minority และพยายามหลีกเลี่ยงตัวอย่างที่มีโอกาสเป็นข้อมูลรวบกวน (Noise หรือ Outlier) ให้มากที่สุด เพื่อให้ได้ตัวอย่างใหม่ที่มีคุณภาพใกล้เคียงกับตัวอย่างเดิมของชุดข้อมูลนั้น เทคนิคการสุ่มตัวอย่างใหม่ที่นำเสนอนี้ ประกอบด้วยการทำงานร่วมกันระหว่างเทคนิคการแบ่งกลุ่มตามแนวคิด DBSCAN (Density Based Spatial Clustering of Applications with Noise) เพื่อแบ่งข้อมูลออกเป็นกลุ่มย่อย แล้วค้นหาคลัสเตอร์หรือกลุ่มของ Minority ที่มีจำนวนสมาชิกมากที่สุด จากนั้นจึงทำการสังเคราะห์ตัวอย่างขึ้นในคลัสเตอร์นั้น ด้วยเทคนิค SMOTE (Synthetic Minority Over-sampling Technique) โดยอ้างอิงจากสมาชิกในคลัสเตอร์ จนได้ผลรวมของจำนวนตัวอย่างของกลุ่ม Minority ใกล้เคียงหรือเท่ากันกับจำนวนตัวอย่างของกลุ่ม Majority ทำให้ชุดข้อมูลตัวอย่างใหม่ที่ได้มีความสมดุลระหว่างสมาชิกทั้งสองกลุ่ม และสามารถนำไปใช้งานในการเรียนรู้ของเครื่องจักรได้ต่อไป

2. ประเมินประสิทธิภาพของกระบวนการ หลังจากที่ออกแบบเทคนิคการสุ่มตัวอย่างใหม่แล้วนั้น จึงได้ทำการทดลองเพื่อวัดคุณภาพของชุดข้อมูลตัวอย่างที่ผ่านการปรับสมดุลแล้ว โดยนำไปใช้เป็นข้อมูลในการฝึกสำหรับสร้างโมเดลการเรียนรู้โดยผ่านอัลกอริธึม Decision Tree และวัดประสิทธิภาพของโมเดลด้วยค่า Accuracy, AUC และ F-measure แล้วเปรียบเทียบกับผลลัพธ์ที่ได้จากการใช้เทคนิคการสุ่มตัวอย่างใหม่อื่นที่มีลักษณะการทำงานคล้ายกัน ได้แก่ การใช้เทคนิค SMOTE เพียงอย่างเดียว เทคนิคแบบผสม DBCS เทคนิคแบบผสม DBSM รวมถึงการใช้ข้อมูลต้นฉบับที่ไม่มีการปรับแต่งใดๆ

จากผลการทดลองเบื้องต้นในบทที่ 4 พบว่าเทคนิคที่นำเสนอสามารถปรับปรุงชุดข้อมูลตัวอย่างที่ไม่สมดุล ให้มีสัดส่วนของจำนวนตัวอย่างของ Minority ใกล้เคียงหรือเท่ากันกับจำนวนตัวอย่างของ Majority ได้ตามที่ต้องการ โดยสามารถค้นหาคลัสเตอร์ของ Minority ที่มีจำนวนสมาชิกมากที่สุดเพื่อทำการสังเคราะห์ตัวอย่างขึ้นในคลัสเตอร์นั้นเพื่อปรับสมดุล และเมื่อนำข้อมูลที่ปรับปรุงแล้วไปใช้ในการฝึกโมเดลและวัดประสิทธิภาพ พบว่าค่า Accuracy ของโมเดลที่ได้มีความแม่นยำในการจำแนกที่ดีขึ้นในหลายชุดข้อมูล เมื่อเปรียบเทียบกับเทคนิคอื่น และมีค่า AUC ที่สอดคล้องไปในทิศทางเดียวกัน ส่วนค่า F-measure ที่แสดงถึงสัดส่วนของความแม่นยำในการจำแนกระหว่างตัวอย่าง positive และ negative มีค่าน้อยกว่าค่า Accuracy ในระดับที่ไม่มากนัก ซึ่งหมายถึงโมเดลสามารถจำแนกระหว่างข้อมูลทั้งสองกลุ่มได้ในระดับที่ใกล้เคียงกัน เมื่อมองในภาพรวมแล้วสามารถประเมินได้ว่าคุณภาพของชุดข้อมูลที่ถูกปรับปรุงด้วยเทคนิคที่นำเสนอ มีความเหมาะสมสำหรับนำไปใช้งานในการเรียนรู้ของเครื่องจักรได้ดีขึ้นอย่างมีนัยสำคัญ

เมื่อทำการทดลองเพิ่มเติมกับข้อมูลที่มีความไม่สมดุลสูง หรือมีค่าอัตราความไม่สมดุลของข้อมูลสูงกว่า 9 จากผลการทดลองพบว่าโมเดลที่เรียนรู้โดยใช้ข้อมูลตัวอย่างที่ผ่านการปรับสมดุลด้วยเทคนิคที่นำเสนอ สามารถให้ประสิทธิภาพการจำแนกที่ดีขึ้นในบางชุดข้อมูลเท่านั้น และให้ค่า F-measure ที่ค่อนข้างต่ำหรือไม่สามารถคำนวณค่าได้ แสดงให้เห็นถึงความเอนเอียง (Bias) ของโมเดลที่ยังไม่สามารถจำแนกระหว่างตัวอย่างในกลุ่ม Majority และ Minority ได้อย่างถูกต้องเพียงพอต่อการนำไปใช้งาน

ในการทดลองส่วนที่สาม เพื่อเป็นการทดสอบสมมติฐานเกี่ยวกับบริเวณที่เหมาะสำหรับการสังเคราะห์ข้อมูลใหม่นั้น ได้ขยายผลจากการทดลองเบื้องต้นที่มุ่งเน้นการสร้างตัวอย่างใหม่เฉพาะในคลัสเตอร์ใหญ่ที่สุดที่ค้นพบเพียงคลัสเตอร์เดียว ออกไปยังคลัสเตอร์อื่น ด้วยเทคนิค DB2SM+Extended SMOTE หรือ DB2SMx แล้ววัดประสิทธิภาพด้วยหลักเกณฑ์และวิธีเดียวกันกับการทดลองเบื้องต้น จากผลการทดลองพบว่า DB2SMx สามารถเพิ่ม Accuracy และ AUC ได้

ดีกว่า DB2SM ในข้อมูลบางชุดข้อมูลเท่านั้น แต่ก็ช่วยปรับปรุง F-measure ของโมเดลให้ดีขึ้นในหลายชุดข้อมูลด้วยเช่นกัน

5.2 อภิปรายผลการศึกษา

จากผลการทดลองเบื้องต้น พบว่ากระบวนการของ DB2SM สามารถตอบสนองตามวัตถุประสงค์ที่ต้องการได้ดีในระดับหนึ่ง นั่นคือ ช่วยแก้ไขปัญหาความไม่สมดุลของชุดข้อมูลตัวอย่างที่ใช้ทดสอบได้ และส่งผลต่อประสิทธิภาพของโมเดลที่ได้จากการใช้ข้อมูลที่ปรับปรุงแล้วนั้นร่วมกับอัลกอริธึมการเรียนรู้แบบ Decision Tree ในเชิงประจักษ์ แต่ด้วยพื้นฐานของเทคนิคที่ออกแบบขึ้นมานั้น ตั้งอยู่บนการทำงานร่วมกันระหว่างเทคนิคการแบ่งกลุ่มด้วยวิธี DBSCAN และเทคนิคการสังเคราะห์ตัวอย่างด้วยวิธี SMOTE ซึ่งเทคนิค DBSCAN จำเป็นจะต้องกำหนดค่าพารามิเตอร์ที่สำคัญ ได้แก่ ค่าระยะห่างระหว่างสมาชิกในคลัสเตอร์หรือค่า epsilon และค่าจำนวนสมาชิกขั้นต่ำของคลัสเตอร์หรือค่า minpnts เพื่อให้ได้ขนาดของคลัสเตอร์ที่เหมาะสม และจะสามารถแบ่งแยกสมาชิกหรือตัวอย่างที่มีโอกาสเป็นตัวรวกวนออกไปให้ได้มากที่สุด โดยเมื่อเปรียบเทียบกับเทคนิคการสุ่มตัวอย่างใหม่ที่นำมาเปรียบเทียบในการศึกษานี้ จะเห็นว่าในเทคนิคอื่นจะพยายามหาคลัสเตอร์ที่ใหญ่ที่สุดที่ยังคงประกอบไปด้วยตัวอย่างกลุ่ม Majority และ Minority อยู่รวมกัน ซึ่งเป็นผลจากการใช้เทคนิค DBSCAN เพียงรอบเดียว จากนั้นจึงทำการสังเคราะห์ตัวอย่างใหม่ขึ้นมาภายในคลัสเตอร์ที่มีตัวอย่างทั้งสองคลาสปะปนกันอยู่นั้น ส่งผลให้ตัวอย่างใหม่ที่สร้างขึ้นมีโอกาสที่จะเกิดจากการอ้างอิงกับเพื่อนบ้านที่เป็นทั้ง Majority หรือ Minority หรือ Noise ก็เป็นได้

ดังนั้น ในเทคนิคที่นำเสนอจึงใช้วิธีการแบ่งกลุ่มตัวอย่างที่ได้จากการทำ DBSCAN ในรอบแรกแล้วซ้ำอีกครั้ง โดยตัดตัวอย่าง Majority ที่ทำหน้าที่เป็นเสมือนจุดเชื่อมต่อระหว่างคลัสเตอร์ของ Minority ออกไป ทำให้เมื่อทำการแบ่งคลัสเตอร์อีกครั้ง จะได้ผลลัพธ์เป็นกลุ่มของคลัสเตอร์ที่ประกอบด้วยตัวอย่างคลาสใดคลาสหนึ่งเพียงอย่างเดียว หรือเป็นตัวอย่างที่อยู่โดดเดี่ยวออกไป ช่วยให้เราสามารถค้นพบกลุ่มของ Minority ที่ใหญ่ที่สุดที่แท้จริงได้ชัดเจนมากขึ้น และเมื่อทำการสังเคราะห์ตัวอย่างใหม่ด้วยเทคนิค SMOTE ได้ทำการทดลองโดยใช่ค่าของจำนวนสมาชิกขั้นต่ำของคลัสเตอร์หรือค่า minpnts จากเทคนิค DBSCAN มาเป็นตัวกำหนดจำนวนเพื่อนบ้านที่เหมาะสมสำหรับการสร้างตัวอย่างใหม่ ทำให้การสร้างตัวอย่างสอดคล้องกับการก่อตัวของคลัสเตอร์ และช่วยลดความซับซ้อนในการคำนวณลงไปได้ระดับหนึ่ง ซึ่งแตกต่างจากบางเทคนิคที่นำเทคนิคที่มีความซับซ้อนเข้ามาช่วยในการปรับค่าพารามิเตอร์ให้เหมาะสม เช่น การใช้ขั้นตอนวิธี

เชิงพันธุกรรม ซึ่งช่วยให้ส่งผลให้ได้ผลลัพธ์ที่ดีขึ้น แต่กระบวนการในการทำงานต้องใช้ Computational Time ที่มากขึ้นด้วย (อนันตพร, 2560)

ในการทดลองกับชุดข้อมูลตัวอย่างที่มีค่าอัตราความไม่สมดุลของข้อมูลสูงกว่า 9 โมเดลกลับมีประสิทธิภาพการจำแนกที่ดีขึ้นเพียงเล็กน้อย ในขณะที่ค่า F-measure ค่อนข้างต่ำหรือไม่สามารถคำนวณค่าได้ แต่เมื่อพิจารณาร่วมกับเทคนิคอื่น พบว่าส่วนใหญ่ให้ค่า F-measure ที่ดีกว่าการใช้ข้อมูลตัวอย่างต้นฉบับที่ไม่มีการปรับสมดุล แต่ที่ให้ค่าน้อยเกิดจากคุณลักษณะโดยปกติของการเกิดความเอนเอียงในการจำแนก เนื่องจากการใช้ข้อมูลที่ไม่สมดุล สะท้อนให้เห็นว่าหากข้อมูลที่นำมาใช้ในการเรียนรู้มีอัตราความไม่สมดุลของข้อมูลที่สูงมาก การแก้ไขในระดับของ Data Level อาจไม่เพียงพอต่อการนำข้อมูลไปใช้ ควรจะต้องมีการปรับปรุงในระดับของ Algorithm Level ควบคู่กันไปด้วย เช่น การถ่วงน้ำหนักตัวอย่างด้วยการทำ Cost-sensitive เพื่อเพิ่มความสำคัญให้กับตัวอย่างในกลุ่ม Minority .ให้มากขึ้นในขณะที่นำมาใช้งาน หรือปรับเปลี่ยนวิธีการที่ใช้ในการเรียนรู้ของโมเดล เช่น การให้โมเดลเรียนรู้ตัวอย่างเพียงด้านเดียวจากตัวอย่าง Majority หรือการทำ One-class Learning เพื่อให้โมเดลสามารถอนุมานได้ว่า ถ้าหากพบตัวอย่างที่แตกต่างจากที่ได้เรียนรู้มา ให้จำแนกตัวอย่างที่พบนั้นเป็นตัวอย่างในกลุ่ม Minority เป็นต้น

นอกจากผลการทดลองทั้งสองส่วนที่กล่าวมาข้างต้น ผู้วิจัยได้ทำการทดลองขยายผลของการกระจายการสร้างตัวอย่างใหม่ไปยังคลัสเตอร์ใกล้เคียง เพื่อพิสูจน์สมมติฐานที่ตั้งไว้ ซึ่งพบว่าเทคนิค DB2SMx ที่ปรับปรุงขึ้นช่วยเพิ่มประสิทธิภาพในการจำแนกของโมเดลเมื่อพิจารณาจากค่า F-measure ในบางชุดข้อมูล และเมื่อพิจารณาในขั้นตอนของการทำ Oversampling ของเทคนิค พบว่าชุดข้อมูลตัวอย่างที่ถูกแบ่งกลุ่มนั้น มีการกระจายตัวของกลุ่ม Minority ออกเป็นคลัสเตอร์ย่อยๆ ที่มีขนาดใกล้เคียงกัน ไม่ได้เกาะกันเป็นกลุ่มคลัสเตอร์ขนาดใหญ่ที่แตกต่างโดดเด่นออกมาอย่างชัดเจนเพียงคลัสเตอร์เดียว ดังนั้น ในแง่ของการนำเทคนิคการสุ่มตัวอย่างใหม่นี้ไปใช้อาจจะต้องเพิ่มขั้นตอนของการตรวจสอบการกระจายตัวของข้อมูลชุดนั้นก่อนว่ามีลักษณะการเกาะกลุ่มกันอยู่อย่างไร เพื่อช่วยให้สามารถตัดสินใจได้ว่าเมื่อใดควรจะใช้เทคนิค DB2SM ที่มุ่งเน้นทำงานกับคลัสเตอร์ที่ใหญ่ที่สุดเพียงคลัสเตอร์เดียว หรือควรใช้เทคนิค DB2SMx เพื่อกระจายการสร้างตัวอย่างใหม่ออกไปยังคลัสเตอร์อื่นๆ ให้ได้ข้อมูลที่สมดุล และเพิ่มประสิทธิภาพของการทำ Oversampling ที่ดียิ่งขึ้น อีกทั้งเป็นการใช้ทรัพยากรในการประมวลผลที่คุ้มค่า อย่างไรก็ตาม ในการวิจัยเบื้องต้นนี้สามารถยืนยันได้ว่า บริเวณที่มีตัวอย่างกลุ่มเดียวกันรวมตัวกันอยู่มากที่สุด มีความเหมาะสมสำหรับการสังเคราะห์ตัวอย่างเทียมขึ้นในพื้นที่นั้นอย่างมีนัยสำคัญตามสมมติฐาน และสามารถนำไปใช้งานได้จริง

5.3 ข้อเสนอแนะ

อย่างไรก็ตาม เนื่องจากการใช้เทคนิคการแบ่งกลุ่มมาเป็นพื้นฐานในการจำแนกตัวอย่างออกจากกันในขั้นต้นของกระบวนการของเทคนิคแบบผสมที่ศึกษาในงานวิจัยนี้ พบว่าสาเหตุหรือปัจจัยหลักที่ส่งผลกระทบต่อประสิทธิภาพของการปรับสมดุลของข้อมูลคือ ลักษณะการกระจายตัวของกลุ่มข้อมูลนั่นเอง ซึ่งโดยธรรมชาติของเหตุการณ์หรือปรากฏการณ์ต่างส่วนใหญ่มักจะให้ข้อมูลที่มีลักษณะเกาะกลุ่มหรืออยู่ในช่วงข้อมูลที่ใกล้เคียงกัน เช่น ข้อมูลการทำธุรกรรมทางการเงิน การฝาก-ถอนเงินมักจะมีมูลค่าในการทำธุรกรรมในวิสัยปกติที่สามารถพบเห็นได้ประจำอยู่ในช่วงมูลค่าเงินจำนวนหนึ่ง ถ้าหากมีการฝาก-ถอนเงินมูลค่ามหาศาลขึ้น หรือมีการโยกย้ายสับเปลี่ยนเงินระหว่างบัญชีในระดับหน่วยของสตางค์ที่เล็กน้อย ย่อมอาจจะสื่อถึงความผิดปกติที่เกิดขึ้นในระบบการธุรกรรมการเงินก็เป็นได้ ซึ่งการจัดกลุ่มของข้อมูลในลักษณะนี้มีความเป็นไปได้ที่จะจำแนกออกกันได้ค่อนข้างชัดเจน หรืออีกตัวอย่างหนึ่ง เช่น ข้อมูลการพบเห็นปรากฏการณ์ UFO ที่เกิดขึ้นทั่วโลก ส่วนใหญ่จะถูกพบในลักษณะตำแหน่งที่พบกระจัดกระจายไปทั่ว และไม่ยึดติดกับตำแหน่งพิกัด สถานที่ เวลาที่พบ ทำให้ข้อมูลตัวอย่างมีลักษณะการกระจายตัวที่สูงมาก โอกาสที่จะสามารถแบ่งข้อมูลออกเป็นกลุ่มจึงทำได้ยาก ทุกตัวอย่างมีความเป็นไปได้ว่าจะเป็นข้อมูลรบกวน หรือเป็น Outlier ได้เท่าๆ กัน การแก้ไขในประเด็นนี้สามารถทำได้โดยศึกษาหาวิธีการที่จะจัดแบ่งกลุ่มของข้อมูลออกให้ได้อย่างมีประสิทธิภาพ เช่น การทำ Feature Selection ร่วมกับการใช้เทคนิคการแบ่งกลุ่ม เพื่อเลือกตัวแปรที่ส่งผลหรือสำคัญต่อการจำแนกอย่างแท้จริงมาใช้งาน หรือการใช้อัลกอริธึมที่มีความซับซ้อนมากขึ้นอย่างโครงข่ายประสาทเทียม (Neural Network) แต่ต้องตระหนักถึงระยะเวลาในการคำนวณเมื่อนำมาใช้งานประกอบกันด้วย

นอกจากนี้ ข้อมูลตัวอย่างที่นำมาใช้ในการศึกษา ถูกกำหนดอยู่ภายใต้เงื่อนไขที่จำกัด ได้แก่ แต่ละชุดข้อมูลตัวอย่างเป็นข้อมูลที่มีตัวแปรประมาณ 4-20 ตัวแปร ประกอบด้วยสมาชิกเพียง 2 กลุ่ม คือ *positive* และ *negative* เท่านั้น และไม่มีข้อมูลที่สูญหาย (Non-Missing Values) แต่ในความเป็นจริง ข้อมูลที่ศึกษาส่วนใหญ่ประกอบไปด้วยตัวแปรจำนวนมาก มีสมาชิกมากกว่า 2 กลุ่ม และที่สำคัญคือโอกาสที่จะพบข้อมูลที่เป็น Null หรือ Missing Values มีค่อนข้างสูง ในขณะที่อัตราความไม่สมดุลของข้อมูลยิ่งมีสูงขึ้นแบบทวีคูณ ดังนั้น การขยายผลการศึกษาในอนาคตเพื่อค้นหากระบวนการที่สามารถรับมือกับปัญหาข้อมูลไม่สมดุลที่มีลักษณะที่ซับซ้อนได้อย่างมีประสิทธิภาพนั้น ยังคงเป็นเรื่องที่ท้าทายสำหรับการทำงานในด้านการเรียนรู้ของเครื่องจักรต่อไป

บรรณานุกรม

บรรณานุกรม

เบญจภรณ์ จันทรกองกุล, สุวรรณา รัศมีขวัญ, สุนิสา ริมเจริญ, ภูสิต กุลเกษม, กฤษณะ ชินสาร, อัณณ์นุพันธ์ รอดทุกข์, … จรรยา อันปันส์. (2557). *วิธีการที่เหมาะสมสำหรับการแบ่งกลุ่มข้อมูลที่ไม่สมดุลสูง*. คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา

อนันตพร หรรษคุณาฒัย. (2560). *วิธีใหม่ในการสุ่มตัวอย่างแบบผสมสำหรับการจำแนกประเภทของชุดข้อมูลที่ไม่สมดุล*. สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

Alcalá-Fdez, J., Fernandez, A., Luengo, J., Derrac, J., García, S., Sánchez, L., & Herrera, F. (2011). *KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework*. Journal of Multiple-Valued Logic and Soft Computing 17:2-3 (2011), pp. 255-287.

Ali, H., Salleh, M., Saedudin, R., Hussain, K., & Mushtaq, M. (2019). *Imbalance class problems in data mining: a review*. In: Indonesian Journal of Electrical Engineering and Computer Sci-ence Vol. 14, No. 3, June 2019.

Chawla, N.V., Bowyer, K.W., Hall, L.O., & Kegelmeyer, W.P. (2002). *SMOTE: Synthetic Minority Over-sampling Technique*. Journal Of Artificial Intelligence Research, Vol. 16, pp. 321-357.

Dua, D. & Graff, C. (2019). *UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]*. Irvine, CA: University of California, School of Information and Computer Science.

Ester, M., Kriegel, H., Sander, J., & Xu, X. (1996). *A Density-Based Algorithm for Discovering Clusters in Large Spatial Database with Noise*. KDD, Vol. 96, No. 34, pp. 226-231, August 1996.

Gan, J., & Tao, Y. (2017). *On the Hardness and Approximation of Euclidean DBSCAN*. ACM Trans Database Syst. 42, 3, Article 14, July 2017.

Josh. (2015). *What does AUC stand for and what is it?*. Retrieved Decemver 26, 2020, from https://stats.stackexchange.com/questions/132777

Longadge, R., Dongre, S., & Malik, L. (2013). *Class Imbalance Problem in Data Mining: Revi*ew. International Journal of Computer Science and Network (IJCSN) Vol. 2, Issue 1, February 2013.

Lv, M., Ren, Y., & Chen, Y. (2019). *Research on imbalanced data : based on SMOTE-AdaBoost algorithm*. The 3rd International Conference on Electronic Information Technology and Computer Engineering (EITCE), pp. 1165-1170.

Netirungroj, N., & Pacharawongsakda, E. (2018). *TOP: An Efficient TwO-levels of Positive Resampling Framework for Class Imbalanced Data*. IEEE International Conference on Big Data, Cloud Computing, Data Science & Engineering (BCD).

Sanguanmak, Y., & Hanskunatai, A. (2016). *DBSM: The combination of DBSCAN and SMOTE for imbalanced data classification.* 13th International Joint Conference on Computer Science and Software Engineering, JCSSE 2016.

Schubert, E., Sander, J., Ester, M., Kriegel, H., & Xu, X. (2017). *DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN*. ACM Trans. Database Syst. 42, 3, Article 19, July 2017.

Verma, M.K., Xaxa, M.K., & Verma, S. (2017). *DBCS: Density based cluster sampling for solving imbalanced classification problem.* International conference of Electronics, Com-munication and Aerospace Technology (ICECA).

Wijaya, C. (2020). *5 SMOTE Techniques for Oversampling your Imbalance Data*. Retrieved September 14, 2020, from https://towardsdatascience.com/5-smote-techniques-for-oversampling-your-imbalance-data-b8155bdbe2b5

Xiaolong, X., Wen, C., & Yenfei, S. (2019). *Over-sampling algorithm for imbalanced data classifica-tion*. Journal of Systems Engineering and Electronics Vol. 30, No. 6, December 2019, pp.1182– 1191.

ภาคผนวก

ภาคผนวก ก

ผลงานตีพิมพ์

DB2SM: An Efficient Resampling Technique for Imbalanced Data Classification

Pharnuphon Jiraamphorn[1] and Eakasit Pacharawongsakda[2]

Big Data Engineering program, College of Innovative Technology and Engineering,
Dhurakij Pundit University, Bangkok, Thailand
[1]6251620200002@dpu.ac.th, [2]eakasit.pac@dpu.ac.th

Abstract. Data classification is the main task of Machine Learning. However, it is not easy to build a good classification model because it requires well-prepared data and efficient classification techniques. One of the data issues that we inevitably have to face is the imbalanced data problem. This problem typically refers to a situation that the classes are not represented equally, e.g., one class is more occur than another class. Therefore, the classifier will be biased to the majority class and give high performance for that class. To deal with this issue, various techniques have been proposed. In this work, we introduce an improvement technique, called DB2SM. Our method employs 2 times DBSCAN clustering to find a proper area and SMOTE for oversample minority instances. For more detail, the first clustering task removes the negative instances (e.g., majority class), which are located close to the positive classes. Then the second clustering task generates clusters that have only positive or negative instances. Finally, SMOTE was applied to create new positive instances. The experimental results with 20 imbalanced datasets showed that our proposed method gives a better performance in terms of accuracy, AUC and F-measure, compared to SMOTE, DBCS and DBSM.
Keywords: Imbalanced Data, Resampling, DBSCAN, SMOTE.

Introduction

Handling imbalanced data is one of the most challenging tasks in Machine Learning. In real-world use-cases, for example, fraud detection, intrusion detection, and disease diagnostics have an imbalanced data issue. There are a few fraudulent cases compared to normal cases and a few patients compared to normal people. A large number of instances in the dataset are called the majority class and the fewer are called the minority class. In this situation, the classification model will be biased to the majority class and give high accuracy. To handle these issues, there are two main approaches: data level and algorithm level [1]. The methods in the first approach are focused on sampling data. It tends to resample instances to a balanced dataset, for example, random undersampling (RUS) or random oversampling (ROS). The second approaches adjust the machine learning techniques to handle the imbalanced dataset. The techniques in this approach including one class SVM, cost sensitive learning.

Each solution has its own limitation. For example, the quality of synthetic instances using SMOTE technique is dependent on the k-nearest neighbor data. If they contain noises or outliers, it will reduce the potential of new instances too [2]. This paper introduces an improvement technique by combining DBSCAN and SMOTE to solve some existing problems and improve oversampling quality on the imbalanced data. By the assumption to generate new instances in the high-density area or largest cluster of minority class will receive a good instance quality as the original data does.

Related Works

In this section, we reviewed previously research works which aimed to address the imbalanced data.

M. Ester et al. [3] presented DBSCAN (A Density-Based Algorithm for Discovering Clusters in Large Spatial Database with Noise) which is a technique that clustered data by exploring every instance. If they have the number of neighbors equal to or greater than the minimum number of points (*minpts*) and located within a constant radius called epsilon (ε) then the algorithm will form a cluster for it and extend the cluster area by applying the same play to their neighbors. The DBSCAN was mostly used because it can handle unlabeled data and can cluster to various shapes instead of an oval.

The Random Over-Sampling (ROS) technique is the simplest method that duplicates the minority class instances. However, this might lead to an over-fitting issue since the new instances are the same as previously. To overcome this problem, Chawla et al. [4] proposed another way by synthetic new instances near the previous minority class instances. This technique is called SMOTE (Synthetic Minority Over-sampling TEchnique) and it's widely used to do over-sampling in the imbalanced dataset.

Y. Sanguanmak and A. Hanskunatai [5] proposed a hybrid technique called DBSM that combining DBSCAN and SMOTE in the same process. After applying DBSCAN to the imbalanced dataset, DBSM removes 50% of negative (majority class) instances in each cluster which have the smallest distance from their centroid. This task is similar to undersampling but it does not random. Then DBSM employs SMOTE to oversampling positive (minority class) instances. This method gave improved performance compared to base-line techniques.

M.K. Verma et al. [6] presented DBCS method which applies DBSCAN to group data into clusters. In the next step, DBCS computes the ratio of positive and negative instances and suggests resampling techniques e.g., undersampling or oversampling to each cluster. For oversampling, it generates the synthetic positive instances within an epsilon (ε) value of each cluster. This method was examined with eleven datasets from KEEL data repository with C4.5 classifier. The results showed a better average AUC and accuracy if model via compared with the using of original data, SMOTE, ROS, and RUS techniques.

X. Xiaolong et al. [7] used the DBSCAN to separate positive instances into three groups including an explicit group of minority class called the core samples, a combination group of majority class, and minority class called the borderline samples, and the isolated minority that located among majorities called the noise samples. The algorithm removed all noise samples before applied different oversampling techniques for remaining groups. The first is applying SMOTE within the core samples, the second is calculating the cluster center in borderline samples and then generated synthetic instances between the cluster center and other instances in the cluster. This work is called DSMOTE. The experiment was done on four UCI datasets which contained two-class and multi-class datasets. The results showed better performances including precision, recall, and F-value while compared with original data, SMOTE, and borderline-SMOTE with J48 decision tree classifier.

M. Lv et al. [8] combined the SMOTE, Boderline SMOTE and boosting techniques such as AdaBoost and cost-sensitive AdaBoost to handle imbalanced credit card clients datasets from UCI. A comparison result show the performance of over-sampled SMOTE-AdaBoost and Borderline SMOTE AdaBoost is slightly worse than the traditional AdaBoost. They noticed on a possibility of using SMOTE with high-dimensional imbalanced data.

N. Netirungroj and E. Pacharawongsakda [9] proposed two steps framework which employs both undersampling and oversampling techniques. This work is called TOP (TwO-levels of Positive Resampling Framework). TOP creates two areas: inner area and outer area. The first area is located near the positive instances and the second area is far from those positive instances. They applied the undersampling technique in the inner area and the oversampling technique to the outer area. An experiment was testing on many UCI imbalanced datasets and evaluated the model performance by using F1, GM, AUROC, and AUPRC. The experimental results were compared with many techniques such as Baseline-SMOTE, ROSE and DBSM. The technique yielded F1 score better than an average improvement across all datasets.

DB2SM Methodology

The goal of this work is to improve an oversampling quality by trying to generate a new instance in the proper area of minority class. The process contains two main function: the first is finding the high density area or largest cluster of minority instances (assumed that this is the best area for the minority class) by applying two rounds of DBSCAN, then the second is creating new instances by applying SMOTE in that area. The process was explained in these following steps:

Step 1. The first round of DBSCAN is applied with parameter: epsilon = e, minimum point = minpnts, and add the result clusters in C. The C contains positive clusters (all instances have minority class), negative clusters (all instances have majority class), and mixed cluster (included positive instances and

negative instances). All members in each cluster connected to their neighbors within distance <= e by DBSCAN algorithm.

Step 2. For each mixed cluster in C, we calculate Euclidean distance between all positive members and all negative members in C.

Step 3. Remove negative instances that have distance less than epsilon (e) from C. This condition reveals a positive group in clusters and separates them from connected by negative instances. (We keep the removed negative instances in an empty dataset called delInstance for later use)

Step 4. Combine all instances from the 3rd step and re-apply the second DBSCAN with the same parameters again. By assumption, C might contain the positive cluster, the negative cluster, and some isolated instances.

Step 5. Choose the largest positive cluster from the 4th step and apply SMOTE with parameter: number of instances = LI from equation (2). Then remove duplicated instances by comparing new instances with members in delInstance. (We assume that positive members will less than negative members to prevent bias and over-fitting from new positive members)

Step 6. Appending original data with new instances as the new training data and use this dataset to build a classification model. The processes are shown in Fig. 1.

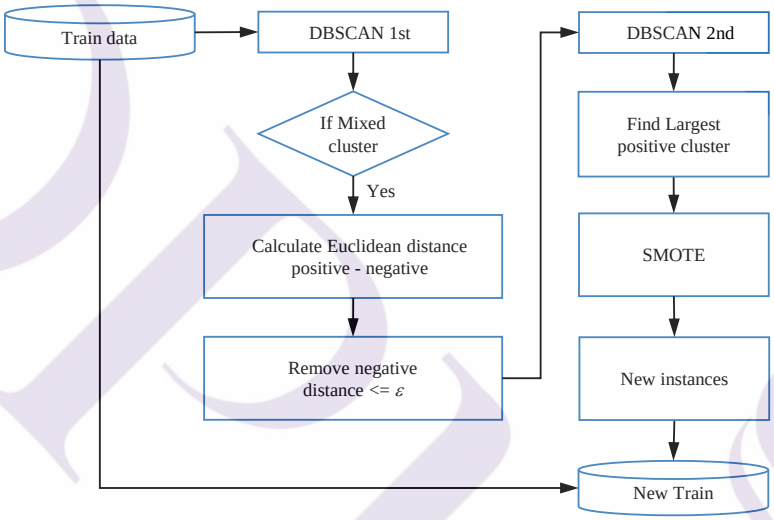


Fig. 1. The methodology

From these steps, the psudo-code was constructed as below.

Algorithm: DB2SM

Input: *T*: Train dataset contains positive & negative,
ε: Epsilon,
minpnts: Minimum Number of Point
LI: Lack of Information for the minority class (LI)
Output: *New_T*: New Balanced train data

```
1.  C = DBSCAN(T, ε, minpnts) // C: set of cluster
2.  for i = 1 to number of cluster in C do
3.      if Ci is mixed cluster then
4.          for j = 1 to number of positive in Ci do
5.              for k = 1 to number of negative in Ci do
6.                  djk = calEuclideanDistance(pj, nk) // pj: positive, nk: negative
7.                  if djk <= ε then
8.                      moveInstance(nk, delInstance)// delInstance: dataset
9.                  end if
```

```
                        end for
                    end for
                    appendInstance(D, Ci) // D:  empty dataset
                end if
                delCluster(C, Ci) // delete Ci from C
            end for
            E = DBSCAN(D, ε, minpnts) // E:  set of cluster
            C = C + E
            Lp = findLargestPositiveCluster(S)
            newInstance = SMOTE(Lp, LI) // newInstance:  dataset
            newInstance = delDuplicate(newInstance, delInstance)
            New_T = T + new_instance
            return(New_T )
```

Experiment and Result

Datasets

In the experiment, we tested with 20 imbalanced datasets from UCI Machine Learning Repository which characteristics were showed as in the table 1. Each dataset contains only two classes (positive, negative) and have no missing values. Imbalanced Ratio (*IR*) and Lack of Information (*LI*) for the minority class are indicators of the relationship between positive and negative in data which are calculated by these equations.

$$IR = \frac{number\ of\ negative}{number\ of\ positive} \quad (1)$$

$$LI = number\ of\ negative - number\ of\ positive \quad (2)$$

Table 1. Datasets characteristics

Dataset	Attribute	Example	IR	LI
ecoli1	8	336	0.77	182
ecoli2	8	336	5.46	232
ecoli3	8	336	8.60	266
glass0	10	214	2.06	74
glass1	10	214	1.82	62
glass6	10	214	6.38	156
haberman	4	306	2.78	254
iris0	5	150	2.00	50
new-thyroid1	6	215	5.14	145
new-thyroid2	6	215	5.14	145
page-blocks0	11	5,472	8.79	4,354
pima	9	768	1.87	232
segment0	20	2,308	6.02	1,650
vehicle0	19	846	3.25	448
vehicle1	19	846	2.90	412
vehicle2	19	846	2.88	410
vehicle3	19	846	2.99	422
wiscosin	10	683	1.86	205
yeast1	9	1,484	2.46	626
yeast3	9	1,484	8.10	1,158

Experimental Design and Evaluation

In the experiment, we used RapidMiner Studio version 9.8 as standard tool. All datasets were split into two parts as training data and test data by ratio 70:30. We used a simple Decision Tree as a classification

model and evaluated performance with three widely used resampling techniques which are SMOTE, DBCS, DBSM.

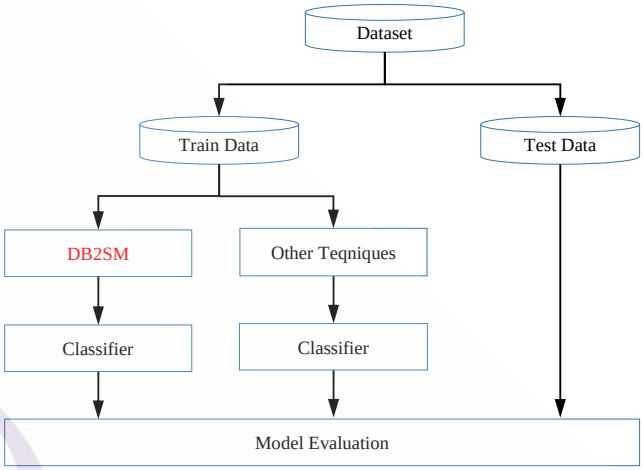


Fig. 2. Design of the experiment

Classification performances are evaluated using statistical measurement tools based on confusion matrix which are accuracy, AUC and F-measure.

Table 2. Confusion Matrix

	Pridicted as *positive*	**Pridicted as *negative***
Actually *positive*	TP	FN
Actually *negative*	FP	TN

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (3)$$

$$TP_{rate} = \frac{TP}{(TP+FN)} \quad (4)$$

$$FP_{rate} = \frac{FP}{(FP+TN)} \quad (5)$$

$$AUC = \frac{1+TP_{rate}+FP_{rate}}{2} \quad (6)$$

$$Precision = \frac{TP}{(TP+FP)} \quad (7)$$

$$Recall = \frac{TP}{(TP+FN)} \quad (8)$$

$$F-measure = \frac{2*Recall*Precision}{Recall+Precision} \quad (9)$$

Results

The results showed in Tables 3, 4 and 5. The best results displayed in bold.

Table 3. Accuracy result comparison

Dataset	Original	SMOTE	DBCS	DBSM	DB2SM
ecoli1	**0.9109**	0.8911	0.8812	0.8911	0.8317
ecoli2	0.9208	0.9307	0.9406	0.9307	**0.9703**
ecoli3	0.8812	0.8614	0.8911	0.7822	**0.9307**
glass0	**0.8438**	0.7500	0.8125	0.7812	**0.8438**
glass1	0.7188	0.6250	0.6875	0.5938	**0.7344**
glass6	0.9844	**1.0000**	0.9688	0.9844	**1.0000**
haberman	**0.7065**	0.6196	0.6522	0.6087	0.6739
iris0	1.0000	1.0000	1.0000	1.0000	1.0000
new-thyroid1	**0.9688**	0.9531	**0.9688**	0.9531	0.9531
new-thyroid2	0.9219	0.9531	0.9531	0.9531	**0.9688**
page-blocks0	0.9604	0.9683	0.9287	0.9562	**0.9714**
pima	0.6957	0.6696	0.5957	0.6696	**0.7000**
segment0	**0.9928**	0.9884	0.9855	0.9884	0.9913
vehicle0	0.9055	0.8031	0.7205	0.8031	**0.9252**
vehicle1	0.7323	0.3858	0.3937	0.3858	**0.7677**
vehicle2	0.8976	0.8858	0.9134	0.8858	**0.9409**
vehicle3	**0.7598**	0.3819	0.4213	0.3819	0.7047
wiscosin	0.9463	0.9512	**0.9659**	0.9610	0.9512
yeast1	0.7326	0.7348	0.4629	0.7348	**0.7551**
yeast3	**0.9618**	0.9528	0.9169	0.9483	**0.9618**
Average	0.8721	0.8153	0.8030	0.8097	**0.8788**

Table 4. AUC result comparison

Dataset	Original	SMOTE	DBCS	DBSM	DB2SM
ecoli1	0.8640	**0.9340**	0.8960	0.8590	0.8270
ecoli2	0.8410	0.8590	0.8910	0.9160	**0.9330**
ecoli3	0.7380	0.8130	0.8840	0.8250	**0.8510**
glass0	0.8390	0.8110	0.7950	0.7820	**0.8890**
glass1	0.6760	0.6320	0.6930	0.6540	**0.7180**
glass6	0.8890	**1.0000**	0.9640	0.9820	**1.0000**
haberman	0.5200	0.5520	**0.5980**	0.5470	0.5710
iris0	0.5000	0.5000	0.5000	0.5000	0.5000
new-thyroid1	0.9750	0.8450	**0.9810**	0.8450	0.8910
new-thyroid2	0.8440	0.9270	0.8850	0.9270	**0.9730**
page-blocks0	0.8230	**0.9460**	0.9330	0.9420	0.9140
pima	0.7040	**0.7460**	0.7360	**0.7460**	0.7340
segment0	**0.9720**	0.9690	0.9680	0.9690	0.9710
vehicle0	0.9170	0.8630	0.8050	0.8630	**0.9390**
vehicle1	0.5000	0.6050	0.5790	0.6050	**0.6370**
vehicle2	0.8870	0.9070	0.8850	0.9070	**0.9180**
vehicle3	**0.6530**	0.5960	0.6400	0.5960	0.6460
wiscosin	0.9280	0.9470	**0.9630**	0.9540	0.9380
yeast1	0.6950	**0.7610**	0.6180	**0.7610**	0.7030
yeast3	0.9050	0.9430	0.8830	**0.9450**	0.9250
Average	0.7835	0.8078	0.8049	0.8063	**0.8239**

Table 5. F-measure result comparison

Dataset	Original	SMOTE	DBCS	DBSM	DB2SM
ecoli1	**0.8000**	0.7843	0.7500	0.7660	0.7018
ecoli2	0.7778	0.8205	0.8500	0.8108	**0.9231**
ecoli3	0.2500	0.4615	**0.5600**	0.3889	0.3636
glass0	**0.7619**	0.6190	0.7273	0.6667	0.7368
glass1	0.5909	0.5556	0.6154	0.5517	**0.6667**
glass6	0.9412	**1.0000**	0.9000	0.9474	**1.0000**
haberman	0.4000	0.4068	0.4286	0.4000	**0.4444**
iris0	1.0000	1.0000	1.0000	1.0000	1.0000
new-thyroid1	0.9000	0.8421	**0.9091**	0.8421	0.8421
new-thyroid2	0.7619	0.8696	0.8696	0.8696	**0.9167**
page-blocks0	0.7735	0.8452	0.7221	0.8125	**0.8554**
pima	0.4068	0.6346	0.6235	0.6346	**0.6634**
segment0	**0.9701**	0.9529	0.9419	0.9529	0.9643
vehicle0	0.8065	0.7024	0.6502	0.7024	0.8652
vehicle1	**0.8110**	0.4730	0.4797	0.4730	0.3059
vehicle2	0.7937	0.7914	0.8333	0.7914	0.8855
vehicle3	0.2469	0.4530	**0.4806**	0.4530	0.4681
wiscosin	0.9209	0.9265	**0.9510**	0.9429	0.9254
yeast1	0.4848	0.6289	0.5286	**0.6312**	0.5856
yeast3	0.8172	0.7879	0.6783	0.7723	**0.8350**
Average	0.7108	0.7278	0.7250	0.7205	**0.7475**

Table 6. Parameters and Results

Dataset	ε	*minpts*	Accuracy	AUC	F-measure
ecoli1	0.50	16	0.8317	0.8275	0.7018
ecoli2	0.15	14	0.9703	0.9327	0.9231
ecoli3	0.10	1	0.9307	0.8508	0.3636
glass0	0.50	5	0.8438	0.8886	0.7368
glass1	0.50	3	0.7344	0.7179	0.6667
glass6	0.35	2	1.0000	1.0000	1.0000
harberman	0.10	2	0.6739	0.5711	0.4444
iris0	0.20	2	1.0000	0.5000	1.0000
new-thyroid1	0.10	11	0.9531	0.8907	0.8421
new-thyroid2	0.10	16	0.9688	0.9734	0.9167
page-blocks0	0.10	8	0.9714	0.9142	0.8554
pima	0.10	14	0.7000	0.7425	0.6634
segment0	0.85	5	0.9913	0.9711	0.9643
vehicle0	0.10	7	0.9252	0.9394	0.8652
vehicle1	0.10	2	0.7677	0.6374	0.3059
vehicle2	0.10	13	0.9409	0.9185	0.8855
vehicle3	0.11	11	0.7047	0.6458	0.4681
wiscosin	0.10	17	0.9512	0.9376	0.9254
yeast1	0.05	1	0.7551	0.7026	0.5856
yeast3	0.10	7	0.9618	0.9248	0.8350

Conclusion and Discussion

In this paper, we proposed DB2SM as one of the good choices to handle imbalanced data. Our method employs two rounds of DBSCAN to find the proper area of minority class and then applying SMOTE technique to generate new minority instances in that area. The experimental results show that AUC and F-measure for all resampling techniques are better than the original train data in various datasets. Moreover, the DB2SM won 10 cases compared to the baseline, SMOTE, DBCS, and DBSM.

Along the process, we changed parameters such as an e and minpnts to see how each dataset forming its appearances. The epsilon was set in the range 0.01-1.00 step by 0.01 from a normalized distance. The minpnts prefered from 1 to the number of positive instance in cluster. We found that changing the sensitive parameters affected performance. This might from the ambiguity between minority instances, noise, and outlier. However, some parameters can caused to generate a lower implicit instance. For example in Table 6, ecoli3 and yeast1 datasets used small epsilon and minpnts that allowed DBSCAN to create a tiny cluster and forced SMOTE to upsampling in this limited area. The result gained a good accuracy but F-measure was poor. It is possible that the model predicted the negative in high corrective rate but very low for positive.

Future Work

In the experiment, we used simple SMOTE as an oversampling method, but there are more variations of SMOTE including: SMOTE-NC, Borderline-SMOTE, SVM-SMOTE, ADASYN [10] that can increase the quality of result data. Furthermore, the experiment was settled on small datasets which consisted of tiny attributes then we can use the simply Euclidean DBSCAN [11] for gaining not much time by the average runtime complexity of DBSCAN is about O(nlog n). [12] In other scenarios such as a high dimensional data or more than two class problem, these will effect to the computational time then we must discover an additional clustering technique to resolve this kind of problem.

References

1. Longadge, R., Dongre, S., Malik, L.: Class Imbalance Problem in Data Mining: Review. In: International Journal of Computer Science and Network (IJCSN) vol. 2, Issue 1, February 2013 (2013).
2. Ali, H., Salleh, M., Saedudin, R., Hussain, K., Mushtaq, M.: Imbalance class problems in data mining: a review. In: Indonesian Journal of Electrical Engineering and Computer Science vol. 14, no. 3, June 2019 (2019).
3. Ester, M., Kriegel, H., Sander, J., Xu, X.: A Density-Based Algorithm for Discovering Clusters in Large Spatial Database with Noise. In: KDD, vol. 96, no. 34, pp. 226-231, August 1996 (1996).
4. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: Synthetic Minority Over-sampling Technique. In: Journal Of Artificial Intelligence Research, Volume 16, pages 321-357 (2002).
5. Sanguanmak, Y., Hanskunatai, A.: DBSM: The combination of DBSCAN and SMOTE for imbalanced data classification. In: 2016 13th International Joint Conference on Computer Science and Software Engineering, JCSSE 2016 (2016).
6. Verma, M.K., Xaxa, M.K., Verma, S.: DBCS: Density based cluster sampling for solving imbalanced classification problem. In: 2017 International conference of Electronics, Communication and Aerospace Technology (ICECA) (2017).
7. Xiaolong, X., Wen, C., Yenfei, S.: Over-sampling algorithm for imbalanced data classification. In: Journal of Systems Engineering and Electronics vol. 30, no. 6, December 2019, pp.1182– 1191 (2019).
8. Lv, M., Ren, Y., Chen, Y.: Research on imbalanced data : based on SMOTE-AdaBoost algorithm. In: The 3rd International Conference on Electronic Information Technology and Computer Engineering (EITCE), pp. 1165-1170 (2019).
9. Netirungroj, N., Pacharawongsakda, E.: TOP: An Efficient TwO-levels of Positive Resampling Framework for Class Imbalanced Data. In: 2018 IEEE International Conference on Big Data, Cloud Computing, Data Science & Engineering (BCD) (2018).
10. Wijaya, C.: 5 SMOTE Techniques for Oversampling your Imbalance Data. In: https://towardsdatascience.com/5-smote-techniques-for-oversampling-your-imbalance-data-b8155bdbe2b5, last accessed 2020/09/14.
11. Gan, J., Tao, Y.: On the Hardness and Approximation of Euclidean DBSCAN. In: ACM Trans Database syst. 42, 3, Article 14, July 2017 (2017).
12. Schubert, E., Sander, J., Ester, M., Kriegel, H., Xu, X.: DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN. In: ACM Trans. Database syst. 42, 3, Article 19, July 2017 (2017).

ประวัติผู้เขียน

ชื่อ–นามสกุล	ภาณุภณ จิระอัมพร
ประวัติการศึกษา	วิทยาศาสตร์บัณฑิต (คณิตศาสตร์) คณะวิทยาศาสตร์ มหาวิทยาลัยเกษตรศาสตร์ พ.ศ.2540
ตำแหน่งและสถานที่ทำงานปัจจุบัน	นักวิชาการศุลกากร กรมศุลกากร กระทรวงการคลัง