การจำแนกข้อมูลไม่สมดุลโดยใช้การปรับปรุงข้อมูลร่วมกับการหาค่าพารามิเตอร์ที่เหมาะสมด้วยขั้นตอนวิธีเชิงพันธุกรรมที่มีการเริ่มต้นใหม่

นายกีระชาติ สุขสุทธิ์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์
มหาวิทยาลัยเทคโนโลยีสุรนารี
ปีการศึกษา 2559

IMBALANCED DATA CLASSIFICATION USING DATA IMPROVEMENT AND PARAMETER OPTIMIZATION WITH RESTARTING GENETIC ALGORITHM

Keerachart Suksut

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in Computer Engineering

Suranaree University of Technology

Academic Year 2016

การจำแนกข้อมูลไม่สมดุลโดยใช้การปรับปรุงข้อมูลร่วมกับการหาค่าพารามิเตอร์ที่เหมาะสมด้วยขั้นตอนวิธีเชิงพันธุกรรมที่มีการเริ่มต้นใหม่

มหาวิทยาลัยเทคโนโลยีสุรนารี อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาดุษฎีบัณฑิต

คณะกรรมการสอบวิทยานิพนธ์

(รศ. ดร.นิตยา เกิดประสพ)
ประธานกรรมการ

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

(ผศ. ดร.ปรเมศวร์ ห่อแก้ว)
กรรมการ

(ผศ. ดร.สายสุนีย์ จับโจร)
กรรมการ

(ผศ. ดร.ศุภกฤษฎิ์ นิวัฒนากูล)
กรรมการ

(ศ. ดร.สันติ แม้นศิริ)
รักษาการแทนรองอธิการบดีฝ่ายวิชาการและพัฒนาความเป็นสากล

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

กีระชาติ สุขสุทธิ์ : การจำแนกข้อมูลไม่สมดุลโดยใช้การปรับปรุงข้อมูลร่วมกับการหาค่าพารามิเตอร์ที่เหมาะสมด้วยขั้นตอนวิธีเชิงพันธุกรรมที่มีการเริ่มต้นใหม่ (IMBALANCED DATA CLASSIFICATION USING DATA IMPROVEMENT AND PARAMETER OPTIMIZATION WITH RESTARTING GENETIC ALGORITHM) อาจารย์ที่ปรึกษา : รองศาสตราจารย์ ดร.กิตติศักดิ์ เกิดประสพ, 152 หน้า.

การทำเหมืองข้อมูล เป็นกระบวนการหาองค์ความรู้จากข้อมูลที่มีขนาดใหญ่ เพื่อค้นหารูปแบบ หรือหาความสัมพันธ์ของข้อมูลที่ซ่อนอยู่ภายในข้อมูลเหล่านั้น เพื่อที่จะนำความรู้ที่ได้รับนั้นมาสร้างโมเดลเพื่อใช้ในการทำนายข้อมูลในอนาคตหรือนำไปใช้จำแนกประเภทข้อมูลที่ยังไม่ทราบกลุ่ม แต่โดยทั่วไปแล้วอัลกอริทึมสำหรับการจำแนกประเภทข้อมูลจะมีประสิทธิภาพและมีความแม่นยำในการจำแนกประเภทข้อมูลสูงถ้าหากข้อมูลที่นำมาใช้ในการเรียนรู้มีจำนวนข้อมูลในแต่ละคลาสสมดุลกัน หรือใกล้เคียงกัน แต่เทคนิคในการจำแนกจะมีประสิทธิภาพในการจำแนกด้อยลงเมื่อข้อมูลที่นำมาใช้ในการเรียนรู้มีความไม่สมดุลเกิดขึ้น เนื่องจากอัลกอริทึมในการจำแนกอาจจะมีความเอนเอียงไปทางกลุ่มของข้อมูลที่มีจำนวนสมาชิกในคลาสนั้นมากกว่าอีกกลุ่ม

ดังนั้นงานวิจัยนี้จึงได้เสนอการปรับปรุงสมดุลข้อมูลร่วมกับการหาค่าพารามิเตอร์ที่เหมาะสมด้วยวิธีการเชิงพันธุกรรมที่มีการเริ่มต้นใหม่โดยได้นำ SMOTE Technique ร่วมกับวิธีการสุ่มลดข้อมูลมาใช้ในการปรับข้อมูลให้เกิดความสมดุล และใช้ขั้นตอนวิธีเชิงพันธุกรรมในการหาค่าพารามิเตอร์ที่เหมาะสมสำหรับสร้างโมเดลจำแนกประเภทข้อมูลด้วยอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีน และได้ใช้เทคนิคการเริ่มต้นขั้นตอนวิธีเชิงพันธุกรรมใหม่เพื่อปรับปรุงประสิทธิภาพในการค้นหาค่าพารามิเตอร์ที่เหมาะสม การทดสอบประสิทธิภาพของวิธีการที่เสนอขึ้นใหม่นี้ ใช้การทดสอบเปรียบเทียบประสิทธิภาพในการจำแนกประเภทข้อมูลด้วยค่าความแม่นยำในการจำแนก (Accuracy) ค่าความเที่ยง (Precision) ค่าความไว (Sensitivity) และค่าการวัดเอฟ (F-measure) โดยผลการทดสอบพบว่าเทคนิคที่นำเสนอมีประสิทธิภาพในการจำแนกข้อมูลจากคลาสส่วนน้อยได้ดีกว่าเทคนิคอื่น ๆ

สาขาวิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนักศึกษา กีระชาติ

ปีการศึกษา 2559 ลายมือชื่ออาจารย์ที่ปรึกษา

KEERACHART SUKSUT : IMBALANCED DATA CLASSIFICATION USING DATA IMPROVEMENT AND PARAMETER OPTIMIZATION WITH RESTARTING GENETIC ALGORITHM. THESIS ADVISOR : ASSOC. PROF. KITTISAK KERDPRASOP, Ph.D., 152 PP.

IMBALANCED DATA CLASSIFICATION/SUPPORT VECTOR MACHINE/ RESTARTING GENETIC ALGORITHM/DATA IMPROVEMENT

Data mining is the process to find knowledge from the huge amount of stored information and use the discovered knowledge to predict or classify the new data item that its class label is unknown. Classification algorithms are normally efficient on classifying balanced data but show poor performance with imbalanced data. Most learning algorithms cannot classify the data within minority class correctly because the model that is learned from imbalanced data tends to bias toward the majority class. In this research we thus propose a hybrid technique for data balancing with random under sampling the data from majority class and synthetically generate the new data from minority class with SMOTE technique, and also finding the optimal parameter with restarting genetic algorithm for model learning with support vector machine algorithm. We test the performance of the proposed method for minority data classification with four measurements: accuracy, precision, recall, and f-measure. The results show that the proposed technique has high performance on classifying the data from minority class and outperforms other techniques.

School of Computer Engineering

Academic Year 2016

Student's Signature _____

Advisor's Signature _____

กิตติกรรมประกาศ

วิทยานิพนธ์นี้สำเร็จลุล่วงด้วยดี ผู้วิจัยขอกราบขอบพระคุณ บุคคล และกลุ่มบุคคลต่าง ๆ ที่ได้กรุณาให้คำปรึกษา แนะนำ ช่วยเหลืออย่างดียิ่ง ทั้งในด้านวิชาการ และด้านการดำเนินงานวิจัย ดังต่อไปนี้

รองศาสตราจารย์ ดร.นิตยา เกิดประสพ และรองศาสตราจารย์ ดร.กิตติศักดิ์ เกิดประสพ ที่ให้คำปรึกษาในการทำงานวิจัย การจัดการรูปแบบ และช่วยตรวจทานความถูกต้องของวิทยานิพนธ์

คุณสายฝน สิบพลกรัง เลขานุการสาขาวิชาวิศวกรรมเครื่องกล และคุณ วิไลลักษณ์ คัมภิรานนท์ เลขานุการสาขาวิชาวิศวกรรมคอมพิวเตอร์ ที่ให้ความช่วยเหลือในการประสานงานด้านเอกสารระหว่างศึกษา

ขอขอบคุณนักศึกษาบัณฑิตสาขาวิชาวิศวกรรมคอมพิวเตอร์ ทุกท่านที่ให้คำปรึกษา ช่วยตรวจทานความถูกต้องและช่วยเหลือด้วยดีมาโดยตลอด

นอกจากนี้ขอขอบคุณครู อาจารย์ทั้งในอดีตและปัจจุบันที่ให้ความรู้แก่ผู้วิจัยจนประสบความสำเร็จในชีวิต

ท้ายที่สุดที่จะลืมไม่ได้ ขอกราบขอบพระคุณ บิดา มารดา ที่ให้กำเนิด อบรม เลี้ยงดู ด้วยความรัก และส่งเสริมการศึกษาเป็นอย่างดีโดยตลอด ทำให้ผู้วิจัยมีความรู้ ความสามารถ มีจิตใจที่เข้มแข็ง รวมทั้งเป็นกำลังใจที่ยิ่งใหญ่แก่ผู้วิจัย จนทำให้ผู้วิจัยประสบความสำเร็จในชีวิตเรื่อยมา

กีระชาติ สุขสุทธิ์

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

ตารางที่ หน้า

สารบัญตาราง (ต่อ)

ตารางที่ หน้า

สารบัญตาราง (ต่อ)

ตารางที่ หน้า

สารบัญตาราง (ต่อ)

ตารางที่ หน้า

สารบัญรูป

รูปที่ หน้า

สารบัญรูป (ต่อ)

รูปที่ หน้า

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มาของปัญหาการวิจัย

การทำเหมืองข้อมูล (Han et al., 2011) เป็นกระบวนการหาองค์ความรู้จากข้อมูลที่มีขนาดใหญ่ เพื่อหารูปแบบ หรือหาความสัมพันธ์ของข้อมูลที่ซ่อนอยู่ภายในข้อมูลเหล่านั้นด้วยวิธีการทางคณิตศาสตร์ สถิติ หรือคอมพิวเตอร์ การทำเหมืองข้อมูลมีหลายประเภทโดยจะขึ้นอยู่กับวัตถุประสงค์ที่จะนำไปใช้งาน เช่น การจำแนกประเภทข้อมูล (Data Classification) การหากฎความสัมพันธ์ของข้อมูล (Association Rule) และการจัดกลุ่มข้อมูล (Clustering) เป็นต้น ปัจจุบันมีการนำเทคนิคการทำเหมืองข้อมูลไปประยุกต์ใช้งานอย่างกว้างขวาง ไม่ว่าจะเป็นการนำไปประยุกต์ใช้งานทางด้านอุตสาหกรรมที่ประยุกต์ใช้การจำแนกประเภทข้อมูลเข้ามาแก้ปัญหาด้วยการนำโมเดล หรือกฎที่ได้จากการจำแนกประเภทข้อมูลไปทำการจำแนกข้อมูลที่ยังไม่ทราบประเภท โดยเทคนิคที่นิยมนำมาใช้ในการจำแนกประเภทข้อมูลมีหลายเทคนิค เช่น การใช้โครงข่ายประสาทเทียม (Artificial Neural Network : ANN) ซึ่งมีแนวคิดพื้นฐานมาจากการจำลองการทำงานของสมองมนุษย์ด้วยการทำให้คอมพิวเตอร์สามารถเรียนรู้ได้เหมือนกับที่มนุษย์เรียนรู้ หรือการใช้การหากฎความสัมพันธ์ของข้อมูลมาจำแนกข้อมูลแต่ละประเภทออกจากกันโดยอาศัยรูปแบบของกฎการเรียนรู้เป็นตัวจำแนก หรือการใช้ต้นไม้ตัดสินใจ (Decision Tree) ซึ่งเป็นเทคนิคในการจำแนกประเภทข้อมูลให้อยู่ในลักษณะคล้ายต้นไม้ โดยมีโหนดราก (Root Node) อยู่บนสุดและมีโหนดใบ (Leaf Node) อยู่ล่างสุด โดยในแต่ละโหนดจะหมายถึงแอตทริบิวต์ (Attribute) ที่นำมาใช้ในการจำแนก และค่าทั้งหมดที่เป็นไปได้จะอยู่ที่โหนดใบ หรือการใช้เทคนิคนาอีฟเบย์ (Naïve Bayes) ซึ่งเป็นการจำแนกประเภทข้อมูลโดยใช้ค่าความน่าจะเป็นของข้อมูลฝึกสอนมาเป็นเกณฑ์ในการตัดสินใจจำแนกข้อมูลที่ไม่ทราบประเภท

เทคนิคเหล่านี้จะมีประสิทธิภาพและความแม่นยำในการจำแนก (Accuracy) ที่สูงเมื่อนำไปใช้ในการจำแนกข้อมูลที่มีความสมดุลกัน (Balanced Data) แต่เทคนิคเหล่านี้จะมีประสิทธิภาพลดลงเมื่อข้อมูลที่ใช้ในการฝึกสอนไม่สมดุลกัน (Imbalanced Data) (Chawla, 2005) โดยข้อมูลไม่สมดุลคือข้อมูลที่มีจำนวนข้อมูลในแต่ละคลาสมีขนาดไม่เท่ากัน โดยมีข้อมูลในคลาสใดคลาสหนึ่งมีจำนวนมากกว่าจำนวนข้อมูลของคลาสอื่น ๆ เป็นจำนวนมาก ๆ เช่น ข้อมูลการทำธุรกรรมผ่านบัตรเครดิต ที่จะมีจำนวนลูกค้าที่ผิดปกติน้อยกว่าจำนวนข้อมูลลูกค้าที่ปกติ หรือข้อมูลการตรวจจับผู้บุกรุกเครือข่ายคอมพิวเตอร์ที่มีจำนวนผู้ใช้งานปกติมากกว่าจำนวนผู้บุกรุก หรือข้อมูลการวินิจฉัยทางการแพทย์ที่มีจำนวนผู้ป่วยโรคร้ายแรงน้อยกว่าจำนวนผู้ที่มีสุขภาพดี เป็นต้น

ข้อมูลไม่สมดุลนั้นจะมีจำนวนข้อมูลในแต่ละคลาสเป้าหมายที่แตกต่างกันมาก ๆ (Chawla et al., 2002) เช่น มีจำนวนข้อมูลทั้งหมด 1,000 ข้อมูล แบ่งเป็นข้อมูลในคลาส A จำนวน 960 ข้อมูล และข้อมูลในคลาส B จำนวน 40 ข้อมูล จะเห็นได้ว่าจำนวนข้อมูลในคลาส A มีจำนวนมากกว่าจำนวนข้อมูลในคลาส B เป็นจำนวนมาก เราเรียกข้อมูลในคลาส A ว่า คลาสส่วนมาก (Majority Class) และเรียกข้อมูลในคลาส B ว่า คลาสส่วนน้อย (Minority Class) โดยปกติแล้วข้อมูลไม่สมดุลจะมีจำนวนข้อมูลในคลาสส่วนน้อยเป็นจำนวน 0.1 % ถึง 10% ของข้อมูลทั้งหมด เมื่อนำข้อมูลที่มีความไม่สมดุลไปทำการจำแนกประเภทข้อมูลด้วยอัลกอริทึมแบบมาตรฐานสำหรับการจำแนกประเภทข้อมูล จะส่งผลให้การจำแนกมีความเอนเอียง (Bias) ไปทางกลุ่มข้อมูลที่มีจำนวนข้อมูลมากกว่า ซึ่งจะส่งผลให้กลุ่มข้อมูลที่อยู่ในคลาสส่วนน้อยเกิดการจัดกลุ่มผิดประเภท (Misclassification) เกิดขึ้น โดยอาจส่งผลให้อัลกอริทึมมาตรฐานสำหรับการจำแนกประเภทไม่สามารถจำแนกประเภทข้อมูลที่อยู่ในคลาสส่วนน้อยได้

จากปัญหาการจำแนกประเภทข้อมูลไม่สมดุลที่กล่าวมาข้างต้น มีเทคนิคและวิธีการต่าง ๆ เพื่อนำมาใช้ในการแก้ปัญหาข้อมูลไม่สมดุล โดยให้ความสำคัญกับข้อมูลที่อยู่ในคลาสส่วนน้อยให้มีการจำแนกประเภทข้อมูลได้แม่นยำมากยิ่งขึ้น เช่นการปรับจำนวนข้อมูลให้สมดุลด้วยการสุ่มข้อมูลซ้ำ (Resampling) ซึ่งมีทั้งการสุ่มข้อมูลเพิ่ม และการสุ่มลดข้อมูลลง ดังในงานวิจัยของ Estabrooks and Japkowicz (2001) หรือแก้ปัญหาข้อมูลไม่สมดุลโดยการแก้ปัญหาในขั้นตอนของการประมวลผลโดยทำการปรับปรุงพารามิเตอร์ที่นำมาใช้สำหรับอัลกอริทึมในการจำแนกประเภทข้อมูล ดังงานวิจัยของ Yu et al. (2015) หรือการประยุกต์ใช้ค่าใช้จ่ายในการจำแนกข้อมูลผิดประเภทเข้ามาเพิ่มน้ำหนักให้ข้อมูลที่มีการจำแนกผิดประเภท ดังงานวิจัยของ Japkowicz and Stephen (2002) หรือการประยุกต์ใช้วิธีการจำแนกประเภทข้อมูลด้วยการใช้โมเดลในการจำแนกมากกว่าหนึ่งตัวหรือที่เรียกว่าเทคนิคการเรียนรู้ร่วมกัน (Ensemble Method) ดังงานวิจัยของ Estabrooks et al. (2004) เข้ามาช่วยในการตัดสินใจแทนการจำแนกประเภทข้อมูลด้วยโมเดลเดียวเพื่อแก้ปัญหาในการจำแนกของข้อมูลที่ใช้ในการเรียนรู้ที่คงที่ ที่อาจจะส่งผลให้เกิดความเอนเอียงในการจำแนกได้ การแก้ปัญหาข้อมูลไม่สมดุลได้รับการนำไปประยุกต์ใช้งานในด้านต่าง ๆ เช่น สังคมศาสตร์ (Bressoux, 2008) การตรวจจับการทุจริตทางเครดิตการ์ด (Shen et al., 2007) การชำระภาษี (Miquel, 2009) กลยุทธ์รักษากลุ่มลูกค้า (Lariviere and Van, 2005) ทำนายการยกเลิกบริการของลูกค้า (Bekkar, 2009) การแบ่งส่วน (Schroff et al., 2008) การวินิจฉัยโรคด้วยรูปภาพ (Bosch et al., 2007) เป็นต้น

ดังนั้นงานวิจัยนี้ผู้วิจัยจึงเสนอเทคนิคในการเพิ่มประสิทธิภาพในการจำแนกข้อมูลไม่สมดุลด้วยการปรับปรุงข้อมูลให้มีความสมดุลขึ้นด้วยการใช้ SMOTE Technique เพื่อสุ่มสร้างข้อมูลคลาสส่วนน้อยใหม่ให้มีความหลากหลายมากขึ้นและใช้เทคนิคการสุ่มลดเพื่อสุ่มลดข้อมูลจากคลาสส่วนมากลง ร่วมกับการหาค่าพารามิเตอร์ที่เหมาะสมสำหรับการจำแนกข้อมูลไม่สมดุลด้วยซัพพอร์ตเวกเตอร์แมชชีนโดยการใช้ขั้นตอนวิธีเชิงพันธุกรรมที่มีการเริ่มต้นใหม่ (Restarting Genetic Algorithm) เพื่อหาค่าพารามิเตอร์ที่เหมาะสมที่สุด และทดสอบเปรียบเทียบประสิทธิภาพ

ในการจำแนกข้อมูลด้วยค่าความแม่นยำในการจำแนก (Accuracy) ค่าความเที่ยง (Precision) ค่าความไว (Sensitivity) และค่าการวัดเอฟ (F-measure)

1.2 วัตถุประสงค์ของการวิจัย

1. เพื่อศึกษาและพัฒนาขั้นตอนวิธีการจำแนกข้อมูลไม่สมดุลให้มีความสามารถในการจำแนกข้อมูลในคลาสส่วนน้อยได้ดียิ่งขึ้น

2. เพื่อเปรียบเทียบประสิทธิภาพในการจำแนกข้อมูลไม่สมดุลระหว่างเทคนิคที่พัฒนาขึ้นกับเทคนิคในการจำแนกข้อมูลไม่สมดุลที่ใช้ในปัจจุบัน

3. เพื่อหาค่าพารามิเตอร์ที่เหมาะสม ที่ดีที่สุดสำหรับนำไปใช้ในการจำแนกข้อมูลไม่สมดุล

1.3 ขอบเขตของการวิจัย

1. ข้อมูลที่นำมาใช้จะต้องเป็นข้อมูลตัวเลขเท่านั้น (ยกเว้นแอตทริบิวต์คลาสที่สามารถเป็นตัวอักษรได้)

2. ข้อมูลที่นำมาใช้จะต้องมีคลาสเป้าหมาย เพียง 2 คลาสเท่านั้น

3. การเปรียบเทียบประสิทธิภาพจะใช้เกณฑ์ความถูกต้องหรือความแม่นยำในการจำแนกข้อมูล ค่าความเที่ยง ค่าความไว และค่าการวัดเอฟ

4. ข้อมูลที่นำมาใช้ในงานวิทยานิพนธ์

ข้อมูลสังเคราะห์ขึ้นจำนวน 1 ชุดข้อมูล

ข้อมูลโรคหอบหืดจำนวน 1 ชุดข้อมูล (พงศกร, 2015)

ข้อมูลโรคหัวใจ 1 ชุดข้อมูล

(http://archive.ics.uci.edu/ml/datasets/heart+Disease)

ข้อมูลโรคตับ 1 ชุดข้อมูล

(https://archive.ics.uci.edu/ml/datasets/ILPD+(Indian+Liver+Patient+Dataset))

1.4 ประโยชน์ที่จะได้รับ

จากการศึกษาและพัฒนางานวิจัยนี้ ผู้วิจัยคาดว่าเทคนิคที่พัฒนาขึ้นจะเกิดประโยชน์ต่อผู้ใช้ในการนำไปจำแนกข้อมูลไม่สมดุลให้มีประสิทธิภาพมากยิ่งขึ้น และยังช่วยหาค่าพารามิเตอร์ที่เหมาะสมสำหรับนำไปใช้ในการสร้างโมเดลการจำแนกสำหรับซัพพอร์ตเวกเตอร์แมชชีนได้

บทที่ 2
ปริทัศน์วรรณกรรม

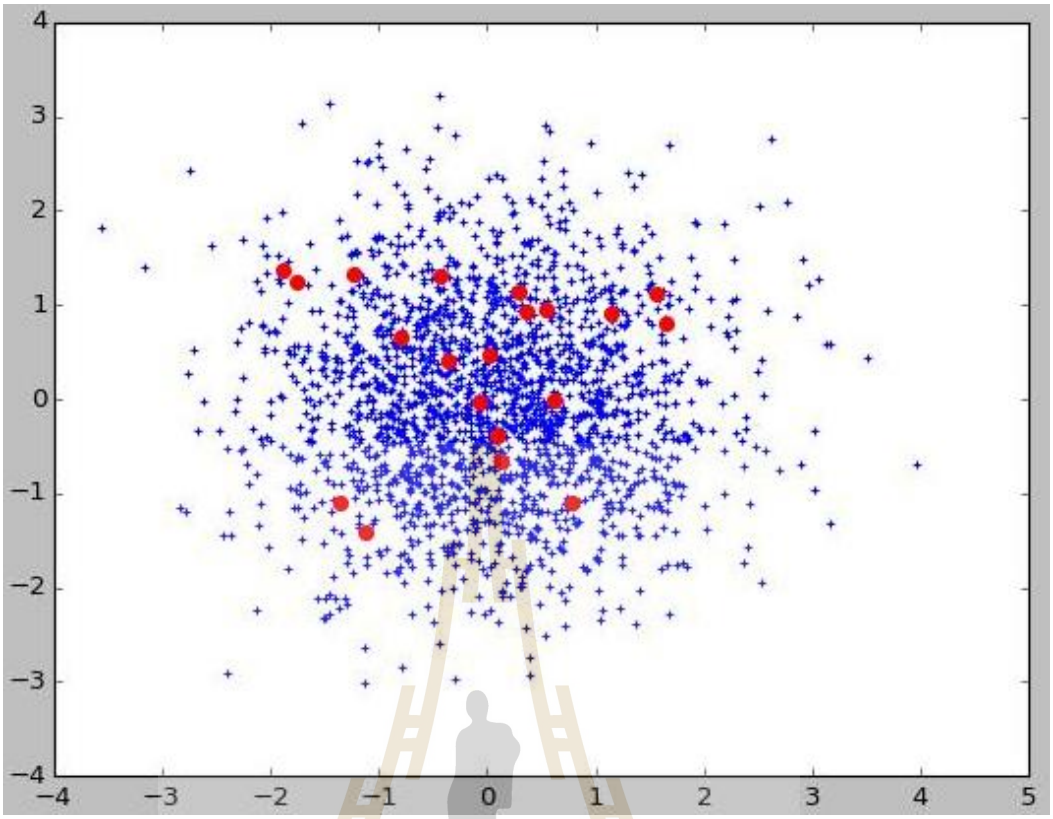
เนื้อหาในบทนี้ประกอบด้วยการทบทวนวรรณกรรมและงานวิจัยที่เกี่ยวข้อง ซึ่งประกอบไปด้วยรายละเอียดข้อมูลไม่สมดุล (Imbalanced Data) เกณฑ์สำหรับการประเมินประสิทธิภาพการจำแนกข้อมูลไม่สมดุลขั้นตอนวิธีเชิงพันธุกรรม (Genetic Algorithm) การจำแนกประเภทข้อมูลด้วยซัพพอร์ตเวกเตอร์แมชชีน และงานวิจัยที่เกี่ยวข้อง

2.1 ข้อมูลไม่สมดุล

ข้อมูลไม่สมดุล หมายถึง ข้อมูลที่มีจำนวนข้อมูลในกลุ่มหนึ่งมีจำนวนมากกว่าข้อมูลในอีกกลุ่มหนึ่งเป็นจำนวนมาก (Chawla et al., 2002; Chawla et al., 2004) ข้อมูลไม่สมดุลนั้นมีสาเหตุมาจากหลายปัจจัย เช่น ข้อมูลไม่สมดุลที่เกิดจากลักษณะทางธรรมชาติของข้อมูลเองดังที่พบในข้อมูลทางการแพทย์ที่พบว่ามีผู้ป่วยที่ป่วยเป็นโรคร้ายแรงน้อยกว่าผู้ที่มีสุขภาพแข็งแรงเป็นจำนวนมาก หรือข้อมูลที่เกิดจากข้อจำกัดในการเก็บข้อมูลเนื่องจากมีค่าใช้จ่ายในการเก็บข้อมูลที่สูงมาก หรือข้อมูลผู้ใช้บัตรเครดิตที่มีข้อมูลลูกค้าที่ใช้จ่ายปกติมากกว่าลูกค้าที่ใช้จ่ายผิดปกติ เป็นต้น นอกจากนี้ข้อมูลไม่สมดุลอาจเกิดจากการเก็บข้อมูลที่ผิดพลาดด้วยเช่นกัน

ตัวอย่างของข้อมูลไม่สมดุล เช่น ข้อมูลในคลาสที่หนึ่ง มีจำนวน 2,000 ข้อมูล ส่วนข้อมูลในคลาสที่สองมีจำนวน 20 ข้อมูล เป็นต้น จะเห็นได้ว่าจำนวนข้อมูลในคลาสที่หนึ่งมีมากกว่าจำนวนข้อมูลในคลาสที่สองเป็นจำนวนมาก เราเรียกข้อมูลชุดนี้ว่า “ข้อมูลไม่สมดุล”แสดงดังรูปที่ 2.1 (ข้อมูลคลาสส่วนมากแทนด้วยจุดเครื่องหมายบวก +, ข้อมูลคลาสส่วนน้อยแทนด้วยเครื่องหมายจุดวงกลม •)โดยทั่วไปเราจะเรียกกลุ่มข้อมูลที่อยู่ในคลาสที่มีจำนวนข้อมูลมากกว่าว่าคลาสส่วนมาก (Majority Class) และเรียกกลุ่มข้อมูลที่อยู่ในคลาสที่มีจำนวนข้อมูลน้อยกว่าว่าคลาสส่วนน้อย (Minority Class) (Boonchuay et al., 2011; Farquad and Bose, 2012; Gao et al., 2012) เนื่องจากจำนวนข้อมูลในแต่ละคลาสมีความแตกต่างกันเป็นจำนวนมาก จึงส่งผลกระทบให้การจำแนกคลาสส่วนน้อยมีความแม่นยำในการจำแนกลดน้อยลงไป เนื่องจากผลกระทบจากข้อมูลในคลาสส่วนมากในขั้นตอนการสร้างโมเดลการเรียนรู้ (Training Model)

ข้อมูลไม่สมดุลส่งผลกระทบต่อการจำแนกคลาสส่วนน้อย เนื่องจากอัลกอริทึมทั่วไปจะทำงานได้อย่างมีประสิทธิภาพสูงสุดก็ต่อเมื่อจำนวนข้อมูลในแต่ละคลาสมีจำนวนใกล้เคียงกัน (ข้อมูลมีความสมดุล) แต่เมื่อข้อมูลมีความไม่สมดุลเกิดขึ้น อัลกอริทึมทั่วไปจะมีความเอนเอียงไปทางกลุ่มข้อมูลที่มีจำนวนมากกว่าอีกกลุ่มหนึ่ง (เอนเอียงไปทางคลาสส่วนมาก) จึงส่งผลให้การจำแนกคลาสส่วนน้อยมีความผิดพลาด



รูปที่ 2.1 ตัวอย่างข้อมูลไม่สมดุล

ระดับของความไม่สมดุล (Imbalanced Degree)

การจำแนกว่าข้อมูลใดมีความสมดุลหรือเป็นข้อมูลไม่สมดุลจากระดับของความไม่สมดุล โดยระดับของความไม่สมดุลจะแสดงอัตราส่วนระหว่างจำนวนของข้อมูลในคลาสส่วนมากเทียบกับจำนวนข้อมูลในคลาสส่วนน้อย (Orriols-Puig et al., 2009; Villar et al., 2011) หากข้อมูลมีระดับของความไม่สมดุลสูง (ระดับความไม่สมดุลมากกว่า 1 มาก ๆ) นั่นหมายความว่าข้อมูลมีความไม่สมดุลสูง หากข้อมูลมีระดับของความไม่สมดุลเท่ากับ 1 นั่นหมายความว่าจำนวนข้อมูลในแต่ละคลาสมีจำนวนเท่ากัน หากข้อมูลมีระดับของความไม่สมดุลต่ำกว่า 1 นั่นหมายความว่าจำนวนข้อมูลของคลาสส่วนน้อยมีจำนวนมากกว่าจำนวนข้อมูลในคลาสส่วนมาก โดยการคำนวณหาระดับของความไม่สมดุลสามารถคำนวณได้จากสมการที่ 2.1

$$Imbalance\ Ratio\ (IR) = \frac{n_{majority}}{n_{minority}} \tag{2.1}$$

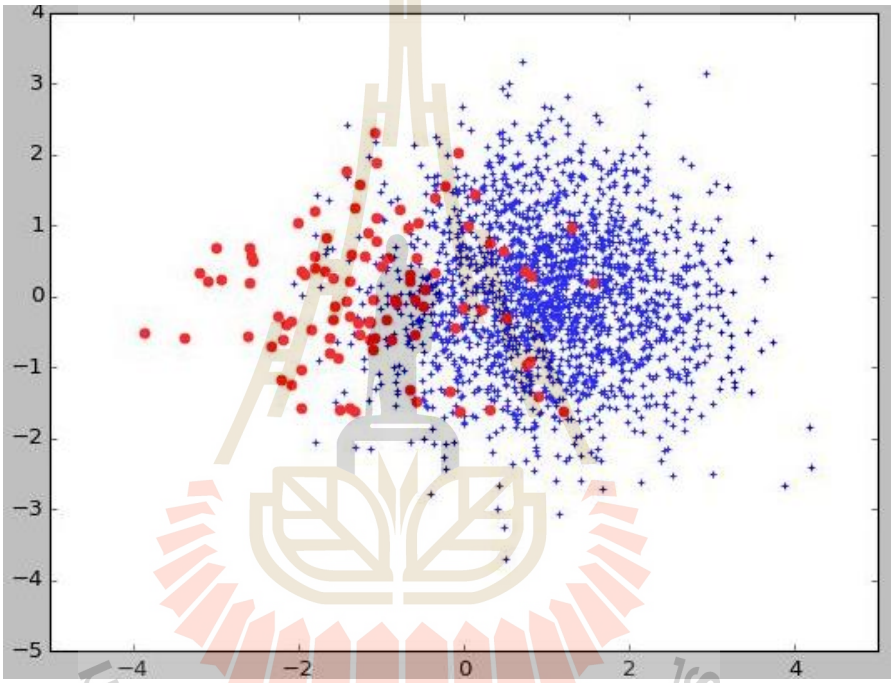
โดยที่ $n_{magority}$ คือจำนวนข้อมูลในคลาสส่วนมาก

$n_{minority}$ คือจำนวนข้อมูลในคลาสส่วนน้อย

ตัวอย่างที่ 1 สมมติว่ามีชุดข้อมูล B ซึ่งมีทั้งหมด 2 คลาสได้แก่คลาส True และคลาส False โดยมีจำนวนข้อมูลทั้งหมดในคลาส True จำนวน 100 ข้อมูล และมีจำนวนข้อมูลทั้งหมดในคลาส False จำนวน 1,900 ข้อมูลแสดงดังตารางที่ 2.1 และรูปที่ 2.2 จะพบว่าข้อมูลชุดนี้มีระดับของความไม่สมดุลที่ 19.0

ตารางที่ 2.1 จำนวนข้อมูลในแต่ละคลาสของชุดข้อมูล B

ชุดข้อมูล	จำนวนข้อมูลในคลาส True	จำนวนข้อมูลในคลาส False
ชุดข้อมูล B	100	1,900



รูปที่ 2.2 จำนวนข้อมูลในแต่ละคลาสของชุดข้อมูล B

แทนค่าตัวแปร $n_{majority}$ ด้วย 1,900 เนื่องจากเป็นข้อมูลคลาสส่วนมาก
แทนค่าตัวแปร $n_{minority}$ ด้วย 100 เนื่องจากเป็นข้อมูลคลาสส่วนน้อย
Imbalanced Degree = 1,900 / 100 = 19.0

นั่นหมายความว่าชุดข้อมูล B มีระดับของความไม่สมดุลเท่ากับ 19.0

การจัดการกับข้อมูลไม่สมดุล

การแก้ปัญหาในระดับข้อมูลเป็นการแก้ไขปัญหาในขั้นตอนก่อนที่จะมีการประมวลผล (Processing Stage) โดยการแก้ไขในระดับนี้จะแก้ไขกับข้อมูลโดยตรง โดยจะทำการปรับปรุงข้อมูล

ที่มีความไม่สมดุลให้กลายเป็นข้อมูลที่มีความสมดุลด้วยเทคนิคการสุ่มเลือกข้อมูล (Data Sampling Technique) หรือเทคนิคการเลือกข้อมูล (Data Selection Technique) (Solberg and Solberg, 1996; Kubat and Matwin, 1997; Ling and Li, 1998; Japkowicz, 2000b; Laurikkala, 2001; Chawla et al., 2002; Weiss and Provost, 2003; Batista et al., 2004; Jo and Japkowicz, 2004; Phua and Alahakoon, 2004) โดยจะแบ่งออกเป็น 3 กลุ่ม ได้แก่

วิธีสุ่มเกิน (Over Sampling)

วิธีสุ่มเกิน (กิตติพงษ์, 2016) เป็นเทคนิค หรือวิธีที่ใช้ในการเพิ่มข้อมูลที่อยู่ในคลาสส่วนน้อย ให้มีจำนวนใกล้เคียง หรือเท่ากับจำนวนข้อมูลในคลาสส่วนมาก โดยการสุ่มเกินเพื่อเพิ่มข้อมูลให้คลาสส่วนน้อยจะเพิ่มข้อมูลโดยการสุ่มเลือกข้อมูลจากข้อมูลเดิม หรือสร้างข้อมูลขึ้นมาใหม่จากตัวอย่างของข้อมูลเดิม

สมมติว่ามีชุดข้อมูล C ซึ่งมีจำนวนคลาสทั้งหมด 2 คลาส ได้แก่คลาส True และคลาส False โดยข้อมูลในคลาส True มีจำนวนทั้งหมด 4 ข้อมูล และข้อมูลในคลาส False มีจำนวนทั้งหมด 10 ข้อมูล แสดงดังตารางที่ 2.2

ตารางที่ 2.2 ชุดข้อมูล C ข้อมูลผู้ป่วยเป็นเนื้อร้าย (คลาส True หมายถึงเนื้อร้าย คลาส False หมายถึงไม่ใช่เนื้อร้าย)

ลำดับข้อมูล	ความกว้าง	ความยาว	น้ำหนักเนื้องอก	ระดับความเข้ม	คลาส
1	1.4”	2.5”	2.0 g	0.9	True
2	0.4”	1.1”	0.1 g	0.1	False
3	0.3”	0.9”	0.2 g	0.2	False
4	1.5”	2.0”	2.1 g	0.8	True
5	0.2”	0.9”	0.2 g	0.1	False
6	0.1”	0.4”	0.3 g	0.1	False
7	1.7”	2.3”	2.2 g	0.9	True
8	1.8”	2.3”	2.5 g	0.7	True
9	0.4”	0.6”	0.3 g	0.2	False
10	0.2”	0.7”	0.2 g	0.3	False
11	0.2”	0.3”	0.5 g	0.1	False
12	0.1”	0.2”	0.3 g	0.2	False
13	0.5”	0.1”	1.8 g	0.6	False
14	0.1”	0.5”	0.7 g	0.3	False

ตารางที่ 2.3 ชุดข้อมูล C หลังจากใช้เทคนิคการสุ่มเลือกข้อมูลจากคลาสส่วนน้อยเพิ่ม 4 ข้อมูล

ลำดับข้อมูล	ความกว้าง	ความยาว	น้ำหนักเนื้องอก	ระดับความเข้ม	คลาส
1	1.4”	2.5”	2.0 g	0.9	True
2	0.4”	1.1”	0.1 g	0.1	False
3	0.3”	0.9”	0.2 g	0.2	False
4	1.5”	2.0”	2.1 g	0.8	True
5	0.2”	0.9”	0.2 g	0.1	False
6	0.1”	0.4”	0.3 g	0.1	False
7	1.7”	2.3”	2.2 g	0.9	True
8	1.8”	2.3”	2.5 g	0.7	True
9	0.4”	0.6”	0.3 g	0.2	False
10	0.2”	0.7”	0.2 g	0.3	False
11	0.2”	0.3”	0.5 g	0.1	False
12	0.1”	0.2”	0.3 g	0.2	False
13	0.5”	0.1”	1.8 g	0.6	False
14	0.1”	0.5”	0.7 g	0.3	False
15	**1.4”**	**2.5”**	**2.0 g**	**0.9**	**True**
16	**1.5”**	**2.0”**	**2.1 g**	**0.8**	**True**
17	**1.7”**	**2.3”**	**2.2 g**	**0.9**	**True**
18	**1.8”**	**2.3”**	**2.5 g**	**0.7**	**True**

ตัวอย่างที่ 3 จากเทคนิคการสุ่มเกินหากเลือกใช้เทคนิคการสร้างข้อมูลใหม่จากข้อมูลเดิม เทคนิคนี้จะทำการเลือกข้อมูลที่อยู่ในคลาส False แล้วทำการเพิ่มค่าขึ้นเล็กน้อยหรือลดค่าลงเล็กน้อย โดยเพิ่มข้อมูลเป็นจำนวน 4 – 6 ข้อมูล เพื่อทำให้ชุดข้อมูล C จากตารางที่ 2.2 มีความสมดุลเกิดขึ้นสมมติว่าทำการสร้างข้อมูลจากคลาสส่วนน้อยเพิ่มขึ้นมา 4 ข้อมูล จะได้ชุดข้อมูลใหม่โดยมีข้อมูลทั้งหมด 18 ข้อมูล แบ่งเป็นข้อมูลในคลาส True จำนวน 8 ข้อมูล และข้อมูลในคลาส False จำนวน 10 ข้อมูล แสดงดังตารางที่ 2.4

ตารางที่ 2.4 ชุดข้อมูล C หลังจากใช้เทคนิคการสร้างข้อมูลใหม่จากคลาสส่วนน้อยเพิ่ม 4 ข้อมูล

ลำดับข้อมูล	ความกว้าง	ความยาว	น้ำหนักเนื้องอก	ระดับความเข้ม	คลาส
1	1.4”	2.5”	2.0 g	0.9	True
2	0.4”	1.1”	0.1 g	0.1	False

ตารางที่ 2.4 ชุดข้อมูล C หลังจากใช้เทคนิคการสร้างข้อมูลใหม่จากคลาสส่วนน้อยเพิ่ม 4 ข้อมูล (ต่อ)

ลำดับข้อมูล	ความกว้าง	ความยาว	น้ำหนักเนื้องอก	ระดับความเข้ม	คลาส
3	0.3”	0.9”	0.2 g	0.2	False
4	1.5”	2.0”	2.1 g	0.8	True
5	0.2”	0.9”	0.2 g	0.1	False
6	0.1”	0.4”	0.3 g	0.1	False
7	1.7”	2.3”	2.2 g	0.9	True
8	1.8”	2.3”	2.5 g	0.7	True
9	0.4”	0.6”	0.3 g	0.2	False
10	0.2”	0.7”	0.2 g	0.3	False
11	0.2”	0.3”	0.5 g	0.1	False
12	0.1”	0.2”	0.3 g	0.2	False
13	0.5”	0.1”	1.8 g	0.6	False
14	0.1”	0.5”	0.7 g	0.3	False
15	**1.45”**	**2.55”**	**2.05 g**	**0.95**	**True**
16	**1.45”**	**1.95”**	**2.05 g**	**0.75**	**True**
17	**1.75”**	**2.35”**	**2.25 g**	**0.95**	**True**
18	**1.75”**	**2.25”**	**2.45 g**	**0.65**	**True**

การสุ่มเกินด้วย SMOTE Technique (Chawla et al., 2002; Chawla, 2003, Chawla et al., 2004, Han et al., 2005) จะทำการสุ่มสร้างข้อมูลจากคลาสส่วนน้อยตามจำนวนที่กำหนด โดยจะสร้างข้อมูลสังเคราะห์จากข้อมูลตัวอย่างด้วยการวัดระยะห่างจากจุดข้อมูลตัวอย่างไปยังจุดข้อมูลใกล้เคียง แล้วสุ่มสร้างข้อมูลสังเคราะห์ขึ้นโดยข้อมูลสังเคราะห์ที่สร้างขึ้นจะอยู่ภายในระยะห่างจากจุดข้อมูลตัวอย่างไปยังจุดข้อมูลเพื่อนบ้านซึ่งจุดใหม่ที่สร้างขึ้นมาแสดงดังสมการที่ 2.2

$$N_{point} = O_{point} + (Random[0,1] * distance(x, y, \dots, z)) \quad (2.2)$$

โดยที่ N_{point} หมายถึง จุดข้อมูลของคลาสส่วนน้อยที่สร้างขึ้นมาใหม่

O_{point} หมายถึง จุดข้อมูลของคลาสส่วนน้อยที่นำไปใช้เป็นตัวตั้งต้นในการหาระยะห่างเทียบกับจุดเพื่อนบ้าน

Random[0,1] หมายถึงการสุ่มค่าระหว่าง 0 ถึง 1

Distance(x, y, …, z) หมายถึงระยะห่างระหว่างจุดตั้งต้นกับจุดเพื่อนบ้านในแอตทริบิวต์ x, y ถึง z

ตัวอย่างที่ 4 จากเทคนิคการสุ่มเกินหากเลือกใช้ SMOTE Technique เทคนิคนี้จะทำการเลือกข้อมูลที่อยู่ในคลาส False ขึ้นมาแล้วทำการเลือกข้อมูลเพื่อนบ้านที่อยู่ในคลาสเดียวกัน แล้วคำนวณระยะห่างระหว่างจุดสองจุดแล้วทำการสุ่มเพิ่มค่าขึ้นเล็กน้อยหรือลดค่าลงเล็กน้อย โดยเพิ่มข้อมูลเป็นจำนวน 3 ข้อมูล โดยใช้ข้อมูลจากลำดับที่ 1 เป็นข้อมูลตั้งต้น เพื่อทำให้ชุดข้อมูล C จากตารางที่ 2.2 มีความสมดุลเกิดขึ้นจะได้ชุดข้อมูลใหม่โดยมีข้อมูลทั้งหมด 17 ข้อมูล แบ่งเป็นข้อมูลในคลาส True จำนวน 7 ข้อมูล และข้อมูลในคลาส False จำนวน 10 ข้อมูล แสดงดังตารางที่ 2.5

ตารางที่ 2.5 ชุดข้อมูล C หลังจากใช้ SMOTE Technique เพิ่มข้อมูลจากคลาสส่วนน้อยเพิ่ม 3 ข้อมูล

ลำดับข้อมูล	ความกว้าง	ความยาว	น้ำหนักเนื้องอก	ระดับความเข้ม	คลาส
1	1.4”	2.5”	2.0 g	0.9	True
2	0.4”	1.1”	0.1 g	0.1	False
3	0.3”	0.9”	0.2 g	0.2	False
4	1.5”	2.0”	2.1 g	0.8	True
5	0.2”	0.9”	0.2 g	0.1	False
6	0.1”	0.4”	0.3 g	0.1	False
7	1.7”	2.3”	2.2 g	0.9	True
8	1.8”	2.3”	2.5 g	0.7	True
9	0.4”	0.6”	0.3 g	0.2	False
10	0.2”	0.7”	0.2 g	0.3	False
11	0.2”	0.3”	0.5 g	0.1	False
12	0.1”	0.2”	0.3 g	0.2	False
13	0.5”	0.1”	1.8 g	0.6	False
14	0.1”	0.5”	0.7 g	0.3	False
15	**1.35”**	**2.75”**	**2.05 g**	**0.95**	**True**
16	**1.34”**	**2.54”**	**1.96 g**	**0.9**	**True**
17	**1.16”**	**2.62”**	**1.7 g**	**1.02**	**True**

วิธีการสุ่มลด (Under Sampling)

วิธีสุ่มลด (Japkowicz, 2000a; Japkowicz and Stephen, 2002 ;กิตติพงษ์, 2016) เป็นเทคนิคหรือวิธีที่ใช้ในการลดจำนวนข้อมูลที่อยู่ในคลาสส่วนมาก ให้มีจำนวนใกล้เคียง หรือเท่ากับจำนวนข้อมูลในคลาสส่วนน้อย โดยการลดจำนวนข้อมูลจากคลาสส่วนมากลง

ตัวอย่างที่ 5 จากเทคนิคการสุ่มลดเทคนิคนี้จะทำการลดจำนวนข้อมูลที่อยู่ในคลาส False โดยลดข้อมูลเป็นจำนวน 4 – 6 ข้อมูล เพื่อทำให้ชุดข้อมูล C จากตารางที่ 2.2 มีความสมดุลเกิดขึ้น สมมติว่าทำการสุ่มลดจำนวนข้อมูลจากคลาสส่วนมากลงเป็นจำนวน 4 ข้อมูล (ลดข้อมูลลำดับที่ 3, 9, 12 และ 14) จะได้ชุดข้อมูลใหม่โดยมีข้อมูลทั้งหมด 10 ข้อมูล แบ่งเป็นข้อมูลในคลาส True จำนวน 4 ข้อมูล และข้อมูลในคลาส False จำนวน 6 ข้อมูล แสดงดังตารางที่ 2.6

ตารางที่ 2.6 ชุดข้อมูล C หลังจากใช้เทคนิคการสุ่มลดข้อมูลจากคลาสส่วนมาก

ลำดับข้อมูล	ความกว้าง	ความยาว	น้ำหนักเนื้องอก	ระดับความเข้ม	คลาส
1	1.4”	2.5”	2.0 g	0.9	True
2	0.4”	1.1”	0.1 g	0.1	False
3	1.5”	2.0”	2.1 g	0.8	True
4	0.2”	0.9”	0.2 g	0.1	False
5	0.1”	0.4”	0.3 g	0.1	False
6	1.7”	2.3”	2.2 g	0.9	True
7	1.8”	2.3”	2.5 g	0.7	True
8	0.2”	0.7”	0.2 g	0.3	False
9	0.2”	0.3”	0.5 g	0.1	False
10	0.5”	0.1”	1.8 g	0.6	False

วิธีผสมผสาน (Hybrid Methods)

วิธีผสมผสาน เป็นวิธีการที่นำเทคนิควิธีสุ่มเกิน และวิธีสุ่มลดมาทำงานร่วมกัน โดยการใช้เทคนิคนี้จะเป็นการสุ่มลดจำนวนข้อมูลจากคลาสส่วนมาก และทำการสุ่มเพิ่มข้อมูลในคลาสส่วนน้อย ให้จำนวนข้อมูลจากทั้งสองคลาสมีจำนวนใกล้เคียงกัน หรือเท่ากัน

ตัวอย่างที่ 6 จากเทคนิคการจัดการข้อมูลแบบผสมผสาน เทคนิคนี้จะทำการสุ่มลดข้อมูลจากคลาสส่วนมากลงจำนวน 2 - 3 ข้อมูล และสุ่มเพิ่มข้อมูลจากคลาสส่วนน้อยเพิ่มขึ้นจำนวน 2 - 3 ข้อมูล เพื่อให้ชุดข้อมูล C จากตารางที่ 2.2 มีความสมดุลเกิดขึ้น สมมติว่าทำการสุ่มลดจำนวนข้อมูลจากคลาสส่วนมากลงจำนวน 2 ข้อมูล (สุ่มลดข้อมูลในลำดับที่ 10 และ 13) และทำการสุ่มเพิ่มข้อมูลจากคลาสส่วนน้อยด้วยการสุ่มเกินจากข้อมูลเดิมจำนวน 2 ข้อมูล (สุ่มเพิ่มข้อมูลจากลำดับที่ 1 และ

7) จะได้จำนวนข้อมูลทั้งหมด 14 ข้อมูล โดยเป็นข้อมูลจากคลาส True มีทั้งหมด 6 ข้อมูล และจำนวนข้อมูลทั้งหมดจากคลาส False มีทั้งหมด 8 ข้อมูล แสดงดังตารางที่ 2.7

ตารางที่ 2.7 ชุดข้อมูล C หลังจากใช้เทคนิคแบบผสมผสานในการแก้ปัญหาข้อมูลไม่สมดุล

ลำดับข้อมูล	ความกว้าง	ความยาว	น้ำหนักเนื้องอก	ระดับความเข้ม	คลาส
1	1.4”	2.5”	2.0 g	0.9	True
2	0.4”	1.1”	0.1 g	0.1	False
3	0.3”	0.9”	0.2 g	0.2	False
4	1.5”	2.0”	2.1 g	0.8	True
5	0.2”	0.9”	0.2 g	0.1	False
6	0.1”	0.4”	0.3 g	0.1	False
7	1.7”	2.3”	2.2 g	0.9	True
8	1.8”	2.3”	2.5 g	0.7	True
9	0.4”	0.6”	0.3 g	0.2	False
10	**1.4”**	**2.5”**	**2.0 g**	**0.9**	**True**
11	0.2”	0.3”	0.5 g	0.1	False
12	0.1”	0.2”	0.3 g	0.2	False
13	**1.7”**	**2.3”**	**2.2 g**	**0.9**	**True**
14	0.1”	0.5”	0.7 g	0.3	False

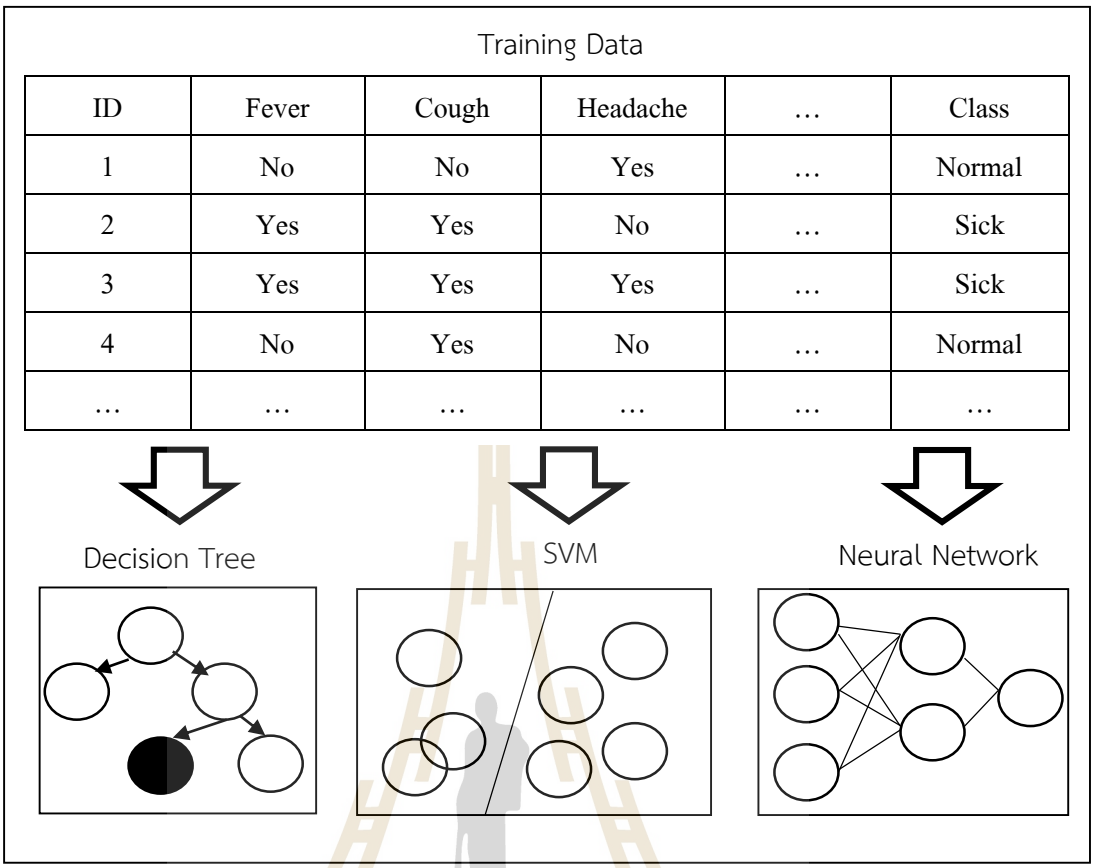
การแก้ปัญหาในระดับอัลกอริทึมเป็นการปรับปรุงการทำงานของอัลกอริทึมต่าง ๆ ให้สามารถจำแนกประเภทข้อมูลไม่สมดุลได้ดียิ่งขึ้น โดยเทคนิคที่นิยมนำมาใช้ได้แก่เทคนิคการทำงานร่วมกัน (Ensemble) และเทคนิคบูสติง (Boosting)

เทคนิคการทำงานร่วมกัน

เทคนิคการทำงานร่วมกันเป็นเทคนิคที่ใช้โมเดลในการจำแนกหลาย ๆ โมเดลมาช่วยในการจำแนกประเภทร่วมกัน ซึ่งทำให้ประสิทธิภาพในการจำแนกสูง (Muhlbaier et al., 2009) โดยจะกล่าวถึง 2 เทคนิคได้แก่ Vote Ensemble และเทคนิคแบ็กกิง (Bagging: Bootstrap Aggregation)

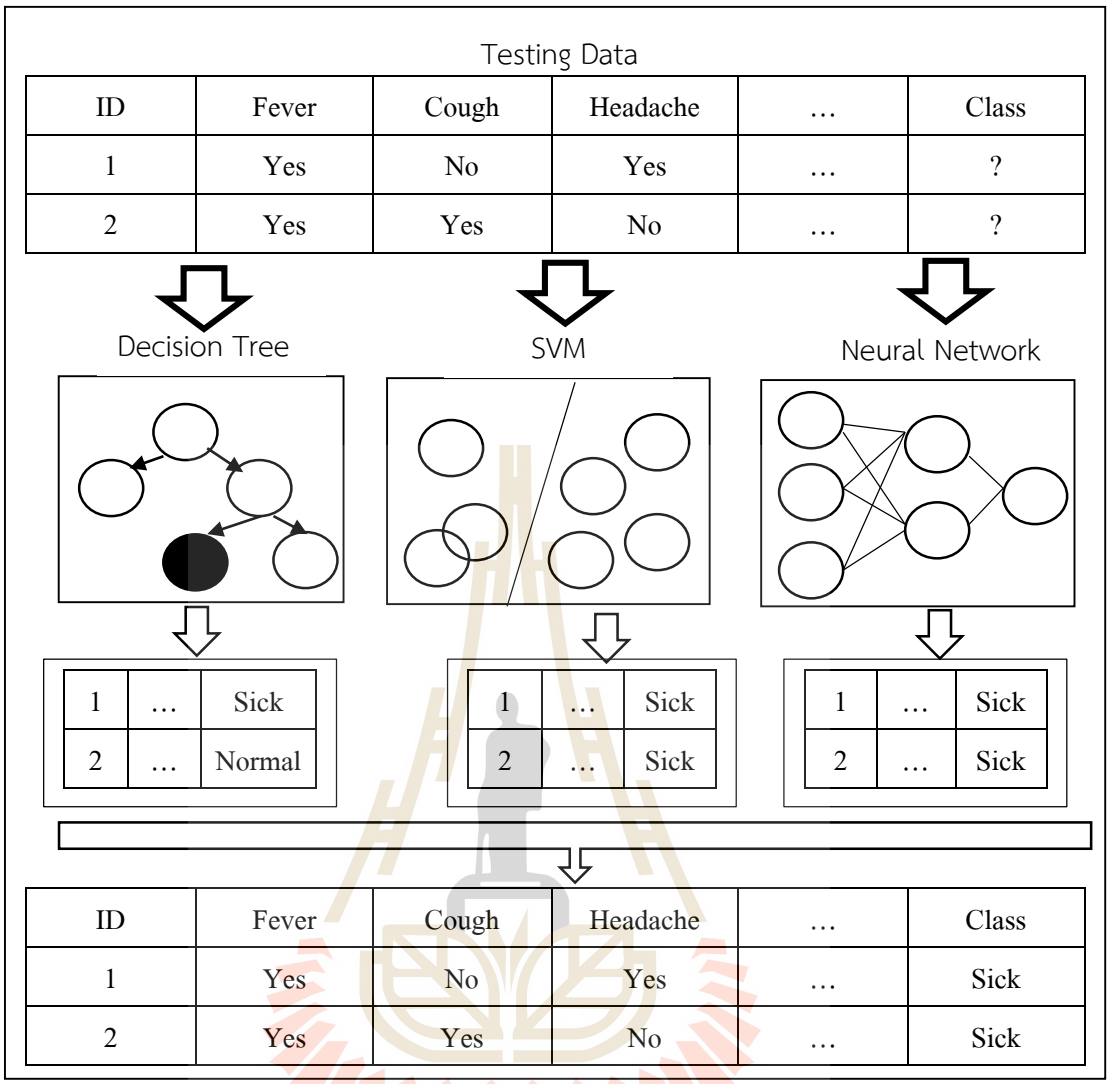
1) Vote Ensemble

เทคนิคนี้เป็นการใช้ชุดข้อมูลในการเรียนรู้ชุดเดียวกันแต่สร้างโมเดลการจำแนกประเภทข้อมูลด้วยอัลกอริทึมที่แตกต่างกัน โดยหลักการสร้างโมเดลสำหรับจำแนกประเภทข้อมูลด้วยเทคนิค Vote Ensemble สามารถแสดงดังรูปที่ 2.3



รูปที่ 2.3 หลักการสร้างโมเดลการจำแนกประเภทด้วย Vote Ensemble

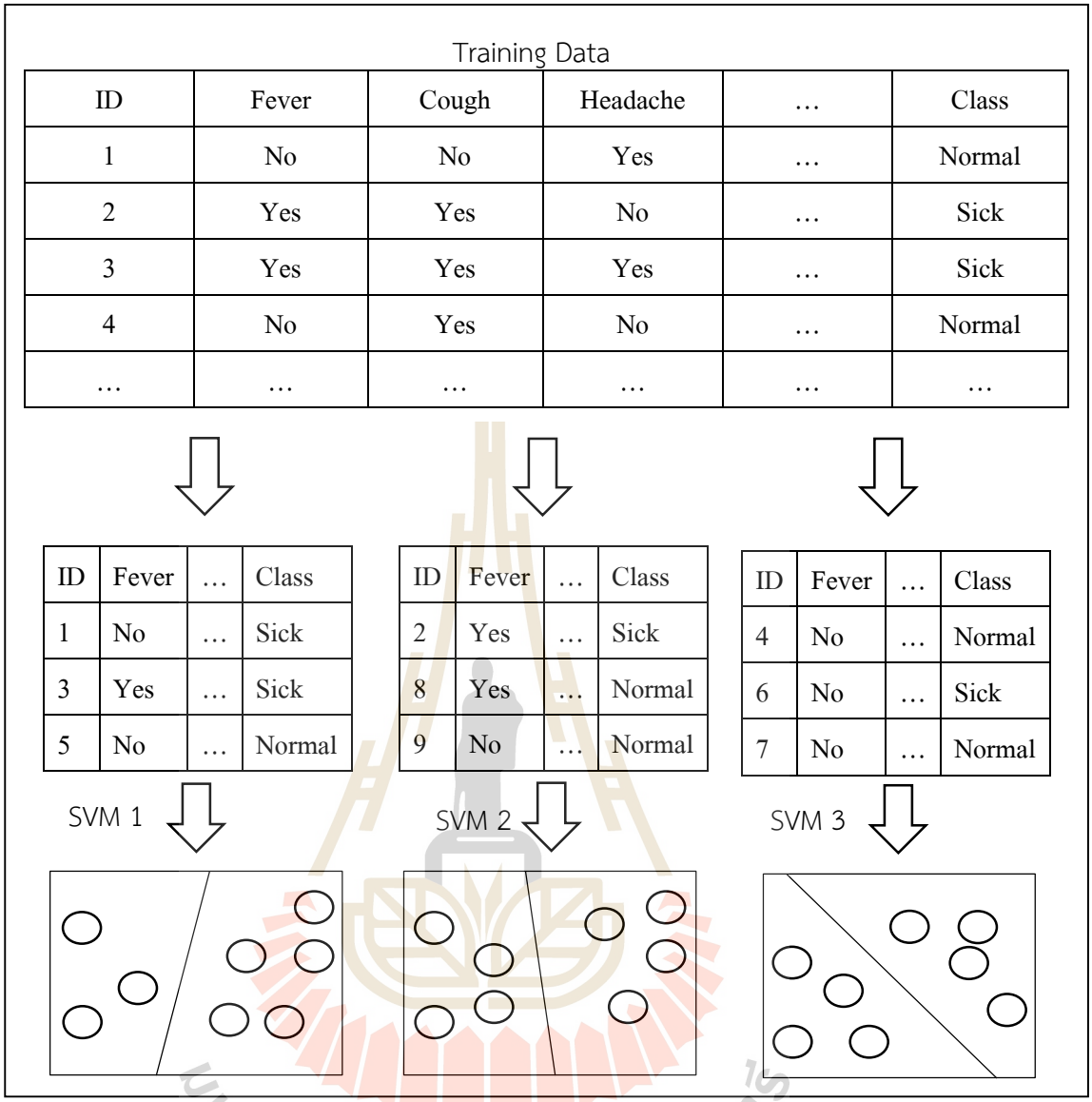
สำหรับการใช้เทคนิค Vote Ensemble เพื่อใช้ในการจำแนกประเภทข้อมูลที่ยังไม่ทราบคลาสโดยอาศัยผลโหวตจากเสียงส่วนมากซึ่งได้รับจากการจำแนกประเภทจากแต่ละอัลกอริทึม เช่น นำข้อมูลที่ยังไม่ทราบคลาสเป้าหมายไปจำแนกด้วยอัลกอริทึม Decision Tree, SVM และ Neural Network แล้วตรวจสอบว่าทั้งสามอัลกอริทึมให้ผลการจำแนกข้อมูลนั้นว่าอยู่ในประเภทไหนมากที่สุด จะทำการจำแนกประเภทข้อมูลใหม่นั้นตามผลโหวตดังกล่าว แสดงหลักการทำงานดังรูปที่ 2.4



รูปที่ 2.4 หลักการจำแนกประเภทข้อมูลด้วย Vote Ensemble

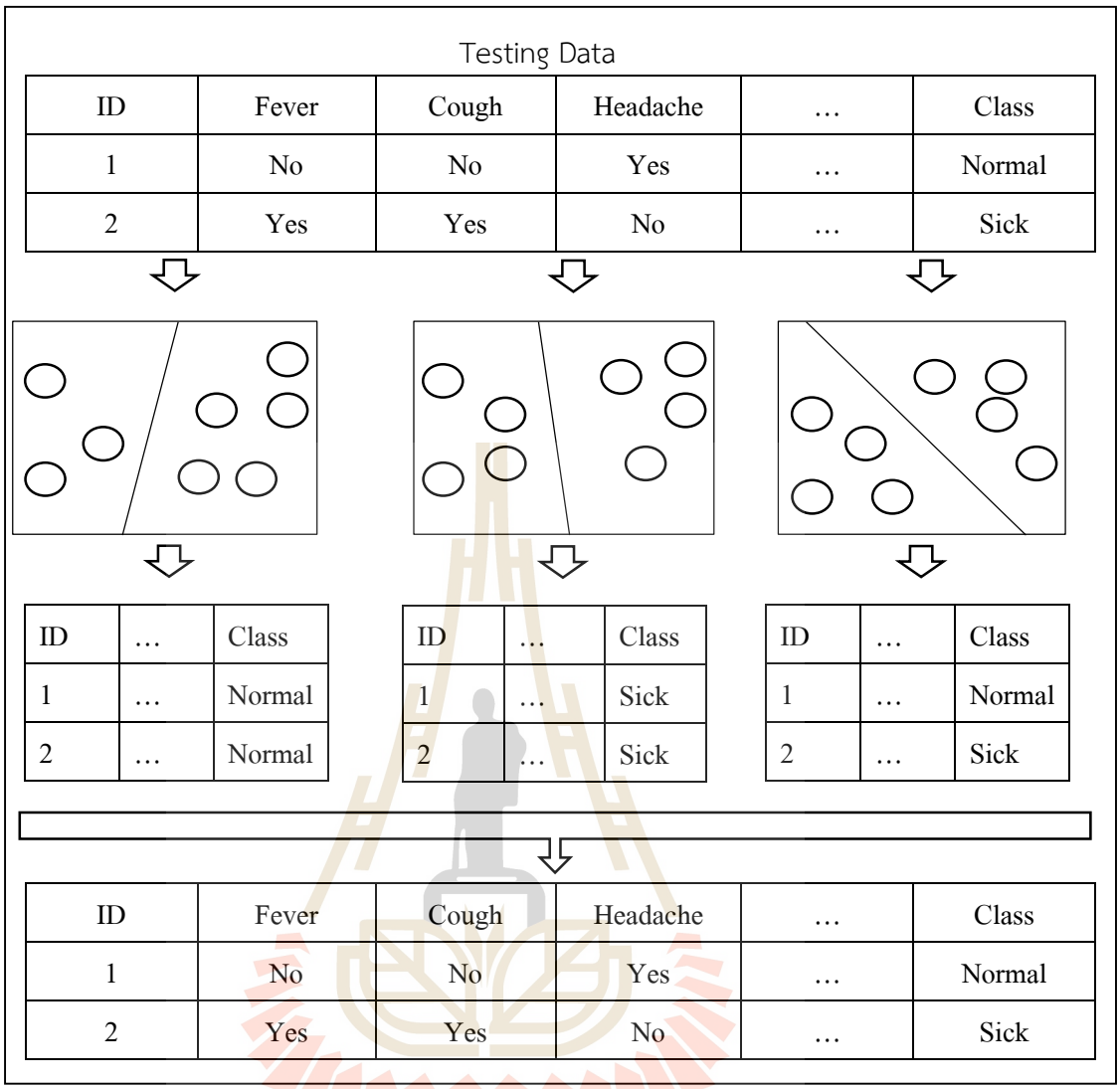
2) แบ็กกิง

สำหรับเทคนิคแบ็กกิงจะใช้เทคนิคการสุ่มข้อมูลตัวอย่างจากชุดข้อมูลเรียนรู้ออกเป็นหลาย ๆ ชุด โดยนำข้อมูลที่แบ่งแยกออกไปสร้างโมเดลด้วยการใช้อัลกอริทึมในการจำแนกประเภทชนิดเดียวกัน (อัลกอริทึมเดียวกัน) (Breiman, 1996) โดยหลักการสร้างโมเดลจำแนกประเภทของเทคนิคแบ็กกิงแสดงดังรูปที่ 2.5



รูปที่ 2.5 หลักการสร้างโมเดลด้วยเทคนิคแบ็กกิง

สำหรับการใช้เทคนิคแบ็กกิงเพื่อใช้ในการจำแนกประเภทข้อมูลที่ยังไม่ทราบคลาสจะอาศัยผลโหวตจากเสียงส่วนมากจากผลของการจำแนกประเภทจากแต่ละโมเดล จะมีหลักการจำแนกประเภทข้อมูลที่ยังไม่ทราบคลาส แสดงดังรูปที่ 2.6

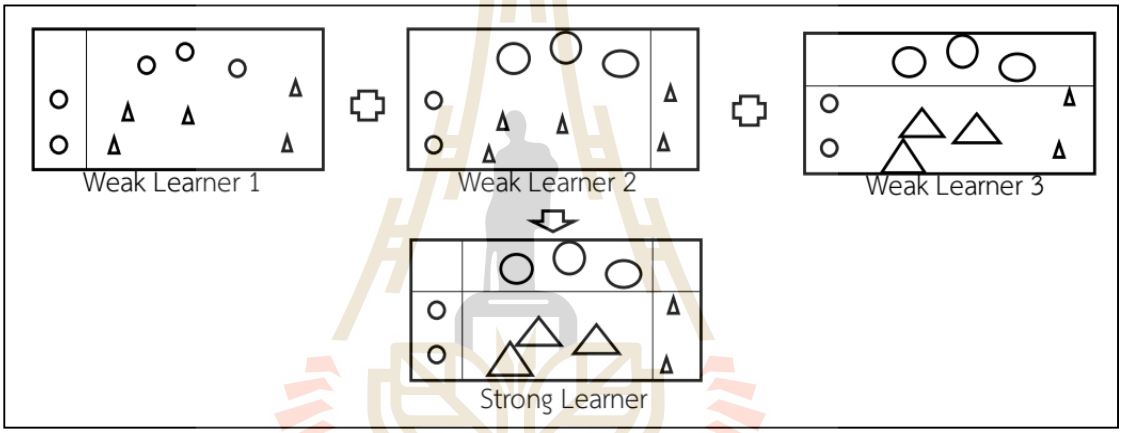


รูปที่ 2.6 หลักการจำแนกประเภทข้อมูลด้วยเทคนิคแบ็กกิง

นอกจากนี้ยังมีการนำเทคนิคแบ็กกิงไปประยุกต์ใช้ร่วมกันกับเทคนิคการปรับปรุงข้อมูล เช่น การนำเทคนิคการสุ่มเพิ่มข้อมูลมาทำงานร่วมกับเทคนิคแบ็กกิง ซึ่งเรียกว่า Over Bagging (Wang and Yao, 2009) หรือการใช้เทคนิคการสร้างตัวอย่างเพิ่มจากข้อมูลส่วนน้อย หรือเรียกว่า SMOTEBagging (Wang and Yao, 2009) การนำเทคนิคการสุ่มลดข้อมูลจากคลาสส่วนมากร่วมกับการใช้เทคนิคแบ็กกิง เรียกว่า Under Bagging (Liu et al., 2006)

3) บูสติง

สำหรับเทคนิคบูสติงคือการประยุกต์ใช้ตัวจำแนกประเภทข้อมูลที่อ่อนแอ (Weak Learner) หลาย ๆ อัลกอริทึมมาทำงานร่วมกัน แล้วสร้างตัวจำแนกที่แข็งแกร่ง (Strong Learner) โดยหลักการสำคัญของเทคนิคนี้คือการปรับเพิ่มค่าน้ำหนักให้แก่ข้อมูลที่ยังมีการจำแนกผิดประเภท แล้วสร้างตัวจำแนกประเภทข้อมูลจากการเรียนรู้ในขั้นตอนก่อนหน้า (Valiant, 1984; Kearns and Valiant, 1994) ซึ่งการนำตัวจำแนกประเภทข้อมูลที่อ่อนแอเหล่านั้นมาทำงานร่วมกันจะทำให้ตัวจำแนกมีความแข็งแกร่งมากยิ่งขึ้น สำหรับเทคนิคหรืออัลกอริทึมทางด้านบูสติงที่ได้รับความนิยมนำไปใช้งานได้แก่อัลกอริทึมเอด้าบูส (Adaboost) (Freund and Schapire, 1996; Freund et al., 1999) โดยมีหลักการทำงาน แสดงดังรูปที่ 2.7



รูปที่ 2.7 หลักการทำงานของอัลกอริทึมเอดาบูส

2.2 การประเมินประสิทธิภาพสำหรับข้อมูลไม่สมดุล

สำหรับการประเมินประสิทธิภาพการจำแนกข้อมูลไม่สมดุล หากใช้วิธีการประเมินประสิทธิภาพด้วยเทคนิคพื้นฐาน ได้แก่ การใช้ค่าความแม่นยำในการจำแนก (Accuracy) อาจจะไม่เพียงพอต่อการประเมินประสิทธิรูปที่แท้จริง เนื่องจากหากเกิดกรณีที่ข้อมูลไม่สมดุลนั้นมีจำนวนข้อมูลในคลาสแต่ละคลาสไม่เท่ากันเป็นจำนวนมาก หากทำนายข้อมูลใหม่เป็นคลาสส่วนมากทั้งหมดก็จะทำให้ค่าความแม่นยำในการจำแนกสูงได้เช่นกัน เช่น ชุดข้อมูลหนึ่งมีจำนวนข้อมูลทั้งหมด 1,000 ข้อมูล แบ่งเป็นจำนวนข้อมูลในคลาสส่วนมากมีจำนวน 940 ข้อมูล และจำนวนข้อมูลในคลาสส่วนน้อยมีจำนวน 60 ข้อมูล หากโมเดลจำแนกว่าข้อมูลทั้ง 1,000 ข้อมูลอยู่ในคลาสส่วนมาก โมเดลนี้จะมีค่าความแม่นยำในการจำแนกอยู่ที่ 94% แต่ในขณะที่ไม่สามารถทำนายข้อมูลในคลาสส่วนน้อยได้แม้เพียงข้อมูลเดียว ดังนั้นการจำแนกข้อมูลไม่สมดุลจึงจำเป็นต้องมีการประเมินประสิทธิภาพในการจำแนกด้วยวิธีอื่น ๆ เพิ่มเติม

เมตริกซ์วัดประสิทธิภาพ (Confusion Matrix) คือเมตริกซ์ที่ใช้แสดงผลการจำแนกข้อมูลจากการทดสอบด้วยชุดข้อมูลออกเป็นแต่ละคลาส เมื่อต้องการประเมินประสิทธิภาพการทำนายคลาสส่วนน้อยเป็นหลัก โดยมีรูปแบบแสดงดังตารางที่ 2.8

ตารางที่ 2.8 เมตริกซ์วัดประสิทธิภาพสำหรับจำแนกข้อมูลสองคลาส

	Actual Minority Class	Actual Majority Class
Minority Class Prediction	True Minority Class	False Minority Class
Majority Class Prediction	False Majority Class	True Majority Class

จากตารางที่ 2.8 แถวของเมตริกซ์จะแสดงจำนวนของข้อมูลจริงของแต่ละคลาส และคอลัมน์ของเมตริกซ์จะแสดงจำนวนที่ทำนายได้ของแต่ละคลาส แบ่งออกเป็น 4 กรณี ดังนี้

กรณีที่ 1 : True Minority Class หมายถึง จำนวนข้อมูลที่อยู่ในคลาสส่วนน้อย แล้วโมเดลสามารถทำนายได้ถูกต้องว่าข้อมูลนั้นอยู่ในคลาสส่วนน้อย

กรณีที่ 2 : False Majority Class หมายถึง จำนวนข้อมูลที่อยู่ในคลาสส่วนมากแต่โมเดลทำนายผิดพลาด โดยทำนายว่าข้อมูลนั้นอยู่ในคลาสส่วนน้อย

กรณีที่ 3 : False Minority Class หมายถึง จำนวนข้อมูลที่อยู่ในคลาสส่วนน้อยแต่โมเดลทำนายผิดพลาด โดยทำนายว่าข้อมูลนั้นอยู่ในคลาสส่วนมาก

กรณีที่ 4 : True Majority Class หมายถึง จำนวนข้อมูลที่อยู่ในคลาสส่วนมาก แล้วโมเดลสามารถทำนายได้ถูกต้องว่าข้อมูลนั้นอยู่ในคลาสส่วนมาก

ค่าความแม่นยำในการจำแนก (Accuracy)

มาตรวัดความแม่นยำในการจำแนกเป็นการประเมินประสิทธิภาพการจำแนกโดยรวมของทุกคลาสของโมเดล แสดงดังสมการที่ 2.3

$$Accuracy = \frac{(\text{True Minority Class} + \text{True Majority Class})}{(Total\ data)} \tag{2.3}$$

ค่าความเที่ยง (Precision)

มาตรวัดความเที่ยง (Buckland and Gey, 1994) เป็นการประเมินความแม่นยำในการทำนายข้อมูลที่อยู่ในคลาสส่วนน้อย โดยคำนวณจากจำนวนข้อมูลที่ทำนายเป็นคลาสส่วนน้อยได้ถูกต้องเทียบกับจำนวนข้อมูลที่ถูกทำนายเป็นคลาสส่วนน้อยทั้งหมด แสดงดังสมการที่ 2.4

$$Precision = \frac{(\text{True Minority Class})}{(\text{True Minority Class} + \text{False Minority Class})} \tag{2.4}$$

ค่าระลึก หรือค่าความไว (Recall / Sensitivity)

มาตรวัดค่าระลึกหรือค่าความไว (Buckland and Gey, 1994) เป็นการประเมินความแม่นยำในการทำนายข้อมูลที่อยู่ในคลาสส่วนน้อยว่าสามารถทำนายได้ถูกต้องแม่นยำเพียงใด โดยคำนวณจากจำนวนข้อมูลที่ทำนายเป็นคลาสส่วนน้อยได้ถูกต้อง เทียบกับจำนวนข้อมูลจริงของคลาสส่วนน้อยทั้งหมด แสดงดังสมการที่ 2.5

$$Sensitivity = Recall = \frac{(\text{True Minority Class})}{(\text{True Minority Class}+\text{False Majority Class})} \quad (2.5)$$

ค่าความจำเพาะ (Specificity)

มาตรวัดความจำเพาะ เป็นการประเมินความแม่นยำในการทำนายข้อมูลที่อยู่ในคลาสส่วนมาก โดยคำนวณจากจำนวนข้อมูลที่ถูกทำนายเป็นคลาสส่วนมากได้ถูกต้อง เทียบกับจำนวนข้อมูลจริงของคลาสส่วนมากทั้งหมด แสดงดังสมการที่ 2.6

$$Specificity = \frac{(\text{True Majority Class})}{(\text{True Majority Class}+\text{False Minority Class})} \quad (2.6)$$

ค่าการวัดเอฟ (F-Measure)

การวัดเอฟ เป็นการประเมินความแม่นยำของการจำแนกคลาสส่วนน้อยโดยดูจากผลเฉลี่ยของ Precision และ Recall แสดงดังสมการที่ 2.7

$$F - measure = \frac{(2*\text{Precision}*\text{Recall})}{(\text{Precision}+\text{Recall})} \quad (2.7)$$

ตัวอย่างที่ 7 สมมติให้การจำแนกข้อมูลไม่สมดุลที่มี 2 คลาส โดยมีข้อมูลทั้งหมด 200 ข้อมูล แบ่งเป็นข้อมูลในคลาสส่วนมากจำนวน 180 ข้อมูล และจำนวนข้อมูลในคลาสส่วนน้อย 20 ข้อมูล โดยมีผลลัพธ์การจำแนกแสดงดังตารางที่ 2.9

ตารางที่ 2.9 ผลลัพธ์การจำแนกข้อมูลไม่สมดุล

	Actual Minority Class	Actual Majority Class
Minority Class Prediction	18	5
Majority Class Prediction	2	175

จะได้ว่าโมเดลการจำแนกนี้ มีประสิทธิภาพในการจำแนกด้วยเกณฑ์ต่าง ๆ ดังนี้

ค่าความแม่นยำในการจำแนก = (175 + 18) / (200) = 0.965

ค่า Precision = (18) / (18 + 5) = 0.783

ค่า Recall	=	(18) / (18+2)	=	0.900
ค่า Specificity	=	(175) / (175 + 5)	=	0.972
ค่า F-measure	=	(2*0.783*0.900) / (0.783 + 0.900)	=	0.837

โดยสามารถสรุปได้ว่า โมเดลนี้มีความสามารถในการจำแนกทั้งสองคลาสอยู่ที่ 96.5% สามารถจำแนกข้อมูลที่อยู่ในคลาสส่วนน้อยได้ถูกต้องเทียบกับการจำแนกข้อมูลว่าเป็นคลาสส่วนน้อยทั้งหมดอยู่ที่ 78.3% สามารถจำแนกข้อมูลของคลาสส่วนน้อยทั้งหมดได้แม่นยำอยู่ที่ 90% และสามารถจำแนกข้อมูลว่าเป็นคลาสส่วนมากได้ถูกต้องทั้งหมดได้แม่นยำอยู่ที่ 97.2% โดยรวมแล้วโมเดลนี้สามารถจำแนกเฉพาะข้อมูลคลาสส่วนน้อยมีความแม่นยำอยู่ที่ 83.7%

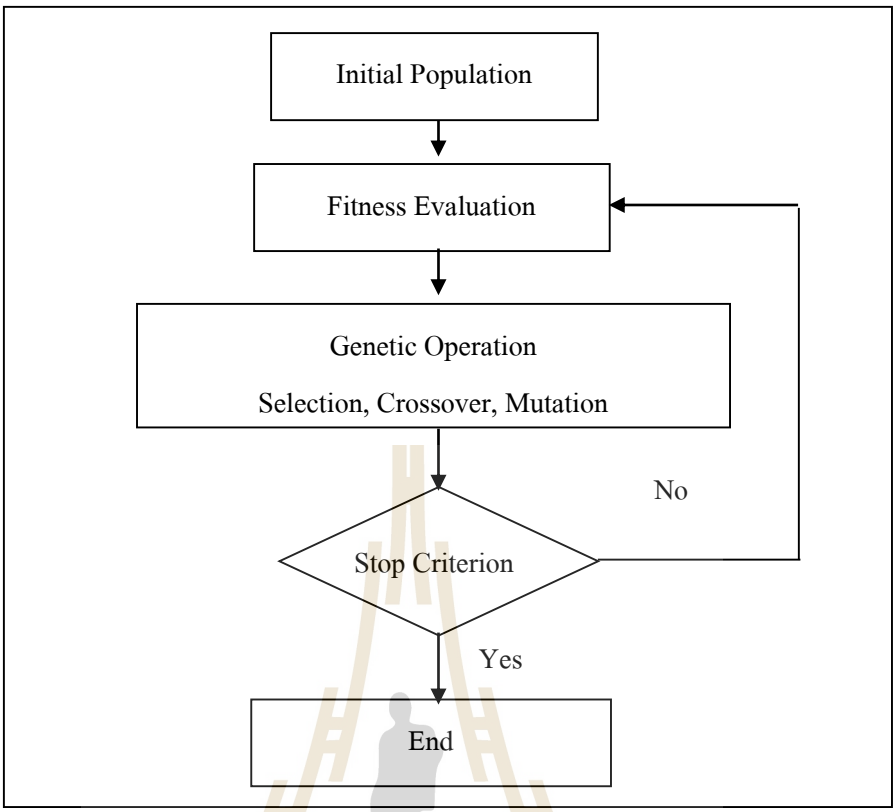
2.3 ขั้นตอนวิธีเชิงพันธุกรรม

ขั้นตอนวิธีเชิงพันธุกรรม เป็นวิธีการค้นหาคำตอบโดยอาศัยการเลียนแบบวิวัฒนาการทางธรรมชาติ โดยมีพื้นฐานแนวคิดมาจากทฤษฎีวิวัฒนาการทางธรรมชาติของ Charie Darwin คือ ผู้ที่แข็งแกร่งกว่าย่อมมีโอกาสในการอยู่รอดมากกว่าผู้ที่อ่อนแอ และมีโอกาสที่จะถ่ายทอดลักษณะทางพันธุกรรมที่แข็งแกร่งเหล่านั้นไปยังรุ่นลูกหลานต่อไป โดยขั้นตอนวิธีเชิงพันธุกรรมเริ่มเป็นที่รู้จักจากงานวิจัยของ John Holland (Holland, 1975) โดยประยุกต์นำเอาการวิวัฒนาการของสิ่งมีชีวิตในระบบชีววิทยามาใช้ในการคำนวณด้วยคอมพิวเตอร์ หลังจากนั้นก็เริ่มมีการนำขั้นตอนวิธีเชิงพันธุกรรมไปประยุกต์ใช้ในงานด้านต่าง ๆ กันอย่างแพร่หลาย

การทำงานของขั้นตอนวิธีเชิงพันธุกรรม จะแบ่งออกเป็น 6 ขั้นตอนหลัก ๆ ได้แก่

1) การเข้ารหัสโครโมโซม (Chromosome Encoding)
2) การสร้างประชากรเริ่มต้น (Population Initialization)
3) การประเมินค่าความเหมาะสม (Fitness Function)
4) การดำเนินการทางขั้นตอนวิธีเชิงพันธุกรรม (Genetic Operations)
5) การแทนที่ (Replacement)
6) การตรวจสอบเงื่อนไขสิ้นสุดการทำงาน (Termination Condition)

โดยแสดงดังรูปที่ 2.8 และสามารถอธิบายแต่ละขั้นตอนได้ดังต่อไปนี้



รูปที่ 2.8 ขั้นตอนการดำเนินงานของขั้นตอนวิธีเชิงพันธุกรรม

จากรูปที่ 2.8 สามารถอธิบายขั้นตอนการทำงานของขั้นตอนวิธีเชิงพันธุกรรมได้ดังนี้ ขั้นตอนการสร้างประชากรเริ่มต้น จะดำเนินการโดยการสุ่มสร้างประชากรจากกลุ่มข้อมูลที่มีอยู่ เพื่อนำประชากรเข้าสู่กระบวนการของขั้นตอนวิธีเชิงพันธุกรรมโดยในการสุ่ม จะสุ่มจำนวนประชากรให้ได้เท่ากับจำนวนประชากร (Population Size) ที่กำหนด หลังจากสุ่มประชากรเริ่มต้นแล้ว จะทำการคำนวณค่าความเหมาะสมของแต่ละประชากร เพื่อค้นหาประชากรที่มีความเหมาะสมตามเกณฑ์ที่กำหนดไปเป็นโครโมโซมตั้งต้นในการสืบทอดพันธุกรรม เมื่อได้โครโมโซมตั้งต้นแล้วจะดำเนินการทางขั้นตอนวิธีเชิงพันธุกรรม ซึ่งได้แก่ การคัดเลือก (Selection) การสลับสายพันธุ์ (Crossover) และทำการกลายพันธุ์ (Mutation) โดยขั้นตอนการคัดเลือกจะทำการคัดเลือกโครโมโซมตั้งต้นที่มีความเหมาะสมสูงที่สุด ส่วนขั้นตอนการสลับสายพันธุ์จะทำการสลับสายพันธุ์ระหว่างโครโมโซมตั้งต้นเพื่อให้เกิดความหลากหลายทางพันธุกรรม หลังจากนั้นจะทำการกลายพันธุ์โครโมโซมที่เกิดจากการสลับสายพันธุ์เพื่อให้เกิดการเปลี่ยนแปลงยีนภายในโครโมโซมที่มีอยู่เดิม หลังจากนั้นจะนำประชากรใหม่ที่ได้ไปแทนที่ประชากรรุ่นก่อนหน้า จนกระทั่งได้ประชากรที่ตรงตามเงื่อนไขที่กำหนด

1. การเข้ารหัสโครโมโซม

การเข้ารหัสโครโมโซมมีความสำคัญสำหรับขั้นตอนวิธีเชิงพันธุกรรมเป็นอย่างมาก เพราะก่อนที่จะเริ่มกระบวนการต่าง ๆ ของขั้นตอนเชิงพันธุกรรมจำเป็นที่จะต้องมีการเข้ารหัสโครโมโซมก่อน โดยในขั้นตอนนี้จะเป็นการออกแบบให้โครโมโซมเป็นตัวแทนของคำตอบของสิ่งที่ต้องการค้นหา โดยจะเลือกใช้วิธีการเข้ารหัสแบบใดก็ได้ซึ่งจะขึ้นอยู่กับความเหมาะสมของการแก้ปัญหา โดยการเข้ารหัสที่นิยมใช้กันมีอยู่ 3 วิธี ได้แก่ การเข้ารหัสแบบเลขฐานสอง (Binary Encoding) การเข้ารหัสแบบค่าจริง (Value Encoding) และการเข้ารหัสแบบเพอร์มิวเตชัน (Permutation Encoding)

1) การเข้ารหัสแบบเลขฐานสอง

การเข้ารหัสแบบเลขฐานสอง (Holland, 1975) เป็นรูปแบบการเข้ารหัสแบบพื้นฐานที่ทำการแปลงค่าให้อยู่ในรูปแบบเลขฐานสอง โดยจะมีรูปร่างของโครโมโซมอยู่ในรูปแบบ Bit String คือ 0 กับ 1 เท่านั้น นั่นหมายความว่ายีนในแต่ละตำแหน่งจะมีค่าเป็น 0 หรือ 1 เท่านั้น แสดงดังรูปที่ 2.9

Chromosome A :	0	1	0	1	1	0
Chromosome B :	1	0	1	1	0	1

รูปที่ 2.9 การเข้ารหัสแบบเลขฐานสอง

2) การเข้ารหัสแบบค่าจริง

การเข้ารหัสแบบค่าจริง (Wright, 1991) เป็นรูปแบบการเข้ารหัสโดยแทนค่าด้วยค่าต่าง ๆ ที่เป็นตัวแทนที่สามารถเชื่อมโยงค่าซึ่งใช้ในการแก้ปัญหา โดยโครโมโซมจะมีรูปร่างตามรูปแบบที่แทนค่า เช่น ตัวอักษร จำนวนจริง หรือคำสั่งต่าง ๆ แสดงดังรูปที่ 2.10

Chromosome A :	0.6	1.5	3.9	6.4	2.4	0.7
Chromosome B :	0.2	1.0	5.6	7.2	1.2	0.3

Chromosome C :	Low	High	Low	Normal	High	Low
Chromosome D :	Normal	High	Low	Low	High	High

รูปที่ 2.10 การเข้ารหัสแบบค่าจริง

3) การเข้ารหัสแบบเพอร์มิวเตชัน

การเข้ารหัสแบบเพอร์มิวเตชัน (Malhotra et al., 2011) เป็นรูปแบบการเข้ารหัสโดยแทนค่าจำนวนนับของตำแหน่งแต่ละลำดับลงไปที่ทุกตำแหน่งของยีนในโครโมโซม แสดงดังรูปที่ 2.11

Chromosome A :	1	6	5	2	4	3
Chromosome B :	6	3	1	4	2	5

รูปที่ 2.11 การเข้ารหัสแบบเพอร์มิวเตชัน

2. การสร้างประชากรเริ่มต้น

การสร้างประชากรเริ่มต้นจะสุ่มสร้างค่าขึ้นมาจากกลุ่มข้อมูลที่มีอยู่ เพื่อนำประชากรเข้าสู่ขั้นตอนเชิงพันธุกรรม โดยในการสุ่มจะต้องสุ่มให้ได้จำนวนเท่ากับขนาดประชากร (Population Size) ที่กำหนดไว้ ซึ่งในขั้นตอนนี้จะยังไม่สนใจค่าความเหมาะสมของแต่ละโครโมโซม

ตัวอย่างที่ 8 กำหนดให้ชุดข้อมูล A มีรายละเอียดแสดงดังตารางที่ 2.10

ตารางที่ 2.10 รายละเอียดชุดข้อมูล A

Chromosome	ยีนที่ 1	ยีนที่ 2	ยีนที่ 3	ยีนที่ 4	ยีนที่ 5	ยีนที่ 6	ยีนที่ 7	ยีนที่ 8
1	0	1	0	1	1	0	1	0
2	1	0	0	1	0	0	1	0
3	1	1	0	1	1	0	1	1
4	0	1	0	1	0	1	0	1
5	1	1	1	0	1	1	1	0
6	1	1	0	0	1	0	0	1
7	0	0	1	1	0	1	0	1
8	1	0	0	1	1	0	1	0
9	0	0	1	0	1	0	1	0
10	1	0	1	1	0	0	0	0
11	0	1	1	0	1	1	1	0
12	0	1	0	1	0	1	0	1
13	0	0	0	0	1	1	1	1
14	1	1	0	1	0	0	1	1
15	0	0	0	0	1	0	0	0

ตารางที่ 2.10 รายละเอียดชุดข้อมูล A (ต่อ)

Chromosome	ยีนที่ 1	ยีนที่ 2	ยีนที่ 3	ยีนที่ 4	ยีนที่ 5	ยีนที่ 6	ยีนที่ 7	ยีนที่ 8
16	1	1	0	0	1	1	1	0
17	0	1	1	1	1	0	0	1
18	1	0	1	1	0	1	1	1
19	1	0	1	0	1	0	1	1
20	0	1	1	1	0	1	1	0

ในขั้นตอนการสุ่มเลือกประชากรเริ่มต้น จะไม่สนใจค่าความเหมาะสมของแต่ละโครโมโซม สมมติว่ากำหนดให้สุ่มเลือกประชากรเริ่มต้นเป็นจำนวน 10 ประชากร จะได้ดังตารางที่ 2.11

ตารางที่ 2.11 ประชากรเริ่มต้นของชุดข้อมูล A จากการสุ่มเลือกตามขนาดประชากร 10 ประชากร

Chromosome	ยีนที่ 1	ยีนที่ 2	ยีนที่ 3	ยีนที่ 4	ยีนที่ 5	ยีนที่ 6	ยีนที่ 7	ยีนที่ 8
1	0	1	0	1	1	0	1	0
2	0	1	1	1	0	1	1	0
3	1	1	0	1	1	0	1	1
4	1	1	1	0	1	1	1	0
5	1	0	1	1	0	1	1	1
6	1	1	0	0	1	0	0	1
7	0	0	0	0	1	0	0	0
8	1	0	0	1	1	0	1	0
9	0	0	1	0	1	0	1	0
10	0	0	0	0	1	1	1	1

3. ฟังก์ชันค่าความเหมาะสม

ฟังก์ชันค่าความเหมาะสม จะเป็นตัวกำหนดค่าความเหมาะสมของแต่ละโครโมโซม เพื่อให้คะแนนความเหมาะสมของแต่ละโครโมโซม โดยโครโมโซมแต่ละตัวจะมีค่าความเหมาะสมของตัวเองเพื่อใช้สำหรับพิจารณาว่าโครโมโซมตัวนั้นเหมาะสมหรือไม่ที่จะนำไปใช้ในการสืบทอดพันธุกรรม โดยวิธีการคำนวณค่าความเหมาะสมนั้นจะใช้สมการที่สอดคล้องกับแต่ละปัญหา

ตัวอย่างที่ 9 คำนวณค่าความเหมาะสมของแต่ละโครโมโซมจากตารางที่ 2.11 โดยกำหนดให้ฟังก์ชันค่าความเหมาะสมคือ “โครโมโซมเลขฐานสองจำนวน 8 บิต ที่มีค่าสูงที่สุดเมื่อ

แปลงเป็นเลขฐาน 10 เป็นโครโมโซมที่เหมาะสมที่สุด” จะได้ค่าความเหมาะสมของแต่ละโครโมโซม แสดงดังตารางที่ 2.12

ตารางที่ 2.12 ค่าความเหมาะสมของแต่ละโครโมโซม

Chromosome	ยีนที่ 1	ยีนที่ 2	ยีนที่ 3	ยีนที่ 4	ยีนที่ 5	ยีนที่ 6	ยีนที่ 7	ยีนที่ 8	ค่าความเหมาะสม
4	1	1	1	0	1	1	1	0	238
3	1	1	0	1	1	0	1	1	219
6	1	1	0	0	1	0	0	1	201
5	1	0	1	1	0	1	1	1	183
8	1	0	0	1	1	0	1	0	154
2	0	1	1	1	0	1	1	0	118
1	0	1	0	1	1	0	1	0	90
9	0	0	1	0	1	0	1	0	42
10	0	0	0	0	1	1	1	1	15
7	0	0	0	0	1	0	0	0	8

โดยทั่วไปแล้วฟังก์ชันค่าความเหมาะสมจะถูกกำหนดให้เหมาะสมกับลักษณะงานที่นำไปใช้งาน เช่น ในด้านการจำแนกข้อมูล อาจจะใช้ฟังก์ชันความเหมาะสมคือ ใช้ค่าความแม่นยำในการจำแนก (Accuracy) แสดงดังสมการที่ 2.8 หรือสำหรับข้อมูลไม่สมดุลอาจจะใช้ค่าความเที่ยงแสดงดังสมการที่ 2.9 มาเป็นเกณฑ์ในการคำนวณค่าความเหมาะสมของแต่ละโครโมโซม เป็นต้น

$$Accuracy = \frac{TP+TN}{Total\ Data} \tag{2.8}$$

เมื่อ TP หมายถึง จำนวนข้อมูลที่อยู่ในคลาส Positive และโมเดลทำนายได้ถูกต้อง
TN หมายถึง จำนวนข้อมูลที่อยู่ในคลาส Negative และโมเดลทำนายได้ถูกต้อง
Total Data หมายถึง จำนวนข้อมูลที่นำมาใช้ทดสอบทั้งหมด

$$Precision = \frac{TP}{TP+FP} \tag{2.9}$$

เมื่อ TP หมายถึง จำนวนข้อมูลที่อยู่ในคลาส Positive และโมเดลสามารถทำนายได้ถูกต้อง
FP หมายถึง จำนวนข้อมูลที่อยู่ในคลาส Negative และโมเดลทำนายเป็นคลาส Positive

4. การดำเนินการทางขั้นตอนวิธีเชิงพันธุกรรม

การดำเนินการทางขั้นตอนวิธีเชิงพันธุกรรม จะประกอบไปด้วย 3 ขั้นตอนสำคัญ (Zheng et al., 2002) ได้แก่ การคัดเลือก (Selection) เพื่อเป็นประชากรในรุ่นถัดไป การสลับสายพันธุ์ (Crossover) และการกลายพันธุ์ (Mutation)

1) การคัดเลือก

การคัดเลือกสายพันธุ์ เป็นวิธีการที่สนับสนุนให้ประชากรที่มีความเหมาะสมในปัจจุบันถูกส่งต่อไปยังรุ่นถัดไป โดยคัดเลือกจากโครโมโซมที่ดีที่สุดจากภายในกลุ่มประชากรทั้งหมดจะถูกนำไปใช้เป็นโครโมโซมพ่อแม่ในการสืบพันธุ์เพื่อใช้ในการให้กำเนิดลูกหลานในรุ่นต่อไป หลักของการอยู่รอดของสิ่งมีชีวิตที่เหมาะสมจะสามารถอยู่รอดได้หากต้นกำเนิดสายพันธุ์มีความเหมาะสม ดังนั้นจึงต้องเลือกโครโมโซมรุ่นพ่อแม่ที่มีค่าความเหมาะสมสูงที่สุดนั่นเอง ในกระบวนการคัดเลือกนี้จะมีเทคนิคในการคัดเลือกอยู่หลายเทคนิคด้วยกัน เช่น การคัดเลือกแบบจัดอันดับ (Ranking Selection) การคัดเลือกแบบจัดการแข่งขัน (Tournament Selection) เป็นต้น

การคัดเลือกแบบจัดอันดับ

วิธีการคัดเลือกแบบจัดอันดับจะกำหนดให้โครโมโซมทุกตัวจะถูกจัดเรียงให้มีอันดับตามค่าความเหมาะสม โดยจะให้ค่าโอกาสในการถูกเลือกของโครโมโซมที่เหมาะสมที่สุดได้มีโอกาสถูกเลือกมากขึ้น

ตัวอย่างที่ 10 การคัดเลือกแบบจัดอันดับจากชุดข้อมูล A จากตารางที่ 2.11 โดยเลือกโครโมโซมที่ดีที่สุด 13 อันดับแรกจากการใช้ฟังก์ชันความเหมาะสมคือ “โครโมโซมเลขฐานสองจำนวน 8 บิต ที่มีค่าสูงที่สุดเมื่อแปลงเป็นเลขฐาน 10 เป็นโครโมโซมที่เหมาะสมที่สุด” จะได้ลำดับของโครโมโซมแต่ละตัวเรียงตามค่าความเหมาะสมจำนวน 13 อันดับแรก แสดงดังตารางที่ 2.13

ตารางที่ 2.13 โครโมโซมที่ดีที่สุดเรียงตามค่าความเหมาะสมของแต่ละโครโมโซม

Chromosome	ยีนที่ 1	ยีนที่ 2	ยีนที่ 3	ยีนที่ 4	ยีนที่ 5	ยีนที่ 6	ยีนที่ 7	ยีนที่ 8	ค่าความเหมาะสม
5	1	1	1	0	1	1	1	0	238
3	1	1	0	1	1	0	1	1	219
14	1	1	0	1	0	0	1	1	211
16	1	1	0	0	1	1	1	0	206
6	1	1	0	0	1	0	0	1	201
18	1	0	1	1	0	1	1	1	183
19	1	0	1	0	1	0	1	1	171
8	1	0	0	1	1	0	1	0	154

ตารางที่ 2.13 โครโมโซมที่ดีที่สุดเรียงตามค่าความเหมาะสมของแต่ละโครโมโซม (ต่อ)

Chromosome	ยีนที่ 1	ยีนที่ 2	ยีนที่ 3	ยีนที่ 4	ยีนที่ 5	ยีนที่ 6	ยีนที่ 7	ยีนที่ 8	ค่าความเหมาะสม
17	0	1	1	1	1	0	0	1	121
20	0	1	1	1	0	1	1	0	118
11	0	1	1	0	1	1	1	0	110
1	0	1	0	1	1	0	1	0	90
12	0	1	0	1	0	1	0	1	85

จะเห็นได้ว่าโครโมโซมที่มีโอกาสถูกเลือกไปใช้เป็นโครโมโซมพ่อแม่มากที่สุดคือ โครโมโซมที่ 5, โครโมโซมที่ 3, โครโมโซมที่ 14, โครโมโซมที่ 16, โครโมโซมที่ 6, โครโมโซมที่ 18, โครโมโซมที่ 19, โครโมโซมที่ 8, โครโมโซมที่ 17, โครโมโซมที่ 20, โครโมโซมที่ 11, โครโมโซมที่ 1 และ 12 เนื่องจากมีค่าความเหมาะสมสูงที่สุดเรียงตามลำดับ

การคัดเลือกแบบจัดการแข่งขัน

การคัดเลือกแบบจัดการแข่งขัน เป็นวิธีที่ใช้ในการคัดเลือกโครโมโซมพ่อแม่ที่ดีที่สุด โดยอาศัยหลักการของการจัดการแข่งขันกีฬาโดยการสุ่มแบ่งกลุ่มคัดเลือกโครโมโซม แล้วเลือกเอาโครโมโซมที่ดีที่สุดในกลุ่มนั้น เพื่อหาโครโมโซมที่ดีที่สุด เป็นต้นกำเนิดสายพันธุ์ให้แก่ลูกหลาน โดยมีหลักการทำงานดังนี้

1) สุ่มเลือกโครโมโซม K ตัว สำหรับจัดการแข่งขันขนาด K (Tournament Size)
2) เลือกโครโมโซมที่มีค่าความเหมาะสมที่สุดจากการแข่งขัน

ตัวอย่างที่ 11 สมมติว่าต้องการเลือกโครโมโซมที่ดีที่สุด 1 โครโมโซมจากชุดข้อมูล A จากตารางที่ 2.11 โดยสุ่มเลือกขนาดประชากรเท่ากับ 10 แสดงดังตารางที่ 2.14

ตารางที่ 2.14 รายละเอียดประชากร 10 ประชากร จากการสุ่มเลือกจากชุดข้อมูล A จากตารางที่ 2.11 และค่าความเหมาะสมของแต่ละประชากร

Chromosome	ยีนที่ 1	ยีนที่ 2	ยีนที่ 3	ยีนที่ 4	ยีนที่ 5	ยีนที่ 6	ยีนที่ 7	ยีนที่ 8	ค่าความเหมาะสม
1	0	1	0	1	1	0	1	0	90
2	1	0	0	1	0	0	1	0	146
3	1	1	0	1	1	0	1	1	219
4	1	1	1	0	1	1	1	0	238
5	0	1	1	1	1	0	0	1	121
6	1	1	0	0	1	0	0	1	201
7	0	0	0	0	1	0	0	0	8
8	1	0	0	1	1	0	1	0	154
9	0	0	1	0	1	0	1	0	42
10	0	0	0	0	1	1	1	1	15

โดยในการแข่งขันรอบแรกจัดการแข่งขันโดย Chromosome 1 พบ Chromosome 2, Chromosome 3 พบ Chromosome 4, Chromosome 5 พบ Chromosome 6, Chromosome 7 พบ Chromosome 8 และ Chromosome 9 พบ Chromosome 10 โดยผลการจัดการแข่งขันแสดงการคัดเลือกโครโมโซมที่ดีที่สุดจากชุดข้อมูล A ในรอบการแข่งขันที่ 1 แสดงดังตารางที่ 2.15

ตารางที่ 2.15 ผลการแข่งขันการคัดเลือกโครโมโซมที่ดีที่สุดรอบที่ 1

Chromosome	ยีนที่ 1	ยีนที่ 2	ยีนที่ 3	ยีนที่ 4	ยีนที่ 5	ยีนที่ 6	ยีนที่ 7	ยีนที่ 8	ค่าความเหมาะสม
2	1	0	0	1	0	0	1	0	146
4	1	1	1	0	1	1	1	0	238
6	1	1	0	0	1	0	0	1	201
8	1	0	0	1	1	0	1	0	154
9	0	0	1	0	1	0	1	0	42

โดยในการแข่งขันรอบที่ 2 จัดการแข่งขันโดย Chromosome 2 พบ Chromosome 4, Chromosome 6 พบ Chromosome 8 และ Chromosome 9 ชนะบายเนื่องจากไม่มีคู่แข่งโดยผลการจัดการแข่งขันแสดงการคัดเลือกโครโมโซมที่ดีที่สุดจากชุดข้อมูล A ในรอบการแข่งขันที่ 2 แสดงดังตารางที่ 2.16

ตารางที่ 2.16 ผลการแข่งขันการคัดเลือกโครโมโซมที่ดีที่สุดรอบที่ 2

Chromosome	ยีนที่ 1	ยีนที่ 2	ยีนที่ 3	ยีนที่ 4	ยีนที่ 5	ยีนที่ 6	ยีนที่ 7	ยีนที่ 8	ค่าความเหมาะสม
4	1	1	1	0	1	1	1	0	238
6	1	1	0	0	1	0	0	1	201
9	0	0	1	0	1	0	1	0	42

โดยในการแข่งขันรอบที่ 3 จัดการแข่งขันโดย Chromosome 4 พบ Chromosome 6 และ Chromosome 9 ชนะบายเนื่องจากไม่มีคู่แข่งโดยผลการจัดการแข่งขันแสดงการคัดเลือกโครโมโซมที่ดีที่สุดจากชุดข้อมูล A ในรอบการแข่งขันที่ 3 แสดงดังตารางที่ 2.17

ตารางที่ 2.17 ผลการแข่งขันการคัดเลือกโครโมโซมที่ดีที่สุดรอบที่ 3

Chromosome	ยีนที่ 1	ยีนที่ 2	ยีนที่ 3	ยีนที่ 4	ยีนที่ 5	ยีนที่ 6	ยีนที่ 7	ยีนที่ 8	ค่าความเหมาะสม
4	1	1	1	0	1	1	1	0	238
9	0	0	1	0	1	0	1	0	42

โดยในการแข่งขันรอบสุดท้ายจัดการแข่งขันโดย Chromosome 4 พบ Chromosome 9 โดยผลการจัดการแข่งขันในรอบสุดท้ายแสดงดังตารางที่ 2.18

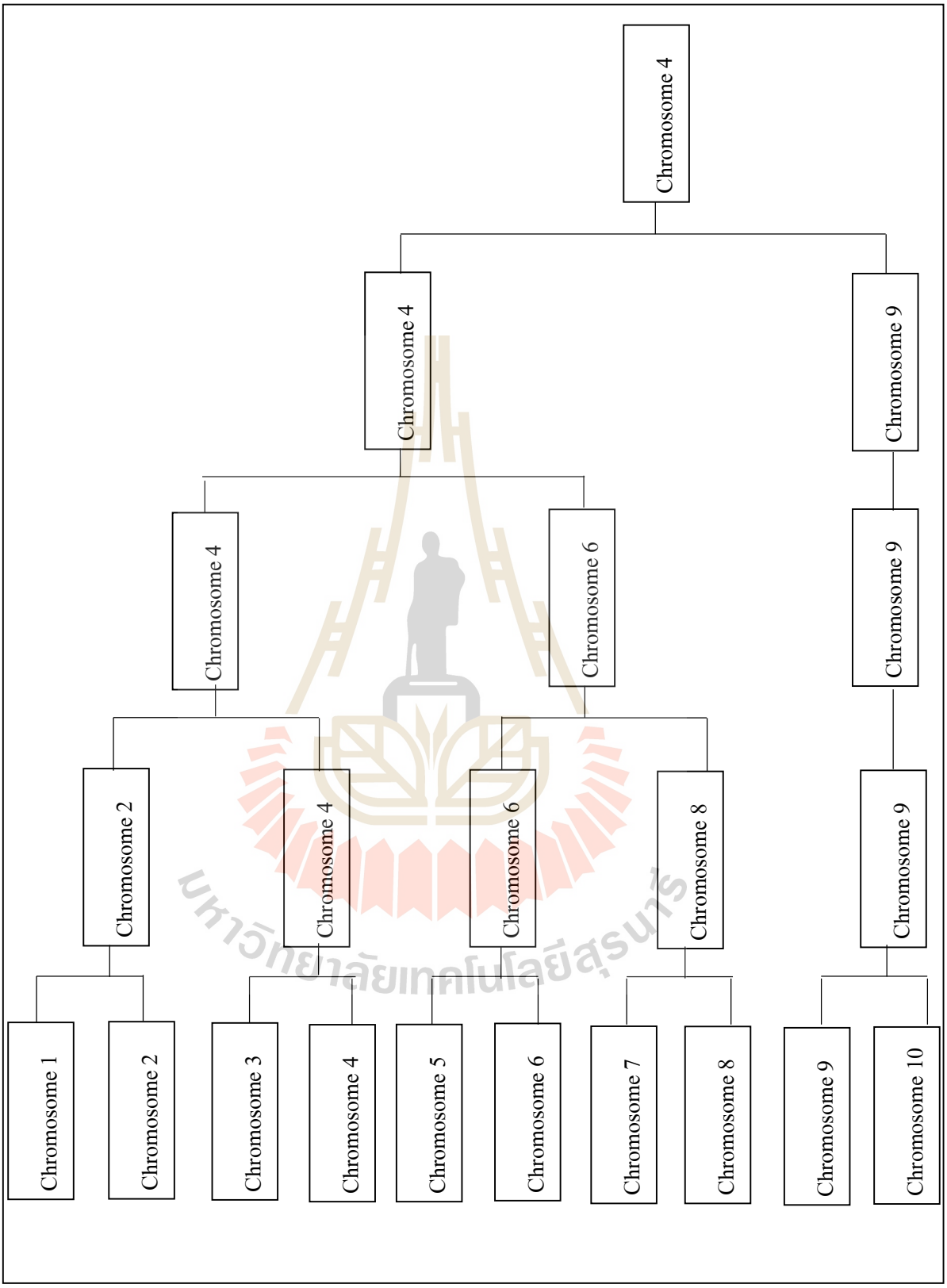
ตารางที่ 2.18 ผลการแข่งขันการคัดเลือกโครโมโซมที่ดีที่สุดรอบสุดท้าย

Chromosome	ยีนที่ 1	ยีนที่ 2	ยีนที่ 3	ยีนที่ 4	ยีนที่ 5	ยีนที่ 6	ยีนที่ 7	ยีนที่ 8	ค่าความเหมาะสม
4	1	1	1	0	1	1	1	0	238

ดังนั้นจะได้โครโมโซมที่ดีที่สุดได้แก่ โครโมโซมที่ 4, โครโมโซมที่ 9, โครโมโซมที่ 6, โครโมโซมที่ 8, โครโมโซมที่ 2, โครโมโซมที่ 3, โครโมโซมที่ 10, โครโมโซมที่ 5, โครโมโซมที่ 1 และโครโมโซมที่ 7 เรียงตามผลการจัดการแข่งขัน โดยผลการแข่งขันการคัดเลือกโครโมโซมทั้งหมดแสดงดังตารางที่ 2.19 และแสดงผลการแข่งขันทั้งหมดดังรูปที่ 2.12

ตารางที่ 2.19 ผลการแข่งขันการคัดเลือกโครโมโซมทั้งหมด

Chromosome	ยีนที่ 1	ยีนที่ 2	ยีนที่ 3	ยีนที่ 4	ยีนที่ 5	ยีนที่ 6	ยีนที่ 7	ยีนที่ 8	ค่าความเหมาะสม
4	1	1	1	0	1	1	1	0	238
9	0	0	1	0	1	0	1	0	42
6	1	1	0	0	1	0	0	1	201
8	1	0	0	1	1	0	1	0	154
2	1	0	0	1	0	0	1	0	146
3	1	1	0	1	1	0	1	1	219
10	0	0	0	0	1	1	1	1	15
5	0	1	1	1	1	0	0	1	121
1	0	1	0	1	1	0	1	0	90
7	0	0	0	0	1	0	0	0	8



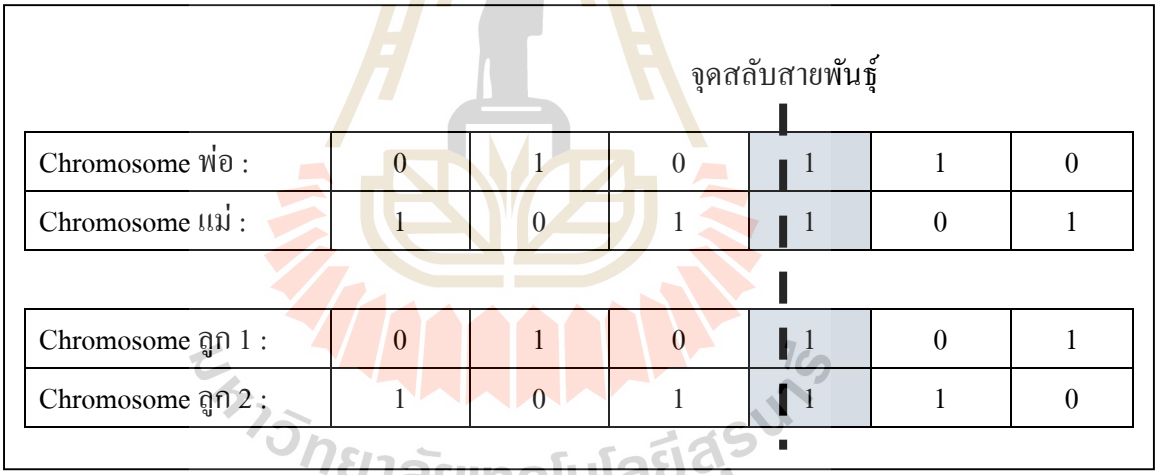
รูปที่ 2.12 ผลการแข่งขันการคัดเลือกแบบจัดการแข่งขัน

2) การสลับสายพันธุ์

การสลับสายพันธุ์ เป็นกระบวนการที่สำคัญของขั้นตอนวิธีเชิงพันธุกรรม โดยเมื่อมีการกลายพันธุ์เกิดขึ้นจะทำให้เกิดการเปลี่ยนแปลงของสิ่งมีชีวิตให้มีความหลากหลายมากยิ่งขึ้น ในขั้นตอนของการกลายพันธุ์จะนำสมาชิกของประชากรที่ผ่านการคัดเลือกมาเป็นคู่ ๆ โดยจะกำหนดให้เป็นสมาชิกรุ่นพ่อแม่ (Parent Individual) นำมาผสมกันเพื่อให้ได้โครโมโซมใหม่ที่เป็นรุ่นลูกขึ้นมา โดยการสุ่มเลือกสมาชิกรุ่นพ่อกับสมาชิกรุ่นแม่มาทำการสลับสายพันธุ์จะถูกกำหนดโดยความน่าจะเป็นในการสลับสายพันธุ์ (Crossover Probability) โดยเทคนิคการสลับสายพันธุ์สามารถทำได้หลายวิธี เช่น การสลับสายพันธุ์แบบจุดเดียว (Single-Point Crossover) การสลับสายพันธุ์แบบหลายจุด (Multiple-Point Crossover) เป็นต้น

การสลับสายพันธุ์แบบจุดเดียว

การสลับสายพันธุ์วิธีนี้ จะทำให้โครโมโซมลูกหลานมีสายพันธุ์ของแต่ละต้นกำเนิดอยู่อย่างละหนึ่งส่วน โดยจุดตัดในการสลับสายพันธุ์นั้นจะได้มาจากการสุ่มเลือกจุดสลับสายพันธุ์หนึ่งจุด แล้วหลังจากนั้นจะทำการผสมยีนระหว่างโครโมโซมพ่อและโครโมโซมแม่แสดงดังรูปที่ 2.13

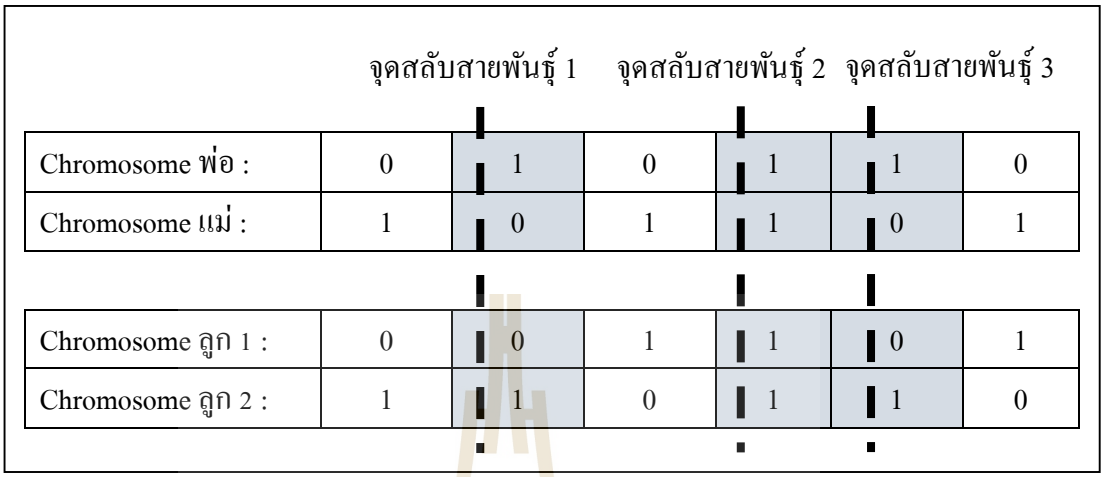


รูปที่ 2.13 วิธีการสลับสายพันธุ์แบบจุดเดียว

การสลับสายพันธุ์แบบหลายจุด

การสลับสายพันธุ์วิธีนี้ จะทำให้โครโมโซมลูกหลานจะมีสายพันธุ์ของต้นกำเนิดมากกว่าหนึ่งส่วน โดยมีหลักการเลือกจุดสลับสายพันธุ์มีอยู่หลายแบบ แต่ละแบบจะส่งผลต่อการเปลี่ยนแปลงโครโมโซมของลูกหลานที่แตกต่างกันออกไปด้วย โดยวิธีที่ง่ายที่สุดที่นิยมใช้ในการเลือกจุดสลับสายพันธุ์คือวิธีการสุ่มจุดสลับสายพันธุ์ การสลับสายพันธุ์แบบหลายจุดจะสามารถทำ

ให้เกิดการเปลี่ยนแปลงของโครโมโซมลูกหลานได้มากกว่าการสลับสายพันธุ์แบบจุดเดียว ดังรูปที่ 2.14



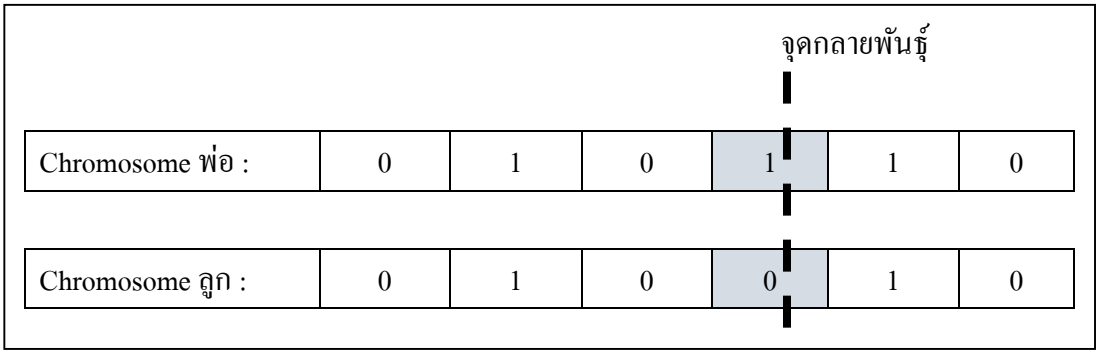
รูปที่ 2.14 การสลับสายพันธุ์แบบ 3 จุด

3) การกลายพันธุ์

การกลายพันธุ์ เป็นวิธีการแปรผันยีนบางตำแหน่ง หรืออาจจะทุกตำแหน่งที่อยู่ในโครโมโซม ซึ่งมีวัตถุประสงค์เพื่อทำให้ค่าของโครโมโซมที่มีอยู่เดิมเกิดการเปลี่ยนแปลง โดยปกติแล้วการกลายพันธุ์จะทำการสุ่มตำแหน่งที่ต้องการกลายพันธุ์จากความน่าจะเป็นในการกลายพันธุ์ (Probability of Mutation) ซึ่งโดยทั่วไปจะมีความน่าจะเป็นในการกลายพันธุ์มีค่าน้อย โดยจะอยู่ระหว่าง 0 ถึง 0.1 โดยเทคนิคการกลายพันธุ์ที่ได้รับความนิยมได้แก่ การกลายพันธุ์แบบกลับบิต (Bit-Flipped Mutation) การกลายพันธุ์แบบผกผัน (Inversion Mutation)

การกลายพันธุ์แบบกลับบิต

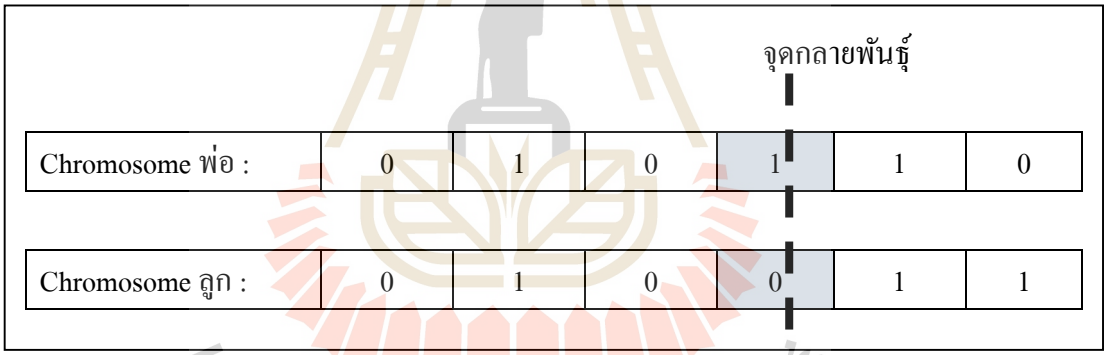
การกลายพันธุ์แบบกลับบิต เป็นการกลายพันธุ์ที่ใช้ในการเข้ารหัสแบบเลขฐานสอง โดยสามารถทำได้โดยการกลับค่าบิตเป็นค่าตรงกันข้ามจากค่าเดิม (Complement) เช่น จากเดิมบิตมีค่าเป็น 0 เมื่อเกิดการกลายพันธุ์แบบกลับบิต ที่ตำแหน่งนั้นบิตจะมีค่าเป็น 1 ดังรูปที่ 2.15



รูปที่ 2.15 การกลายพันธุ์แบบกลับบิต

การกลายพันธุ์แบบผกผัน

การกลายพันธุ์แบบผกผัน เป็นการสลับตำแหน่งยีนหลังจุดกลายพันธุ์แบบหลังไปหน้า โดยจะทำการสุ่มเลือกโครโมโซมรุ่นพ่อแม่ขึ้นมาหนึ่งโครโมโซม หลังจากนั้นจะทำการสุ่มเลือกจุดกลายพันธุ์ และทำการสลับตำแหน่งของยีนหลังจุดกลายพันธุ์ แสดงดังรูปที่ 2.16



รูปที่ 2.16 การกลายพันธุ์แบบผกผัน

5. การแทนที่

การแทนที่ เป็นขั้นตอนหลังจากที่ขั้นตอนทางพันธุกรรมได้โครโมโซมรุ่นลูกหลานเรียบร้อยแล้ว โดยจะนำโครโมโซมลูกหลานใหม่นี้ไปแทนที่ในประชากรรุ่นเก่า โดยมีวัตถุประสงค์ในการแทนที่ประชากรรุ่นเก่าเพื่อทำให้ประชากรรุ่นใหม่เป็นโครโมโซมที่ดีกว่าเพราะได้สายพันธุ์ที่ดีจากต้นกำเนิดสายพันธุ์ โดยจะทำให้โครโมโซมรุ่นใหม่ประกอบไปด้วยโครโมโซมใหม่ ๆ ที่สืบสายพันธุ์มาจากโครโมโซมรุ่นพ่อแม่ที่ผ่านการคัดเลือกแล้ว โดยการแทนที่ประชากรสามารถทำได้ 2 วิธี ได้แก่ การแทนที่ประชากรทั้งรุ่น (Generational Genetic Algorithm) และการแทนที่ประชากรบางส่วน (Partial Genetic Algorithm)

การแทนที่ประชากรทั้งรุ่น

การแทนที่ประชากรทั้งรุ่น เป็นวิธีการแทนที่ประชากรที่ง่าย เนื่องมาจากไม่จำเป็นที่จะต้องคัดเลือกว่าประชากรรุ่นพ่อแม่ส่วนไหนจะถูกแทนที่ด้วยประชากรรุ่นลูกหลาน เป็นการนำเอาโครโมโซม หรือประชากรรุ่นลูกหลานไปแทนที่ประชากรรุ่นพ่อแม่ทั้งหมด ดังนั้นถ้าในระบบหนึ่งมีจำนวนประชากรเท่ากับ N ประชากร จำนวนโครโมโซมของรุ่นลูกหลานที่จะนำมาแทนที่ต้องมีจำนวน N ประชากรเช่นเดียวกัน ซึ่งการแทนที่ประชากรทั้งรุ่นส่งผลให้โครโมโซมที่ดีในรุ่นพ่อแม่ถูกแทนที่ไปด้วยโครโมโซมรุ่นลูกหลานด้วย แต่สามารถแก้ไขได้โดยการคัดเลือกเก็บโครโมโซมที่ดีที่สุด 2-3 ตัวแรกเอาไว้โดยอาจจะใช้วิธีการคัดเลือกหัวกะทิ (Elitist Strategy) โดยหลักการคือหากไม่มีโครโมโซมใหม่ที่ดีกว่าเกิดขึ้น โครโมโซมที่ดีที่สุดจากรุ่นพ่อแม่จะถูกเก็บไว้อยู่ต่อไป

การแทนที่ประชากรบางส่วน

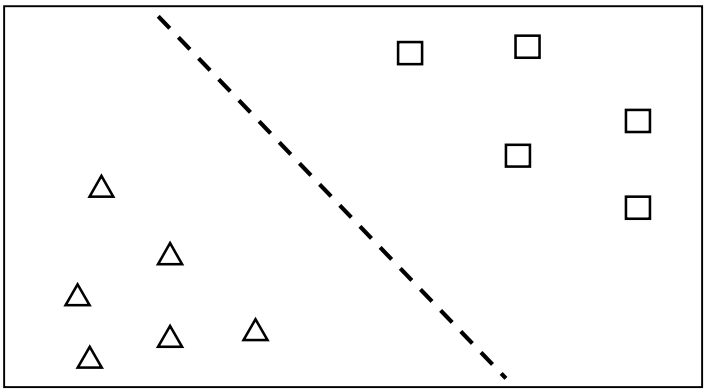
การแทนที่ประชากรบางส่วน เป็นวิธีการแทนที่ประชากรเดิมเพียงบางส่วนเท่านั้น โดยจะทำการคัดเลือกประชากรรุ่นพ่อแม่ที่จะถูกแทนที่ด้วยประชากรรุ่นลูกหลานที่พิจารณาจากค่าความเหมาะสมของโครโมโซม การแทนที่ประชากรบางส่วนจะทำให้ประชากรรุ่นพ่อแม่ถูกแทนที่ด้วยโครโมโซมรุ่นลูกหลานเพียง 1 หรือ 2 ตัวเท่านั้น โดยมีวิธีในการแทนที่ประชากรอยู่หลายวิธี แต่วิธีที่ได้รับความนิยมและง่ายที่สุดคือการแทนที่ประชากรรุ่นพ่อแม่ที่ด้อยที่สุด หรือการแทนที่ประชากรโดยการสุ่ม

6. การตรวจสอบเงื่อนไขสิ้นสุดการทำงาน

การตรวจสอบเงื่อนไขสิ้นสุดการทำงาน เป็นขั้นตอนของการตรวจสอบว่าจบกระบวนการทางพันธุกรรมแล้วหรือยัง ซึ่งการทำงานของขั้นตอนวิธีเชิงพันธุกรรมจะทำงานวนเวียนเป็นวัฏจักรหมุนเวียนอยู่เช่นเดิม จนกระทั่งถึงจุดหนึ่งตามเงื่อนไขที่กำหนดไว้ เช่น ได้จำนวนประชากรที่มีค่าความเหมาะสมตามจำนวนที่กำหนด หรือพบคำตอบที่ดีที่สุดตามเกณฑ์ที่ตั้งเป้าไว้เรียบร้อยแล้ว เป็นต้น ซึ่งหากยังไม่เข้าเงื่อนไขสิ้นสุดการทำงาน ขั้นตอนวิธีเชิงพันธุกรรมก็จะทำการกลับไปทำงานวนเวียนจนกว่าจะเป็นไปตามเงื่อนไขสิ้นสุดการทำงาน

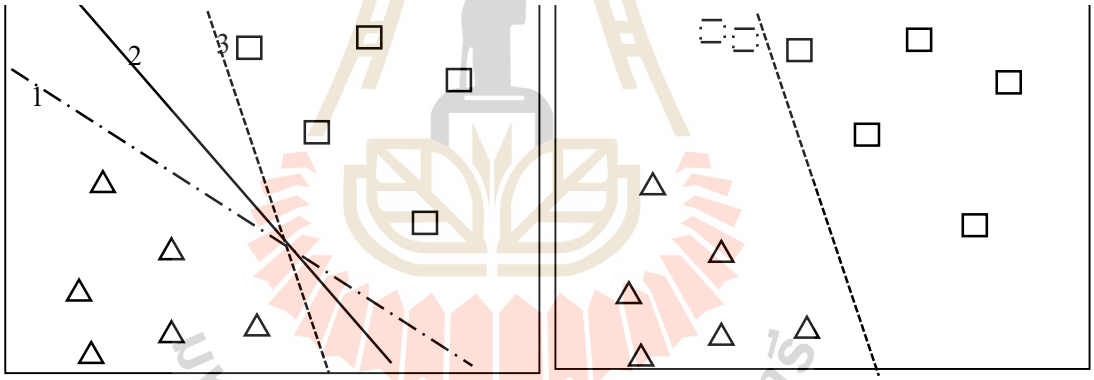
2.4 การจำแนกข้อมูลด้วยอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีน

ซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine) (Cortes and Vapnik, 1995) เป็นอัลกอริทึมที่ใช้ในการจำแนกประเภทข้อมูลในแต่ละคลาสที่ได้รับความนิยมมาก เนื่องจากความสามารถในการจำแนกประเภทข้อมูลในแต่ละคลาสมีความแม่นยำสูง โดยหลักการสำคัญของอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนคือการสร้างเส้นแบ่ง (Hyperplane) เพื่อแบ่งแยกประเภทข้อมูลออกจากกัน แสดงดังรูปที่ 2.17



รูปที่ 2.17 เส้นแบ่ง (Hyperplane) เพื่อแบ่งแยกข้อมูลออกเป็น 2 กลุ่ม

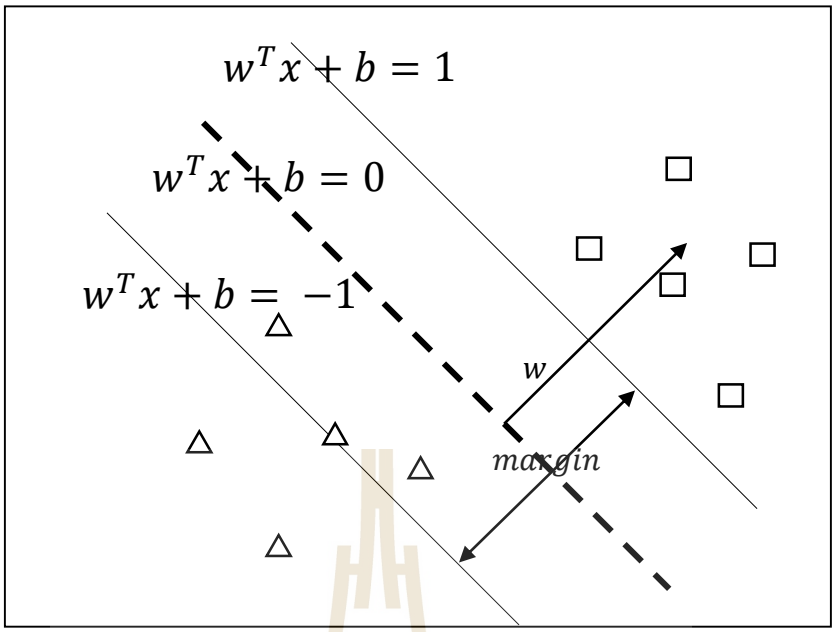
การสร้างเส้นแบ่งสำหรับอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีน สามารถสร้างเส้นแบ่งได้มากกว่าหนึ่งเส้น แสดงดังรูปที่ 2.18 (ก) โดยจะเห็นว่าเมื่อทำการเลือกเส้นแบ่งเส้นที่ 3 ไปใช้ในการจำแนกประเภทข้อมูลในอนาคต ยังมีความผิดพลาดเกิดขึ้นแสดงดังรูปที่ 2.18(ข)



(ก) เส้นแบ่งที่เป็นไปได้ (ข) จำแนกข้อมูลด้วยเส้นแบ่งที่ 3

รูปที่ 2.18 เส้นแบ่งที่เป็นไปได้สำหรับการจำแนกข้อมูล

การเลือกเส้นแบ่งที่เหมาะสมสำหรับอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีน คือการสร้างเส้นแบ่งที่มีขนาดความกว้างของขอบ (Margin) ระหว่างเส้นแบ่งไปยังจุดข้อมูลในแต่ละคลาสมากที่สุด เพื่อเพิ่มประสิทธิภาพในการจำแนกประเภทข้อมูลที่ยังไม่ทราบคลาสของข้อมูล แสดงดังรูปที่ 2.19



รูปที่ 2.19 เส้นแบ่งข้อมูลที่มีระยะห่างระหว่างข้อมูลมากที่สุด

จากรูปที่ 2.19 เส้นแบ่งนั้นจะทำการแบ่งข้อมูลทั้งสองคลาสออกจากกันด้วยระยะห่างระหว่างข้อมูลทั้งสองคลาสมากที่สุด และมีเวกเตอร์ถ่วงน้ำหนัก w (Weight Vector) เป็นตัวกำหนดทิศทางและความเอียงของไฮเปอร์เพลน ซึ่งเวกเตอร์ w จะตั้งฉากกับเส้นแบ่ง และข้อมูลจะถูกแปลงให้อยู่ในรูปแบบเวกเตอร์ x ส่วน y จะเป็นตัวกำหนดว่าข้อมูลจุดนั้นจะเป็นคลาส 1 หรือคลาส -1 โดยสามารถเขียนเป็นสมการได้ดังสมการที่ 2.10

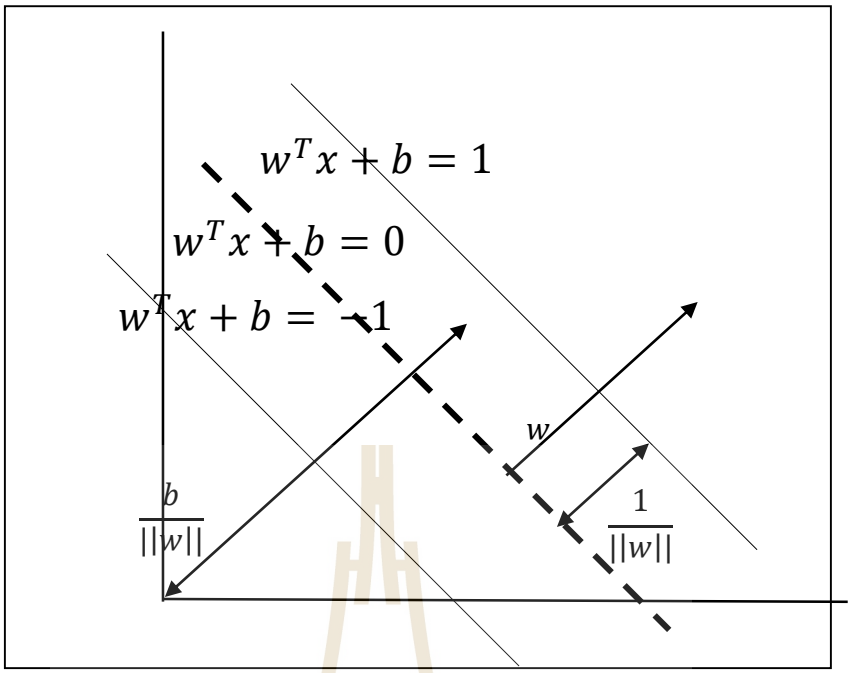
$$w^T x + b \geq 1, when\ y_i = +1$$
$$w^T x + b \leq 1, when y_i = -1 \quad (2.10)$$

เมื่อ

w คือเวกเตอร์ถ่วงน้ำหนัก (Weight Vector)

b คือค่าไบแอส (Bias)

การหาค่าเวกเตอร์ถ่วงน้ำหนัก สามารถหาได้จากความชันของเส้นแบ่งที่สร้างขึ้น นั่นคือเวกเตอร์ถ่วงน้ำหนักคือเส้นที่ลากไปตั้งฉากกับเส้นแบ่ง และค่าไบแอสจะเป็นตัวกำหนดระยะห่างระหว่างเส้นแบ่งกับจุดกำเนิด (Origin) แสดงดังรูปที่ 2.20



รูปที่ 2.20 เวกเตอร์ถ่วงน้ำหนัก และค่าไบแอส

เมื่อพิจารณาข้อมูล 2 มิติ โดยที่เส้นแบ่งเป็นเส้นตรง และกำหนดให้จุดทุกจุด $X = (X_1, X_2)^T$ จะได้สมการของเส้นแบ่ง ดังสมการที่ 2.11

$$h(x) = w^T x + b = w_1 x_1 + w_2 x_2 + b = 0$$

$$x_2 = -\frac{w_1}{w_2} x_1 - \frac{b}{w_2} \tag{2.11}$$

เมื่อพิจารณาจุด 2 จุดบนเส้นแบ่ง ได้แก่จุด $A = (A_1, A_2) = (2, 0)$ และจุด $B = (B_1, B_2) = (4, 6)$ และเวกเตอร์ถ่วงน้ำหนักหมายถึงความชันของเส้นแบ่ง สามารถคำนวณได้จากสมการที่ 2.12

$$weight\ vector = -\frac{w_1}{w_2} = -\frac{(B_2 - A_2)}{(B_1 - A_1)} = -\frac{(6-0)}{(4-2)} = -\frac{6}{2} \tag{2.12}$$

จะได้ $w_1 = 6$ และ $w_2 = 2$ สำหรับจุด (2, 0) บนเส้นแบ่ง และสามารถคำนวณหาค่าไบแอสได้ดังสมการที่ 2.13

$$b = -6x_1 - 2x_2 = -6(2) - 2(0) = -12 \tag{2.13}$$

ซึ่งสามารถคำนวณหาความกว้างของเส้นขอบ (Margin) ได้จากสมการที่ 2.14

$$margin = \frac{2}{||w||} = \frac{2}{\sqrt{40}} \tag{2.14}$$

โดยที่ขนาดของ w สามารถคำนวณได้จากสมการที่ 2.15

$$||w|| = \sqrt{w_1^2 + w_2^2} = \sqrt{6^2 + 2^2} = \sqrt{36 + 4} = \sqrt{40} \tag{2.15}$$

ในบางข้อมูลการใช้วิธีการจำแนกประเภทข้อมูลด้วยอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนแบบใช้เคอร์เนลเส้นตรง (Linear Kernel) นั้นไม่สามารถที่จะจำแนกข้อมูลได้ จึงมีการพัฒนาเคอร์เนล (Chistianini and Shawe-Taylor, 2000; Muller et al, 2001) เพื่อใช้ร่วมกับอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนสำหรับจำแนกประเภทข้อมูลที่ไม่สามารถจำแนกด้วยเส้นตรงได้ โดยเคอร์เนลแต่ละเคอร์เนลจะเป็นการประยุกต์ใช้สมการต่าง ๆ เพื่อสร้างเส้นแบ่งในรูปแบบอื่นที่ไม่ใช่เส้นตรง โดยเคอร์เนลที่นิยมใช้ในปัจจุบัน แสดงดังตารางที่ 2.20

ตารางที่ 2.20 เคอร์เนลฟังก์ชันสำหรับอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีน

Kernel	Inner Product Kernel
Linear	$x^T x_i$
Polynomial	$(x^T x_i + n)^d$
Radial-basis function	$\exp(-\gamma\|x - x_i\|^2), \gamma > 0$
Sigmoidal	$\tanh(\gamma(x^T . x_i + \eta)), \gamma > 0$

สำหรับการจำแนกคลาสของข้อมูลด้วยอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีน จะประกอบไปด้วยพารามิเตอร์ต่าง ๆ มากมาย เพื่อนำไปตั้งค่าให้อัลกอริทึมนั้นมีความสามารถในการจำแนกประเภทข้อมูลได้มีประสิทธิภาพมากยิ่งขึ้น โดยพารามิเตอร์ที่นิยมปรับค่าเพื่อเพิ่มประสิทธิภาพให้กับอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีน ได้แก่ พารามิเตอร์ C, พารามิเตอร์ Epsilon และพารามิเตอร์ Gamma (สำหรับจำแนกด้วยเคอร์เนลเรเดียลเบสิสฟังก์ชัน)

พารามิเตอร์ C ทำหน้าที่เป็นตัวควบคุมค่าใช้จ่ายในการจำแนกข้อมูลผิดคลาสในชุดข้อมูลฝึกสอน เนื่องจากอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนจะทำการหาไฮเปอร์เพลนที่มีความกว้างของขอบระหว่างแต่ละคลาสมากที่สุด โดยจะเกิดการแลกเปลี่ยนระหว่างการหาขอบที่มีระยะห่างน้อยแต่ไม่มีความผิดพลาดในการจำแนกข้อมูลเลย หรืออาจจะมีการจำแนกข้อมูลผิดพลาดเพียงเล็กน้อยกับการหาขอบที่มีระยะห่างของขอบกว้างมากแต่แลกกับการจำแนกข้อมูลผิดพลาดมากขึ้น โดยหากค่าพารามิเตอร์ C มีค่าน้อยจะทำให้ความกว้างของขอบมีขนาดใหญ่ ทำให้โมเดลมีความยืดหยุ่นต่อข้อมูลในอนาคตสูง และหากค่าพารามิเตอร์ C มีค่ามากจะส่งผลให้ความกว้างของขอบมีขนาดเล็ก อาจจะทำให้เกิดปัญหา Over Fitting ได้

พารามิเตอร์ Epsilon ทำหน้าที่ควบคุมการเพิ่มประสิทธิภาพในการจำแนกข้อมูล โดยจะช่วยในการลู่เข้าหาค่าไฮเปอร์เพลนที่ดีที่สุดได้ดียิ่งขึ้น เช่น สำหรับการทำงานในแต่ละรอบเพื่อจำแนกข้อมูล ค่าใช้จ่าย (Cost) หรือที่เรียกว่า Loss จะมีการคำนวณค่าใช้จ่ายจากการจำแนกผิดคลาสเกิดขึ้น สำหรับการทำงานในรอบถัดไป ไฮเปอร์เพลนจะมีการปรับปรุงตามข้อผิดพลาดที่พบในรอบที่ผ่านมา จนกระทั่งได้ไฮเปอร์เพลนที่มีระยะห่างตามที่กำหนด หรือดีที่สุด การกำหนดค่าพารามิเตอร์ Epsilon จะช่วยให้ลู่เข้าหาคำตอบที่ทำให้ได้ไฮเปอร์เพลนที่ดีที่สุดได้เร็วขึ้น

พารามิเตอร์ Gamma เป็นพารามิเตอร์สำหรับ Kernel Radial Basis Function (RBF Kernel เป็น Kernel ที่ใช้จำแนกข้อมูลที่มีลักษณะไม่เชิงเส้น) สำหรับการจำแนกในสองมิติที่ไม่สามารถแบ่งแยกข้อมูลออกเป็นสองคลาสได้ โดยจะใช้การแก้ปัญหาไม่เชิงเส้น (Non-Linear) เข้ามาจำแนกข้อมูลออกเป็นสองคลาส โดยพารามิเตอร์ Gamma จะทำหน้าที่ควบคุมความโค้งของเส้นไฮเปอร์เพลน หากค่าพารามิเตอร์ Gamma มีค่าน้อยส่งผลให้ความกว้างของขอบมีขนาดกว้างขึ้นทำให้มีความยืดหยุ่นในการจำแนกข้อมูลได้ แต่หากค่าพารามิเตอร์ Gamma มีค่าสูงจะส่งผลให้ความกว้างของขอบมีขนาดแคบทำให้อาจจะเกิดปัญหา Over Fitting ได้

2.5 งานวิจัยที่เกี่ยวข้องกับการจำแนกข้อมูลไม่สมดุล และหาค่าพารามิเตอร์ที่เหมาะสมด้วยเทคนิคต่าง ๆ

Krawczyk et al. (2014) ได้เสนอวิธีการจำแนกข้อมูลไม่สมดุลโดยการใช้วิธีการเรียนรู้ร่วมกันด้วยอัลกอริทึมต้นไม้ตัดสินใจ (Decision Tree) ร่วมกับการเรียนรู้แบบมีค่าใช้จ่ายเพื่อจำแนกข้อมูลไม่สมดุล ในงานวิจัยนี้ได้เสนออัลกอริทึมสำหรับการปรับปรุงวิธีการเรียนรู้ร่วมกันด้วยอัลกอริทึมต้นไม้ตัดสินใจ โดยการกำหนดตารางค่าใช้จ่าย (Cost) จากค่าที่ได้จากการวิเคราะห์พื้นที่ใต้กราฟ ROC โดยทำการทดลองกับชุดข้อมูล 6 ชุดข้อมูลจาก Keel และใช้มาตรวัด 2 ชนิด ได้แก่ Sensitivity และ Specificity โดยทำการเปรียบเทียบประสิทธิภาพกับการจำแนกด้วยอัลกอริทึม SingleCTTree, MCS, SMOTEBagging, SMOTEBoost, IIvotes และ EasyEnsemble โดยผลการทดลองที่ได้ปรากฏว่าอัลกอริทึมที่นำเสนอให้ประสิทธิภาพในการจำแนกที่ดีกว่าอัลกอริทึมอื่น

Liao et al. (2014) ได้นำเสนอวิธีการจำแนกข้อมูลไม่สมดุลโดยการนำอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine) มาใช้ในขั้นตอนก่อนการประมวลผลเพื่อจัดการกับข้อมูลที่ไม่สมดุลให้มีความสมดุลมากขึ้น หลังจากนั้นทำการคัดเลือกคุณลักษณะ (Feature Selection) และนำคุณลักษณะที่ได้มาทำการคัดเลือกเข้าสู่กระบวนการการเรียนรู้ร่วมกันด้วยอัลกอริทึมโครงข่ายประสาทเทียมแบบย้อนกลับ (Back-Propagation Neural Network :BPNN)โดยมีขั้นตอนดำเนินงานวิจัยในขั้นตอนแรกจะทำการตรวจสอบข้อมูล หรือเป็นการเตรียมข้อมูล กำจัดค่าสูญหาย (Missing Values) หลังจากนั้นจะปรับขนาดของข้อมูลด้วยการใช้ซัพพอร์ตเวกเตอร์แมชชีน หลังจากนั้นจะทำการแบ่งข้อมูลออกเป็น 2 ชุดข้อมูล ได้แก่ ชุดข้อมูลเรียนรู้และชุดข้อมูลทดสอบ สำหรับชุดข้อมูลเรียนรู้จะแบ่งออกเป็น 4 ส่วนสำหรับการสร้างโมเดลการจำแนกด้วยโครงข่ายประสาทเทียมแบบย้อนกลับ 4 โมเดล หลังจากนั้นทำการโหวตผลลัพธ์จากแต่ละโมเดลเพื่อจำแนกข้อมูลทดสอบ และนำไปสร้างฐานความรู้ (Knowledge Base) โดยการใช้ทฤษฎีเซต

อย่างหยาบ (Rough Set) และนำฐานความรู้ที่ได้ไปสร้างกฎ และทำการลดกฎเพื่อให้มีจำนวนกฎน้อย ๆ แต่ครอบคลุม แล้วนำกฎที่ได้ไปประยุกต์ใช้งาน โดยงานวิจัยนี้ได้ทำการทดลองกับข้อมูลของบริษัทที่จดทะเบียนในตลาดหลักทรัพย์ ประกอบไปด้วย 63 บริษัทที่เกิดวิกฤตทางการเงิน และ 2,680 บริษัทที่ไม่เกิดวิกฤตทางการเงิน ผลที่ได้จากการทดลองปรากฏว่าอัลกอริทึมที่นำเสนอมีประสิทธิภาพในการจำแนกข้อมูลและมีความแม่นยำมากกว่าวิธีอื่น ๆ

Cateni et al. (2014) ได้นำเสนอวิธีการสำหรับแก้ปัญหาข้อมูลไม่สมดุล โดยการนำสองเทคนิคการสุ่มเลือกข้อมูล ได้แก่ วิธีสุ่มเกิน และวิธีสุ่มลดมาทำงานร่วมกัน โดยนำไปประยุกต์ใช้กับ 4 อัลกอริทึมสำหรับจำแนกประเภทข้อมูล ได้แก่ ซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine) ต้นไม้ตัดสินใจ (Decision Tree) SOM (Self-Organizing Map) และ Bayesian Classifiers โดยมีขั้นตอนดำเนินงานวิจัยในขั้นตอนแรกจะนำข้อมูลไม่สมดุลมาแบ่งแยกออกเป็น 2 ชุดข้อมูล ได้แก่ชุดข้อมูลเรียนรู้จำนวน 75% จากข้อมูลทั้งหมด และชุดข้อมูลสำหรับทดสอบจำนวน 25% จากข้อมูลทั้งหมด โดยในชุดข้อมูลเรียนรู้จะทำการสุ่มเกินข้อมูลจากคลาสส่วนน้อย และทำการสุ่มลดข้อมูลจากคลาสส่วนมากลง และนำชุดข้อมูลเรียนรู้ชุดใหม่ไปสร้างโมเดลในการจำแนกด้วยตัวจำแนก Bayesian (Bayesian Classifier) หลังจากนั้นประเมินประสิทธิภาพโมเดลด้วยการใช้ข้อมูลทดสอบ โดยงานวิจัยนี้ได้ทำการทดลองกับข้อมูล 4 ชุดข้อมูล ประกอบไปด้วย ชุดข้อมูลสังเคราะห์ ชุดข้อมูลมะเร็งเต้านม (UCI Wisconsin) และสองชุดข้อมูลจากอุตสาหกรรมโลหะ โดยทำการแบ่งข้อมูลออกเป็นชุดฝึกสอน 75% และชุดทดสอบประสิทธิภาพ 25% ผลลัพธ์ที่ได้พบว่าวิธีการที่นำเสนอสามารถจำแนกข้อมูลไม่สมดุลได้อย่างมีประสิทธิภาพ

Zhang and Li (2014) ได้นำเสนอเทคนิคการสุ่มเพิ่มข้อมูลจากคลาสส่วนน้อยที่มีชื่อว่า Random Walk Over-Sampling (RWO-Sampling) เพื่อลดปัญหาข้อมูลไม่สมดุล โดยการสุ่มเพิ่มข้อมูลด้วยเทคนิคนี้จะไม่ทำให้ค่าเบี่ยงเบนมาตรฐาน และค่าเฉลี่ยของข้อมูลเดิมเปลี่ยนแปลง ทำให้ข้อมูลเดิมไม่ได้รับผลกระทบจากการสุ่มข้อมูลเพิ่ม โดยในการดำเนินงานวิจัยจะทำการนำข้อมูลไม่สมดุลมาสุ่มข้อมูลจากคลาสส่วนน้อยเพิ่ม เพื่อทำให้ข้อมูลมีความสมดุล หลังจากนั้นจะแบ่งข้อมูลออกเป็น 10 ก้อน (10 Fold) โดยทำการจำแนกด้วยต้นไม้ตัดสินใจ ซัพพอร์ตเวกเตอร์แมชชีน และนาอีฟเบย์ ดำเนินการทดสอบประสิทธิภาพของโมเดลด้วยวิธีการ 10 Fold Cross Validation โดยใช้ชุดข้อมูลมาตรฐานจากฐานข้อมูล UCI โดยผลลัพธ์ที่ได้พบว่าวิธีการที่นำเสนอสามารถจำแนกข้อมูลไม่สมดุลได้ดียิ่งขึ้น

Yin et al. (2011) ได้นำเสนอวิธีการปรับค่าพารามิเตอร์ที่เหมาะสมสำหรับกระบวนการฉีดพลาสติกด้วยการใช้เทคนิค Back Propagation Neural Network ร่วมกับ Genetic Algorithm โดยการทดลองจะทำการทดลองด้วยการจำลองสถานการณ์ของกระบวนการฉีดพลาสติกโดยมีเป้าหมายเพื่อหาพารามิเตอร์ที่เหมาะสมเพิ่มประสิทธิภาพให้กับขั้นตอนการผลิตด้วยการประยุกต์ใช้ขั้นตอนวิธีเชิงพันธุกรรมเข้ามาเพิ่มประสิทธิภาพของกระบวนการผลิตด้วยการหาค่าพารามิเตอร์ที่เหมาะสมโดยวัดจากความโก่งงอ รวมไปถึงแรงยึดระหว่างการฉีดพลาสติกจะเป็นตัวชี้วัดว่าพารามิเตอร์นั้นดีหรือไม่โดยใช้พารามิเตอร์ อุณหภูมิแม่พิมพ์ อุณหภูมิที่ละลาย ความดัน และเวลาบรรจุ มาเป็นตัว

กำหนดการทำงานของขั้นตอนการผลิต ผลลัพธ์ที่ได้พบว่าวิธีการที่นำเสนอสามารถปรับพารามิเตอร์ของทั้งกระบวนการการผลิตได้อย่างถูกต้องและมีประสิทธิภาพสูงกว่าวิธีอื่น

Chen et al. (2014) ได้เสนอวิธีการใหม่สำหรับจำแนกโรคมะเร็งที่ชื่อว่า PSOC4.5 (Particle Swarm Optimization with C4.5) โดยประยุกต์นำเทคนิค Particle Swarm Optimization มาใช้งานร่วมกับการจำแนกด้วยต้นไม้ตัดสินใจเพื่อนำไปคัดเลือกยีนโดยในการดำเนินงานวิจัยจะทำการสุ่มสร้างประชากรเริ่มต้น หลังจากนั้นจะคำนวณค่าความเหมาะสมของแต่ละประชากรด้วยค่าความแม่นยำในการจำแนกจากการใช้ต้นไม้ตัดสินใจ C4.5 เป็นตัวจำแนกแล้วนำไปแทนที่ประชากรรุ่นเก่า จนกระทั่งได้ประชากรที่ดีที่สุด หรือครบรอบตามที่กำหนด โดยการทดลองจะนำไปประยุกต์ใช้กับข้อมูลโรคมะเร็งต่าง ๆ จำนวน 13 ข้อมูล โดยทำการเปรียบเทียบกับอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีน Self-Organizing Map, BPNN (Back Propagation Neural Network) และต้นไม้ตัดสินใจ C4.5 แบบปกติ โดยผลลัพธ์ที่ได้พบว่าวิธีการที่นำเสนอมีความสามารถในการจำแนกได้แม่นยำกว่าเทคนิคอื่น ๆ

Dao et al. (2016) ได้นำเสนอแนวคิดสำหรับเพิ่มประสิทธิภาพให้ขั้นตอนวิธีเชิงพันธุกรรมด้วยการประยุกต์นำเทคนิคการเริ่มต้นใหม่ (Restart) เข้ามาช่วยในการสร้างประชากรเริ่มต้นให้มีประสิทธิภาพมากยิ่งขึ้น โดยวิธีที่นำเสนอจะตรวจสอบว่าประชากรรุ่นใหม่ที่สร้างขึ้นมีความเหมาะสมมากขึ้นหรือน้อยลงจากประชากรรุ่นก่อนหน้า หากประชากรรุ่นใหม่มีความเหมาะสมน้อยกว่าประชากรรุ่นเก่าติดต่อกันตามจำนวนที่กำหนดจะทำการสร้างประชากรเริ่มต้นใหม่โดยการนำประชากรรุ่นเก่าที่มีความเหมาะสมตามจำนวนที่กำหนดไปเป็นประชากรเริ่มต้นของขั้นตอนวิธีเชิงพันธุกรรมด้วย โดยประเมินประสิทธิภาพด้วยการวิเคราะห์ความแปรปรวน (ANOVA Test) และเวลาที่ใช้ในการประมวลผล โดยการทดลองจะทำการทดลองกับข้อมูลสังเคราะห์ที่มีจำนวนประชากร ความน่าจะเป็นในการสลับสายพันธุ์ ความน่าจะเป็นในการกลายพันธุ์ จำนวนรุ่นที่ประชากรไม่ดีขึ้นที่แตกต่างกัน จำนวนประชากรที่นำไปเป็นต้นกำเนิดที่แตกต่างกัน โดยผลลัพธ์ที่ได้พบว่าวิธีการที่นำเสนอสามารถลู่เข้าหาคำตอบได้ดีกว่าวิธีแบบดั้งเดิม

แนวคิดของวิทยานิพนธ์ฉบับนี้ใช้หลักการพื้นฐานเหมือนงานวิจัยของ Dao et al. (2016) ที่เสนอแนวคิดการเริ่มต้นขั้นตอนวิธีเชิงพันธุกรรมใหม่เมื่อประชากรรุ่นใหม่มีความเหมาะสมหรือด้อยกว่าประชากรรุ่นเก่า โดยนำมาประยุกต์ใช้ในการหาค่าพารามิเตอร์ที่เหมาะสมสำหรับการจำแนกประเภทข้อมูลด้วยซัพพอร์ตเวกเตอร์แมชชีนด้วยการใช้ขั้นตอนวิธีเชิงพันธุกรรมในการหาค่าพารามิเตอร์ต่าง ๆ โดยในงานวิจัยนี้นำเสนอวิธีการปรับปรุงข้อมูลไม่สมดุลให้ข้อมูลเกิดความสมดุลมากขึ้นก่อนที่จะสร้างโมเดลในการจำแนกข้อมูลด้วยอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีน

ตารางที่ 2.21 สรุปเปรียบเทียบงานวิจัยที่เกี่ยวข้องกับการจำแนกประเภทข้อมูลส่วนน้อยในข้อมูลไม่สมดุล และการหาค่าพารามิเตอร์ที่เหมาะสม

กระบวนการทำงาน	งานวิจัยที่เกี่ยวข้อง						
	ก	ข	ค	ง	จ	ฉ	ช*
อัลกอริทึมที่เกี่ยวข้อง							
Support Vector Machine		✓	✓	✓		✓	✓
Decision Tree	✓		✓	✓		✓	
Genetic Algorithm					✓		✓
Back-Propagation Neural Network		✓			✓	✓	
Bagging / Boosting	✓						
Restarting Algorithm							✓
เสนออัลกอริทึมใหม่						✓	✓
เทคนิคที่ใช้จัดการกับข้อมูล							
Over-Sampling		✓	✓	✓			✓
Under-Sampling			✓				✓
SMOTE Technique	✓						✓
มาตรวัดที่ใช้วัดประสิทธิภาพของโมเดล							
Accuracy		✓	✓	✓		✓	✓
Precision							✓
Sensitivity or Recall	✓						✓
Specificity	✓						
F-Measure							✓
วัตถุประสงค์ของการวิจัย							
เพื่อทดสอบประสิทธิภาพของโมเดล	✓	✓	✓	✓	✓	✓	✓
เพื่อทดสอบความถูกต้อง	✓	✓	✓	✓	✓	✓	✓
เพื่อเสนอแนวคิดใหม่	✓	✓	✓	✓	✓	✓	✓

ก หมายถึง งานวิจัยของ Krawczyk et al. (2014)

ข หมายถึง งานวิจัยของ Liao et al. (2014)

ค หมายถึง งานวิจัยของ Cateni et al. (2014)

ง หมายถึง งานวิจัยของ Zhang and Li (2014)

จ หมายถึง งานวิจัยของ Yin et al. (2011)

ฉ หมายถึง งานวิจัยของ Chen et al. (2014)

ช หมายถึง งานวิจัยของวิทยานิพนธ์ฉบับนี้

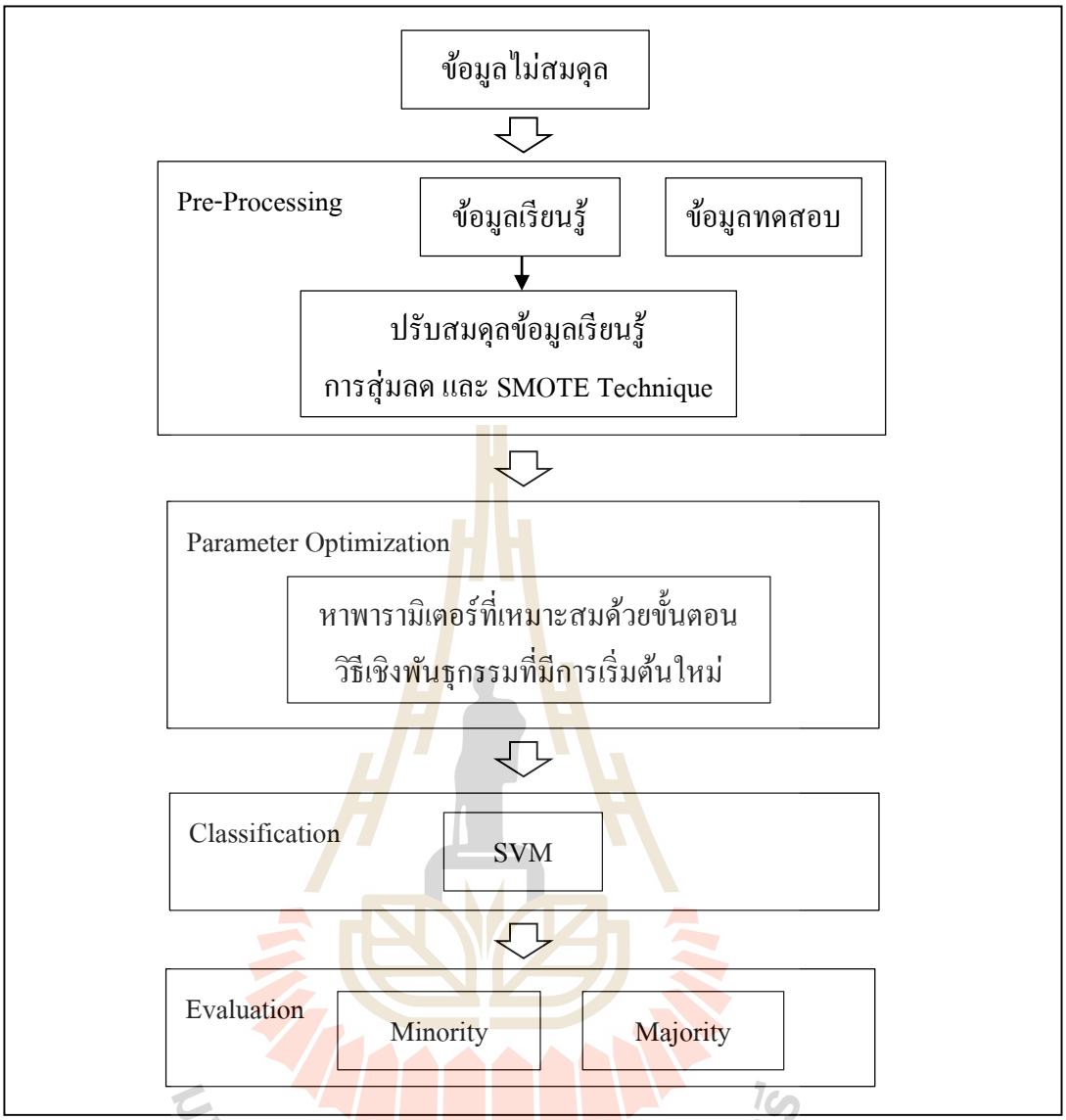
บทที่ 3
วิธีดำเนินงานวิจัย

ในงานวิจัยนี้มีวัตถุประสงค์เพื่อพัฒนาอัลกอริทึมที่ใช้ในการจำแนกประเภทข้อมูลที่มีจำนวนข้อมูลในแต่ละคลาสไม่สมดุลกัน เพื่อเพิ่มประสิทธิภาพในการจำแนกข้อมูลคลาสส่วนน้อยให้ดียิ่งขึ้น โดยขอบเขตของงานวิจัยจะใช้ข้อมูลประเภทตัวเลข และมีคลาสเป้าหมายจำนวน 2 คลาส ซึ่งในบทนี้จะกล่าวถึงรายละเอียดวิธีดำเนินงานวิจัย และขั้นตอนต่าง ๆ ในงานวิจัย

3.1 กรอบแนวคิดของการวิจัย

แนวคิดหลักของงานวิจัยนี้ คือ การจำแนกข้อมูลไม่สมดุล โดยอาศัยการปรับข้อมูลโดยการลดจำนวนข้อมูลจากคลาสส่วนมากลงด้วยเทคนิคการสุ่มลด และสังเคราะห์สร้างข้อมูลจากคลาสส่วนน้อยเพิ่มด้วย SMOTE Technique เพื่อให้ข้อมูลมีความสมดุลมากขึ้น และใช้เทคนิคขั้นตอนวิธีเชิงพันธุกรรมในการหาค่าพารามิเตอร์ที่เหมาะสมสำหรับซัพพอร์ตเวกเตอร์แมชชีน ร่วมกับการประยุกต์ใช้เทคนิคการเริ่มต้นใหม่ (Restart Technique) ในขั้นตอนวิธีเชิงพันธุกรรมเพื่อปรับปรุงการสร้างประชากรเริ่มต้นให้มีประสิทธิภาพดียิ่งขึ้นซึ่งทำให้ประสิทธิภาพในการจำแนกข้อมูลมีความแม่นยำมากขึ้น

ขั้นตอนวิธีของการวิจัยนี้ประกอบไปด้วย 4 ส่วนหลัก คือ 1. การปรับข้อมูลให้มีความสมดุลด้วยการสุ่มลดข้อมูลจากคลาสส่วนมาก และสังเคราะห์สร้างข้อมูลจากคลาสส่วนน้อยเพิ่มด้วย SMOTE Technique 2. การหาค่าพารามิเตอร์ที่เหมาะสมด้วยขั้นตอนวิธีเชิงพันธุกรรมที่มีการเริ่มต้นใหม่ 3. การจำแนกข้อมูลไม่สมดุล และ 4. การประเมินประสิทธิภาพของการจำแนก แสดงขั้นตอนการทำงานดังรูปที่ 3.1



รูปที่ 3.1 กรอบแนวคิดงานวิจัย

3.1.1 การปรับข้อมูลให้มีความสมดุล

สำหรับขั้นตอนนี้จะทำการปรับข้อมูลให้มีความสมดุล โดยใช้เทคนิคการสุ่มลดจำนวนข้อมูลจากคลาสส่วนมากลงเพื่อทำให้จำนวนข้อมูลระหว่างคลาสส่วนมากและคลาสส่วนน้อยมีจำนวนใกล้เคียงกันมากขึ้น และทำการสังเคราะห์สร้างข้อมูลจากคลาสส่วนน้อยให้มีจำนวนเพิ่มมากขึ้นจนใกล้เคียงกับจำนวนข้อมูลในคลาสส่วนมาก

การปรับข้อมูลให้มีความสมดุลด้วย SMOTE Technique จะทำการสุ่มสร้างข้อมูลจากคลาสส่วนน้อยตามจำนวนที่กำหนด โดยการวัดระยะห่างจากจุดข้อมูลตัวอย่างไปยังจุดข้อมูลใกล้เคียง แล้วสุ่มสร้างข้อมูลสังเคราะห์ขึ้นโดยข้อมูลสังเคราะห์ที่สร้างขึ้นจะอยู่ภายในระยะห่างจากจุดข้อมูลตัวอย่างไปยังจุดข้อมูลเพื่อนบ้านแสดงดังรูปที่ 3.2

การปรับข้อมูลให้มีความสมดุลด้วย SMOTE Technique

ข้อมูลเข้า : ข้อมูลไม่สมดุล D, จำนวนคลาสส่วนน้อยที่ต้องการสังเคราะห์เพิ่ม N, จำนวนเพื่อนบ้าน k

ผลลัพธ์ : ข้อมูลที่มีความสมดุลระหว่างสองคลาส

วิธีการ :

1. สุ่มเลือกข้อมูลจากคลาสส่วนน้อยเพื่อเป็นจุดศูนย์กลาง C ในการสร้างข้อมูลสังเคราะห์
2. คำนวณระยะห่างระหว่างจุดศูนย์กลาง C ไปยังจุดข้อมูลเพื่อนบ้าน k จุด (ข้อมูลคลาสส่วนน้อยรอบ ๆ จุดศูนย์กลาง)
3. สำหรับรอบที่ $i \leq N$
    1) สุ่มตัวเลขระหว่าง 1 ถึง k, ในขั้นตอนนี้จะสุ่มเลือกตัวเลขขึ้นมา 1 ตัว เพื่อใช้เป็นข้อมูลเริ่มต้นสำหรับสร้างข้อมูลสังเคราะห์ร่วมกับข้อมูลที่เป็นจุดศูนย์กลาง
    2) คำนวณระยะห่างระหว่างข้อมูลจุดศูนย์กลางกับข้อมูลจุดที่ถูกสุ่มเลือก
    3) สุ่มตัวเลขระหว่าง 0 ถึง 1

รูปที่ 3.2 อัลกอริทึมปรับข้อมูลให้มีความสมดุลด้วย SMOTE Technique

ตัวอย่างการปรับข้อมูลด้วย SMOTE Technique เมื่อสุ่มจุดข้อมูลจากคลาสส่วนน้อยที่นำมาใช้เป็นข้อมูลจุดศูนย์กลางคือข้อมูลลำดับที่ 3 และพิจารณาจุดเพื่อนบ้านจำนวน 3 จุดที่ใกล้ข้อมูลจุดศูนย์กลางมากที่สุด เพื่อสร้างข้อมูลสังเคราะห์จำนวน 3 จุด และสมมติให้ค่าสุ่มตัวเลขระหว่าง 0 ถึง 1 ของข้อมูลทั้ง 3 ชุดเป็น 0.5 เมื่อสมมติให้ข้อมูลตัวอย่าง ดังตารางที่ 3.1

ตารางที่ 3.1 ตัวอย่างข้อมูลของคลาสส่วนน้อย

ลำดับข้อมูล	ความกว้าง	ความยาว	น้ำหนักเนื้องอก	ระดับความเข้ม	คลาส
1	1.5”	2.2”	2.0g	0.9	True
2	1.5”	2.1”	2.1 g	0.8	True
3	1.7”	2.3”	2.2 g	0.9	True
4	1.6”	2.2”	2.0 g	0.8	True
5	1.8”	2.7”	2.5 g	0.7	True

จากการใช้ข้อมูลที่ 3 เป็นข้อมูลจุดศูนย์กลาง และใช้จำนวนเพื่อนบ้าน 3 จุดเพื่อสร้างข้อมูลสังเคราะห์ขึ้นจำนวน 3 จุด จะได้ข้อมูลที่นำมาใช้สร้างร่วมกับข้อมูลจุดศูนย์กลาง ได้แก่ ข้อมูลลำดับที่ 1, ข้อมูลลำดับที่ 2 และข้อมูลลำดับที่ 4 โดยแต่ละข้อมูลจะมีระยะห่างจากข้อมูลจุดศูนย์กลางดังนี้

ข้อมูลที่ 1 อยู่ห่างจากข้อมูลจุดศูนย์กลาง = (1.5 – 1.7, 2.2 – 2.3, 2.0 – 2.2, 0.9 – 0.9) = (-0.2, -0.1, -0.2, 0)

ข้อมูลที่ 2 อยู่ห่างจากข้อมูลจุดศูนย์กลาง = (1.5 - 1.7, 2.1 - 2.3, 2.1 - 2.2, 0.8 - 0.9) = (-0.2, -0.2, -0.1, -0.1)

ข้อมูลที่ 4 อยู่ห่างจากข้อมูลจุดศูนย์กลาง = (1.6 - 1.7, 2.2 - 2.3, 2.0 - 2.2, 0.8 - 0.9) = (-0.1, -0.1, -0.2, -0.1)

จะได้ข้อมูลสังเคราะห์ที่เกิดจากการสร้างโดยใช้ข้อมูลจุดศูนย์กลาง (ข้อมูลลำดับที่ 3) ร่วมกับการใช้ข้อมูลลำดับที่ 1 ดังนี้(1.7 + (0.5*-0.2), 2.3 + (0.5*-0.1), 2.2 + (0.5*-0.2), 0.9 + (0.5*0) = (1.6, 2.25, 2.1, 0.9)

จะได้ข้อมูลสังเคราะห์ที่เกิดจากการสร้างโดยใช้ข้อมูลจุดศูนย์กลาง (ข้อมูลลำดับที่ 3) ร่วมกับการใช้ข้อมูลลำดับที่ 2 ดังนี้ (1.7 + (0.5*-0.2), 2.3 + (0.5*-0.2), 2.2 + (0.5*-0.1), 0.9 + (0.5*-0.1) = (1.6, 2.2, 2.15, 0.85)

จะได้ข้อมูลสังเคราะห์ที่เกิดจากการสร้างจากการใช้ข้อมูลจุดศูนย์กลาง (ข้อมูลลำดับที่ 3) ร่วมกับการใช้ข้อมูลลำดับที่ 4 ดังนี้ (1.7 + (0.5*-0.1), 2.3 + (0.5*-0.1), 2.2 + (0.5*-0.2), 0.9+(0.5*0.1) = (1.65, 2.25, 2.1, 0.85)
ซึ่งชุดข้อมูลใหม่แสดงให้เห็นดังตารางที่ 3.2

ตารางที่ 3.2 ชุดข้อมูลหลังจากใช้ SMOTE Technique สร้างข้อมูลสังเคราะห์ 3 ข้อมูล

ลำดับข้อมูล	ความกว้าง	ความยาว	น้ำหนักเนื้องอก	ระดับความเข้ม	คลาส
1	1.5”	2.2”	2.0 g	0.9	True
2	1.5”	2.1”	2.1 g	0.8	True
3	1.7”	2.3”	2.2 g	0.9	True
4	1.6”	2.2”	2.0 g	0.8	True
5	1.8”	2.7”	2.5 g	0.7	True
6	1.6”	2.25”	2.1 g	0.9	True
7	1.6”	2.2”	2.15 g	0.85	True
8	1.65”	2.25”	2.1 g	0.85	True

3.1.2 การหาค่าพารามิเตอร์ที่เหมาะสมด้วยขั้นตอนวิธีเชิงพันธุกรรมที่มีการเริ่มต้นใหม่

สำหรับการหาค่าพารามิเตอร์ที่เหมาะสมด้วยขั้นตอนวิธีเชิงพันธุกรรมที่มีการเริ่มต้นใหม่ จะทำการเลือกพารามิเตอร์ทั้ง 3 พารามิเตอร์ที่เหมาะสมที่สุดเพื่อนำไปสร้างโมเดลในการจำแนกประเภทข้อมูลด้วยอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีน โดยมีหลักการทำงานของขั้นตอนวิธีเชิงพันธุกรรมอย่างง่าย แสดงดังรูปที่ 3.3

การหาค่าพารามิเตอร์ที่เหมาะสมด้วยขั้นตอนวิธีเชิงพันธุกรรม

ข้อมูลเข้า : พารามิเตอร์ C, พารามิเตอร์ Epsilon, พารามิเตอร์ Gamma, จำนวนรุ่นของประชากรที่ต้องการสร้าง T, ข้อมูลที่มีความสมดุล

ผลลัพธ์ : พารามิเตอร์ C, พารามิเตอร์ Epsilon, พารามิเตอร์ Gamma ที่มีความเหมาะสมที่สุด

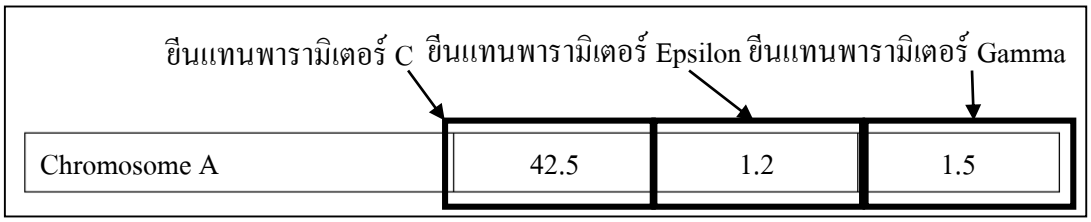
วิธีการ :

1. เข้ารหัสโครโมโซม
2. สุ่มสร้างประชากรเริ่มต้นจำนวน p
3. สำหรับรอบที่ $t \leq T$
    1) ประเมินความเหมาะสมของแต่ละโครโมโซมด้วยการใช้ค่าความถูกต้องในการจำแนกจากอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีน
    2) ดำเนินการทางขั้นตอนวิธีเชิงพันธุกรรม (การคัดเลือก การสลับสายพันธุ์ การกลายพันธุ์)
    3) แทนที่ประชากรรุ่นเก่า
4. ประเมินค่าความเหมาะสมของแต่ละโครโมโซมเพื่อคัดเลือกโครโมโซมที่เหมาะสมที่สุด

รูปที่ 3.3 การหาค่าพารามิเตอร์ที่เหมาะสมด้วยขั้นตอนวิธีเชิงพันธุกรรม

1. การเข้ารหัสโครโมโซม

ในแต่ละประชากรจะประกอบไปด้วยโครโมโซมที่มีการเข้ารหัสแบบใช้ค่าจริง โดยแต่ละยีนในโครโมโซมแสดงถึงค่าพารามิเตอร์ C พารามิเตอร์ Epsilon พารามิเตอร์ Gamma แสดงดังรูปที่ 3.4



รูปที่ 3.4 พารามิเตอร์ในรูปแบบโครโมโซม

2. สร้างประชากรเริ่มต้น

สำหรับขั้นตอนการสุ่มสร้างประชากรเริ่มต้น จะทำการสุ่มสร้างประชากรที่ประกอบไปด้วยโครโมโซมที่มีลักษณะเช่นเดียวกับโครโมโซมในรูปที่ 3.4 โดยจะทำการสุ่มสร้างประชากรเริ่มต้นเป็นจำนวน p ตัวอย่างการสุ่มสร้างประชากรเริ่มต้นจำนวน 5 ประชากร แสดงดังตารางที่ 3.3

ตารางที่ 3.3 ตัวอย่างการสุ่มสร้างประชากร

Chromosome A	42.5	1.2	1.5
Chromosome B	33.7	1.4	1.7
Chromosome C	41.5	0.7	0.2
Chromosome D	17.7	0.4	0.9
Chromosome E	26.4	0.1	0.1

3. ตรวจสอบเงื่อนไขการสิ้นสุดการทำงาน

ในขั้นตอนตรวจสอบเงื่อนไขสิ้นสุดการทำงานของขั้นตอนวิธีเชิงพันธุกรรม จะตรวจสอบจากจำนวนรุ่นประชากรที่สร้างขึ้น เมื่อดำเนินการทางพันธุกรรมเพื่อสร้างประชากรรุ่นใหม่ครบตามจำนวนที่กำหนดจะสิ้นสุดการทำงาน

4. การประเมินความเหมาะสมของแต่ละโครโมโซม

ในขั้นตอนการประเมินความเหมาะสมของแต่ละโครโมโซม จะทำการประเมินว่าโครโมโซมใดที่นำไปประยุกต์ใช้ในการสร้างโมเดลด้วยอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีน แล้วนำไปจำแนกประเภทข้อมูลแล้วได้ค่าความแม่นยำในการจำแนกประเภทข้อมูลสูงที่สุด (ค่า Accuracy สูงที่สุด) โดยค่าความแม่นยำในการจำแนกประเภทข้อมูลสามารถคำนวณได้จากสมการที่ 8

ตัวอย่างการประเมินความเหมาะสมของแต่ละโครโมโซม โดยสมมติให้โครโมโซมจากตารางที่ 3.3 มีค่าความแม่นยำในการจำแนกประเภทข้อมูลของแต่ละโครโมโซมแสดงดังตารางที่ 3.4

ตารางที่ 3.4 ค่าความแม่นยำในการจำแนกประเภทข้อมูลของแต่ละโครโมโซม

Chromosome	C	Epsilon	Gamma	Accuracy
Chromosome A	42.5	1.2	1.5	82.75
Chromosome B	33.7	1.4	1.7	73.98
Chromosome C	41.5	0.7	0.2	80.17
Chromosome D	17.7	0.4	0.9	75.79
Chromosome E	26.4	0.1	0.1	85.23

เมื่อเรียงตามค่าความถูกต้องในการจำแนกประเภทข้อมูล (ค่า Accuracy ยิ่งสูง แสดงว่าโมเดลนั้นมีประสิทธิภาพยิ่งดี) จะได้ว่า โครโมโซม E มีความเหมาะสมที่จะนำไปใช้ในการสร้างโมเดลจำแนกประเภทข้อมูลด้วยอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนมากที่สุด แสดงดังตารางที่ 3.5

ตารางที่ 3.5 โครโมโซมที่มีค่าความเหมาะสมในการนำไปเป็นโครโมโซมพ่อแม่

Chromosome	C	Epsilon	Gamma	Accuracy
Chromosome E	26.4	0.1	0.1	85.23
Chromosome A	42.5	1.2	1.5	82.75
Chromosome C	41.5	0.7	0.2	80.17
Chromosome D	17.7	0.4	0.9	75.79
Chromosome B	33.7	1.4	1.7	73.98

5. การดำเนินการทางขั้นตอนวิธีเชิงพันธุกรรม

ในขั้นตอนนี้จะทำการเพิ่มโอกาสในการถูกคัดเลือกให้กับโครโมโซมที่เหมาะสมเพื่อเพิ่มโอกาสในการถูกคัดเลือกซึ่งจะนำไปใช้เป็นต้นกำเนิดสายพันธุ์ให้แก่โครโมโซมรุ่นลูกหลาน โดยจัดสรรด้วยค่าความเหมาะสมของแต่ละโครโมโซม หลังจากนั้นจะทำการสุ่มเลือกโครโมโซมพ่อแม่แล้วนำไปเป็นต้นกำเนิดในการสลับสายพันธุ์เพื่อให้เกิดโครโมโซมรุ่นลูกหลาน สมมติสุ่มเลือกโครโมโซม C และโครโมโซม D เป็นต้นกำเนิดสายพันธุ์โดยมีการสลับสายพันธุ์แบบจุดเดียวที่ตำแหน่งที่ 2 แสดงดังรูปที่ 3.5 และโครโมโซมใหม่ที่ได้จากการสลับสายพันธุ์แสดงดังตารางที่ 3.6

Chromosome C	41.5	0.7	0.2
Chromosome D	17.7	0.4	0.9

Chromosome C(offspring 1)	41.5	0.4	0.9
Chromosome D (offspring 2)	17.7	0.7	0.2

รูปที่ 3.5 การสลับสายพันธุ์แบบจุดเดียวที่ตำแหน่งที่ 2

ตารางที่ 3.6 โครโมโซมหลังจากเกิดการสลับสายพันธุ์

Chromosome A	42.5	1.2	1.5
Chromosome B	33.7	1.4	1.7
Chromosome C (offspring 1)	41.5	0.4	0.9
Chromosome D (offspring 2)	17.7	0.7	0.2
Chromosome E	26.4	0.4	0.9

หลังจากนั้นเพื่อเพิ่มความหลากหลายทางสายพันธุ์มากยิ่งขึ้น จะทำการกระตุ้นให้เกิดการกลายพันธุ์ โดยทำการสุ่มเลือกยีนที่จะถูกกระตุ้น แล้วทำการสุ่มค่าขึ้นมาแทนที่ค่าเดิมของยีนนั้น สมมติให้สุ่มเลือกโครโมโซม A โดยที่ยีนที่ 2 ถูกกระตุ้นให้เกิดการกลายพันธุ์ด้วยการแทนที่ด้วยค่า 1.0 การกลายพันธุ์แสดงดังรูปที่ 3.6 และได้โครโมโซมใหม่ ดังตารางที่ 3.7

Chromosome A	42.5	1.2	1.5

Chromosome A (offspring)	42.5	1.0	1.5

รูปที่ 3.6 การกลายพันธุ์ของโครโมโซม

ตารางที่ 3.7 โครโมโซมหลังจากเกิดการกลายพันธุ์

Chromosome A (offspring)	42.5	1.0	1.5
Chromosome B	33.7	1.4	1.7
Chromosome C (offspring 1)	41.5	0.4	0.9
Chromosome D (offspring 2)	17.7	0.7	0.2
Chromosome E	26.4	0.4	0.9

6. การแทนที่ประชากร

สำหรับการแทนที่ประชากรจะเป็นการนำประชากรรุ่นใหม่ที่ได้ไปแทนที่ประชากรรุ่นเก่า โดยตัวอย่างของการแทนที่ประชากรเดิมทั้งรุ่น โดยการนำประชากรรุ่นใหม่ทั้งหมด แทนที่ประชากรรุ่นเก่าทั้งหมด แสดงดังตารางที่ 3.8

ตารางที่ 3.8 การแทนที่ประชากรรุ่นเก่าด้วยประชากรรุ่นใหม่ทั้งหมด

Chromosome A	42.5	1.0	1.5
Chromosome B	33.7	1.4	1.7
Chromosome C	41.5	0.4	0.9
Chromosome D	17.7	0.7	0.2
Chromosome E	26.4	0.4	0.9

หลังจากนั้นจะทำการประเมินค่าความเหมาะสมของแต่ละประชากรใหม่ เพื่อเลือกโครโมโซมที่มีค่าความเหมาะสมที่สุดไปใช้งาน

ขั้นตอนวิธีเชิงพันธุกรรมที่มีการเริ่มต้นใหม่

สำหรับขั้นตอนวิธีเชิงพันธุกรรมที่มีการเริ่มต้นใหม่ (Dao et al. ,2016) จะเป็นวิธีการตรวจสอบว่าโครโมโซมรุ่นใหม่ที่ได้จากการดำเนินการทางพันธุกรรมมีความเหมาะสมกว่าโครโมโซมรุ่นก่อนหน้าหรือไม่ หากโครโมโซมรุ่นใหม่ที่สร้างจากประชากรรุ่นเก่าไม่ได้ดีขึ้นกว่าเดิมติดต่อกันตามจำนวนรุ่นที่กำหนด นั่นหมายความว่าการดำเนินการทางพันธุกรรมโดยอาศัยเฉพาะการค้นหาแบบท้องถิ่น (Local Search) ของโครโมโซมที่มีอยู่ไม่เพียงพอต่อการเพิ่มประสิทธิภาพในการหาค่าพารามิเตอร์ที่เหมาะสม โดยมีหลักการทำงานของขั้นตอนวิธีเชิงพันธุกรรมที่มีการเริ่มต้นใหม่ แสดงดังรูปที่ 3.7

ขั้นตอนวิธีเชิงพันธุกรรมที่มีการเริ่มต้นใหม่

ข้อมูลเข้า : พารามิเตอร์ C, พารามิเตอร์ Epsilon, พารามิเตอร์ Gamma, จำนวนรุ่นของประชากรที่ต้องการสร้าง T, จำนวนรุ่นของประชากรที่ไม่ดีขึ้นติดกัน S, ข้อมูลที่มีความสมดุล

ผลลัพธ์ : พารามิเตอร์ C, พารามิเตอร์ Epsilon, พารามิเตอร์ Gamma ที่มีความเหมาะสมที่สุด

วิธีการ :

1. เข้ารหัสโครโมโซม
2. สุ่มสร้างประชากรเริ่มต้นจำนวน p
3. ประเมินความเหมาะสมของแต่ละโครโมโซมด้วยการใช้ค่าความถูกต้องในการจำแนกจากอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีน
4. สำหรับรอบที่ $t \leq T$
    1) สำหรับจำนวนรุ่นประชากรรุ่นใหม่ด้อยกว่าประชากรรุ่นเก่า $i \leq S$
        1) ดำเนินการทางขั้นตอนวิธีเชิงพันธุกรรม (การคัดเลือก การสลับสายพันธุ์ การกลายพันธุ์)
        2) แทนที่ประชากร
        3) ประเมินค่าความเหมาะสมของแต่ละโครโมโซม
    2) สำหรับจำนวนรุ่นประชากรรุ่นใหม่ด้อยกว่าประชากรรุ่นเก่า $i \geq S$
        1) สุ่มสร้างประชากรเริ่มต้นใหม่ โดยใช้โครโมโซมที่ดีที่สุดจากขั้นตอนวิธีเชิงพันธุกรรมเป็นประชากรเริ่มต้นด้วย
        2) ประเมินค่าความเหมาะสมของแต่ละโครโมโซม

รูปที่ 3.7 หลักการทำงานของขั้นตอนวิธีเชิงพันธุกรรมที่มีการเริ่มต้นใหม่

3.1.3 การจำแนกข้อมูลไม่สมดุล

ในงานวิจัยนี้ผู้วิจัยได้ใช้อัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนในการจำแนกโดยใช้ค่าพารามิเตอร์ C, พารามิเตอร์ Epsilon และพารามิเตอร์ Gamma ที่ได้รับจากหัวข้อที่ 3.1.2 มาเป็นพารามิเตอร์สำหรับการสร้างแบบจำลองการจำแนกประเภทข้อมูลไม่สมดุล โดยในงานวิจัยครั้งนี้ผู้วิจัยได้ใช้ข้อมูลสังเคราะห์จำนวน 1 ข้อมูล (สังเคราะห์ด้วยโปรแกรม R Studio Desktop Version 1.0.143) ชุดข้อมูลโรคหอบหืด จากโรงพยาบาลแห่งหนึ่งในจังหวัดนครราชสีมา โดยทำการรวบรวมข้อมูล ณ วันที่ 8 พฤศจิกายน พ.ศ. 2557 ข้อมูลโรคหัวใจ และข้อมูลผู้ป่วยโรคตับทางตะวันออกเฉียงเหนือของประเทศอินเดีย ซึ่งได้รับจากฐานข้อมูล UCI (UCI Machine Learning

Repository) สำหรับการเตรียมข้อมูลนั้นหากข้อมูลมีค่าสูญหาย (Missing Value) ผู้วิจัยใช้วิธีการลบข้อมูลทั้งแถวของข้อมูลที่มีค่าสูญหายทิ้ง หลังจากนั้นจะแบ่งข้อมูลออกเป็นสองส่วน ได้แก่ ส่วนข้อมูลทดสอบ และส่วนข้อมูลเรียนรู้ แล้วนำข้อมูลเรียนรู้มาปรับสมดุลข้อมูลด้วยเทคนิคการสุ่มลด และสังเคราะห์สร้างข้อมูลด้วย SMOTE Technique เมื่อข้อมูลเรียนรู้เกิดความสมดุลขึ้นแล้วจะนำข้อมูลเรียนรู้ไปหาค่าพารามิเตอร์ที่เหมาะสมสำหรับการจำแนกโดยการใช้ขั้นตอนวิธีเชิงพันธุกรรมที่มีการเริ่มต้นใหม่ หลังจากนั้นนำค่าพารามิเตอร์ที่ได้ไปสร้างโมเดลการจำแนกด้วยอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีน

3.2 เครื่องมือที่ใช้ในการวิจัย

เครื่องมือที่ใช้ในการพัฒนางานวิจัยนี้ ประกอบด้วย

1) เครื่องคอมพิวเตอร์สำหรับพัฒนา มีรายละเอียดดังนี้
   หน่วยประมวลผลกลาง : Intel Core i3-4160 3.6 GHz
   หน่วยความจำสำรอง : 1 TB
   หน่วยความจำหลัก : 12 GB
2) ระบบปฏิบัติการและโปรแกรมประยุกต์สำหรับพัฒนา ประกอบด้วย
   ระบบปฏิบัติการ : Windows 10 Pro 64 Bits
   เครื่องมือที่ใช้ในการพัฒนา : R Studio Desktop Version 1.0.143

บทที่ 4
การทดสอบและอภิปรายผล

การทดสอบประสิทธิภาพของระบบนั้น จะทดสอบประสิทธิภาพด้วยค่าความแม่นยำในการจำแนก ค่าความเที่ยง ค่าความไวหรือค่าระลึก และค่าการวัดเอฟ ในการจำแนกข้อมูลไม่สมดุลระหว่างคลาสส่วนมากและคลาสส่วนน้อย โดยเปรียบเทียบกับการจำแนกด้วยวิธีการดั้งเดิม (ไม่ปรับสมดุลข้อมูล และไม่ปรับค่าพารามิเตอร์) การจำแนกด้วยการใช้ขั้นตอนวิธีเชิงพันธุกรรมร่วมกับอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีน และเปรียบเทียบกับอัลกอริทึมในการจำแนกประเภทข้อมูลไม่สมดุลอีก 2 ประเภทได้แก่ เอดาบูส และรัสบูส สำหรับเนื้อหาในบทนี้จะประกอบด้วยข้อมูลที่ใช้ในการทดสอบ การทดสอบประสิทธิภาพการจำแนกประเภทข้อมูลไม่สมดุลโดยแบ่งเป็นการทดสอบประสิทธิภาพการจำแนกด้วยค่าความแม่นยำในการจำแนก ค่าความเที่ยง ค่าความไวหรือค่าระลึก และค่าการวัดเอฟ และในหัวข้อสุดท้ายเป็นการอภิปรายผล

4.1 ข้อมูลที่ใช้ในการทดสอบ

การทดสอบการจำแนกข้อมูลไม่สมดุลโดยใช้การปรับปรุงข้อมูลร่วมกับการหาค่าพารามิเตอร์ที่เหมาะสมด้วยขั้นตอนวิธีเชิงพันธุกรรมที่มีการเริ่มต้นใหม่จะใช้ข้อมูลสังเคราะห์จำนวน 1 ชุดข้อมูล (สังเคราะห์ด้วยโปรแกรม R Studio Desktop Version 1.0.143) ชุดข้อมูลโรคหอบหืด จากโรงพยาบาลแห่งหนึ่งในจังหวัดนครราชสีมา โดยทำการรวบรวมข้อมูล ณ วันที่ 8 พฤศจิกายน พ.ศ. 2557 ข้อมูลโรคหัวใจ และข้อมูลผู้ป่วยโรคตับทางตะวันออกเฉียงเหนือของประเทศอินเดีย ซึ่งได้รับจากฐานข้อมูล UCI (UCI Machine Learning Repository) โดยรายละเอียดของแต่ละข้อมูลแสดงดังตารางที่ 4.1

ตารางที่ 4.1 รายละเอียดชุดข้อมูลที่นำมาใช้ในงานวิจัย

ชุดข้อมูล	Attributes	จำนวนตัวอย่าง			Imbalanced Ratio
		Majority	Minority	Total	
Synthetic Dataset	16	600	100	700	6.00
Asthma Dataset	12	570	128	698	4.45
Hearth Disease	14	150	120	270	1.25
Indian Liver Patient Dataset	11	414	165	579	2.51

จากตารางที่ 4.1 สามารถอธิบายรายละเอียดของแต่ละชุดข้อมูลได้ดังนี้ สำหรับชุดข้อมูลสังเคราะห์จะประกอบไปด้วย 16 คอลัมน์ (15 คอลัมน์สำหรับใช้ในการจำแนก และ 1 คอลัมน์สำหรับคลาส) โดยมีจำนวนข้อมูลทั้งหมด 700 ข้อมูล มีจำนวนข้อมูลในคลาสส่วนมากทั้งหมด 600 ข้อมูล มีจำนวนข้อมูลในคลาสส่วนน้อย 100 ข้อมูล และมีระดับความไม่สมดุลอยู่ที่ 6.00 สำหรับชุดข้อมูลโรคหอบหืดจะประกอบไปด้วย 12 คอลัมน์ (11 คอลัมน์สำหรับใช้ในการจำแนก และ 1 คอลัมน์สำหรับคลาส) โดยมีจำนวนข้อมูลในคลาสส่วนมากทั้งหมด 570 ข้อมูล มีข้อมูลในคลาสส่วนน้อยทั้งหมด 120 ข้อมูล รวมทั้งหมด 698 ข้อมูล และมีระดับความไม่สมดุลอยู่ที่ 4.45 มีรายละเอียดแต่ละคอลัมน์แสดงดังตารางที่ 4.2

ตารางที่ 4.2 ข้อมูลโรคหอบหืด

ลำดับ	Attribute (คอลัมน์)	ค่า
1	Age Respondents (Year)	35-64
2	Gender of Respondents	0, 1
3	Highest Education Level	1, 2, 3, 4
4	Marital Status	1, 2, 3, 4, 5
5	Religion	1, 2, 3
6	Smoking	0, 1, 2
7	Exercise	0, 1
8	Weight (kg.)	37.2-113.3
9	Weight (cm.)	141-192
10	Waist (cm.)	57-119
11	Percent Body Fat	11.7-47.6
12	Class	Low, High

สำหรับชุดข้อมูลโรคหัวใจจะประกอบไปด้วย 14 คอลัมน์ (13 คอลัมน์สำหรับใช้ในการจำแนก และ 1 คอลัมน์สำหรับคลาส) โดยมีจำนวนข้อมูลทั้งหมด 270 ข้อมูล มีจำนวนข้อมูลในคลาสส่วนมากทั้งหมด 150 ข้อมูล มีจำนวนข้อมูลในคลาสส่วนน้อย 120 ข้อมูล และมีระดับความไม่สมดุลอยู่ที่ 1.25 มีรายละเอียดแต่ละคอลัมน์แสดงดังตารางที่ 4.3

ตารางที่ 4.3 ข้อมูลโรคหัวใจ

ลำดับ	Attribute (คอลัมน์)	ค่า
1	Age	29-77
2	Sex	0, 1
3	Chest Pain Type	0, 1, 2, 3
4	Resting Blood Pressure	94-200
5	Serum Cholesterol	126-564
6	Fasting Blood Sugar > 120 mg/dl	0, 1
7	Resting Electrocardiographic Results	0, 1
8	Maximum Heart Rate Achieved	71-202
9	Exercise Induced Angina	0, 1
10	Old peak	0-6.2
11	The Slope of The Peak Exercise ST Segment	0, 1, 2
12	Number of Major Vessels	0, 1, 2, 3
13	Thal	0, 1, 2
14	Class	Absence, Presence

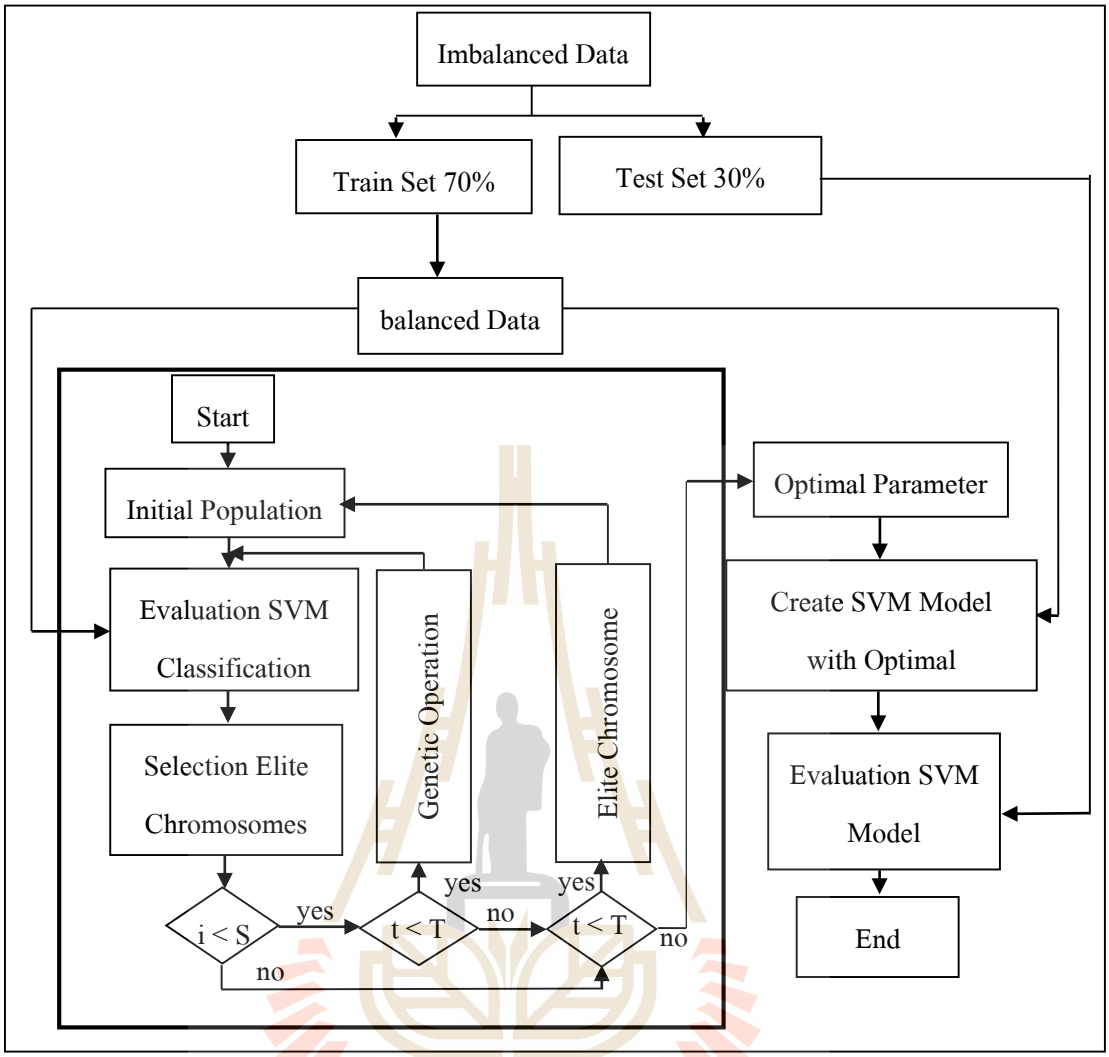
สำหรับชุดข้อมูลผู้ป่วยโรคตับจะประกอบไปด้วย 11 คอลัมน์ (10 คอลัมน์สำหรับใช้ในการจำแนก และ 1 คอลัมน์สำหรับคลาส) โดยมีจำนวนข้อมูลในคลาสส่วนมากทั้งหมด 414 ข้อมูล มีข้อมูลในคลาสส่วนน้อยทั้งหมด 165 ข้อมูล รวมทั้งหมด 579 ข้อมูล และมีระดับความไม่สมดุลอยู่ที่ 2.51 มีรายละเอียดแต่ละคอลัมน์แสดงดังตารางที่ 4.4

ตารางที่ 4.4 ข้อมูลผู้ป่วยโรคตับ

ลำดับ	Attribute (คอลัมน์)	ค่า
1	Age of the patient	4-90
2	Gender of the patient	1, 2
3	Total Bilirubin	0.4-75
4	Direct Bilirubin	0.1-19.7
5	Alkaline Phosphatase	63-2,110
6	Alanine Aminotransferase	10-2,000
7	Aspartate Aminotransferase	10-4,929
8	Total Proteins	2.7-9.6
9	Albumin	0.9-5.5
10	Albumin and Globulin Ratio	0.3-2.8
11	Class	Healthy, Sick

4.2 การออกแบบวิธีทดสอบ

การทดสอบประสิทธิภาพการจำแนกข้อมูลไม่สมดุลนี้จะทำการเปรียบเทียบประสิทธิภาพค่าความแม่นยำในการจำแนก ค่าความเที่ยง ค่าความไวหรือค่าระลึก และค่าการวัดเอฟ ในการจำแนกประเภทข้อมูลไม่สมดุลระหว่างการจำแนกด้วยวิธีการดั้งเดิม (ไม่ปรับสมดุลข้อมูล และไม่ปรับค่าพารามิเตอร์) การใช้ขั้นตอนวิธีเชิงพันธุกรรมร่วมกับอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีน และเปรียบเทียบกับอัลกอริทึมในการจำแนกประเภทข้อมูลไม่สมดุลอีก 2 ประเภทได้แก่ เอดาบูส และรัสบูส โดยผู้วิจัยได้ออกแบบการทดลองแสดงดังรูปที่ 4.1



รูปที่ 4.1 กรอบแนวคิดของการวิจัย

จากรูปที่ 4.1 สามารถอธิบายกรอบแนวคิดของการวิจัยได้ดังนี้ เริ่มต้นนำข้อมูลมาแบ่งออกเป็นสองส่วนในอัตราส่วน 70% ต่อ 30% โดยในส่วนแรกถูกแบ่งไว้ 70% จากข้อมูลทั้งหมดสำหรับเป็นข้อมูลเรียนรู้ และส่วนที่สองถูกแบ่งไว้ 30% จากข้อมูลทั้งหมดสำหรับเป็นข้อมูลทดสอบ และนำข้อมูลเรียนรู้มาปรับให้มีความสมดุลด้วยการใช้วิธีการผสมผสานระหว่างการสุ่มลดข้อมูลจากคลาสส่วนมากและสังเคราะห์ข้อมูลจากคลาสส่วนน้อยด้วย SMOTE Technique หลังจากนั้นจะได้ข้อมูลเรียนรู้ที่มีความสมดุลเพื่อนำไปใช้ในการหาค่าพารามิเตอร์ที่เหมาะสมสำหรับการจำแนกประเภทข้อมูลด้วยอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนและนำไปใช้ในการ สร้างโมเดลจำแนกประเภทข้อมูลไม่สมดุล หลังจากนั้นเริ่มกระบวนการขั้นตอนวิธีเชิงพันธุกรรมเพื่อหาค่าพารามิเตอร์ที่เหมาะสมจำนวน 3 พารามิเตอร์ ได้แก่พารามิเตอร์ C พารามิเตอร์ Epsilon และ

พารามิเตอร์ Gamma สำหรับอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีน โดยกำหนดขอบเขตในการค้นหาค่าพารามิเตอร์ที่เหมาะสมแสดงดังตารางที่ 4.26 ซึ่งจะใช้ความแม่นยำในการจำแนกที่ได้จากการใช้อัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีน โดยใช้ค่าพารามิเตอร์ที่ได้รับจากขั้นตอนวิธีเชิงพันธุกรรมร่วมกับข้อมูลเรียนรู้มาเป็นตัวประเมินค่าความเหมาะสมของแต่ละประชากรที่สร้างขึ้น และทำการเลือกประชากรระดับหัวกะทิเป็นจำนวน 10 อันดับ (ประชากรที่มีค่าความเหมาะสมสูงเป็นจำนวน 10 อันดับแรก) แล้วทำการดำเนินการทางสายพันธุ์จนกระทั่งได้ประชากรรุ่นใหม่ หากประชากรรุ่นใหม่มีค่าความเหมาะสมที่ด้อยกว่าประชากรรุ่นเก่าติดต่อกันเป็นจำนวน 2 รอบ ให้ทำการเริ่มต้นกระบวนการขั้นตอนวิธีเชิงพันธุกรรมใหม่ด้วยการนำประชากรระดับหัวกะทิจำนวน 10 ตัวไปสร้างเป็นประชากรเริ่มต้นด้วย จนกระทั่งครบรอบการทำงานที่กำหนด หลังจากนั้นจะนำพารามิเตอร์ที่เหมาะสมที่สุดไปสร้างโมเดลการจำแนกข้อมูลไม่สมดุลด้วยอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีน หลังจากนั้นทดสอบโมเดลด้วยการใช้ข้อมูลทดสอบ และประเมินประสิทธิภาพด้วยค่าความแม่นยำในการจำแนก ค่าความเที่ยง ค่าความไวหรือค่าระลึก และค่าการวัดเอฟ

4.3 การทดสอบประสิทธิภาพ

สำหรับการทดสอบประสิทธิภาพของการจำแนกประเภทข้อมูลไม่สมดุลนั้น ในงานวิจัยนี้สนใจประสิทธิภาพในการจำแนกประเภทข้อมูลส่วนน้อยให้มีประสิทธิภาพมากยิ่งขึ้น จึงได้กำหนดให้ Positive Class หมายถึงคลาสของข้อมูลส่วนน้อย ส่วน Negative Class หมายถึงคลาสของข้อมูลส่วนมาก ดังนั้น TP Rate จึงหมายถึงประสิทธิภาพในการจำแนกข้อมูลจากคลาสส่วนน้อย และ TN Rate หมายถึงประสิทธิภาพในการจำแนกข้อมูลจากคลาสส่วนมาก สมมุติให้ตารางที่ 4.5 คือตัวอย่างเมตริกซ์วัดประสิทธิภาพ (Confusion Matrix) ในการจำแนกประเภทข้อมูลของโมเดลจำแนก

ตารางที่ 4.5 เมตริกซ์วัดประสิทธิภาพการจำแนกประเภทข้อมูล

		Actual	
		Positive	Negative
Prediction	Positive	44 (TP)	16 (FP)
	Negative	6 (FN)	134 (TN)

ตัวอย่างการคำนวณประสิทธิภาพการจำแนกประเภทข้อมูลไม่สมดุล สามารถคำนวณได้ดังต่อไปนี้

ค่าความแม่นยำในการจำแนก = (TP+TN) / (TP+TN+FP+FN) = (44+134) / (44+134+16+6) = 0.89 หรือ 89.00%

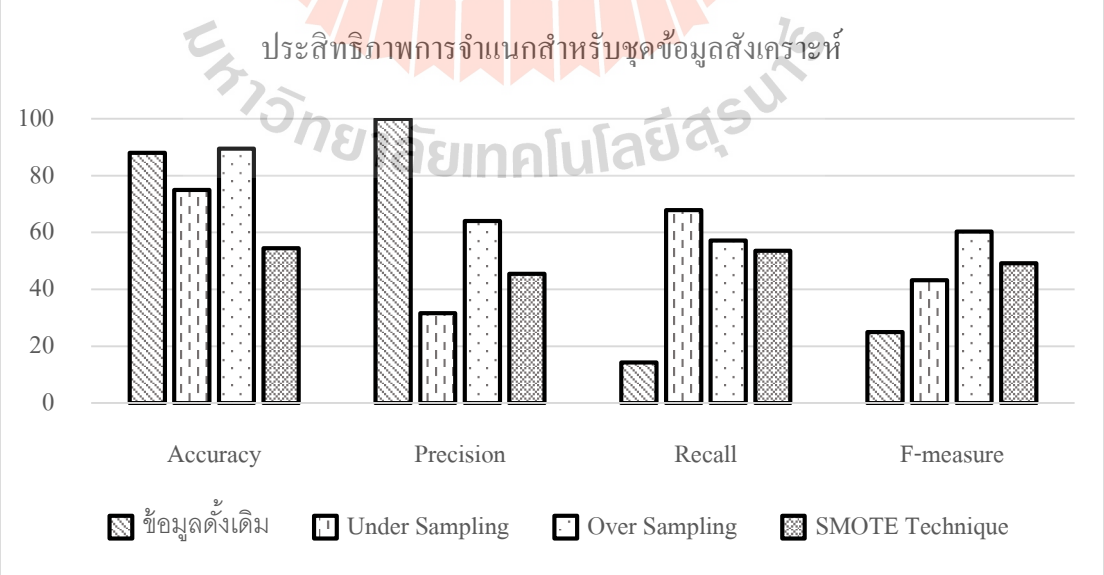
ค่าความเที่ยง = (TP) / (TP+FP) = (44) / (44+16) = 0.7333 หรือ 73.33%

ค่าระลึกหรือค่าความไว = (TP) / (TP+FN) = (44) / (44+6) = 0.88 หรือ 88.00%

ค่าการวัดเอฟ = (2*ค่าความเที่ยง*ค่าระลึกหรือค่าความไว) / (ค่าความเที่ยง+ค่าระลึกหรือค่าความไว) = (2*0.7333*0.88) / (0.7333+0.88) = 0.7273 หรือ 72.73%

4.4 ผลการทดสอบประสิทธิภาพ

สำหรับการทดสอบประสิทธิภาพการจำแนกประเภทข้อมูลไม่สมดุลนั้น จะใช้ข้อมูลทั้งหมด 4 ชุดข้อมูล โดยเป็นข้อมูลที่ได้จากการสังเคราะห์ข้อมูลจำนวน 1 ชุดข้อมูล ข้อมูลจากโรงพยาบาลแห่งหนึ่งในจังหวัดนครราชสีมาจำนวน 1 ชุดข้อมูล และข้อมูลจริงจากฐานข้อมูลมาตรฐานจำนวน 2 ชุดข้อมูล เมื่อทำการเปรียบเทียบประสิทธิภาพการจำแนกข้อมูลไม่สมดุลระหว่างการใช้ข้อมูลแบบดั้งเดิมเปรียบเทียบกับวิธีการปรับสมดุลให้แก่ข้อมูลเรียนรู้สำหรับชุดข้อมูลสังเคราะห์ ประสิทธิภาพการจำแนกแสดงดังรูปที่ 4.2 โดยมีรายละเอียดค่าประสิทธิภาพตามเกณฑ์ต่าง ๆ แสดงดังตารางที่ 4.6 สำหรับเมตริกซ์วัดประสิทธิภาพของเทคนิคแบบดั้งเดิมแสดงดังตารางที่ 4.7 เมตริกซ์วัดประสิทธิภาพของเทคนิคการสุ่มลดแสดงดังตารางที่ 4.8 เมตริกซ์วัดประสิทธิภาพของเทคนิคการสุ่มเพิ่มแสดงดังตารางที่ 4.9 และเมตริกซ์วัดประสิทธิภาพของเทคนิค SMOTE แสดงดังตารางที่ 4.10



รูปที่ 4.2 ประสิทธิภาพการจำแนกด้วยวิธีดั้งเดิมกับการปรับสมดุลของชุดข้อมูลสังเคราะห์

ตารางที่ 4.6 ประสิทธิภาพการจำแนกระหว่างวิธีดั้งเดิมกับการปรับสมดุลข้อมูลของชุดข้อมูลสังเคราะห์

เทคนิคที่ใช้	Accuracy (%)	Precision (%)	Recall (%)	F-measure (%)
ใช้ข้อมูลแบบดั้งเดิม	88.00	**100.00**	14.29	25.01
Under Sampling	75.00	31.67	**67.86**	43.19
Over Sampling	**89.50**	64.00	57.14	**60.38**
SMOTE Technique	84.50	45.46	53.57	49.18

ตารางที่ 4.7 เมตริกซ์วัดประสิทธิภาพของข้อมูลแบบดั้งเดิม (ไม่ปรับสมดุล) สำหรับชุดข้อมูลสังเคราะห์

		Actual	
		Positive	Negative
Prediction	Positive	4	0
	Negative	24	172

ตารางที่ 4.8 เมตริกซ์วัดประสิทธิภาพของเทคนิคสุ่มลดสำหรับชุดข้อมูลสังเคราะห์

		Actual	
		Positive	Negative
Prediction	Positive	19	41
	Negative	9	131

ตารางที่ 4.9 เมตริกซ์วัดประสิทธิภาพของเทคนิคสุ่มเกินสำหรับชุดข้อมูลสังเคราะห์

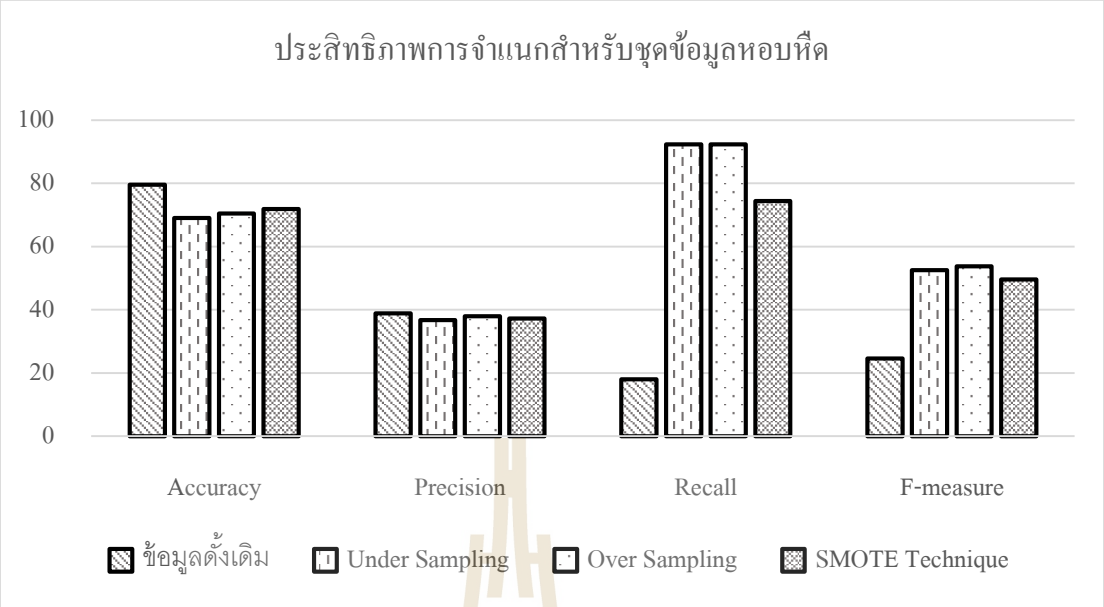
		Actual	
		Positive	Negative
Prediction	Positive	16	9
	Negative	12	163

ตารางที่ 4.10 เมตริกซ์วัดประสิทธิภาพของเทคนิค SMOTE สำหรับชุดข้อมูลสังเคราะห์

		Actual	
		Positive	Negative
Prediction	Positive	15	18
	Negative	13	154

จากผลการทดลองจากตารางที่ 4.6 ในการคำนวณหาค่าความแม่นยำในการจำแนกสำหรับข้อมูลไม่สมดุลพบว่าการปรับปรุงข้อมูลด้วยเทคนิคการสุ่มเกินให้ประสิทธิภาพดีที่สุดที่ 89.50% ลำดับถัดมาได้แก่การใช้ข้อมูลดั้งเดิมที่ 88.00% การใช้เทคนิค SMOTE มีค่าความแม่นยำเป็นอันดับสาม ที่ 84.50% และการสุ่มลดข้อมูลมีความแม่นยำน้อยที่สุดที่ 75.00% สำหรับค่าความเที่ยงพบว่าการใช้ข้อมูลดั้งเดิมให้ประสิทธิภาพสูงที่สุดที่ 100.00% การใช้การสุ่มเกินให้ประสิทธิภาพดีรองลงมาที่ 64.00% ขณะที่การใช้เทคนิค SMOTE มีค่าความแม่นยำเป็นอันดับสาม ที่ 45.46% และการสุ่มลดข้อมูลมีความแม่นยำด้อยที่สุดที่ 31.67% ในขณะที่ค่าความไวหรือค่าระลึกพบว่าการใช้วิธีปรับข้อมูลด้วยการสุ่มลดให้ประสิทธิภาพดีกว่าการใช้วิธีการสุ่มเกิน และดีกว่าการใช้เทคนิค SMOTE และการใช้ข้อมูลดั้งเดิม ที่ 67.86%, 57.14%, 53.57% และ 14.29% ตามลำดับ สำหรับค่าการวัดเอฟพบว่าการใช้วิธีการสุ่มเกินให้ประสิทธิภาพดีที่สุดที่ 60.38% การใช้เทคนิค SMOTE ที่ 49.18% ใช้วิธีการสุ่มลดที่ 43.19% และการใช้ข้อมูลดั้งเดิมที่ 25.01%

สำหรับชุดข้อมูลโรคหอบหืด เปรียบเทียบประสิทธิภาพการจำแนกระหว่างการใช้ข้อมูลแบบดั้งเดิมเปรียบเทียบกับวิธีการปรับสมดุลให้แก่ข้อมูลเรียนรู้แสดงประสิทธิภาพการจำแนกแสดงดังรูปที่ 4.3 และมีรายละเอียดค่าประสิทธิภาพการจำแนกตามเกณฑ์ต่าง ๆ ดังตารางที่ 4.11 สำหรับเมตริกซ์วัดประสิทธิภาพของเทคนิคแบบดั้งเดิมดังตารางที่ 4.12 เมตริกซ์วัดประสิทธิภาพของเทคนิคการสุ่มลดดังตารางที่ 4.13 เมตริกซ์วัดประสิทธิภาพของเทคนิคการสุ่มเพิ่มดังตารางที่ 4.14 และเมตริกซ์วัดประสิทธิภาพของเทคนิค SMOTE ดังตารางที่ 4.15



รูปที่ 4.3 ประสิทธิภาพการจำแนกด้วยวิธีดั้งเดิมกับการปรับสมดุลของชุดข้อมูลโรคหอบหืด

ตารางที่ 4.11 ประสิทธิภาพการจำแนกระหว่างวิธีดั้งเดิมกับการปรับสมดุลข้อมูลของชุดข้อมูลโรคหอบหืด

เทคนิคที่ใช้	Accuracy (%)	Precision (%)	Recall (%)	F-measure (%)
ใช้ข้อมูลแบบดั้งเดิม	**79.52**	**38.89**	17.95	24.56
Under Sampling	69.05	36.74	**92.31**	52.56
Over Sampling	70.48	37.90	**92.31**	**53.74**
SMOTE Technique	71.90	37.18	74.36	49.57

ตารางที่ 4.12 เมตริกซ์วัดประสิทธิภาพของข้อมูลแบบดั้งเดิม (ไม่ปรับสมดุล) สำหรับชุดข้อมูลโรคหอบหืด

		Actual	
		Positive	Negative
Prediction	Positive	7	11
	Negative	32	160

ตารางที่ 4.13 เมตริกซ์วัดประสิทธิภาพของเทคนิคสุ่มลดสำหรับชุดข้อมูลโรคหอบหืด

		Actual	
		Positive	Negative
Prediction	Positive	36	62
	Negative	3	109

ตารางที่ 4.14 เมตริกซ์วัดประสิทธิภาพของเทคนิคสุ่มเกินสำหรับชุดข้อมูลโรคหอบหืด

		Actual	
		Positive	Negative
Prediction	Positive	36	59
	Negative	3	112

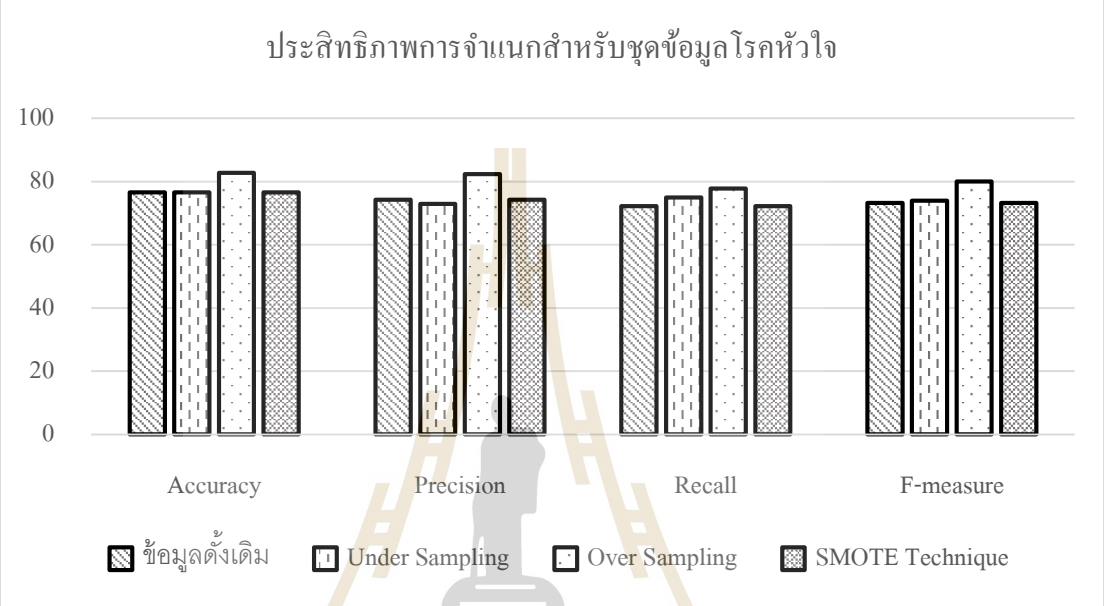
ตารางที่ 4.15 เมตริกซ์วัดประสิทธิภาพของเทคนิค SMOTE สำหรับชุดข้อมูลโรคหอบหืด

		Actual	
		Positive	Negative
Prediction	Positive	29	49
	Negative	10	122

จากผลการทดลองจากตารางที่ 4.11 เมื่อพิจารณาที่ค่าความแม่นยำในการจำแนกพบว่าการใช้ข้อมูลดั้งเดิมมีประสิทธิภาพสูงที่สุดที่ 79.52% รองลงมาเป็นการใช้เทคนิค SMOTE ที่ 71.90% และการสุ่มเกินมีความแม่นยำที่ 70.48% ขณะที่การสุ่มลดข้อมูลมีประสิทธิรูปที่ด้อยที่สุดที่ 69.05% เมื่อพิจารณาที่ค่าความเที่ยงพบว่าการใช้ข้อมูลดั้งเดิมให้ประสิทธิภาพดีที่สุดที่ 38.89% และการสุ่มเกินมีค่าความเที่ยงที่ 37.90% ขณะที่การใช้เทคนิค SMOTE มีค่าความเที่ยงที่ 37.18% ส่วนการใช้วิธีการสุ่มลดมีค่าความเที่ยงที่ 36.74% ในขณะที่ค่าความไวหรือค่าระลึกพบว่าการใช้วิธีปรับข้อมูลด้วยการสุ่มลดและการสุ่มเกินให้ประสิทธิภาพดีเท่ากันที่ 92.31% การใช้เทคนิค SMOTE มีประสิทธิภาพอยู่ที่ 74.36% และด้อยที่สุดคือการใช้ข้อมูลดั้งเดิมที่ 17.95% สำหรับค่าการวัดเอฟพบว่าการใช้วิธีการสุ่มเกินให้ประสิทธิภาพดีกว่าการใช้การสุ่มลด การใช้เทคนิค SMOTE และการใช้ข้อมูลดั้งเดิมที่ 53.74%, 52.56%, 49.57% และ 24.56% ตามลำดับ

ผลการเปรียบเทียบประสิทธิภาพการจำแนกของชุดข้อมูลโรคหัวใจ การใช้ข้อมูลแบบดั้งเดิมเปรียบเทียบกับวิธีการปรับสมดุลให้แก่ข้อมูลเรียนรู้ ประสิทธิภาพการจำแนกแสดงดังรูปที่

4.4 และมีรายละเอียดค่าประสิทธิภาพการจำแนกตามเกณฑ์ต่าง ๆ ดังตารางที่ 4.16 สำหรับเมตริกซ์วัดประสิทธิภาพของเทคนิคแบบดั้งเดิมแสดงดังตารางที่ 4.17 เมตริกซ์วัดประสิทธิภาพของเทคนิคการสุ่มลดแสดงดังตารางที่ 4.18 เมตริกซ์วัดประสิทธิภาพของเทคนิคการสุ่มเพิ่มแสดงดังตารางที่ 4.19 และเมตริกซ์วัดประสิทธิภาพของเทคนิค SMOTE แสดงดังตารางที่ 4.20



รูปที่ 4.4 ประสิทธิภาพการจำแนกด้วยวิธีดั้งเดิมกับการปรับสมดุลของชุดข้อมูลโรคหัวใจ

ตารางที่ 4.16 ประสิทธิภาพการจำแนกระหว่างวิธีดั้งเดิมกับการปรับสมดุลข้อมูลของชุดข้อมูลโรคหัวใจ

เทคนิคที่ใช้	Accuracy (%)	Precision (%)	Recall (%)	F-measure (%)
ใช้ข้อมูลแบบดั้งเดิม	76.54	74.29	72.22	73.24
Under Sampling	76.54	72.94	75.00	73.97
Over Sampling	**82.71**	**82.35**	**77.78**	**80.00**
SMOTE Technique	76.54	74.29	72.22	73.24

ตารางที่ 4.17 เมตริกซ์วัดประสิทธิภาพของข้อมูลแบบดั้งเดิม (ไม่ปรับสมดุล) สำหรับชุดข้อมูลโรคหัวใจ

		Actual	
		Positive	Negative
Prediction	Positive	26	9
	Negative	10	36

ตารางที่ 4.18 เมตริกซ์วัดประสิทธิภาพของเทคนิคสุ่มลดสำหรับชุดข้อมูลโรคหัวใจ

		Actual	
		Positive	Negative
Prediction	Positive	27	10
	Negative	9	35

ตารางที่ 4.19 เมตริกซ์วัดประสิทธิภาพของเทคนิคสุ่มเกินสำหรับชุดข้อมูลโรคหัวใจ

		Actual	
		Positive	Negative
Prediction	Positive	28	6
	Negative	8	39

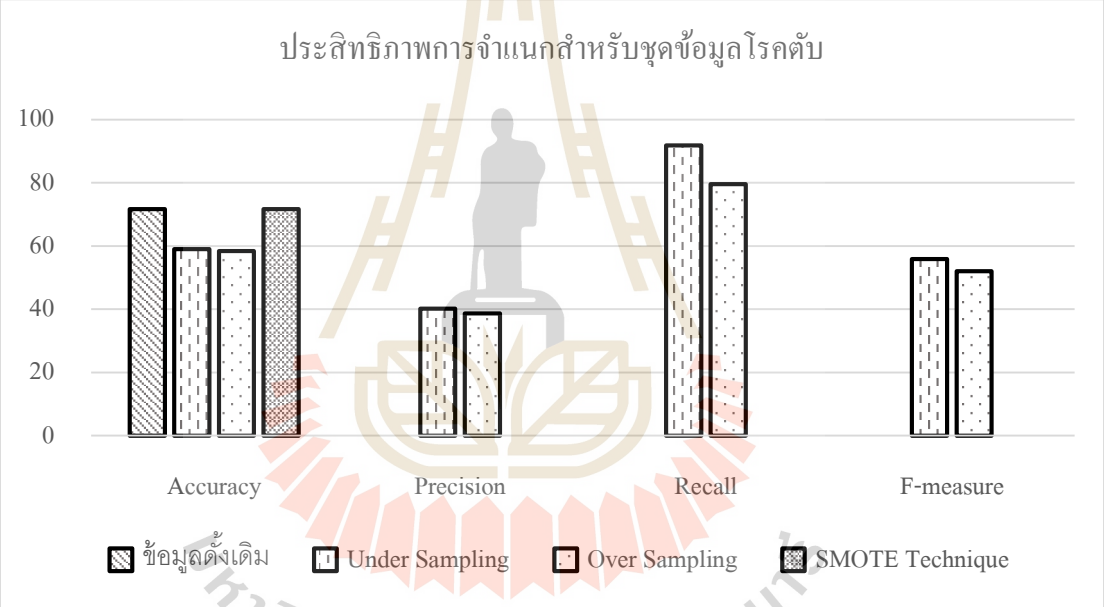
ตารางที่ 4.20 เมตริกซ์วัดประสิทธิภาพของเทคนิค SMOTE สำหรับชุดข้อมูลโรคหัวใจ

		Actual	
		Positive	Negative
Prediction	Positive	26	9
	Negative	10	36

จากผลการทดลองจากตารางที่ 4.16 พบว่าการใช้เทคนิคการสุ่มเกินให้ประสิทธิภาพในการจำแนกข้อมูลไม่สมดุลที่ดีกว่าเทคนิคอื่น ๆ ในทุก ๆ กรณี ที่ค่าความแม่นยำในการจำแนกที่ 82.71% ในขณะที่การใช้เทคนิคอื่น ๆ มีค่าความแม่นยำที่ 76.54% สำหรับค่าความเที่ยงที่ 82.35% ขณะที่การใช้ข้อมูลดั้งเดิมและการใช้เทคนิค SMOTE มีค่าความเที่ยงที่ 74.29% และการใช้การสุ่มลดมีประสิทธิภาพด้อยที่สุดที่ 72.94% ในส่วนของค่าระลึกหรือค่าความไวที่ 77.78% การใช้วิธีการสุ่ม

ลดมีประสิทธิภาพรองลงมาที่ 75.00% การใช้ข้อมูลดั้งเดิมและการใช้เทคนิค SMOTE มีประสิทธิรูปที่ 72.22% และสำหรับค่าการวัดเอฟที่ 80.00% โดยการใช้วิธีการสุ่มลดมีประสิทธิภาพรองลงมาที่ 73.97% และการใช้ข้อมูลดั้งเดิมกับการใช้เทคนิค SMOTE มีประสิทธิภาพเท่ากันที่ 73.24%

สำหรับชุดข้อมูลโรคตับเปรียบเทียบประสิทธิภาพการจำแนกระหว่างการใช้ข้อมูลแบบดั้งเดิมเปรียบเทียบกับวิธีการปรับสมดุลให้แก่ข้อมูลเรียนรู้ ประสิทธิภาพการจำแนกแสดงดังรูปที่ 4.5 และมีรายละเอียดค่าประสิทธิภาพการจำแนกตามเกณฑ์ต่าง ๆ ดังตารางที่ 4.21 สำหรับเมตริกซ์วัดประสิทธิภาพของเทคนิคแบบดั้งเดิมแสดงดังตารางที่ 4.22 เมตริกซ์วัดประสิทธิภาพของเทคนิคการสุ่มลดแสดงดังตารางที่ 4.23 เมตริกซ์วัดประสิทธิภาพของเทคนิคการสุ่มเพิ่มแสดงดังตารางที่ 4.24 และเมตริกซ์วัดประสิทธิภาพของเทคนิค SMOTE แสดงดังตารางที่ 4.25



รูปที่ 4.5 ประสิทธิภาพการจำแนกด้วยวิธีดั้งเดิมกับการปรับสมดุลของชุดข้อมูลโรคตับ

ตารางที่ 4.21 ประสิทธิภาพการจำแนกระหว่างวิธีดั้งเดิมกับการปรับสมดุลข้อมูลของชุดข้อมูลโรคตับ

เทคนิคที่ใช้	Accuracy (%)	Precision (%)	Recall (%)	F-measure (%)
ใช้ข้อมูลแบบดั้งเดิม	**71.68**	00.00	00.00	00.00
Under Sampling	58.96	**40.18**	**91.84**	**55.90**
Over Sampling	58.38	38.61	79.59	52.00
SMOTE Technique	**71.68**	00.00	00.00	00.00

ตารางที่ 4.22 เมตริกซ์วัดประสิทธิภาพของข้อมูลแบบดั้งเดิม (ไม่ปรับสมดุล) สำหรับชุดข้อมูลโรคตับ

		Actual	
		Positive	Negative
Prediction	Positive	0	0
	Negative	49	124

ตารางที่ 4.23 เมตริกซ์วัดประสิทธิภาพของเทคนิคสุ่มลดสำหรับชุดข้อมูลโรคตับ

		Actual	
		Positive	Negative
Prediction	Positive	45	67
	Negative	4	57

ตารางที่ 4.24 เมตริกซ์วัดประสิทธิภาพของเทคนิคสุ่มเกินสำหรับชุดข้อมูลโรคตับ

		Actual	
		Positive	Negative
Prediction	Positive	39	62
	Negative	10	62

ตารางที่ 4.25 เมตริกซ์วัดประสิทธิภาพของเทคนิค SMOTE สำหรับชุดข้อมูลโรคตับ

		Actual	
		Positive	Negative
Prediction	Positive	0	0
	Negative	49	124

จากผลการทดลองจากตารางที่ 4.21 เมื่อพิจารณาที่ค่าความแม่นยำในการจำแนกพบว่าการไม่ปรับสมดุลข้อมูลและการปรับสมดุลข้อมูลด้วยเทคนิค SMOTE ให้ประสิทธิภาพดีที่สุดที่ 71.68% รองลงมาคือการใช้วิธีการสุ่มลดที่ 58.96% และวิธีการสุ่มเกินที่ 58.38% สำหรับค่าความเที่ยง ค่าระลึกหรือค่าความไว และการวัดเอฟ พบกว่าการใช้เทคนิคการสุ่มลดให้ประสิทธิภาพดีที่สุดที่ 40.18%, 91.84% และ 55.90% ตามลำดับ รองลงมาคือการใช้วิธีการสุ่มเกิน มีประสิทธิภาพ

อยู่ที่ 38.61%, 79.59% และ 52.00% ในขณะที่การใช้ข้อมูลดั้งเดิมและการใช้เทคนิค SMOTE ไม่สามารถจำแนกข้อมูลจากคลาสส่วนน้อยได้เลย

จากผลการทดลองของทั้ง 4 ชุดข้อมูล จะเห็นได้ว่าการปรับข้อมูลเรียนรู้ให้เกิดความสมดุลระหว่างคลาสจะทำให้สามารถสร้างโมเดลที่มีความสามารถในการจำแนกข้อมูลจากคลาสส่วนน้อยได้ดียิ่งขึ้น โดยพิจารณาจากค่าการวัดเอฟเป็นหลัก หากพิจารณาจากค่าความแม่นยำในการจำแนกจะเห็นได้ว่าการไม่ปรับสมดุลจะให้ประสิทธิรูปที่ดีกว่าแต่เมื่อพิจารณาความสามารถในการจำแนกคลาสส่วนน้อยได้ถูกต้องทั้งหมดจะเห็นว่าการไม่ปรับสมดุลข้อมูลให้ประสิทธิรูปที่แย่

ในงานวิจัยนี้จึงทำการปรับข้อมูลให้มีความสมดุลก่อนนำไปหาค่าพารามิเตอร์ที่เหมาะสมด้วยขั้นตอนวิธีเชิงพันธุกรรมที่มีการเริ่มต้นใหม่ โดยเลือกใช้วิธีการแบบผสมผสานระหว่างเทคนิคการสุ่มลดข้อมูลจากคลาสส่วนมากและสังเคราะห์สร้างข้อมูลจากคลาสส่วนน้อย เนื่องจากการสุ่มลดข้อมูลจากคลาสส่วนมากลงเพียงอย่างเดียวอาจจะทำให้เกิดการสูญเสียบางข้อมูลที่สำคัญมากเกินไป (หากอัตราความไม่สมดุลแตกต่างกันมาก หากใช้การสุ่มลดเพียงอย่างเดียวจะทำการลดข้อมูลลงเป็นจำนวนมาก) ในขณะที่การสุ่มเพิ่มข้อมูลด้วยการใช้วิธีการสุ่มเพิ่มข้อมูลจากคลาสส่วนน้อยจะทำให้ข้อมูลมีการเอนเอียง และไม่มีความหลากหลายของข้อมูลในคลาสส่วนน้อยดังนั้นจึงเลือกใช้การสังเคราะห์สร้างข้อมูลคลาสส่วนน้อยเพิ่มโดยใช้ข้อมูลจากคลาสส่วนน้อยเป็นต้นแบบในการสุ่มสร้างเพื่อให้เกิดความหลากหลายมากยิ่งขึ้น

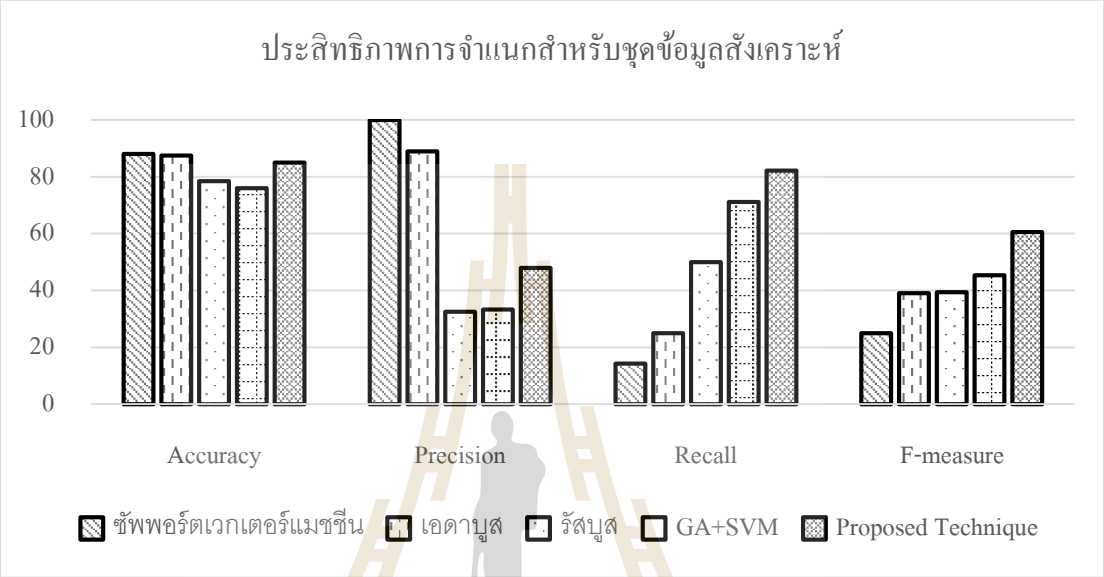
เมื่อปรับปรุงข้อมูลเรียนรู้ให้มีความสมดุลเกิดขึ้นแล้ว หลังจากนั้นทำการหาค่าพารามิเตอร์ที่เหมาะสมด้วยขั้นตอนวิธีเชิงพันธุกรรมที่มีการเริ่มต้นใหม่ โดยในงานวิจัยนี้ได้กำหนดพารามิเตอร์เริ่มต้นสำหรับขั้นตอนวิธีเชิงพันธุกรรมที่มีการเริ่มต้นใหม่ แสดงดังตารางที่ 4.26

ตารางที่ 4.26 พารามิเตอร์เริ่มต้นสำหรับขั้นตอนวิธีเชิงพันธุกรรมที่มีการเริ่มต้นใหม่

Cost, C	$10^{-4} – 10^{2}$	Prob. of mutation	0.01
Gamma	$10^{-3} – 2$	Iteration	100
Epsilon	$10^{-2} – 2$	Restart GA	2
Population size	100	Elite chromosome	10
Prob. of crossover	0.80		

สำหรับประสิทธิภาพการจำแนกของข้อมูลสังเคราะห์โดยเปรียบเทียบระหว่างเทคนิคที่นำเสนอกับวิธีการที่ใช้ในการจำแนกข้อมูลไม่สมดุลในปัจจุบันแสดงดังรูปที่ 4.6 และมีรายละเอียดประสิทธิภาพตามเกณฑ์ต่าง ๆ แสดงดังตารางที่ 4.27 สำหรับเมตริกซ์วัดประสิทธิภาพของการใช้

อัลกอริทึมเอดาบูส แสดงดังตารางที่ 4.28 เมตริกซ์วัดประสิทธิภาพของการใช้อัลกอริทึมรัสบูส แสดงดังตารางที่ 4.29 เมตริกซ์วัดประสิทธิภาพของการใช้ขั้นตอนวิธีเชิงพันธุกรรมร่วมกับอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีน แสดงดังตารางที่ 4.30 และเมตริกซ์วัดประสิทธิภาพของเทคนิคที่นำเสนอแสดงดังตารางที่ 4.31



รูปที่ 4.6 ประสิทธิภาพการจำแนกด้วยเทคนิคต่าง ๆ ของชุดข้อมูลสังเคราะห์

ตารางที่ 4.27 ประสิทธิภาพการจำแนกแต่ละอัลกอริทึมของชุดข้อมูลสังเคราะห์

อัลกอริทึม	Accuracy (%)	Precision (%)	Recall (%)	F-measure (%)
ซัพพอร์ตเวกเตอร์แมชชีน	**88.00**	**100.00**	14.29	25.01
เอดาบูส	87.50	88.89	25.00	39.03
รัสบูส	78.50	32.56	50.00	39.44
Genetic Algorithm + SVM	76.00	33.33	71.14	45.39
Proposed Technique	85.00	47.92	**82.14**	**60.53**

ตารางที่ 4.28 เมตริกซ์วัดประสิทธิภาพของอัลกอริทึมเอดาบูส สำหรับชุดข้อมูลสังเคราะห์

		Actual	
		Positive	Negative
Prediction	Positive	8	24
	Negative	24	167

ตารางที่ 4.29 เมตริกซ์วัดประสิทธิภาพของอัลกอริทึมรัสบูส สำหรับชุดข้อมูลสังเคราะห์

		Actual	
		Positive	Negative
Prediction	Positive	14	29
	Negative	14	143

ตารางที่ 4.30 เมตริกซ์วัดประสิทธิภาพของขั้นตอนวิธีเชิงพันธุกรรมร่วมกับอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีน สำหรับชุดข้อมูลสังเคราะห์

		Actual	
		Positive	Negative
Prediction	Positive	20	40
	Negative	8	132

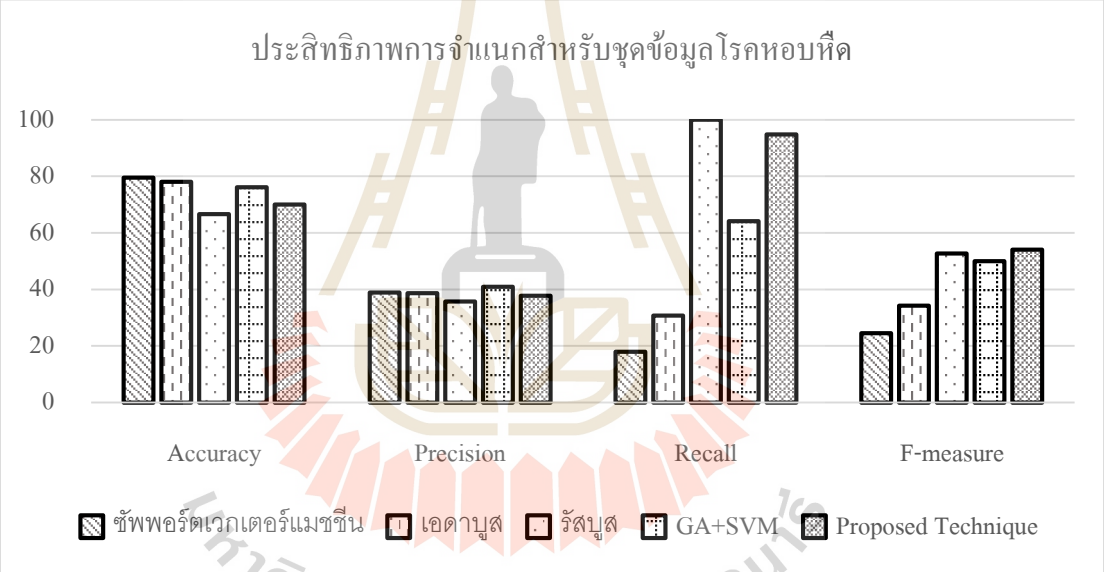
ตารางที่ 4.31 เมตริกซ์วัดประสิทธิภาพของเทคนิคที่นำเสนอ สำหรับชุดข้อมูลสังเคราะห์

		Actual	
		Positive	Negative
Prediction	Positive	23	5
	Negative	5	147

จากผลการทดลองในตารางที่ 4.27 พบว่าอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนแบบดั้งเดิมให้ประสิทธิรูปที่ดีกว่าการใช้อัลกอริทึมเอดาบูสดีกว่าเทคนิคที่นำเสนอและดีกว่าการใช้อัลกอริทึมรัสบูสรวมไปถึงดีกว่าการใช้ขั้นตอนวิธีเชิงพันธุกรรมร่วมกับอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีน ทางด้านค่าความแม่นยำในการจำแนกที่ 88.00%, 87.50%, 85.00%, 78.50% และ 76.00% ตามลำดับ สำหรับค่าความเที่ยง อัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนแบบดั้งเดิมมีประสิทธิรูปที่ 100.00% อัลกอริทึมเอดาบูสที่ 88.89% เทคนิคที่นำเสนอที่ 47.92% ขั้นตอนวิธีเชิงพันธุกรรมร่วมกับอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนที่ 33.33% และอัลกอริทึมรัสบูสที่ 32.56% เมื่อพิจารณาที่ค่าระลึกหรือค่าความไว และค่าการวัดเอฟ พบว่าเทคนิคที่นำเสนอมีประสิทธิภาพดีกว่าการใช้ขั้นตอนวิธีเชิงพันธุกรรมร่วมกับอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีน การใช้อัลกอริทึมรัสบูส การใช้อัลกอริทึมเอดาบูส และการใช้อัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีน ที่ค่าระลึกหรือค่าความไวที่ 82.14%, 71.14%, 50.00%, 25.00% และ 14.29% ตามลำดับ และค่าการวัดเอฟที่ 60.53%, 45.39%,

39.44%, 39.03% และ 25.01% ตามลำดับ โดยค่าพารามิเตอร์เริ่มต้นสำหรับอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนมีค่า cost = 1, epsilon = 0.1 และพารามิเตอร์ gamma = 0.0625 เมื่อทำการหาค่าพารามิเตอร์ที่เหมาะสมด้วยเทคนิคที่นำเสนอพบว่าพารามิเตอร์ที่เหมาะสมที่สุดคือ cost = 56.97464, epsilon = 1.720305 และพารามิเตอร์ gamma = 0.004282877

สำหรับชุดข้อมูลโรคหอบหืด ประสิทธิภาพการจำแนกของแต่ละอัลกอริทึมแสดงดังรูปที่ 4.7 และมีรายละเอียดประสิทธิภาพตามเกณฑ์ต่าง ๆ แสดงดังตารางที่ 4.32 และเมตริกซ์วัดประสิทธิภาพของอัลกอริทึมเอดาบูส แสดงดังตารางที่ 4.33 เมตริกซ์วัดประสิทธิภาพของอัลกอริทึมรัสบูส แสดงดังตารางที่ 4.34 เมตริกซ์วัดประสิทธิภาพของการใช้ขั้นตอนวิธีเชิงพันธุกรรมร่วมกับอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีน แสดงดังตารางที่ 4.35 และเมตริกซ์วัดประสิทธิภาพของเทคนิคที่นำเสนอแสดงดังตารางที่ 4.36



รูปที่ 4.7 ประสิทธิภาพการจำแนกด้วยเทคนิคต่าง ๆ ของชุดข้อมูลโรคหอบหืด

ตารางที่ 4.32 ประสิทธิภาพการจำแนกแต่ละอัลกอริทึมของชุดข้อมูลโรคหอบหืด

อัลกอริทึม	Accuracy (%)	Precision (%)	Recall (%)	F-measure (%)
ซัพพอร์ตเวกเตอร์แมชชีน	**79.52**	38.89	17.95	24.56
เอดาบูส	78.10	38.71	30.77	34.29
รัสบูส	66.67	35.78	**100.00**	52.70
Genetic Algorithm + SVM	76.19	**40.98**	64.10	50.00
Proposed Technique	70.00	37.76	94.87	**54.02**

ตารางที่ 4.33 เมตริกซ์วัดประสิทธิภาพของอัลกอริทึมเอดาบูส สำหรับชุดข้อมูลโรคหอบหืด

		Actual	
		Positive	Negative
Prediction	Positive	12	19
	Negative	27	152

ตารางที่ 4.34 เมตริกซ์วัดประสิทธิภาพของอัลกอริทึมรัสบูส สำหรับชุดข้อมูลโรคหอบหืด

		Actual	
		Positive	Negative
Prediction	Positive	39	70
	Negative	0	101

ตารางที่ 4.35 เมตริกซ์วัดประสิทธิภาพของขั้นตอนวิธีเชิงพันธุกรรมร่วมกับอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีน สำหรับชุดข้อมูลโรคหอบหืด

		Actual	
		Positive	Negative
Prediction	Positive	25	36
	Negative	14	135

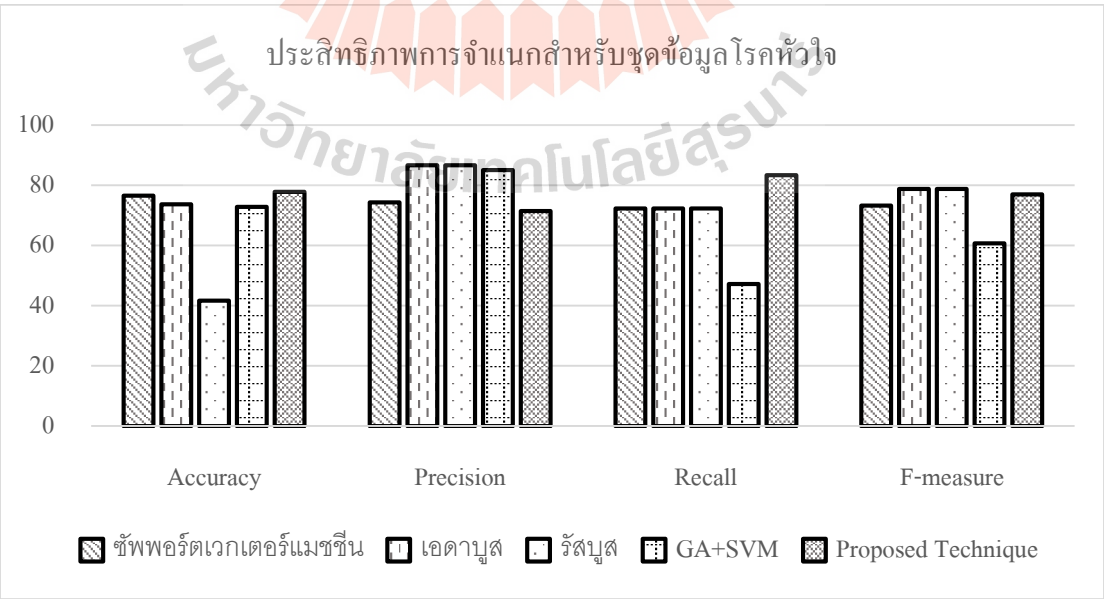
ตารางที่ 4.36 เมตริกซ์วัดประสิทธิภาพของเทคนิคที่นำเสนอ สำหรับชุดข้อมูลโรคหอบหืด

		Actual	
		Positive	Negative
Prediction	Positive	37	61
	Negative	2	110

จากผลการทดลองในตารางที่ 4.31 พบว่าเมื่อพิจารณาที่ค่าความแม่นยำในการจำแนกอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนให้ประสิทธิรูปที่ดีกว่าการใช้อัลกอริทึมเอดาบูส การใช้ขั้นตอนวิธีเชิงพันธุกรรมร่วมกับอัลกอริทึมซัพพอร์ตวกเตอร์แมชชีน เทคนิคที่นำเสนอและการใช้อัลกอริทึมรัสบูสที่ค่าความแม่นยำ 79.52%, 78.10%, 76.19%, 70.00% และ 66.67% ตามลำดับ สำหรับค่าความเที่ยงพบว่าการใช้ขั้นตอนวิธีเชิงพันธุกรรมร่วมกับอัลกอริทึมซัพพอร์ตเวกเตอร์แม

ชชีนมีประสิทธิภาพสูงที่สุดที่ 40.98% อัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนที่ 38.89% อัลกอริทึมเอดาบูสที่ 38.71% เทคนิคที่นำเสนอที่ 37.76% และอัลกอริทึมรัสบูสที่ 35.78% สำหรับค่าความไวหรือค่าระลึกอัลกอริทึมรัสบูส มีประสิทธิรูปที่ดีที่สุดที่ 100.00% รองลงมาคือเทคนิคที่นำเสนอที่ 94.87% การใช้ขั้นตอนวิธีเชิงพันธุกรรมร่วมกับอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนที่ 64.10% การใช้อัลกอริทึมเอดาบูสที่ 30.77% และที่ด้อยที่สุดคือการใช้อัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนที่ 17.95% ขณะที่ค่าการวัดเอฟพบว่าเทคนิคที่นำเสนอให้ประสิทธิภาพดีกว่าวิธีการอื่น ๆ ที่ 54.02% รองลงมาคือการใช้อัลกอริทึมรัสบูสที่ 52.70% การใช้ขั้นตอนวิธีเชิงพันธุกรรมร่วมกับอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนที่ 50.00% การใช้อัลกอริทึมเอดาบูสที่ 34.29% และการใช้อัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนที่ 24.56% โดยค่าพารามิเตอร์เริ่มต้นสำหรับอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนมีค่า cost = 1, epsilon = 0.1 และพารามิเตอร์ gamma = 0.083333 เมื่อทำการหาค่าพารามิเตอร์ที่เหมาะสมด้วยเทคนิคที่นำเสนอพบว่าพารามิเตอร์ที่เหมาะสมที่สุดคือ cost = 8.374748, epsilon = 0.8360468 และพารามิเตอร์ gamma = 0.2513825

ประสิทธิภาพการจำแนกของชุดข้อมูลโรคหัวใจแสดงดังรูปที่ 4.8 และมีรายละเอียดประสิทธิภาพตามเกณฑ์ต่าง ๆ แสดงดังตารางที่ 4.37 และเมตริกซ์วัดประสิทธิภาพของอัลกอริทึมเอดาบูส แสดงดังตารางที่ 4.38 เมตริกซ์วัดประสิทธิภาพของอัลกอริทึมรัสบูส แสดงดังตารางที่ 4.39 เมตริกซ์วัดประสิทธิภาพของการใช้ขั้นตอนวิธีเชิงพันธุกรรมร่วมกับอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีน แสดงดังตารางที่ 4.40 และเมตริกซ์วัดประสิทธิภาพของเทคนิคที่นำเสนอแสดงดังตารางที่ 4.41



รูปที่ 4.8 ประสิทธิภาพการจำแนกด้วยเทคนิคต่าง ๆ ของชุดข้อมูลโรคหัวใจ

ตารางที่ 4.37 ประสิทธิภาพการจำแนกแต่ละอัลกอริทึมของชุดข้อมูลโรคหัวใจ

อัลกอริทึม	Accuracy (%)	Precision (%)	Recall (%)	F-measure (%)
ซัพพอร์ตเวกเตอร์แมชชีน	76.54	74.29	72.22	73.24
เอดาบูส	73.63	**86.67**	72.22	**78.79**
รัสบูส	41.63	**86.67**	72.22	**78.79**
Genetic Algorithm + SVM	72.84	85.00	47.22	60.71
Proposed Technique	**77.78**	71.43	**83.33**	76.92

ตารางที่ 4.38 เมตริกซ์วัดประสิทธิภาพของอัลกอริทึมเอดาบูส สำหรับชุดข้อมูลโรคหัวใจ

		Actual	
		Positive	Negative
Prediction	Positive	26	4
	Negative	10	41

ตารางที่ 4.39 เมตริกซ์วัดประสิทธิภาพของอัลกอริทึมรัสบูส สำหรับชุดข้อมูลโรคหัวใจ

		Actual	
		Positive	Negative
Prediction	Positive	26	4
	Negative	10	41

ตารางที่ 4.40 เมตริกซ์วัดประสิทธิภาพของขั้นตอนวิธีเชิงพันธุกรรมร่วมกับอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีน สำหรับชุดข้อมูลโรคหัวใจ

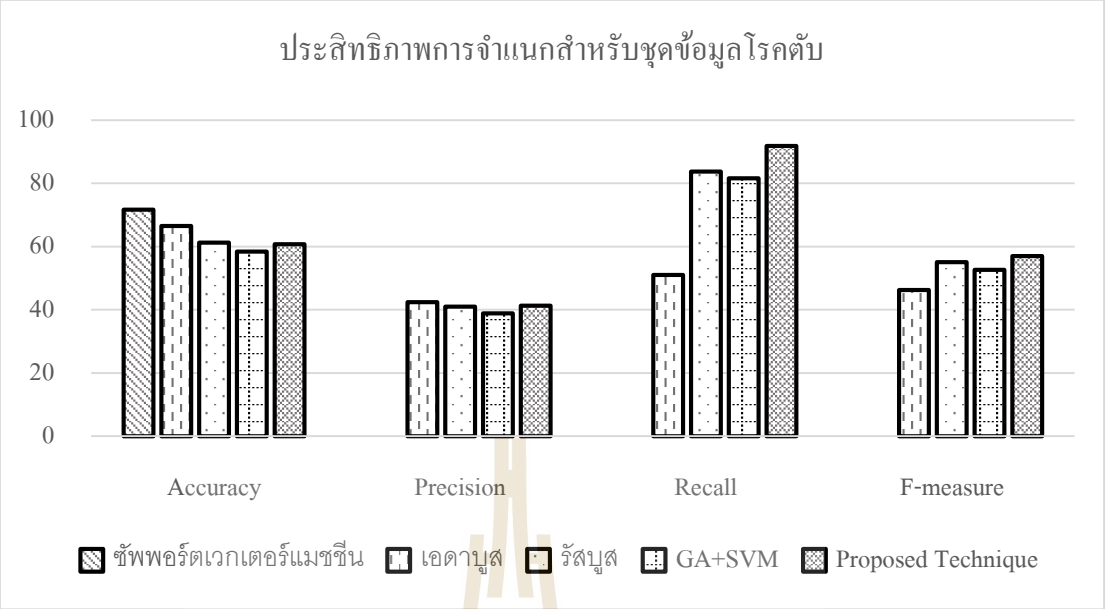
		Actual	
		Positive	Negative
Prediction	Positive	17	3
	Negative	19	42

ตารางที่ 4.41 เมตริกซ์วัดประสิทธิภาพของเทคนิคที่นำเสนอ สำหรับชุดข้อมูลโรคหัวใจ

		Actual	
		Positive	Negative
Prediction	Positive	30	12
	Negative	6	33

จากผลการทดลองในตารางที่ 4.37 พบว่าเทคนิคที่นำเสนอมีความแม่นยำในการจำแนกดีที่สุดที่ 77.78% รองลงมาคือการใช้อัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนที่ 76.54% การใช้อัลกอริทึมเอดาบูสที่ 73.63% การใช้ขั้นตอนวิธีเชิงพันธุกรรมร่วมกับอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนที่ 72.84% และอัลกอริทึมรัสบูสที่ 41.63% เมื่อพิจารณาที่ค่าระลึกหรือค่าความไวพบว่าเทคนิคที่นำเสนอมีประสิทธิภาพดีว่าการใช้อัลกอริทึมอื่น ๆ ที่ 83.33% ขณะที่อัลกอริทึมอื่น ๆ มีประสิทธิภาพอยู่ที่ 72.22% และการใช้ขั้นตอนวิธีเชิงพันธุกรรมร่วมกับอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนที่ 47.22% สำหรับค่าความเที่ยงพบว่าการใช้อัลกอริทึมเอดาบูสและอัลกอริทึมรัสบูสให้ประสิทธิภาพสูงที่สุดที่ 86.67% รองลงมาคือการใช้ขั้นตอนวิธีเชิงพันธุกรรมร่วมกับอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนที่ 85.00% การใช้อัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนที่ 74.29% และเทคนิคที่นำเสนอที่ 71.43% และค่าการวัดเอฟพบว่าอัลกอริทึมเอดาบูส และอัลกอริทึมรัสบูส มีประสิทธิภาพดีที่สุดที่ 78.79% เทคนิคที่นำเสนอมีประสิทธิรูปที่ 76.92% การใช้อัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนที่ 73.24% และการใช้ขั้นตอนวิธีเชิงพันธุกรรมร่วมกับอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนมีประสิทธิภาพด้อยที่สุดที่ 60.71% โดยค่าพารามิเตอร์เริ่มต้นสำหรับอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนมีค่า cost = 1, epsilon = 0.1 และพารามิเตอร์ gamma = 0.0714286 เมื่อทำการหาค่าพารามิเตอร์ที่เหมาะสมด้วยเทคนิคที่นำเสนอพบว่าพารามิเตอร์ที่เหมาะสมที่สุดคือ cost = 4.319116, epsilon = 0.012312 และพารามิเตอร์ gamma = 0.182505

สำหรับชุดข้อมูลโรคตับ ประสิทธิภาพการจำแนกของแต่ละอัลกอริทึมในด้านของความแม่นยำในการจำแนก ค่าความเที่ยง ค่าระลึกหรือค่าความไว และค่าการวัดเอฟ แสดงดังรูปที่ 4.9 และมีรายละเอียดประสิทธิภาพตามเกณฑ์ต่าง ๆ แสดงดังตารางที่ 4.42 และเมตริกซ์วัดประสิทธิภาพของอัลกอริทึมเอดาบูส แสดงดังตารางที่ 4.43 เมตริกซ์วัดประสิทธิภาพของอัลกอริทึมรัสบูส แสดงดังตารางที่ 4.44 เมตริกซ์วัดประสิทธิภาพของการใช้ขั้นตอนวิธีเชิงพันธุกรรมร่วมกับอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีน แสดงดังตารางที่ 4.45 และเมตริกซ์วัดประสิทธิภาพของเทคนิคที่นำเสนอแสดงดังตารางที่ 4.46



รูปที่ 4.9 ประสิทธิภาพการจำแนกด้วยเทคนิคต่าง ๆ ของชุดข้อมูลโรคตับ

ตารางที่ 4.42 ประสิทธิภาพการจำแนกแต่ละอัลกอริทึมของชุดข้อมูลโรคตับ

อัลกอริทึม	Accuracy (%)	Precision (%)	Recall (%)	F-measure (%)
ซัพพอร์ตเวกเตอร์แมชชีน	**71.68**	00.00	00.00	00.00
เอดาบูส	66.47	**42.37**	51.02	46.29
รัสบูส	61.27	41.00	83.67	55.03
Genetic Algorithm + SVM	58.38	38.84	81.63	52.64
Proposed Technique	60.69	41.28	**91.84**	**56.96**

ตารางที่ 4.43 เมตริกซ์วัดประสิทธิภาพของอัลกอริทึมเอดาบูส สำหรับชุดข้อมูลโรคตับ

		Actual	
		Positive	Negative
Prediction	Positive	25	34
	Negative	24	90

ตารางที่ 4.44 เมตริกซ์วัดประสิทธิภาพของอัลกอริทึมรัสบูส สำหรับชุดข้อมูลโรคตับ

		Actual	
		Positive	Negative
Prediction	Positive	41	59
	Negative	8	65

ตารางที่ 4.45 เมตริกซ์วัดประสิทธิภาพของขั้นตอนวิธีเชิงพันธุกรรมร่วมกับอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีน สำหรับชุดข้อมูลโรคตับ

		Actual	
		Positive	Negative
Prediction	Positive	40	61
	Negative	9	63

ตารางที่ 4.46 เมตริกซ์วัดประสิทธิภาพของเทคนิคที่นำเสนอ สำหรับชุดข้อมูลโรคตับ

		Actual	
		Positive	Negative
Prediction	Positive	45	64
	Negative	4	60

จากตารางที่ 4.42 เมื่อเปรียบเทียบประสิทธิภาพการจำแนกในด้านของค่าความแม่นยำ พบว่าการใช้อัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนมีประสิทธิรูปที่ดีกว่าการใช้อัลกอริทึมเอดาบูส การใช้อัลกอริทึมรัสบูส การใช้เทคนิคที่นำเสนอ และการใช้ขั้นตอนวิธีเชิงพันธุกรรมร่วมกับอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนตามลำดับ ที่ 71.68%, 66.47%, 61.27%, 60.69% และ 58.38% แต่เมื่อพิจารณาในด้านอื่น ๆ พบว่าอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนไม่สามารถจำแนกข้อมูลจากคลาสส่วนน้อยได้ ในขณะที่ค่าความเที่ยงพบว่าอัลกอริทึมเอดาบูสมีประสิทธิภาพสูงที่สุดที่ 42.37% เทคนิคที่นำเสนอมีประสิทธิภาพรองลงมาที่ 41.28% อัลกอริทึมรัสบูสที่ 41.00% และการใช้ขั้นตอนวิธีเชิงพันธุกรรมร่วมกับอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนที่ 38.84% สำหรับค่าระลึกหรือค่าความไวพบว่าเทคนิคที่นำเสนอมีประสิทธิภาพดีกว่าอัลกอริทึมรัสบูส การใช้ขั้นตอนวิธีเชิงพันธุกรรมร่วมกับอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนและอัลกอริทึมเอดาบูสตามลำดับ ที่ 91.84%, 83.67%, 81.63% และ 51.02% เมื่อพิจารณาที่ค่าการวัดเอฟพบว่าเทคนิคที่นำเสนอมี

ประสิทธิภาพดีที่สุดที่ 56.96% รองลงมาคืออัลกอริทึมรัสบูสที่ 55.03% การใช้ขั้นตอนวิธีเชิงพันธุกรรมร่วมกับอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนที่ 52.64% และอัลกอริทึมเอดาบูสที่ 46.29% โดยค่าพารามิเตอร์เริ่มต้นสำหรับอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนมีค่า cost = 1, epsilon = 0.1 และพารามิเตอร์ gamma = 0.090909 เมื่อทำการหาค่าพารามิเตอร์ที่เหมาะสมด้วยเทคนิคที่นำเสนอ พบว่าพารามิเตอร์ที่เหมาะสมที่สุดคือ cost = 9.0821964, epsilon = 1.0846983 และพารามิเตอร์ gamma = 0.0154112

4.5 อภิปรายผล

จากผลการทดสอบประสิทธิภาพการจำแนกข้อมูลไม่สมดุลโดยใช้การปรับปรุงข้อมูลร่วมกับการหาค่าพารามิเตอร์ที่เหมาะสมด้วยขั้นตอนวิธีเชิงพันธุกรรมที่มีการเริ่มต้นใหม่ ได้ทำการทดสอบกับข้อมูลจำนวน 4 ชุดข้อมูล โดยแต่ละชุดข้อมูลประกอบไปด้วยคลาส 2 คลาส กระบวนการปรับสมดุลข้อมูลถูกนำมาใช้เพื่อปรับจำนวนข้อมูลในแต่ละคลาสให้มีขนาดใกล้เคียงกันเพื่อไม่ให้เกิดการเอนเอียงไปทางคลาสที่มีจำนวนข้อมูลมากกว่า (คลาสส่วนมาก) ก่อนนำข้อมูลไปหาพารามิเตอร์ที่เหมาะสมด้วยวิธีเชิงพันธุกรรมที่มีการเริ่มต้นใหม่เพื่อสร้างโมเดลในการจำแนกประเภท และประเมินประสิทธิภาพการจำแนก สามารถสรุปผลการทดสอบเปรียบเทียบได้ดังนี้

1) การปรับสมดุลข้อมูลเรียนรู้ เพื่อเตรียมข้อมูลก่อนการนำไปหาพารามิเตอร์ที่เหมาะสมด้วยขั้นตอนวิธีเชิงพันธุกรรมที่มีการเริ่มต้นใหม่ ช่วยให้โมเดลการจำแนกมีความสามารถในการจำแนกข้อมูลจากคลาสส่วนน้อยได้ดียิ่งขึ้น (สามารถจำแนกคลาสส่วนน้อยได้แม่นยำมากขึ้น)

2) การใช้อัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนแบบดั้งเดิม (ไม่ปรับสมดุลข้อมูล รวมถึงใช้ค่าพารามิเตอร์เริ่มต้น) ให้ประสิทธิรูปที่ดีเมื่อวัดประสิทธิภาพด้วยค่าความแม่นยำในการจำแนก เนื่องจากหากทำนายข้อมูลทดสอบทั้งหมดให้เป็นคลาสส่วนมากทั้งหมดก็ส่งผลให้ความแม่นยำในการจำแนกสูง แต่ในขณะที่ความสามารถในการจำแนกข้อมูลจากคลาสส่วนน้อยจะต่ำมาก

3) อัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนมีประสิทธิรูปที่ดีมากในด้านค่าความเที่ยง เนื่องจากจำนวนที่จำแนกประเภทข้อมูลเป็นคลาสส่วนน้อยมีการทำนายในปริมาณที่น้อยและในจำนวนที่ทำนายสามารถทำนายได้ถูกจึงส่งผลให้ค่าความเที่ยงมีค่าสูง แต่เมื่อพิจารณาว่าสามารถจำแนกจำนวนข้อมูลที่อยู่ในคลาสส่วนน้อยได้ทั้งหมดพบว่ามีประสิทธิรูปที่ต่ำ (พิจารณาจากค่าระลึกหรือค่าความไว)

4) เมื่อพิจารณาความสามารถในการจำแนกประเภทข้อมูลจากคลาสส่วนน้อยโดยรวม พบว่าเทคนิคที่นำเสนอมีประสิทธิรูปที่ดีที่สุดเป็นจำนวน 3 ชุดข้อมูล จากทั้งหมด 4 ชุดข้อมูล โดยพิจารณาจากค่าการวัดเอฟ (ค่าการวัดเอฟเป็นอัตราเฉลี่ยระหว่างจำนวนที่จำแนกข้อมูลว่าเป็นคลาส

ส่วนน้อยได้ถูกต้อง และจำนวนข้อมูลที่อยู่ในคลาสส่วนน้อยทั้งหมด โดยพิจารณาจากค่าความเที่ยงและค่าระลึกหรือค่าความไว) หากค่าการวัดเอฟมีค่าที่สูงหมายความว่าโมเดลมีจำนวนการจำแนกประเภทข้อมูลที่อยู่ในคลาสส่วนน้อยในปริมาณที่สูง และการจำแนกข้อมูลในคลาสส่วนน้อยนั้นแม่นยำสูงด้วย

5) เมื่อพิจารณาประสิทธิภาพการจำแนกข้อมูลไม่สมดุล พบว่าเทคนิคที่นำเสนอเหมาะสำหรับนำไปจำแนกประเภทข้อมูลไม่สมดุลที่มีระดับความไม่สมดุลตั้งแต่ 2 ขึ้นไป เมื่อข้อมูลมีระดับความไม่สมดุลต่ำกว่า 2 พบว่าเทคนิคที่นำเสนอให้ประสิทธิภาพด้อยกว่าอัลกอริทึมเอดาบูสและอัลกอริทึมรัสบูส

บทที่ 5
สรุปผลการวิจัยและข้อเสนอแนะ

ในปัจจุบันข้อมูลถูกเก็บให้อยู่ในรูปแบบดิจิตอลซึ่งสามารถใช้เทคนิคการทำเหมืองข้อมูลเพื่อหารูปแบบ หรือหาความสัมพันธ์ที่ซ่อนเร้นอยู่ภายในข้อมูลเหล่านั้น โดยหนึ่งในเทคนิคการทำเหมืองข้อมูลที่ได้รับความนิยมคือการจำแนกประเภทข้อมูล ซึ่งการจำแนกประเภทข้อมูลสามารถจำแนกด้วยการประยุกต์ใช้อัลกอริทึมต่าง ๆ มากมาย แต่อัลกอริทึมเหล่านั้นจะทำงานได้อย่างเต็มประสิทธิภาพก็ต่อเมื่อข้อมูลที่นำมาใช้มีความสมดุลกัน หากข้อมูลเกิดความไม่สมดุลจะทำให้โมเดลการจำแนกประเภทข้อมูลมีความเอนเอียงไปในกลุ่มข้อมูลที่มีจำนวนข้อมูลมากกว่า ซึ่งเรียกกลุ่มข้อมูลประเภทนี้ว่าข้อมูลไม่สมดุล โดยที่ข้อมูลที่มีจำนวนมากกว่าข้อมูลอีกกลุ่มหนึ่งจะเรียกว่าข้อมูลคลาสส่วนมาก และข้อมูลที่มีจำนวนน้อยกว่าข้อมูลในอีกกลุ่มหนึ่งจะเรียกว่าข้อมูลคลาสส่วนน้อย จากปัญหาข้อมูลไม่สมดุลสามารถแก้ปัญหาได้ด้วยการปรับปรุงข้อมูลให้สมดุล ไม่ว่าจะเป็นการสุ่มลดข้อมูลจากคลาสส่วนมากลง หรือจะทำการสุ่มเพิ่มข้อมูลจากคลาสส่วนน้อยขึ้น หรืออาจจะใช้ทั้งสองวิธีที่กล่าวมาข้างต้นทำงานร่วมกัน ซึ่งงานวิจัยบางประเภทมุ่งเน้นไปที่การปรับปรุงอัลกอริทึมในการจำแนก โดยทำการปรับค่าน้ำหนักให้แก่ข้อมูลที่จำแนกผิดประเภทให้มีความสำคัญเพิ่มมากขึ้นเพื่อเพิ่มโอกาสในการจำแนกได้ถูกประเภทมากยิ่งขึ้น ซึ่งวิธีการดังกล่าวสามารถเพิ่มประสิทธิภาพการจำแนกข้อมูลจากคลาสส่วนน้อยได้ดียิ่งขึ้น หรือจะเป็นการปรับปรุงอัลกอริทึมด้วยการปรับค่าพารามิเตอร์เพื่อเพิ่มประสิทธิภาพในการจำแนกประเภทข้อมูลได้ดียิ่งขึ้น

ดังนั้นวัตถุประสงค์ของวิทยานิพนธ์ฉบับนี้คือการเพิ่มประสิทธิภาพในการจำแนกข้อมูลจากคลาสส่วนน้อยให้สามารถจำแนกข้อมูลจากคลาสส่วนน้อยได้ดียิ่งขึ้น จึงเสนอเทคนิคการปรับปรุงข้อมูล ร่วมกับการหาค่าพารามิเตอร์ที่เหมาะสมด้วยขั้นตอนวิธีเชิงพันธุกรรมที่มีการเริ่มต้นใหม่ โดยในขั้นตอนการปรับปรุงข้อมูลจะใช้เทคนิคการผสมผสานระหว่างการสุ่มลดข้อมูลจากคลาสส่วนมากลง และสังเคราะห์ข้อมูลจากคลาสส่วนน้อยเพิ่มขึ้นด้วยเทคนิค SMOTE เมื่อข้อมูลมีความสมดุลแล้วจะหาค่าพารามิเตอร์ที่เหมาะสมด้วยขั้นตอนวิธีเชิงพันธุกรรมที่มีการเริ่มต้นใหม่ โดยทำการหาค่าพารามิเตอร์จำนวน 3 ค่าพารามิเตอร์ ได้แก่ พารามิเตอร์ Cost หรือ C พารามิเตอร์ Epsilon และพารามิเตอร์ Gamma โดยในขั้นตอนการหาค่าพารามิเตอร์ที่เหมาะสมด้วยการใช้ขั้นตอนวิธีเชิงพันธุกรรมที่มีการเริ่มต้นใหม่

ได้กำหนดพารามิเตอร์เริ่มต้นของขั้นตอนวิธีเชิงพันธุกรรมที่มีการเริ่มต้นใหม่ดังนี้ ค่าพารามิเตอร์ Cost หรือ C อยู่ระหว่าง 10^{-4} ถึง 10^{2} ค่าพารามิเตอร์ Epsilon อยู่ระหว่าง 10^{-2} ถึง 2 ค่าพารามิเตอร์ Gamma อยู่ระหว่าง 10^{-3} ถึง 2 จำนวนประชากรในการสุ่มสร้าง 100 ประชากร เลือกประชากรระดับหัวกะทิจำนวน 10 ประชากร หากประชากรรุ่นใหม่มีประสิทธิภาพด้อยกว่าประชากรรุ่นเก่าติดต่อกันจำนวน 2 รอบให้สุ่มสร้างประชากรเริ่มต้นใหม่โดยใช้ประชากรระดับหัวกะทิเป็นประชากรเริ่มต้นด้วย จำนวนรอบในการทำงานทั้งหมด 100 รอบ เมื่อได้ค่าพารามิเตอร์ที่เหมาะสมที่สุดแล้วหลังจากนั้นจะทำการสร้างโมเดลการจำแนกประเภทด้วยการใช้พารามิเตอร์ที่ดีที่สุดร่วมกับอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีน หลังจากนั้นประเมินประสิทธิภาพการจำแนกประเภทข้อมูลไม่สมดุลโดยเปรียบเทียบกับอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนแบบดั้งเดิม อัลกอริทึมเอดาบูส และอัลกอริทึมรัสบูส โดยพบว่าการปรับสมดุลข้อมูลและการปรับค่าพารามิเตอร์ช่วยให้สามารถจำแนกข้อมูลจากคลาสส่วนน้อยได้ดียิ่งขึ้น

5.1 สรุปผลการวิจัย

จากผลการทดสอบประสิทธิภาพในการจำแนกข้อมูลที่แสดงในบทที่ 4 นั้น วิทยานิพนธ์ฉบับนี้ได้ใช้ข้อมูลไม่สมดุลจำนวนทั้งหมด 4 ชุดข้อมูล ได้แก่ ชุดข้อมูลสังเคราะห์ 1 ชุดข้อมูล ชุดข้อมูลโรคหอบหืด ชุดข้อมูลโรคหัวใจ และชุดข้อมูลโรคตับ โดยชุดข้อมูลสังเคราะห์ทำการสังเคราะห์ด้วยโปรแกรม R Studio 1.0.143 ชุดข้อมูลโรคหอบหืดได้รับจากโรงพยาบาลแห่งหนึ่งในจังหวัดนครราชสีมา ชุดข้อมูลโรคหัวใจและชุดข้อมูลโรคตับได้รับจากฐานข้อมูลมาตรฐาน สามารถสรุปได้ดังต่อไปนี้

1) จากการพัฒนาขั้นตอนการจำแนกข้อมูลไม่สมดุล โดยอาศัยการปรับปรุงข้อมูลไม่สมดุลด้วยการใช้การผสมผสานระหว่างการสุ่มลดข้อมูลจากคลาสส่วนมากลง และสังเคราะห์เพิ่มข้อมูลจากคลาสส่วนน้อยขึ้น หลังจากนั้นนำข้อมูลที่สมดุลไปหาค่าพารามิเตอร์ที่เหมาะสมแล้วนำไปสร้างโมเดลจำแนกประเภทข้อมูล สามารถเพิ่มประสิทธิภาพการจำแนกประเภทข้อมูลจากคลาสส่วนน้อยให้มีประสิทธิภาพสูงขึ้นมากกว่าการใช้ข้อมูลดั้งเดิม

2) ในขั้นตอนการหาค่าพารามิเตอร์ที่เหมาะสมด้วยขั้นตอนวิธีเชิงพันธุกรรมที่มีการเริ่มต้นใหม่ สามารถหาค่าพารามิเตอร์ที่เหมาะสมสำหรับการจำแนกประเภทข้อมูลด้วยอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนของแต่ละชุดข้อมูล โดยพารามิเตอร์ที่ได้มานั้นสามารถช่วยเพิ่มประสิทธิภาพในการจำแนกด้วยอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนได้ดียิ่งขึ้น

3) เมื่อเปรียบเทียบกับเทคนิคที่ใช้ในการจำแนกประเภทข้อมูลในปัจจุบัน ได้แก่ อัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีน การใช้ขั้นตอนวิธีเชิงพันธุกรรมร่วมกับอัลกอริทึมซัพพอร์ต

เวกเตอร์แมชชีน อัลกอริทึมเอดาบูส และอัลกอริทึมรัสบูส เทคนิคที่นำเสนอมีประสิทธิภาพโดยรวมของการจำแนกประเภทข้อมูลจากคลาสส่วนน้อยดีที่สุดเป็นจำนวน 3 ชุดข้อมูล จากทั้งหมด 4 ชุดข้อมูล เมื่อพิจารณาค่าการวัดเอฟ จะพบว่าเทคนิคที่นำเสนอเหมาะสำหรับการจำแนกประเภทข้อมูลไม่สมดุลที่มีระดับความไม่สมดุลตั้งแต่ 2 ขึ้นไป ในขณะที่เมื่อพิจารณาที่ค่าความแม่นยำในการจำแนก พบกว่าอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนมีประสิทธิภาพสูงที่สุด แต่ความสามารถในการจำแนกประเภทข้อมูลจากคลาสส่วนน้อยอยู่ในระดับที่ต่ำ

5.2 ปัญหาและข้อเสนอแนะ

การจำแนกประเภทข้อมูลของคลาสส่วนน้อยบนข้อมูลที่ไม่สมดุลนั้น ยังไม่มีวิธีที่สามารถแก้ปัญหาได้อย่างแน่นอน โดยจะเห็นจากวิธีที่นำเสนอในวิทยานิพนธ์ฉบับนี้เมื่อมีการปรับสมดุลข้อมูลจะส่งผลให้ประสิทธิภาพทางด้านความแม่นยำในการจำแนกประเภทลดลง แต่สามารถเพิ่มประสิทธิภาพในการจำแนกข้อมูลจากคลาสส่วนน้อยได้ดียิ่งขึ้น

สำหรับอีกหนึ่งปัญหาคือการหาค่าพารามิเตอร์ด้วยขั้นตอนวิธีเชิงพันธุกรรมที่มีการเริ่มต้นใหม่ ในขั้นตอนการหาค่าพารามิเตอร์จะต้องมีการกำหนดขอบเขตการค้นหาของแต่ละพารามิเตอร์ด้วยตนเอง หากกำหนดขอบเขตของพารามิเตอร์ไม่ครอบคลุม หรือหากกำหนดขอบเขตพารามิเตอร์กว้างมากเกินไปก็จะทำให้พารามิเตอร์ที่ได้รับมาไม่ใช่พารามิเตอร์ที่ดีที่สุด และหากชุดข้อมูลมีจำนวนข้อมูลในปริมาณมากจะส่งผลให้ระยะเวลาในการทำงานในแต่ละรอบของขั้นตอนวิธีเชิงพันธุกรรมที่มีการเริ่มต้นใหม่จะใช้เวลานานยิ่งขึ้น

ดังนั้นสิ่งที่จะเสนอแนะคือ การกำหนดขอบเขตของค่าพารามิเตอร์ต่าง ๆ แบบอัตโนมัติเพื่อลดระยะเวลาในการทำงาน และการปรับปรุงขั้นตอนวิธีเชิงพันธุกรรมที่มีการเริ่มต้นใหม่ ให้สามารถสิ้นสุดการทำงานเมื่อได้ค่าความเหมาะสมของประชากรตามที่ต้องการ

รายการอ้างอิง

กิตติพงษ์ ชมบุญ. (2016). เทคนิคการค้นหาคลาสที่ค้นพบได้ยากสำหรับข้อมูลที่ขนาดแตกต่างกันมาก. **วิทยานิพนธ์วิศวกรรมศาสตร์ดุษฎีบัณฑิต.** สาขาวิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเทคโนโลยีสุรนารี.

พงศกร ธีรรัศมี. (2015). วิธีการหาค่า เค ที่เหมาะสมในการจำแนกแบบเคเนียร์เรสเนเบอร์กับข้อมูลทางการแพทย์. **วิทยานิพนธ์วิศวกรรมศาสตร์มหาบัณฑิต.** สาขาวิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเทคโนโลยีสุรนารี.

Batista G. E. A. P. A., Prati R. C., and Monard M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. **SIGKDD Explorations.** 6(1).

Bekkar M. (2009). "Developpement d'un modele de prediction du churn clientele en telecommunication". **Master 2 theis stat et econometrie. Unive Toulouse 1 Sciences Sociales**.

Boonchuay K., Sinapiromsaran K. and Lursinsap C. (2011). Minority Split and Gain Ratio for a Class Imbalance. **Eight International Conference on Fuzzy Systems and Knowledge Discovery (FSKD) 2011**. pp. 2060-3064.

Bosch A., Zisserman X., and Muñoz. (2007). Image Classification Using Random Forests and Ferns. **IEEE International Conference on Computer Vision.** Rio de Janeiro. Brazil.

Breiman L. (1996). **Bagging predictors. Machine learning.** 24(2). 123-140.

Bressoux P. (2008). Modelisation statistique appliquee aux sciences sociales. **De Boeck, Bruxelle.**

Buckland M. and Gey F. (1994). The Relationship Between Recall and Precision. **Journal of the American Society for Information Science.** 45(1): 12-19.

Chawla N.V., Bowyer K.W., Hall L.O., and Kegelmeyer W.P. (2002). SMOTE: synthetic minority over-sampling technique. **Journal of Artificial Intelligence Research.** 16:321–357.

Chawla N.V. (2003). C4.5 and Imbalanced Data Sets: Investigating the Effect of Sampling Method, Probabilistic Estimate, and Decision Tree Structure. **Proc. Int'l Conf. Machine Learning, Workshop Learning from Imbalanced Data Sets II.**

Chawla N.V., Japkowicz N., and Kotcz A. (2004). Editorial: special issue on learning from imbalanced data sets. **ACM Sigkdd Explorations Newsletter.** 6(1): 1-6.

Chen K. H., Wang K. J., Wang K. M., and Angelia M. A. (2014). Applying particle swarm optimization-based decision tree classifier for cancer classification on gene expression data. **Applied Soft Computing**. 24. 773-780

Chistianini N., and Shawe-Taylor J. (2000). An introduction to support vector machines, and other kernel-based learning methods. **Cambridge University Press.**

Cortes C, and Vapnik V. (1995). Support vector network. **Machine Learning.** 20(3):273–297.

Dao S. D., Abhary K., and Marian R. (2016). An improved structure of genetic algorithms for global optimisation. **Progress in Artificial Intelligence.** 1-9.

Estabrooks A., and Japkowicz N. (2001). A mixture-of-experts framework for learning from unbalanced data sets. **In Proceedings of the 2001 Intelligent Data Analysis Conference.** pp34-43.

Estabrooks A., Jo, T., and Japkowicz N. (2004). A Multiple Resampling Method for Learning from Imbalanced Data Sets. **Computational Intelligence.** 20:18-36.

Farquad M. A. H., and Bose I. (2012). Preprocessing unbalanced data using support vector machine. **Decision Support Systems.** 53(1):226-233.

Freund Y., Schapire R., (1996). Experiments with a new Boosting Algorithm; **In Proc. 13th International Conference on Machine Learning 148–146.** San Francisco: Morgan Kaufmann.

Freund Y., Schapire R., and Abe N. (1999). A short introduction to boosting. **Journal-Japanese Society For Artificial Intelligence.** 14(771-780). 1612.

Gao M., Hong X., Chen S., and Harris, C. J. (2012). Probability density function estimation based over-sampling for imbalanced two-class problems. In Neural Networks (IJCNN). **The 2012 International Joint Conference on IEEE.** 1-8.

Han H., Wang W. Y., and Mao B. H. (2005). Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning; **Proc. Int'l Conf. Intelligent Computing.** 878-887.

Han J., Pei J., and Kamber M. (2011). Data mining: concepts and techniques. **Elsevier.**

Holland H. (1975). Adaptation in Natural and Artificial Systems. **Ann Arbor: The University of Michigan Press.** Michigan.

Japkowicz N. (2000a). Learning from imbalanced data sets: a comparison of various strategies. **AAAI Tech Report WS-00-05. AAAI.**

Japkowicz N. (2000b). The Class Imbalance Problem: Significance and Strategies. **In Proceedings of the 2000 International Conference on Artificial Intelligence (IC-A1'2000): Special Track on Inductive Learning.** Las Vegas. Nevada.

Japkowicz N., and Stephen S. (2002). The class imbalance problem: A systematic study. **Intelligent Data Analysis.** 6(5):203-231.

Jo T. and Japkowicz N. (2004). Class imbalances versus small disjuncts. **SIGKDD Explorations.** 6(1).

Krawczyk B., Wozniak M., and Schaefer G. (2014). Cost-sensitive decision tree ensembles for effective imbalanced classification. **Applied Soft Computing.** 14:554-562.

Kearns M., and Valiant L. G. (1994). Cryptographic limitations on learning Boolean formulae and finite automata; **Journal of the Association for Computing Machinery.** 41(1):67–95.

Kubat M. and Matwin S. (1997). Addressing the Curse of Imbalanced Training Sets: One Sided Selection. **In Proceedings of the Fourteenth Intemational Conference on Machine Learning.** pages 179-186. Nashville. Tennesse. Morgan Kaufmann.

Lariviere B., and Van d. P. D. (2005). Predicting customer retention and profitability by using random forests and regression forests techniques. **Expert Systems With Applications.** 29.

Laurikkala J. (2001). Improving Identification of Difficult Small Classes by Balancing Class Distribution. **Technical Report A-2001-2.** University of Tampere.

Liao J. J., Shih C. H., Chen T. F., and Hsu M. F. (2014). An ensemble-based model for two class imbalanced financial problem. **Economic Modelling.** 37. 175-183.

Ling C. and Li C. (1998). Data Mining for Direct Marketing Problems and Solutions. **In Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98).** New York. NY. AAAI Press.

Liu Y., Chawla N. V., Harper M., Shriberg E., and Stolcke A. (2006). A Study in Machine Learning from Imbalanced Data for Sentence Boundary Detection in Speech. **Computer Speech and Language.** 20. 468-494.

Malhotra R., Singh N., and Singh Y. (2011). Genetic algorithms: Concepts, design for optimization of process controllers. **Computer and Information Science.** 4(2). 39.

Miguel P. S. A., (2009), Classification for Fraud Detection with Social Network Analysis. **Master's Degree Dissertation, Engenharia Informatica e de Computadores; Instituto Superior Technico.** Universidade Tecnica de Lisboa. Portugal.

Muhlbaier M. D., Topalis A., and Polikar R. (2009). Learn $^{++} $. NC: Combining Ensemble of Classifiers With Dynamically Weighted Consult-and-Vote for Efficient Incremental Learning of New Classes. **IEEE transactions on neural networks.** 20(1). 152-168.

Muller KR., Mika S., Ratsch G., Tsuda K., and Scholkopf B. (2001). An Introduction to kernel-based learning algorithms. **IEEE Trans Neural Networks.** 12(2):199–222.

Orriols-Puig A., and Bernadó-Mansilla E. (2009). Evolutionary rule-based systems for imbalanced data sets. **Soft Computing.** 13(3):213-225.

Phua C. and Alahakoon D. (2004). Minority report in fraud detection: Classification of skewed data. **SIGKDD Explorations.** 6(1)

Schroff F., Criminisi A., and Zisserman A. (2008). Object Class Segmentation using Random Forests. **Dept. of Engineering Science.** University of Oxford.

Shen A., Tong R., and Deng Y. (2007). Application of Classification Models on Credit Card Fraud Detection. **International Conference on Service Systems and Service Management. IEEE.**

Solberg A. H. and Solberg R. (1996). A Large-Scale Evaluation of Features for Automatic Detection of Oil Spills in ERS SAR Images. **In International Geoscience and Remote Sensing Symposium,** pages 1484-1486. Lincoln. NE.

Valiant L G. (1984). A theory of the learnable. **Communication of the ACM.** 27(11).

Wang S., and Yao. X. (2009). Diversity Analysis on Imbalanced Data Sets by using Ensemble Models. **In Proc. of The IEEE Symposium on Computational Intelligence and Data Mining.**

Weiss G. and Provost F. (2003). Learning when Training Data are Costly: The Effect of Class Distribution on Tree Induction. **Journal of ArtiJicial Intelligence Research.** 19:315-354.

Wright A. H. (1991). Foundations of Genetic Algorithms. **chapter Genetic Algorithms for Real Parameter Optimization.** pages 205--218. Morgan Kaufmann.

Yin F., Mao H., and Hua, L. (2011). A hybrid of back propagation neural network and genetic algorithm for optimization of injection molding process parameters. **Materials & Design.** 32(6). 3457-3464.

Yu H., Mu C., Sun C., Yang W., Yang X., and Zuo X. (2015). Support vector machine-based optimized decision threshold adjustment strategy for classifying imbalanced data. **Knowledge-Based Systems.** 76. 67-78.

Zheng H., Kong L. X., and Nahavandi S. (2002). Automatic inspection of metallic surface defects using genetic algorithms. **Journal of materials processing technology.** 125. 427-433.

ภาคผนวก ก

รหัสต้นฉบับของโปรแกรม

โปรแกรมการจำแนกประเภทข้อมูลไม่สมดุลโดยใช้การปรับปรุงข้อมูลร่วมกับการหาค่าพารามิเตอร์ที่เหมาะสมด้วยขั้นตอนวิธีเชิงพันธุกรรมที่มีการเริ่มต้นใหม่

```
#ชุดข้อมูลสังเคราะห์#
#การจำแนกด้วยอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีน#
library("e1071")
library("sampling")
library("SamplingStrata")
library("caret")
#generate data 2 class
set.seed(1)
datatrain <- twoClassSim(500, intercept = -13)
datatest <- twoClassSim(200, intercept = -13)
table(datatrain$Class)
table(datatest$Class)
#create model with SVM
model  <- svm(Class~., data = datatrain)
#Predict model with test set
prediction <- predict(model, datatest)
#confusion matrix
table(prediction, datatest$Class)
#accuracy
sum(prediction == datatest$Class)/nrow(datatest)

#การจำแนกด้วยอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนร่วมกับการสุ่มลดข้อมูลจากคลาสส่วนมาก#
library("e1071")
library("ROSE")
library("sampling")
library("SamplingStrata")
#generate data 2 class
set.seed(1)
datatrain <- twoClassSim(500, intercept = -13)
datatest <- twoClassSim(200, intercept = -13)
table(datatrain$Class)
#balance Data
```

```
balancedata <- ovun.sample(Class~., data = datatrain, method = "under", N = 144, seed = 1)$data
table(balancedata$Class)
#create model with SVM
model  <- svm(Class~., data = balancedata)
#Predict model with test set
prediction <- predict(model, datatest)
#confusion matrix
table(prediction, datatest$Class)
#accuracy
sum(prediction == datatest$Class)/nrow(datatest)

#การจำแนกด้วยอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนร่วมกับการสุ่มเพิ่มข้อมูลจากคลาสส่วนน้อย#
library("e1071")
library("ROSE")
library("sampling")
library("SamplingStrata")
#generate data 2 class
set.seed(1)
datatrain <- twoClassSim(500, intercept = -13)
datatest <- twoClassSim(200, intercept = -13)
table(datatrain$Class)
#balance Data
balancedata <- ovun.sample(Class~., data = datatrain, method = "over", N = 856, seed = 1)$data
table(balancedata$Class)
#create model with SVM
model  <- svm(Class~., data = balancedata)
#Predict model with test set
prediction <- predict(model, datatest)
#confusion matrix
table(prediction, datatest$Class)
#accuracy
sum(prediction == datatest$Class)/nrow(datatest)

#การจำแนกด้วยอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนร่วมกับการสังเคราะห์ข้อมูลจากคลาสส่วนน้อย#
library("e1071")
```

```
library("sampling")
library("SamplingStrata")
library("DMwR")
#generate data 2 class
set.seed(1)
datatrain <- twoClassSim(500, intercept = -13)
datatest <- twoClassSim(200, intercept = -13)
table(datatrain$Class)
#balancedata
balancedata <- SMOTE(Class ~ ., datatrain, perc.under=120, perc.over = 500)
table(balancedata$Class)
#create model with SVM
model  <- svm(Class~., data = balancedata)
#Predict model with test set
prediction <- predict(model, datatest)
#confusion matrix
table(prediction, datatest$Class)
#accuracy
sum(prediction == datatest$Class)/nrow(datatest)
#การจำแนกด้วยอัลกอริทึมเอดาบูส#
library("e1071")
library("sampling")
library("SamplingStrata")
library("adabag")
library("rpart")
#generate data 2 class
set.seed(1)
datatrain <- twoClassSim(500, intercept = -13)
datatest <- twoClassSim(200, intercept = -13)
table(datatrain$Class)
table(datatest$Class)
#Create model
datatrainboots <- boosting(Class~., data = datatrain, mfinal = 10, control = rpart.control(maxdepth = 1))
#prediction
prediction <- predict(datatrainboots, datatest)
```

```
#confusion matrix
prediction

#การจำแนกด้วยอัลกอริทึมรัสบูส#
library("e1071")
library("sampling")
library("SamplingStrata")
library("adabag")
library("rpart")
library("ROSE")
#generate data 2 class
set.seed(1)
datatrain <- twoClassSim(500, intercept = -13)
datatest <- twoClassSim(200, intercept = -13)
table(datatrain$Class)
table(datatest$Class)
#balance Data
balancedata <- ovun.sample(Class~., data = datatrain, method = "under", N = 144, seed = 1)$data
table(balancedata$Class)
#Create model
datatrainboots <- boosting(Class~., data = balancedata, mfinal = 10, control = rpart.control(maxdepth = 1))
#prediction
prediction <- predict(datatrainboots, datatest)
#confusion matrix
prediction

#การหาค่าพารามิเตอร์ที่เหมาะสมด้วยขั้นตอนวิธีเชิงพันธุกรรมที่มีการเริ่มต้นใหม่#
library("e1071")
library("sampling")
library("SamplingStrata")
library("DMwR")
library("GA")
#generate data 2 class
set.seed(1)
datatrain <- twoClassSim(500, intercept = -13)
```

```
datatest <- twoClassSim(200, intercept = -13)
table(datatrain$Class)
#Under-sampling
underdata <- ovun.sample(Class~., data = datatrain, method = "under", N = 300, seed = 1)$data
table(underdata$Class)
#balance
balancedata <- SMOTE(Class ~ ., underdata, perc.under=160, perc.over = 200)
table(balancedata$Class)
# Setup the data for cross-validation
K = 5 # 5-fold cross-validation
fold_inds <- sample(1:K, nrow(balancedata), replace = TRUE)
lst_CV_data <- lapply(1:K, function(i) list(train_data = balancedata[fold_inds != i, , drop = FALSE], test_data
= balancedata[fold_inds == i, , drop = FALSE]))
# Given the values of parameters 'cost', 'gamma' and 'epsilon', return the rmse of the model over the test data
evalParams <- function(train_data, test_data, cost, gamma, epsilon) {
 # Train
 model <- svm(Class ~ ., data = train_data, cost = cost, gamma = gamma, epsilon = epsilon, type = "C-
classification", kernel = "radial")
 # Test
 prediction <- predict(model, test_data)
 acc <- sum(prediction == test_data$Class)/nrow(test_data)
 return (acc)
}
# Fitness function (to be maximized)
# Parameter vector x is: (cost, gamma, epsilon)
fitnessFunc <- function(x, Lst_CV_Data) {
 # Retrieve the SVM parameters
 cost_val <- x[1]
 gamma_val <- x[2]
 epsilon_val <- x[3]
 # Use cross-validation to estimate the RMSE for each split of the dataset
 rmse_vals <- sapply(Lst_CV_Data, function(in_data) with(in_data, evalParams(train_data, test_data, cost_val,
gamma_val, epsilon_val)))
 # As fitness measure, return minus the average rmse (over the cross-validation folds),
 # so that by maximizing fitness we are minimizing the rmse
```

```
 return (-mean(rmse_vals))
}
# Range of the parameter values to be tested
# Parameters are: (cost, gamma, epsilon)
theta_min <- c(cost = 1e-4, gamma = 1e-3, epsilon = 1e-2)
theta_max <- c(cost = 100, gamma = 2, epsilon = 2)
# Run the genetic algorithm
results <- ga(type = "real-valued", fitness = fitnessFunc, lst_CV_data, names = names(theta_min), min =
theta_min, max = theta_max, popSize = 100, maxiter = 100)
summary(results)

#การจำแนกด้วยอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนด้วยพารามิเตอร์ที่เหมาะสม#
library("e1071")
library("sampling")
library("SamplingStrata")
library("DMwR")
#generate data 2 class
set.seed(1)
datatrain <- twoClassSim(500, intercept = -13)
datatest <- twoClassSim(200, intercept = -13)
table(datatrain$Class)
#Under-sampling
underdata <- ovun.sample(Class~., data = datatrain, method = "under", N = 300, seed = 1)$data
table(underdata$Class)
#balance
balancedata <- SMOTE(Class ~ ., underdata, perc.under=160, perc.over = 200)
table(balancedata$Class)
#create model with SVM
model  <- svm(Class ~ ., data = balancedata, cost = 56.97464 , gamma = 0.004282877 , epsilon = 1.720305,
type = "C-classification", kernel = "radial")
#Predict model with test set
prediction <- predict(model, datatest)
#confusion matrix
table(prediction, datatest$Class)
#accuracy
```

```
sum(prediction == datatest$Class)/nrow(datatest)


#ชุดข้อมูลโรคหอบหืด#
#การจำแนกด้วยอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีน#
library("e1071")
library("sampling")
library("SamplingStrata")
Data <- read.csv("D:/Rtmp/dataset_for_PHD/asthma_dataset/asthma_default.csv")
table(Data$Class)
#Function Split Data
#split data into train and test
set.seed(0)
stsampling <- function(X,target,test){
 output <- list()
 idtrain <- strata(X,stratanames=target,size=table(X[,target])*(1-test), method="srswor")$ID_unit
 output$train <- X[idtrain,]
 output$test <- X[(idtrain*-1),]
 return (output)
}
#Sprit Train-Test
dataSplit <- stsampling(Data,"Class",0.30)
datatrain <- dataSplit$train
datatest <- dataSplit$test
#create model with SVM
model  <- svm(Class~., data = datatrain)
#Predict model with test set
prediction <- predict(model, datatest)
#confusion matrix
table(prediction, datatest$Class)
#accuracy
sum(prediction == datatest$Class)/nrow(datatest)


#การจำแนกด้วยอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนร่วมกับการสุ่มลดข้อมูลจากคลาสส่วนมาก#
library("e1071")
library("sampling")
```

```
library("SamplingStrata")
library("ROSE")
Data <- read.csv("D:/Rtmp/dataset_for_PHD/asthma_dataset/asthma_default.csv")
table(Data$Class)
#Function Split Data
#split data into train and test
set.seed(0)
stsampling <- function(X,target,test){
 output <- list()
 idtrain <- strata(X,stratanames=target,size=table(X[,target])*(1-test), method="srswor")$ID_unit
 output$train <- X[idtrain,]
 output$test <- X[(idtrain*-1),]
 return (output)
}
#Sprit Train-Test
dataSplit <- stsampling(Data,"Class",0.30)
datatrain <- dataSplit$train
datatest <- dataSplit$test
#balance Data
balancedata <- ovun.sample(Class~., data = datatrain, method = "under", N = 178, seed = 1)$data
#create model with SVM
model  <- svm(Class~., data = balancedata)
#Predict model with test set
prediction <- predict(model, datatest)
#confusion matrix
table(prediction, datatest$Class)
#accuracy
sum(prediction == datatest$Class)/nrow(datatest)

#การจำแนกด้วยอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนร่วมกับการสุ่มเพิ่มข้อมูลจากคลาสส่วนน้อย#
library("e1071")
library("sampling")
library("SamplingStrata")
library("ROSE")
Data <- read.csv("D:/Rtmp/dataset_for_PHD/asthma_dataset/asthma_default.csv")
```

```
table(Data$Class)
#Function Split Data
#split data into train and test
set.seed(0)
stsampling <- function(X,target,test){
 output <- list()
 idtrain <- strata(X,stratanames=target,size=table(X[,target])*(1-test), method="srswor")$ID_unit
 output$train <- X[idtrain,]
 output$test <- X[(idtrain*-1),]
 return (output)
}
#Sprit Train-Test
dataSplit <- stsampling(Data,"Class",0.30)
datatrain <- dataSplit$train
datatest <- dataSplit$test
#balance Data
balancedata <- ovun.sample(Class~., data = datatrain, method = "over", N = 798, seed = 1)$data
#create model with SVM
model  <- svm(Class~., data = balancedata)
#Predict model with test set
prediction <- predict(model, datatest)
#confusion matrix
table(prediction, datatest$Class)
#accuracy
sum(prediction == datatest$Class)/nrow(datatest)


#การจำแนกด้วยอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนร่วมกับการสังเคราะห์ข้อมูลจากคลาสส่วนน้อย#
library("e1071")
library("sampling")
library("SamplingStrata")
library("ROSE")
library("DMwR")
Data <- read.csv("D:/Rtmp/dataset_for_PHD/asthma_dataset/asthma_default.csv")
table(Data$Class)
#Function Split Data
```

```
#split data into train and test
set.seed(0)
stsampling <- function(X,target,test){
 output <- list()
 idtrain <- strata(X,stratanames=target,size=table(X[,target])*(1-test), method="srswor")$ID_unit
 output$train <- X[idtrain,]
 output$test <- X[(idtrain*-1),]
 return (output)
}
#Sprit Train-Test
dataSplit <- stsampling(Data,"Class",0.30)
datatrain <- dataSplit$train
datatest <- dataSplit$test
#balance
balancedata <- SMOTE(Class ~ ., datatrain, perc.under=135, perc.over = 300)
table(balancedata$Class)
#create model with SVM
model  <- svm(Class~., data = balancedata)
#Predict model with test set
prediction <- predict(model, datatest)
#confusion matrix
table(prediction, datatest$Class)
#accuracy
sum(prediction == datatest$Class)/nrow(datatest)

#การจำแนกด้วยอัลกอริทึมเอดาบูส#
library("e1071")
library("sampling")
library("SamplingStrata")
library("ROSE")
library("adabag")
library("rpart")
Data <- read.csv("D:/Rtmp/dataset_for_PHD/asthma_dataset/asthma_default.csv")
table(Data$Class)
#Function Split Data
```

```
#split data into train and test
set.seed(0)
stsampling <- function(X,target,test){
 output <- list()
 idtrain <- strata(X,stratanames=target,size=table(X[,target])*(1-test), method="srswor")$ID_unit
 output$train <- X[idtrain,]
 output$test <- X[(idtrain*-1),]
 return (output)
}
#Sprit Train-Test
dataSplit <- stsampling(Data,"Class",0.30)
datatrain <- dataSplit$train
datatest <- dataSplit$test
table(datatrain$Class)
table(datatest$Class)
#Create model
datatrainboots <- boosting(Class~., data = datatrain, mfinal = 5, control = rpart.control(maxdepth = 1))
#prediction
prediction <- predict(datatrainboots, datatest)
#confusion matrix
prediction


#การจำแนกด้วยอัลกอริทึมรัสบูส#
library("e1071")
library("sampling")
library("SamplingStrata")
library("ROSE")
library("adabag")
library("rpart")
Data <- read.csv("D:/Rtmp/dataset_for_PHD/asthma_dataset/asthma_default.csv")
table(Data$Class)
#Function Split Data
#split data into train and test
set.seed(0)
stsampling <- function(X,target,test){
```

```
 output <- list()
 idtrain <- strata(X,stratanames=target,size=table(X[,target])*(1-test), method="srswor")$ID_unit
 output$train <- X[idtrain,]
 output$test <- X[(idtrain*-1),]
 return (output)
}
#Sprit Train-Test
dataSplit <- stsampling(Data,"Class",0.30)
datatrain <- dataSplit$train
datatest <- dataSplit$test
table(datatrain$Class)
table(datatest$Class)
#balance Data
balancedata <- ovun.sample(Class~., data = datatrain, method = "under", N = 178, seed = 1)$data
#Create model
datatrainboots <- boosting(Class~., data = balancedata, mfinal = 5, control = rpart.control(maxdepth = 1))
#prediction
prediction <- predict(datatrainboots, datatest)
#confusion matrix
prediction


#การหาค่าพารามิเตอร์ที่เหมาะสมด้วยขั้นตอนวิธีเชิงพันธุกรรมที่มีการเริ่มต้นใหม่#
library("e1071")
library("GA")
library("sampling")
library("SamplingStrata")
library("ROSE")
Data <- read.csv("D:/Rtmp/dataset_for_PHD/asthma_dataset/asthma_default.csv")
table(Data$Class)
#Function Split Data
#split data into train and test
set.seed(0)
stsampling <- function(X,target,test){
 output <- list()
 idtrain <- strata(X,stratanames=target,size=table(X[,target])*(1-test), method="srswor")$ID_unit
```

```
 output$train <- X[idtrain,]
 output$test <- X[(idtrain*-1),]
 return (output)
}
#Sprit Train-Test
dataSplit <- stsampling(Data,"Class",0.30)
datatrain <- dataSplit$train
datatest <- dataSplit$test
#Under sampling
#underdata <- ovun.sample(Class~., data = datatrain, method = "under", N = 438, seed = 1)$data
balancedata <- ovun.sample(Class~., data = datatrain, method = "under", N = 178, seed = 1)$data
table(balancedata$Class)
#balance
#balancedata <- SMOTE(Class ~ ., underdata, perc.under=135, perc.over = 300)
#table(balancedata$Class)
# Setup the data for cross-validation
K = 5 # 5-fold cross-validation
fold_inds <- sample(1:K, nrow(balancedata), replace = TRUE)
lst_CV_data <- lapply(1:K, function(i) list(train_data = balancedata[fold_inds != i, , drop = FALSE], test_data
= balancedata[fold_inds == i, , drop = FALSE]))
# Given the values of parameters 'cost', 'gamma' and 'epsilon', return the rmse of the model over the test data
evalParams <- function(train_data, test_data, cost, gamma, epsilon) {
 # Train
 model <- svm(Class ~ ., data = train_data, cost = cost, gamma = gamma, epsilon = epsilon, type = "C-
classification", kernel = "radial")
 # Test
 prediction <- predict(model, test_data)
 acc <- sum(prediction == test_data$Class)/nrow(test_data)
 return (acc)
}
# Fitness function (to be maximized)
# Parameter vector x is: (cost, gamma, epsilon)
fitnessFunc <- function(x, Lst_CV_Data) {
 # Retrieve the SVM parameters
 cost_val <- x[1]
```

```
 gamma_val <- x[2]
 epsilon_val <- x[3]
 # Use cross-validation to estimate the RMSE for each split of the dataset
 rmse_vals <- sapply(Lst_CV_Data, function(in_data) with(in_data, evalParams(train_data, test_data, cost_val,
gamma_val, epsilon_val)))
 # As fitness measure, return minus the average rmse (over the cross-validation folds),
 # so that by maximizing fitness we are minimizing the rmse
 return (-mean(rmse_vals))
}
# Range of the parameter values to be tested
# Parameters are: (cost, gamma, epsilon)
theta_min <- c(cost = 1e-4, gamma = 1e-3, epsilon = 1e-2)
theta_max <- c(cost = 100, gamma = 2, epsilon = 2)
# Run the genetic algorithm
results <- ga(type = "real-valued", fitness = fitnessFunc, lst_CV_data, names = names(theta_min), min =
theta_min, max = theta_max, popSize = 100, maxiter = 100)

summary(results)

#การจำแนกด้วยอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนด้วยพารามิเตอร์ที่เหมาะสม#
library("e1071")
library("sampling")
library("SamplingStrata")
Data <- read.csv("D:/Rtmp/dataset_for_PHD/asthma_dataset/asthma_default.csv")
#Function Split Data
#split data into train and test
set.seed(0)
stsampling <- function(X,target,test){
 output <- list()
 idtrain <- strata(X,stratanames=target,size=table(X[,target])*(1-test), method="srswor")$ID_unit
 output$train <- X[idtrain,]
 output$test <- X[(idtrain*-1),]
 return (output)
}
#Sprit Train-Test
```

```
dataSplit <- stsampling(Data,"Class",0.30)
datatrain <- dataSplit$train
datatest <- dataSplit$test
#create model with SVM
model  <- svm(Class ~ ., data = datatrain, cost = 8.374748, gamma = 0.2513825, epsilon = 0.8360468, type =
"C-classification", kernel = "radial")
#Predict model with test set
prediction <- predict(model, datatest)
#confusion matrix
table(prediction, datatest$Class)
#accuracy
sum(prediction == datatest$Class)/nrow(datatest)
#ชุดข้อมูลโรคหัวใจ#
#การจำแนกด้วยอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีน#
library("e1071")
library("sampling")
library("SamplingStrata")
library("caret")
Data <- read.csv("D:/Rtmp/dataset_for_PHD/heart_disease_dataset/HD_default.csv")
table(Data$Class)
#Function Split Data
#split data into train and test
set.seed(3456)
trainIndex <- createDataPartition(Data$Class, p = 0.7,list = FALSE, times = 1)
head(trainIndex)
datatrain <- Data[trainIndex,]
datatest <- Data[-trainIndex,]
#create model with SVM
model  <- svm(Class~., data = datatrain, type = "C-classification", kernel = "radial")
#Predict model with test set
prediction <- predict(model, datatest)
#confusion matrix
table(prediction, datatest$Class)
#accuracy
sum(prediction == datatest$Class)/nrow(datatest)
```

```
#การจำแนกด้วยอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนร่วมกับการสุ่มลดข้อมูลจากคลาสส่วนมาก#
library("e1071")
library("sampling")
library("SamplingStrata")
library("ROSE")
library("caret")
Data <- read.csv("D:/Rtmp/dataset_for_PHD/heart_disease_dataset/HD_default.csv")
table(Data$Class)
#Function Split Data
#split data into train and test
set.seed(3456)
trainIndex <- createDataPartition(Data$Class, p = 0.7,list = FALSE, times = 1)
head(trainIndex)
datatrain <- Data[trainIndex,]
datatest <- Data[-trainIndex,]
table(datatrain$Class)
table(datatest$Class)
#balance Data
balancedata <- ovun.sample(Class~., data = datatrain, method = "under", N = 168, seed = 1)$data
table(balancedata$Class)
#create model with SVM
model  <- svm(Class~., data = balancedata)
#Predict model with test set
prediction <- predict(model, datatest)
#confusion matrix
table(prediction, datatest$Class)
#accuracy
sum(prediction == datatest$Class)/nrow(datatest)

#การจำแนกด้วยอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนร่วมกับการสุ่มเพิ่มข้อมูลจากคลาสส่วนน้อย#
library("e1071")
library("ROSE")
library("caret")
#Load data
```

```
Data <- read.csv("D:/Rtmp/dataset_for_PHD/heart_disease_dataset/HD_default.csv")
table(Data$Class)
#Function Split Data
#split data into train and test
set.seed(3456)
trainIndex <- createDataPartition(Data$Class, p = 0.7,list = FALSE, times = 1)
head(trainIndex)
datatrain <- Data[trainIndex,]
datatest <- Data[-trainIndex,]
table(datatrain$Class)
table(datatest$Class)
#balance Data
balancedata <- ovun.sample(Class~., data = datatrain, method = "over", N = 210, seed = 1)$data
table(balancedata$Class)
#create model with SVM
model  <- svm(Class~., data = balancedata)
#Predict model with test set
prediction <- predict(model, datatest)
#confusion matrix
table(prediction, datatest$Class)
#accuracy
sum(prediction == datatest$Class)/nrow(datatest)



#การจำแนกด้วยอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนร่วมกับการสังเคราะห์ข้อมูลจากคลาสส่วนน้อย#
library("e1071")
library("sampling")
library("SamplingStrata")
library("DMwR")
library("ROSE")
library("caret")
#Load data
Data <- read.csv("D:/Rtmp/dataset_for_PHD/heart_disease_dataset/HD_default.csv")
table(Data$Class)
#Function Split Data
```

```
#split data into train and test
set.seed(3456)
trainIndex <- createDataPartition(Data$Class, p = 0.7,list = FALSE, times = 1)
head(trainIndex)
datatrain <- Data[trainIndex,]
datatest <- Data[-trainIndex,]
table(datatrain$Class)
table(datatest$Class)
#balance
balancedata <- SMOTE(Class ~ ., datatrain, perc.under=500, perc.over = 25)
table(balancedata$Class)
#create model with SVM
model  <- svm(Class~., data = datatrain)
#Predict model with test set
prediction <- predict(model, datatest)
#confusion matrix
table(prediction, datatest$Class)
#accuracy
sum(prediction == datatest$Class)/nrow(datatest)


#การจำแนกด้วยอัลกอริทึมเอดาบูส#
library("e1071")
library("adabag")
library("caret")
#Load data
Data <- read.csv("D:/Rtmp/dataset_for_PHD/heart_disease_dataset/HD_default.csv")
table(Data$Class)
#Function Split Data
#split data into train and test
set.seed(3456)
trainIndex <- createDataPartition(Data$Class, p = 0.7,list = FALSE, times = 1)
head(trainIndex)
datatrain <- Data[trainIndex,]
datatest <- Data[-trainIndex,]
table(datatrain$Class)
```

```
table(datatest$Class)
#Create model
datatrainboots <- boosting(Class~., data = datatrain, mfinal = 10, control = rpart.control(maxdepth = 1))
#prediction
prediction <- predict(datatrainboots, datatest)
#confusion matrix
prediction

#การจำแนกด้วยอัลกอริทึมรัสบูส#
library("e1071")
library("sampling")
library("SamplingStrata")
library("ROSE")
library("adabag")
library("caret")
Data <- read.csv("D:/Rtmp/dataset_for_PHD/heart_disease_dataset/HD_default.csv")
table(Data$Class)
#Function Split Data
#split data into train and test
set.seed(3456)
trainIndex <- createDataPartition(Data$Class, p = 0.7,list = FALSE, times = 1)
head(trainIndex)
datatrain <- Data[trainIndex,]
datatest <- Data[-trainIndex,]
table(datatrain$Class)
table(datatest$Class)
#balance Data
balancedata <- ovun.sample(Class~., data = datatrain, method = "under", N = 168, seed = 1)$data
table(balancedata$Class)
datatrainboots <- boosting(Class~., data = balancedata, mfinal = 10, control = rpart.control(maxdepth = 1))
#prediction
prediction <- predict(datatrainboots, datatest)
#confusion matrix
prediction
```

```
#การหาค่าพารามิเตอร์ที่เหมาะสมด้วยขั้นตอนวิธีเชิงพันธุกรรมที่มีการเริ่มต้นใหม่#
library("e1071")
library("sampling")
library("SamplingStrata")
library("DMwR")
library("GA")
library("caret")
#Load data
Data <- read.csv("D:/Rtmp/dataset_for_PHD/heart_disease_dataset/HD_default.csv")
table(Data$Class)
#Function Split Data
#split data into train and test
set.seed(3456)
trainIndex <- createDataPartition(Data$Class, p = 0.7,list = FALSE, times = 1)
head(trainIndex)
datatrain <- Data[trainIndex,]
datatest <- Data[-trainIndex,]
table(datatrain$Class)
table(datatest$Class)
#Under-sampling
underdata <- ovun.sample(Class~., data = datatrain, method = "under", N = 180, seed = 1)$data
table(underdata$Class)
#balance
balancedata <- SMOTE(Class ~ ., underdata, perc.under=700, perc.over = 17)
table(balancedata$Class)
balancedata <- SMOTE(Class ~ ., Dataold, perc.over = 25,perc.under=500)
# Setup the data for cross-validation
K = 5 # 5-fold cross-validation
fold_inds <- sample(1:K, nrow(balancedata), replace = TRUE)
lst_CV_data <- lapply(1:K, function(i) list(train_data = balancedata[fold_inds != i, , drop = FALSE], test_data
= balancedata[fold_inds == i, , drop = FALSE]))
# Given the values of parameters 'cost', 'gamma' and 'epsilon', return the rmse of the model over the test data
evalParams <- function(train_data, test_data, cost, gamma, epsilon) {
 # Train
```

```
 model <- svm(Class ~ ., data = train_data, cost = cost, gamma = gamma, epsilon = epsilon, type = "C-
classification", kernel = "radial")
 # Test
 prediction <- predict(model, test_data)
 acc <- sum(prediction == test_data$Class)/nrow(test_data)
 return (acc)
}
# Fitness function (to be maximized)
# Parameter vector x is: (cost, gamma, epsilon)
fitnessFunc <- function(x, Lst_CV_Data) {
 # Retrieve the SVM parameters
 cost_val <- x[1]
 gamma_val <- x[2]
 epsilon_val <- x[3]
 # Use cross-validation to estimate the RMSE for each split of the dataset
 rmse_vals <- sapply(Lst_CV_Data, function(in_data) with(in_data, evalParams(train_data, test_data, cost_val,
gamma_val, epsilon_val)))
 # As fitness measure, return minus the average rmse (over the cross-validation folds),
 # so that by maximizing fitness we are minimizing the rmse
 return (-mean(rmse_vals))
}
# Range of the parameter values to be tested
# Parameters are: (cost, gamma, epsilon)
theta_min <- c(cost = 1e-4, gamma = 1e-3, epsilon = 1e-2)
theta_max <- c(cost = 100, gamma = 2, epsilon = 2)
# Run the genetic algorithm
results <- ga(type = "real-valued", fitness = fitnessFunc, lst_CV_data, names = names(theta_min), min =
theta_min, max = theta_max, popSize = 100, maxiter = 100)
summary(results)

#การจำแนกด้วยอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนด้วยพารามิเตอร์ที่เหมาะสม#
library("e1071")
library("sampling")
library("SamplingStrata")
Data <- read.csv("D:/Rtmp/dataset_for_PHD/heart_disease_dataset/HD_default.csv")
```

```
#Function Split Data
#split data into train and test
set.seed(0)
stsampling <- function(X,target,test){
 output <- list()
 idtrain <- strata(X,stratanames=target,size=table(X[,target])*(1-test), method="srswor")$ID_unit
 output$train <- X[idtrain,]
 output$test <- X[(idtrain*-1),]
 return (output)
}
#Sprit Train-Test
dataSplit <- stsampling(Data,"Class",0.30)
datatrain <- dataSplit$train
datatest <- dataSplit$test
#create model with SVM
model  <- svm(Class ~ ., data = datatrain, cost = 4.319116, gamma = 0.182505, epsilon = 0.012312, type = "C-
classification", kernel = "radial")
#Predict model with test set
prediction <- predict(model, datatest)
#confusion matrix
table(prediction, datatest$Class)
#accuracy
sum(prediction == datatest$Class)/nrow(datatest)


#ชุดข้อมูลสังเคราะห์#
#การจำแนกด้วยอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีน#
library("e1071")
library("sampling")
library("SamplingStrata")
library("caret")
Data <- read.csv("D:/Rtmp/dataset_for_PHD/indian_liver_patient_dataset/IndianLiverPatientDataset.csv")
table(Data$Class)
#Function Split Data
#split data into train and test
set.seed(3456)
```

```
trainIndex <- createDataPartition(Data$Class, p = 0.7,list = FALSE, times = 1)
head(trainIndex)
datatrain <- Data[trainIndex,]
datatest <- Data[-trainIndex,]
table(datatrain$Class)
table(datatest$Class)
#create model with SVM
model  <- svm(Class~., data = datatrain)
#Predict model with test set
prediction <- predict(model, datatest)
#confusion matrix
table(prediction, datatest$Class)
#accuracy
sum(prediction == datatest$Class)/nrow(datatest)

#การจำแนกด้วยอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนร่วมกับการสุ่มลดข้อมูลจากคลาสส่วนมาก#
library("e1071")
library("sampling")
library("SamplingStrata")
library("ROSE")
library("caret")
Data <- read.csv("D:/Rtmp/dataset_for_PHD/indian_liver_patient_dataset/IndianLiverPatientDataset.csv")
table(Data$Class)
#Function Split Data
#split data into train and test
set.seed(3456)
trainIndex <- createDataPartition(Data$Class, p = 0.7,list = FALSE, times = 1)
head(trainIndex)
datatrain <- Data[trainIndex,]
datatest <- Data[-trainIndex,]
table(datatrain$Class)
table(datatest$Class)
#balance Data
balancedata <- ovun.sample(Class~., data = datatrain, method = "under", N = 232, seed = 1)$data
table(balancedata$Class)
```

```
#create model with SVM
model  <- svm(Class~., data = balancedata)
#Predict model with test set
prediction <- predict(model, datatest)
#confusion matrix
table(prediction, datatest$Class)
#accuracy
sum(prediction == datatest$Class)/nrow(datatest)

#การจำแนกด้วยอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนร่วมกับการสุ่มเพิ่มข้อมูลจากคลาสส่วนน้อย#
library("e1071")
library("sampling")
library("SamplingStrata")
library("ROSE")
library("caret")
Data <- read.csv("D:/Rtmp/dataset_for_PHD/indian_liver_patient_dataset/IndianLiverPatientDataset.csv")
table(Data$Class)
#Function Split Data
#split data into train and test
set.seed(3456)
trainIndex <- createDataPartition(Data$Class, p = 0.7,list = FALSE, times = 1)
head(trainIndex)
datatrain <- Data[trainIndex,]
datatest <- Data[-trainIndex,]
table(datatrain$Class)
table(datatest$Class)
#balance Data
balancedata <- ovun.sample(Class~., data = datatrain, method = "over", N = 580, seed = 1)$data
table(balancedata$Class)
#create model with SVM
model  <- svm(Class~., data = balancedata)
#Predict model with test set
prediction <- predict(model, datatest)
#confusion matrix
table(prediction, datatest$Class)
```

```
#accuracy
sum(prediction == datatest$Class)/nrow(datatest)

#การจำแนกด้วยอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนร่วมกับการสังเคราะห์ข้อมูลจากคลาสส่วนน้อย#
library("e1071")
library("sampling")
library("SamplingStrata")
library("DMwR")
library("ROSE")
library("caret")
Data <- read.csv("D:/Rtmp/dataset_for_PHD/indian_liver_patient_dataset/IndianLiverPatientDataset.csv")
table(Data$Class)
#Function Split Data
#split data into train and test
set.seed(3456)
trainIndex <- createDataPartition(Data$Class, p = 0.7,list = FALSE, times = 1)
head(trainIndex)
datatrain <- Data[trainIndex,]
datatest <- Data[-trainIndex,]
table(datatrain$Class)
table(datatest$Class)
#balance
balancedata <- SMOTE(Class ~ ., datatrain, perc.under=250, perc.over = 100)
table(balancedata$Class)
#create model with SVM
model  <- svm(Class~., data = datatrain)
#Predict model with test set
prediction <- predict(model, datatest)
#confusion matrix
table(prediction, datatest$Class)
#accuracy
sum(prediction == datatest$Class)/nrow(datatest)

#การจำแนกด้วยอัลกอริทึมเอดาบูส#
library("e1071")
```

```
library("sampling")
library("SamplingStrata")
library("ROSE")
library("adabag")
library("caret")
Data <- read.csv("D:/Rtmp/dataset_for_PHD/indian_liver_patient_dataset/IndianLiverPatientDataset.csv")
table(Data$Class)
#Function Split Data
#split data into train and test
set.seed(3456)
trainIndex <- createDataPartition(Data$Class, p = 0.7,list = FALSE, times = 1)
head(trainIndex)
datatrain <- Data[trainIndex,]
datatest <- Data[-trainIndex,]
table(datatrain$Class)
table(datatest$Class)
#create model
datatrainboots <- boosting(Class~., data = datatrain, mfinal = 10, control = rpart.control(maxdepth = 1))
#prediction
prediction <- predict(datatrainboots, datatest)
#confusion matrix
prediction


#การจำแนกด้วยอัลกอริทึมรัสบูส#
library("e1071")
library("sampling")
library("SamplingStrata")
library("ROSE")
library("adabag")
library("caret")
Data <- read.csv("D:/Rtmp/dataset_for_PHD/indian_liver_patient_dataset/IndianLiverPatientDataset.csv")
table(Data$Class)
#Function Split Data
#split data into train and test
set.seed(3456)
```

```
trainIndex <- createDataPartition(Data$Class, p = 0.7,list = FALSE, times = 1)
head(trainIndex)
datatrain <- Data[trainIndex,]
datatest <- Data[-trainIndex,]
table(datatrain$Class)
table(datatest$Class)
#balance Data
balancedata <- ovun.sample(Class~., data = datatrain, method = "under", N = 232, seed = 1)$data
table(balancedata$Class)
#create model
datatrainboots <- boosting(Class~., data = balancedata, mfinal = 10, control = rpart.control(maxdepth = 1))
#prediction
prediction <- predict(datatrainboots, datatest)
#confusion matrix
prediction


#การหาค่าพารามิเตอร์ที่เหมาะสมด้วยขั้นตอนวิธีเชิงพันธุกรรมที่มีการเริ่มต้นใหม่#
library("e1071")
library("sampling")
library("SamplingStrata")
library("DMwR")
library("ROSE")
library("GA")
library("caret")
Data <- read.csv("D:/Rtmp/dataset_for_PHD/indian_liver_patient_dataset/IndianLiverPatientDataset.csv")
table(Data$Class)
#Function Split Data
#split data into train and test
set.seed(3456)
trainIndex <- createDataPartition(Data$Class, p = 0.7,list = FALSE, times = 1)
head(trainIndex)
datatrain <- Data[trainIndex,]
datatest <- Data[-trainIndex,]
table(datatrain$Class)
table(datatest$Class)
```

```
#balance Data
underdata <- ovun.sample(Class~., data = datatrain, method = "under", N = 319, seed = 1)$data
table(underdata$Class)
balancedata <- SMOTE(Class ~ ., underdata, perc.under=234, perc.over = 75)
table(balancedata$Class)
# Setup the data for cross-validation
K = 5 # 5-fold cross-validation
fold_inds <- sample(1:K, nrow(balancedata), replace = TRUE)
lst_CV_data <- lapply(1:K, function(i) list(train_data = balancedata[fold_inds != i, , drop = FALSE], test_data
= balancedata[fold_inds == i, , drop = FALSE]))
# Given the values of parameters 'cost', 'gamma' and 'epsilon', return the rmse of the model over the test data
evalParams <- function(train_data, test_data, cost, gamma, epsilon) {
 # Train
 model <- svm(Class ~ ., data = train_data, cost = cost, gamma = gamma, epsilon = epsilon, type = "C-
classification", kernel = "radial")
 # Test
 prediction <- predict(model, test_data)
 acc <- sum(prediction == test_data$Class)/nrow(test_data)
 return (acc)
}
# Fitness function (to be maximized)
# Parameter vector x is: (cost, gamma, epsilon)
fitnessFunc <- function(x, Lst_CV_Data) {
 # Retrieve the SVM parameters
 cost_val <- x[1]
 gamma_val <- x[2]
 epsilon_val <- x[3]
 # Use cross-validation to estimate the RMSE for each split of the dataset
 rmse_vals <- sapply(Lst_CV_Data, function(in_data) with(in_data, evalParams(train_data, test_data, cost_val,
gamma_val, epsilon_val)))
 # As fitness measure, return minus the average rmse (over the cross-validation folds),
 # so that by maximizing fitness we are minimizing the rmse
 return (-mean(rmse_vals))
}
# Range of the parameter values to be tested
```

```
# Parameters are: (cost, gamma, epsilon)
theta_min <- c(cost = 1e-4, gamma = 1e-3, epsilon = 1e-2)
theta_max <- c(cost = 100, gamma = 2, epsilon = 2)
# Run the genetic algorithm
results <- ga(type = "real-valued", fitness = fitnessFunc, lst_CV_data, names = names(theta_min), min =
theta_min, max = theta_max, popSize = 100, maxiter = 100)
summary(results)

#การจำแนกด้วยอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนด้วยพารามิเตอร์ที่เหมาะสม#
library("e1071")
library("sampling")
library("SamplingStrata")
library("DMwR")
library("ROSE")
library("GA")
library("caret")
Data <- read.csv("D:/Rtmp/dataset_for_PHD/indian_liver_patient_dataset/IndianLiverPatientDataset.csv")
table(Data$Class)
#Function Split Data
#split data into train and test
set.seed(3456)
trainIndex <- createDataPartition(Data$Class, p = 0.7,list = FALSE, times = 1)
head(trainIndex)
datatrain <- Data[trainIndex,]
datatest <- Data[-trainIndex,]
table(datatrain$Class)
table(datatest$Class)
#balance Data
underdata <- ovun.sample(Class~., data = datatrain, method = "under", N = 319, seed = 1)$data
table(underdata$Class)
balancedata <- SMOTE(Class ~ ., underdata, perc.under=234, perc.over = 75)
table(balancedata$Class)
#create model with SVM
model  <- svm(Class ~ ., data = balancedata, cost = 9.0821964 , gamma = 0.0154112 , epsilon = 1.0846983,
type = "C-classification", kernel = "radial")
```

```
#Predict model with test set
prediction <- predict(model, datatest)
#confusion matrix
table(prediction, datatest$Class)
#accuracy
sum(prediction == datatest$Class)/nrow(datatest)
```

ภาคผนวก ข

บทความวิจัยที่ได้รับการตีพิมพ์เผยแพร่ในระหว่างศึกษา

รายชื่อบทความวิจัยที่ได้รับการตีพิมพ์เผยแพร่ในระหว่างศึกษา

Keerachart Suksut, Nittaya Kerdprasop and Kittisak Kerdprasop. (2015). A Comparative Study of Time Series Classification by Using Decision Tree and Support Vector Machine. SEATUC'2015 the 9th South East Asia Technical University Consortium (SEATUC) Symposium 2015, Suranaree University of Technology, Thailand. 27-30 July 2015.

กีระชาติ สุขสุทธิ์, กิตติศักดิ์ เกิดประสพ และ นิตยา เกิดประสพ. (2560). การจำแนกชนิดของป่าด้วยซัพพอร์ตเวกเตอร์แมชชีนและขั้นตอนวิธีเชิงพันธุกรรมที่มีการเริ่มต้นใหม่. วารสารสมาคมสำรวจข้อมูลระยะไกลและสารสนเทศภูมิศาสตร์แห่งประเทศไทย, ปีที่ 18, ฉบับพิเศษ, หน้า 231-242.

Keerachart Suksut, Ratiporn Chanklan, Nuntawut Kaoungku, Kedkarn Chaiyakhan, Nittaya Kerdprasop and Kittisak Kerdprasop. (2017). Parameter Optimization for Mammogram Image Classification with Support Vector Machine. The International MultiConference of Engineers and Computer Scientists 2017. Hong Kong. 15-17 March 2017.

Keerachart Suksut, Kittisak Kerdprasop and Nittaya Kerdprasop. (2017). Support Vector Machine with Restarting Genetic Algorithm for Classifying Imbalanced Data. International Journal of Future Computer and Communication. vol. 6. no. 3. pp. 92-96.

A COMPARATIVE STUDY OF TIME SERIES CLASSIFICATION BY USING DECISION TREE AND SUPPORT VECTOR MACHINE

Keerachart Suksut*, Nittaya Kerdprasop, Kittisak Kerdprasop
School of Computer Engineering, Suranaree University of Technology, Thailand

ABSTRACT

Time series analysis can be used to forecast future events. There are many techniques for classifying time series data such as CTREE, Random Forest, and SVM. In this research, we perform a comparative study in terms of accuracy and time usage for classification with CTREE, Random Forests, and SVM. From the result we found that SVM has higher accuracy than Random Forests and CTREE respectively. CTREE can classify faster than SVM and Random Forests, respectively. We also applied Discrete Wavelet Transform (DWT) to extract features and classify with CTREE, Random Forests, and SVM. We found that with DWT Random Forests can classify better than SVM and CTREE. In terms of time usage, we found that CTREE can classify faster than Random Forests, and SVM respectively.

1. INTRODUCTION

At present, business is often highly competitive since there are many new companies. By different companies competing for their efforts as one or as a leader of the company, and is known as a successful business or organization will need to plan for the future. In general, it will use statistical knowledge to apply to their business or organization.

In planning to forecast future events can have a many techniques. Popular techniques, using time series analysis Such as the daily closing value of the Dow Jones industrial average or used to predict air travel in the next two years.

Time series analysis can be applied to many field. Wong, et al. (2014), propose blind biosignal classification model for automatically identify the type (ECG, EEG, EMG or others) of a blind biosignal, and thus can classify a disease or symptom without knowing the type of the source biosignal. Roumani, et al. (2015), propose model for predict the number of vulnerabilities using time series models and to find whether vulnerabilities have trends, levels, and seasonality components.

In this research, we show comparative time series classification with different technique (CTREE, Random Forests, and Support Vector Machine) in term of accuracy and time usage.

2. BACKGROUND

2.1 Time Series

Time series data are data that has relation with time such as the daily closing value of the Dow Jones industrial average. Time series data may be in the manner which the annual data,quarterly or monthly. This all depends on the proper implementation of the benefits.

Time series are used in many fields such as statistics, signal processing, pattern recognition, mathematical finance, weather forecasting and earthquake prediction (Hamilton & Douglas, 1994).

Elements of time series consists of four parts(trend, season, cyclical, irregular component) (Brockwell, et al., 2009).

- Trend is data changes are smooth straight line or curve in the increase or decrease. The value trend of the data movement in a relatively long period.
- Season is data changes are an increase or decrease in the same manner of the period, one that certainly. Also called seasonal changes.The unit of time may be for hourly, daily, weekly, monthly, quarterly, annual data no seasonal variation.Seasonal changes that define the duration of a single iteration in past quite certainly.
- Cyclical are similarities to the seasonal changes. The different is that changes in cycles, each cycle takes longer.
- Irregular component is the change of time series of events that we can not predict such as earthquake, flood or fire.

2.2 Classification Method

2.2.1 Decision Tree Classification

Decision tree are more techniques to classify such as Iterative Dichotomiser 3 (id3), successor of ID3 (C4.5), Classification And Regression Tree (CART), Conditional Inference Trees (CTREE), Random Forest etc. In this research we used CTREE and Random Froests to classify time series data.

CTREE (Hothorn, et al., 2006) is a non-parametric class of regression trees embed tree-structured regression models into a well defined theory of conditional inference procedures. Ctree recursive partitioning for continuous, censored, ordered, nominal and multivariate response variables in a conditional inference framework.

Random forests (Svetnik, et al., 2003) improve predictive accuracy by generate a large number of bootstrap trees (based on random sample of variable), classify a case by use each tree in this new forest, and decide a final predict outcome by combining the results across all of the trees (an average in regression, a majority vote in classification).

2.2.2 Support Vector Machine Classification

SVM (Support Vector Machine) algorithm (Cortes & Vapnik, 1995) is algorithm for classify that has been widely applied to many fields. The concept of SVM are to provide input on practice as a vector in space N dimension, then create hyperplane to separate groups of input vector into various class example for separate data into two class show in figure 1.

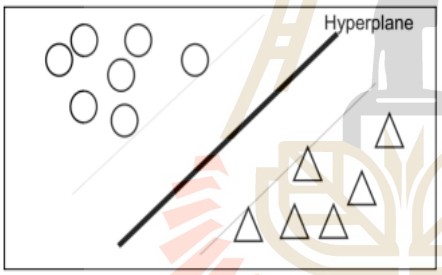


Figure 1. Hyperplane for separate data two dimension.

2.3 Extracted Features

2.3.1 Discrete Wavelet Transform

Wavelet transform (Burrus, et al., 1998) is a mathematical process to describe the structure of the signal system that contain multiple individual signals combined into one signal by signal, the signal is only a small wave called "wavelet". It is a wave that is changing continuously.

The format of the wavelet transform. In general, it can be divided into two types of wavelet (continuous wavelet transform and discrete wavelet transform).

- Continuous wavelet transform formats are similar to the signal analysis for every value of the frequency.

- Discrete wavelet transform (DTW) is the wavelet transform with analysis features by developing patterns to scale and position in a range of not continuous.

In this research we use Discrete Wavelet Transform to extract features from time series and create classification model.

3. METHODOLOGY

Researchers have designed the process of the comparative time series classification with different techniques as show in figure 2.

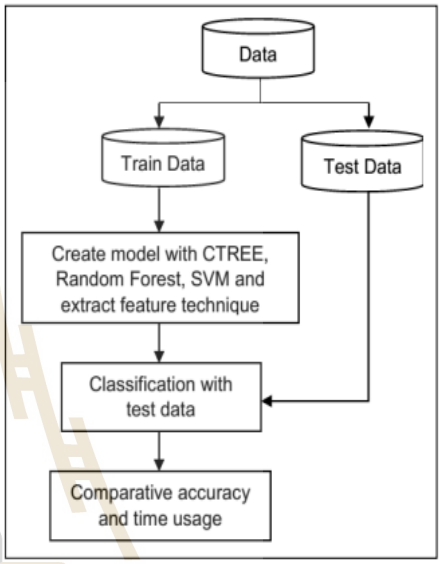


Figure 2. A research framework.

In create model process, we split task into two part, the first part we create model with train data by using CTREE, Random Forests and SVM. Second part we create model with applied Discrete Wavelet Transform to extract features from train data and create model by using CTREE, Random Forest and SVM.

In classification process, we used test data for the evaluation of the accuracy and time usage.

4. EXPERIMENTION AND RESULTS

4.1 Dataset

Our experiment used synthetic control chart time series, japanese vowels and spoken arabic digits from UCI machine learning repository.

- synthetic control chart time series dataset has 600 instances with 60 attributes and the target class has 6 class show in figure 3.

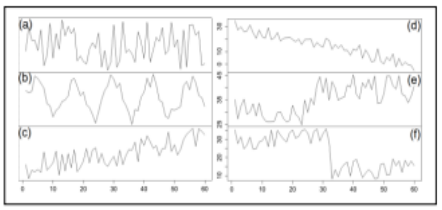


figure 3. class of synthetic control chart time series dataset (a) Normal, (b) Cyclic, (c) Increasing trend, (d) Decreasing trend (e) Upward shift and (f) Downward shift.

- japanese vowels dataset has 9,684 instances with 12 attributes and the target class has 10 class but in this research we used 3,005 instances with 12 attributes classify with 3 class (speaker 1, speaker 2, speaker 3) The target class show in figure 4.

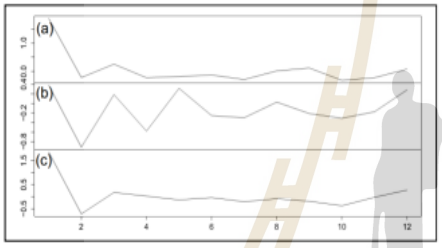


figure 4. class of japanese vowels dataset (a) Speaker 1, (b) Speaker 2, (c) Speaker 3.

- spoken arabic digits dataset has 359,118 instances with 13 attributes and the target class has 10 class but in this research we used 5,288 instances with 13 attributes classify with 2 class (male speaker, female speaker) because this dataset have large data and out of memory. The target class show in figure 5.

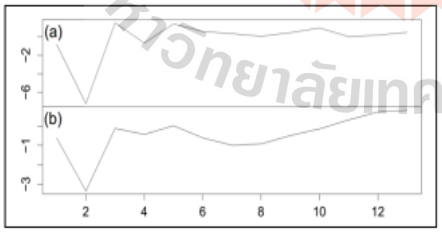


figure 5. class of spoken arabic digits dataset (a) Male speaker, (b) Female speaker.

4.2 Classification Result

In this section, we used 3 classification technique (CTREE, Random Forest, Support Vector Machine). The result show accuracy between classify with original data and used dynamic time warping before classify of each classification technique.

4.2.1 Classification result

Table 1. Comparative accuracy and time usage of synthetic control chart time series dataset.

	CTREE		RF		SVM	
	Acc.	Time	Acc.	Time	Acc.	Time
Ori	80.00	**0.41**	97.79	0.73	**98.3**	0.88
DTW	87.78	**3.2**	**96.11**	3.35	70.0	3.83

*ori = original data, DTW = dynamic time warping, Acc = accuracy, RF = random forest, SVM = support vector machine

From table 1. show the comparative accuracy and time usage for synthetic control chart time series dataset. It can be seen that when we create model with train data and classifying with different classification technique, in term of accuracy we found that SVM (98.30%) can classify better than Random Forest (97.79%) and CTREE (80.00%) respectively. In terms of time usage we found that CTREE (0.41s) can classify faster than Random Forest (0.73s) and SVM (0.88s) respectively. And when we create model with applied DTW to extract features from train data and classifying with different classification technique, in terms of accuracy we found that with DTW Random Forest (96.11%) can classify more than CTREE (87.78%) and SVM (70.00%) respectively. In terms of time usage we found that CTREE (3.20s) can classify faster than Random Forest (3.35s) and SVM (3.83s) respectively.

4.2.2 Classification japanese vowels

Table 2. Comparative accuracy and time usage of japanese vowels dataset.

	CTREE		RF		SVM	
	Acc.	Time	Acc.	Time	Acc.	Time
Ori	92.49	**0.28**	96.50	1.07	**97.58**	0.36
DTW	86.82	**7.8**	92.43	9.22	**95.41**	8.08

*ori = original data, DTW = dynamic time warping, Acc = accuracy, RF = random forest, SVM = support vector machine

From table 2. show the comparative accuracy and time usage for japanese vowels dataset. It can be seen that when we create model with train data and classifying with different classification technique, in terms of accuracy we found that SVM (97.58%) can classify better than Random Forest (96.50%) and CTREE (92.49%) respectively. In terms of time usage we found that CTREE (0.28s) can classify faster than SVM (0.36s) and Random Forest (1.07s) respectively. And when we create model with applied DTW to extract features from train data and classifying with different classification technique, in terms of accuracy we found that SVM (95.41%) has higher accuracy than Random Forest (92.43%) and CTREE (86.82%) respectively. In terms of time usage we found that CTREE (7.80s) can classify faster than SVM (8.08s) and Random Forest (9.22s) respectively.

4.2.3 Classification spoken arabic digits

Table 3. Comparative accuracy and time usage of spoken arabic digits dataset.

	CTREE		RF		SVM	
	Acc.	Time	Acc.	Time	Acc.	Time
Ori	68.96	**0.6**	74.63	3.15	**77.55**	2.23
DTW	69.29	**14.26**	**80.79**	17.05	79.69	17.86

*ori = original data, DTW = dynamic time warping, Acc = accuracy, RF = random forest, SVM = support vector machine

From table 3. show the comparative accuracy and time usage for spoken arabic digits dataset. It can be seen that when we create model with train data and classifying with different classification technique, in terms of accuracy we found that SVM (77.55%) has higher accuracy than Random Forest (74.63%) and CTREE (68.96%) respectively. In terms of time usage we found that CTREE (0.60s) can classify faster than SVM (2.23s) and Random Forest (3.15s) respectively. Then we create model with applied DWT to extract features from train data and classifying with different classification technique, in terms of accuracy we found that Random Forest (80.79%) can classify better than SVM (79.65%) and CTREE (69.29%) respectively. In terms of time usage we found that CTREE (14.26s) can classify faster than Random Forest (17.05s) and SVM (17.86s) respectively.

5. CONCLUSIONS

In this research, we show comparative in terms of accuracy and time usage for time series classification with CTREE, Random Forest, and SVM. In terms of accuracy we found that SVM can classification better than Random Forest and CTREE respectively. In term of times usage we found that CTREE can classify faster than SVM and Random Forest respectively.

We use Discrete Wavelet Transform (DWT) to extract features and create model we found that when we used extract features technique can increase accuracy for some dataset and some classification technique but as a result, it take more time. In terms of accuracy we found that with DTW Random Forest has higher accuracy than SVM and CTREE respectively. In term of times usage we found that CTREE can classify faster than Random Forest and SVM respectively.

REFERENCES

Brockwell, Peter, J. and Richard, A. D., Time series: theory and methods, Springer Science & Business Media, 2009.

Burrus, C.S., Gopinath, R.A. and Guo, H., Introduction to Wavelets and Wavelet Transforms: A Primer, Prentice-Hall, Inc., 1998.

Cortes, C. and Vapnik, V., Support-vector networks, Machine learning, vol. 20, no. 3, pp. 273-297, 1995.

Hamilton and Douglas, J., Time series analysis, Princeton: Princeton university press, Vol. 2, 1994.

Hothorn, T., Hornik, K. and Zeileis, A., Unbiased recursive partitioning: A conditional inference framework, Journal of Computational and Graphical statistics, vol. 15, no. 3, pp. 651-674, 2006.

Roumani, Y., Nwankpa, J. K., Roumani, Y. F., Time series modeling of vulnerabilities, Computers & Security, vol. 51, pp. 32-40, 2015.

Svetnik, V., Liaw, A., Tong, C., Culberson, J. C., Sheridan, R. P. and Feuston, B. P., Random forest: a classification and regression tool for compound classification and QSAR modeling, Journal of chemical information and computer sciences, vol. 43, no. 6, pp. 1947-1958, 2003.

Wong, D. F., Chao, L. S., Zeng, X., Vai, M. I. and Lam, H. L., Time series for blind biosignal classification model, Computers in biology and medicine, vol. 54, pp. 32-36, 2014.

Keerachart Suksut is currently a PHD. student with the School of Computer Engineering, Suranaree University of Technology, Thailand. He received his bachelor degree and master degree in Computer Engineering from Suranaree University of Technology in 2011 and 2013 respectively. In his current research includes classification time series.

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

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

การจำแนกชนิดของป่าด้วยซัพพอร์ตเวกเตอร์แมชชีนและขั้นตอนวิธีเชิงพันธุกรรมที่มีการเริ่มต้นใหม่

Forest type classification using support vector machine with restarting genetic algorithm

กีระชาติ สุขสุทธิ์, กิตติศักดิ์ เกิดประสพ และ นิตยา เกิดประสพ

Keerachart Suksut, Kittisak Kerdprasop and Nittaya Kerdprasop

สาขาวิชาวิศวกรรมคอมพิวเตอร์ สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

โทรศัพท์ 08-7961-9062 e-mail: mikaiterng@gmail.com

บทคัดย่อ

การทำเหมืองข้อมูลในลักษณะของการเรียนรู้ด้วยวิธีอัตโนมัติเพื่อสร้างโมเดลสำหรับจำแนกประเภทข้อมูลมีจุดประสงค์เพื่อนำโมเดลที่ได้ไปใช้ทำนายประเภทให้กับข้อมูลใหม่ที่ไม่ทราบค่าว่าเป็นประเภทใด ในปัจจุบันวิธีการจำแนกประเภทข้อมูลด้วยเทคนิคซัพพอร์ตเวกเตอร์แมชชีนเริ่มเป็นที่นิยมนำมาประยุกต์ใช้เนื่องจากให้โมเดลที่มีความแม่นยำสูงในการจำแนก ในงานวิจัยนี้จึงเสนอวิธีการใหม่ในการปรับปรุงการจำแนกประเภทข้อมูลของซัพพอร์ตเวกเตอร์แมชชีนให้มีประสิทธิภาพการจำแนกดียิ่งขึ้นด้วยการเพิ่มขั้นตอนวิธีเชิงพันธุกรรมที่มีเทคนิคการเริ่มต้นใหม่ ทั้งนี้ขั้นตอนวิธีเชิงพันธุกรรมที่มีเทคนิคการเริ่มต้นใหม่จะทำหน้าที่หาค่าพารามิเตอร์ที่เหมาะสมสำหรับซัพพอร์ตเวกเตอร์แมชชีน เทคนิคที่นำเสนอใหม่นี้จะนำมาใช้ในการจำแนกชนิดของป่า ในการจำแนกอัตโนมัตินี้จะใช้ข้อมูลภาพถ่ายพื้นที่ป่าในประเทศญี่ปุ่นจากดาวเทียมแอสเตอร์ จากผลการทดลองพบว่าวิธีที่นำเสนอสามารถจำแนกประเภทป่าได้แม่นยำกว่าวิธีดั้งเดิม (วิธีการจำแนกโดยไม่มีการปรับค่าพารามิเตอร์)

คำสำคัญ: การทำเหมืองข้อมูล การจำแนกประเภทข้อมูล ซัพพอร์ตเวกเตอร์แมชชีน ขั้นตอนวิธีเชิงพันธุกรรมที่มีการเริ่มต้นใหม่

ABSTRACT

The objective of data mining that applies automatic learning technique to induce classification model is to apply the model to predict class of the new data with unknown type. Currently, data classification with support vector machine is gaining popularity due to high classification accuracy of the induced model. In this paper, we propose a new technique to improve classification accuracy of the support vector machine. The improvement is achieved through the incorporating of genetic algorithm with restarting concept. The restarting genetic algorithm can help support vector machine by learning an appropriate set of parameters. The power of the proposed classification technique is demonstrated through its application for image-based forest type classification over the forest area in Japan. We use the satellite image data from the ASTER satellite. The results show that the proposed technique can classify the forest types with higher accuracy than the traditional techniques (default parameter).

KEY WORDS: Data mining, data classification, support vector machine, restarting genetic algorithm

1.บทนำ

การทำเหมืองข้อมูล (Han and Kamber, 2006) เป็นกระบวนการหาองค์ความรู้จากข้อมูลที่มีขนาดใหญ่ เพื่อหารูปแบบ หรือหาความสัมพันธ์ของข้อมูลที่ซ่อนอยู่ภายในข้อมูลเหล่านั้นด้วยวิธีการทางคณิตศาสตร์ สถิติ หรือคอมพิวเตอร์ การทำเหมืองข้อมูลมีหลายประเภทโดยจะขึ้นอยู่กับวัตถุประสงค์ที่จะนำไปใช้งาน เช่น การจำแนกประเภทข้อมูล (Data Classification) การหากฎความสัมพันธ์ของข้อมูล (Association Rule) และการจัดกลุ่มข้อมูล (Clustering) เป็นต้น ปัจจุบันมีการนำเทคนิคการทำเหมืองข้อมูลไปประยุกต์ใช้งานอย่างกว้างขวาง ไม่ว่าจะเป็นการนำไปประยุกต์ใช้งานทางด้านอุตสาหกรรมที่ประยุกต์ใช้การจำแนกประเภทข้อมูลเข้ามาแก้ปัญหาด้วยการนำโมเดลหรือกฎที่ได้จากการจำแนกประเภทข้อมูลไปทำการจำแนกข้อมูลที่ยังไม่ทราบประเภท

เทคนิคที่นิยมนำมาใช้ในการจำแนกประเภทข้อมูลมีหลายเทคนิค เช่น การใช้โครงข่ายประสาทเทียม (Artificial Neural Network : ANN) ซึ่งมีแนวคิดพื้นฐานมาจากการจำลองการทำงานของสมองมนุษย์ด้วยการทำให้คอมพิวเตอร์สามารถเรียนรู้ได้เหมือนกับที่มนุษย์เรียนรู้ หรือการใช้การหากฎความสัมพันธ์ของข้อมูลมาจำแนกข้อมูลแต่ละประเภทออกจากกันโดยอาศัยรูปแบบของกฎการเรียนรู้เป็นตัวจำแนก หรือการใช้ต้นไม้ตัดสินใจ (Decision Tree) ซึ่งเป็นเทคนิคในการจำแนกประเภทข้อมูลให้อยู่ในลักษณะคล้ายต้นไม้ โดยมีโหนดราก (Root Node) อยู่บนสุดและมีโหนดใบ (Leaf Node) อยู่ล่างสุด โดยในแต่ละโหนดจะหมายถึงแอททริบิวต์ (Attribute) ที่นำมาใช้ในการจำแนก และค่าทั้งหมดที่เป็นไปได้จะอยู่ที่โหนดใบ หรือการใช้เทคนิคนาอีฟเบย์ (Naïve Bayes) ซึ่งเป็นการจำแนกประเภทข้อมูลโดยใช้ค่าความน่าจะเป็นของข้อมูลฝึกสอนมาเป็นเกณฑ์ในการตัดสินใจจำแนกข้อมูลที่ไม่ทราบประเภท หรือวิธีการที่ได้รับความนิยมในปัจจุบันสูงคือการใช้อัลกอริทึมซัพพอร์ทเวกเตอร์แมชชีน ซึ่งเป็นการสร้างเส้นแบ่งสำหรับจำแนกประเภทข้อมูลโดยที่เส้นแบ่งประเภทข้อมูลนั้นจะมีระยะห่างระหว่างกลุ่มข้อมูลแต่ละกลุ่มจึงทำให้สามารถจำแนกข้อมูลที่ยังไม่ทราบค่าว่าเป็นประเภทใดได้ดี เนื่องจากการใช้อัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนมาใช้ในการจำแนกมีประสิทธิภาพในการจำแนกประเภทข้อมูลสูงจึงเป็นที่นิยมในการนำไปประยุกต์ใช้งานในการจำแนกด้านต่าง ๆ ดังงานวิจัยของ Liao et al. (2014) และ Cateni et al. (2014) โดยการนำไปประยุกต์ใช้ในการจำแนกข้อมูลไม่สมดุล โดยผลที่ได้พบว่าการใช้อัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนให้ประสิทธิภาพในการจำแนกดีกว่าการใช้อัลกอริทึมอื่น ๆ

การเพิ่มประสิทธิภาพในการจำแนกประเภทข้อมูลสามารถเพิ่มประสิทธิภาพด้วยการปรับปรุงอัลกอริทึมในการจำแนกประเภทข้อมูล ซึ่งวิธีที่ได้รับความนิยมในการนำมาปรับปรุงอัลกอริทึมสำหรับการจำแนกประเภทข้อมูลคือการหาค่าพารามิเตอร์ที่เหมาะสมที่จะใช้ในการจำแนกประเภทข้อมูลนั้น ๆ โดยเทคนิคในการหาค่าพารามิเตอร์ที่นิยมใช้ในปัจจุบันได้แก่ การใช้ขั้นตอนวิธีเชิงพันธุกรรม (Genetic algorithm) ดังงานวิจัยของ Yin et al. (2011), Jamshidi et al. (2015) และ Shiff et al. (2016) ที่ได้ประยุกต์ใช้ขั้นตอนวิธีเชิงพันธุกรรมในการหาค่าพารามิเตอร์เพื่อเพิ่มประสิทธิภาพการจำแนกให้กับอัลกอริทึมสำหรับจำแนกประเภท แต่การใช้ขั้นตอนวิธีเชิงพันธุกรรมมักจะพบปัญหาการสุ่มสร้างประชากรเริ่มต้นไม่ครอบคลุมกับช่วงของคำตอบที่ต้องการ (เมื่อมีการกระตุ้นให้เกิดการกลายพันธุ์หรือการสลับสายพันธุ์ แต่ประสิทธิภาพ หรือความเหมาะสมของประชากรรุ่นใหม่ไม่เปลี่ยนแปลง หรือด้อยกว่าประชากรดั้งเดิม) ดังนั้นงานวิจัยนี้จึงเสนอเทคนิคสำหรับการหาค่าพารามิเตอร์ที่เหมาะสมด้วยขั้นตอนวิธีเชิงพันธุกรรมที่มีการเริ่มต้นใหม่เพื่อเริ่มขั้นตอนการสุ่มสร้างประชากรเริ่มต้นใหม่โดยการนำประชากรระดับหัวกะทิตามจำนวนที่กำหนดเป็นประชากรเริ่มต้นด้วย

2. วัตถุประสงค์

ในงานวิจัยนี้มีวัตถุประสงค์เพื่อปรับปรุงประสิทธิภาพการจำแนกประเภทข้อมูลด้วยการหาค่าพารามิเตอร์ที่เหมาะสมด้วยขั้นตอนวิธีเชิงพันธุกรรมที่มีการเริ่มต้นใหม่เพื่อหาค่าพารามิเตอร์ที่เหมาะสมสำหรับอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีน

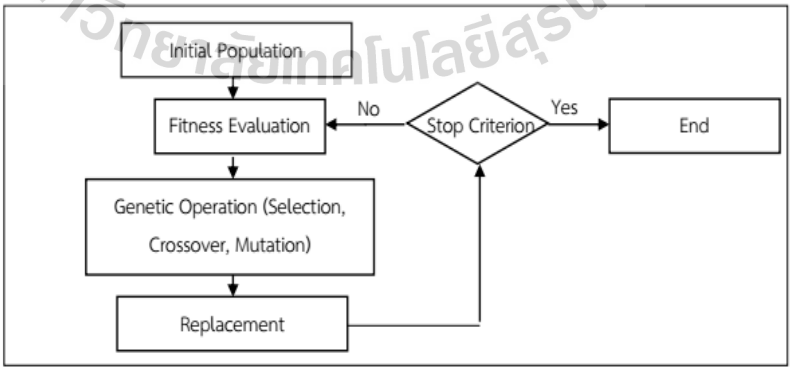
3. ทฤษฎีที่เกี่ยวข้อง

เนื้อหาในบทนี้ประกอบด้วยทฤษฎีที่เกี่ยวข้องสำหรับการจำแนกชนิดของป่าด้วยซัพพอร์ตเวกเตอร์แมชชีน และขั้นตอนวิธีเชิงพันธุกรรมที่มีการเริ่มต้นใหม่ ซึ่งประกอบไปด้วยขั้นตอนวิธีเชิงพันธุกรรม (Genetic Algorithm) การจำแนกประเภทข้อมูลด้วยอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีน และเกณฑ์สำหรับการประเมินประสิทธิภาพการจำแนกข้อมูล

3.1 ขั้นตอนวิธีเชิงพันธุกรรม

ขั้นตอนเชิงพันธุกรรม เป็นวิธีการค้นหาคำตอบโดยอาศัยการเลียนแบบวิวัฒนาการทางธรรมชาติ โดยมีพื้นฐานแนวคิดมาจากทฤษฎีวิวัฒนาการทางธรรมชาติของ Charie Darwin คือ ผู้ที่แข็งแกร่งกว่าย่อมมีโอกาสในการอยู่รอดมากกว่าผู้ที่อ่อนแอกว่า และมีโอกาสที่จะถ่ายทอดลักษณะทางพันธุกรรมที่แข็งแกร่งเหล่านั้นไปยังรุ่นลูกหลานต่อไป โดยขั้นตอนวิธีเชิงพันธุกรรมเริ่มเป็นที่รู้จักจากงานวิจัยของ John Holland (Holland, 1975) โดยประยุคนำเอาการวิวัฒนาการของสิ่งมีชีวิตในระบบชีววิทยามาใช้ในการคำนวณด้วยคอมพิวเตอร์ หลังจากนั้นก็เริ่มมีการนำขั้นตอนวิธีเชิงพันธุกรรมไปประยุกต์ใช้ในงานด้านต่าง ๆ กันอย่างแพร่หลาย การทำงานของขั้นตอนวิธีเชิงพันธุกรรมสามารถแสดงดังรูปที่ 1

จากรูปที่ 1 สามารถอธิบายขั้นตอนการทำงานของขั้นตอนวิธีเชิงพันธุกรรมได้ดังนี้ สำหรับขั้นตอนการสร้างประชากรเริ่มต้น จะดำเนินการโดยการสุ่มสร้างประชากรจากกลุ่มข้อมูลที่มีอยู่เพื่อนำประชากรเข้าสู่กระบวนการของ Genetic Algorithm โดยในการสุ่ม จะสุ่มจำนวนประชากรให้ได้เท่ากับจำนวนประชากร (Population Size) ที่กำหนด หลังจากสุ่มประชากรเริ่มต้นแล้ว จะทำการคำนวณค่าความเหมาะสมของแต่ละประชากร เพื่อค้นหาประชากรที่มีความเหมาะสมตามเกณฑ์ที่กำหนดไปเป็นโครโมโซมตั้งต้นในการสืบทอดพันธุกรรม เมื่อได้โครโมโซมตั้งต้นแล้วจะดำเนินการทาง Genetic Algorithm ซึ่งได้แก่การคัดเลือก (Selection) การสลับสายพันธุ์ (Crossover) และทำการกลายพันธุ์ (Mutation) โดยในการคัดเลือกจะทำการคัดเลือกโครโมโซมตั้งต้นที่มีความเหมาะสมสูงที่สุด ส่วนขั้นตอน



รูปที่ 1 ผังการทำงานของขั้นตอนวิธีเชิงพันธุกรรมอย่างง่าย

การสลับสายพันธุ์จะทำการสลับสายพันธุ์ระหว่างโครโมโซมตั้งต้นเพื่อให้เกิดความหลากหลายทางพันธุกรรม หลังจากนั้นจะทำการกลายพันธุ์โครโมโซมที่เกิดจากการสลับสายพันธุ์เพื่อให้เกิดการเปลี่ยนแปลงยีนภายในโครโมโซมที่มีอยู่เดิม หลังจากนั้นจะนำประชากรใหม่ที่ได้ไปแทนที่ประชากรรุ่นก่อนหน้า จนกระทั่งได้ประชากรที่ตรงตามเงื่อนไขที่กำหนด

1. การเข้ารหัสโครโมโซม

การเข้ารหัสโครโมโซมมีความสำคัญสำหรับขั้นตอนวิธีเชิงพันธุกรรมเป็นอย่างมาก เพราะก่อนที่จะ เริ่มกระบวนการต่าง ๆ ของขั้นตอนเชิงพันธุกรรมจำเป็นที่จะต้องมีการเข้ารหัสโครโมโซมก่อน โดยในขั้นตอนนี้จะเป็นการออกแบบให้โครโมโซมเป็นตัวแทนของคำตอบของสิ่งที่ต้องการค้นหา โดยจะเลือกใช้วิธีการเข้ารหัสแบบใดก็ได้ซึ่งจะขึ้นอยู่กับความเหมาะสมของการแก้ปัญหา

2 การสร้างประชากรเริ่มต้น

การสร้างประชากรเริ่มต้นจะดำเนินการโดยการสุ่มสร้างค่าขึ้นมาจากกลุ่มข้อมูลที่มีอยู่ เพื่อนำประชากรเข้าสู่กระบวนการขั้นตอนเชิงพันธุกรรม โดยในการสุ่มจะต้องสุ่มให้ได้จำนวนเท่ากับขนาดประชากร (Population Size) ที่กำหนดไว้ซึ่งในขั้นตอนนี้จะยังไม่สนใจค่าความเหมาะสมของแต่ละโครโมโซม

3 ฟังก์ชันค่าความเหมาะสม

ฟังก์ชันค่าความเหมาะสม จะเป็นตัวกำหนดค่าความเหมาะสมของแต่ละโครโมโซม เพื่อให้คะแนนความเหมาะสมของแต่ละโครโมโซม โดยโครโมโซมแต่ละตัวจะมีค่าความเหมาะสมของตัวเองเพื่อใช้สำหรับพิจารณาว่าโครโมโซมตัวนั้นเหมาะสมหรือไม่ที่จะนำไปใช้ในการสืบทอดพันธุกรรม โดยวิธีการคำนวณค่าความเหมาะสมนั้นจะใช้สมการที่สอดคล้องกับแต่ละปัญหา โดยทั่วไปแล้วฟังก์ชันค่าความเหมาะสมจะถูกกำหนดให้เหมาะสมกับลักษณะงานที่นำไปใช้งาน เช่นในด้านการจำแนกข้อมูล อาจจะใช้ฟังก์ชันความเหมาะสมคือ ใช้ค่าความแม่นยำในการจำแนก (Accuracy)

4 การดำเนินการทางขั้นตอนวิธีเชิงพันธุกรรม

การดำเนินการทางขั้นตอนวิธีเชิงพันธุกรรม จะประกอบไปด้วย 3 ขั้นตอนสำคัญ (Zheng et al., 2002) ได้แก่ การคัดเลือกสายพันธุ์ (Selection) เพื่อเป็นประชากรในรุ่นถัดไป การสลับสายพันธุ์ (Crossover) และการกลายพันธุ์ (Mutation)

- การคัดเลือกสายพันธุ์ เป็นวิธีการที่สนับสนุนให้ประชากรที่มีความเหมาะสมในปัจจุบันถูกส่งต่อไปยังรุ่นถัดไป โดยคัดเลือกจากโครโมโซมที่ดีที่สุดจากภายในกลุ่มประชากรทั้งหมด ซึ่งโครโมโซมที่ได้จะถูกนำไปใช้เป็นโครโมโซมพ่อแม่ในการสืบพันธ์เพื่อใช้ในการให้กำเนิดลูกหลายในรุ่นต่อไป หลักของการอยู่รอดของสิ่งมีชีวิตที่เหมาะสมจะสามารถอยู่รอดได้หากต้นกำเนิดสายพันธ์มีความเหมาะสม ดังนั้นจึงต้องเลือกโครโมโซมรุ่นพ่อแม่ที่มีค่าความเหมาะสมสูงที่สุดนั่นเอง

- การสลับสายพันธุ์ เป็นกระบวนการที่สำคัญของขั้นตอนวิธีเชิงพันธุกรรม โดยเมื่อมีการสลับสายพันธุ์เกิดขึ้นจะทำให้เกิดการเปลี่ยนแปลงของสิ่งมีชีวิตให้มีความหลากหลายมากยิ่งขึ้น ในขั้นตอนของการสลับสายพันธุ์จะนำสมาชิกของประชากรที่ผ่านการคัดเลือกมาเป็นคู่ ๆ โดยจะกำหนดให้เป็นสมาชิกรุ่นพ่อแม่ (Parent Individual) นำมาผสมกันเพื่อให้ได้โครโมโซมใหม่ที่เป็นรุ่นลูกขึ้นมา โดยการสุ่มเลือกสมาชิกรุ่นพ่อกับสมาชิกรุ่นแม่มาทำการสลับสายพันธุ์จะถูกกำหนดโดยความน่าจะเป็นในการสลับสายพันธุ์ (Crossover Probability)

- การกลายพันธุ์ เป็นวิธีการแปรผันยีนบางตำแหน่ง หรืออาจจะทุกตำแหน่งที่ อยู่ในโครโมโซม ซึ่งมีวัตถุประสงค์เพื่อทำให้ค่าของโครโมโซมที่มีอยู่เดิมเกิดการเปลี่ยนแปลง โดยปกติแล้วการกลายพันธุ์จะทำการสุ่มตำแหน่งที่ต้องการกลายพันธุ์จากความน่าจะเป็นในการกลายพันธุ์ (Probability of Mutation) ซึ่งโดยทั่วไปจะมีความน่าจะเปลี่ยนในการกลายพันธุ์มีค่าน้อย โดยจะอยู่ระหว่าง 0 ถึง 0.1

5.การแทนที่

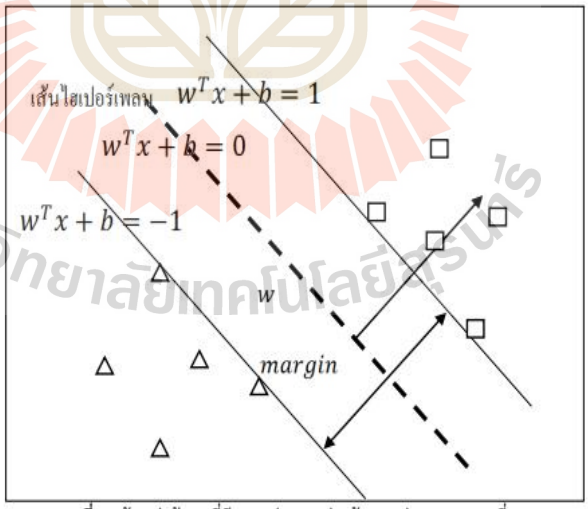
การแทนที่ เป็นขั้นตอนหลังจากที่ขั้นตอนทางพันธุกรรมได้โครโมโซมรุ่นลูกหลานเรียบร้อยแล้ว โดยจะนำโครโมโซมลูกหลานใหม่นี้ไปแทนที่ในประชากรรุ่นเก่า โดยมีวัตถุประสงค์ในการแทนที่ประชากรรุ่นเก่าเพื่อทำให้ประชากรรุ่นใหม่เป็นโครโมโซมที่ดีกว่าเพราะได้สายพันธุ์ที่ดีจากต้นกำเนิดสายพันธุ์ โดยจะทำให้โครโมโซมรุ่นใหม่ประกอบไปด้วยโครโมโซมใหม่ ๆ ที่สืบสายพันธุ์มาจากโครโมโซมรุ่นพ่อแม่ที่ผ่านการคัดเลือกแล้ว

6.การตรวจสอบเงื่อนไขสิ้นสุดการทำงาน

การตรวจสอบเงื่อนไขสิ้นสุดการทำงาน เป็นขั้นตอนของการตรวจสอบว่าจบกระบวนการทางพันธุกรรมแล้วหรือยัง ซึ่งการทำงานของขั้นตอนวิธีเชิงพันธุกรรมจะทำงานวนเวียนเป็นวัฏจักรหมุนเวียนอยู่เช่นเดิม จนกระทั่งถึงจุดหนึ่งตามเงื่อนไขที่กำหนดไว้ เช่น ได้จำนวนประชากรที่มีค่าความเหมาะสมตามจำนวนที่กำหนด หรือพบคำตอบที่ดีที่สุดตามเกณฑ์ที่ตั้งเป้าไว้เรียบร้อยแล้ว เป็นต้น ซึ่งหากยังไม่เข้าเงื่อนไขสิ้นสุดการทำงาน ขั้นตอนวิธีเชิงพันธุกรรมก็จะทำการกลับไปทำงานวนเวียนจนกว่าจะเป็นไปตามเงื่อนไขสิ้นสุดการทำงาน

3.2 การจำแนกประเภทข้อมูลด้วยอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีน

ซัพพอร์ตเวกเตอร์แมชชีน(Support Vector Machine) (Cortes and Vapnik, 1995) เป็นอัลกอริทึมที่ใช้ในการจำแนกประเภทข้อมูลในแต่ละคลาสที่ได้รับความนิยมมาก เนื่องจากความสามารถในการจำแนกประเภทข้อมูลในแต่ละคลาสมีความแม่นยำสูง โดยหลักการสำคัญของอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนคือการสร้างเส้นแบ่ง (Hyper plane) เพื่อแบ่งแยกประเภทข้อมูลออกจากกัน การเลือกเส้นแบ่งที่เหมาะสมสำหรับอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีน คือการสร้างเส้นแบ่งที่มีขนาดความกว้างของขอบ (Margin) ระหว่างเส้นแบ่งไปยังจุดข้อมูลในแต่ละคลาสมากที่สุด เพื่อเพิ่มประสิทธิภาพในการจำแนกประเภทข้อมูลที่ยังไม่ทราบคลาสของข้อมูล แสดงดังรูปที่ 2



รูปที่ 2 เส้นแบ่งข้อมูลที่มีระยะห่างระหว่างข้อมูลแต่ละคลาสมากที่สุด

จากภาพที่ 2.12 เส้นแบ่งนั้นจะทำการแบ่งข้อมูลทั้งสองคลาสออกจากกันด้วยระยะห่างระหว่างข้อมูลทั้งสองคลาสมากที่สุด และมีเวกเตอร์ถ่วงน้ำหนัก w (Weight Vector) เป็นตัวกำหนดทิศทางและความเอียงของไฮเปอร์เพลน ซึ่งเวกเตอร์ w จะตั้งฉากกับเส้นแบ่ง และข้อมูลจะถูกแปลงให้อยู่ในรูปแบบเวกเตอร์ x ส่วน y จะเป็นตัวกำหนดว่าข้อมูลจุดนั้นจะเป็นคลาส 1 หรือคลาส -1 โดยสามารถเขียนเป็นสมการได้ดังสมการที่ 1

$$w^T x + b \geq 1, when\ y_i = +1$$
$$w^T x + b \leq 1, when y_i = -1 \quad (1)$$

เมื่อ

w คือเวกเตอร์ถ่วงน้ำหนัก (Weight Vector)

b คือค่าไบแอส (Bias)

การหาค่าเวกเตอร์ถ่วงน้ำหนัก สามารถหาได้จากความชันของเส้นแบ่งที่สร้างขึ้น นั่นคือ เวกเตอร์ถ่วงน้ำหนักคือเส้นที่ลากไปตั้งฉากกับเส้นแบ่ง และค่าไบแอสจะเป็นตัวกำหนดระยะห่างระหว่างเส้นแบ่งกับจุดกำเนิด (Origin) เมื่อพิจารณาข้อมูล 2 มิติ โดยที่เส้นแบ่งเป็นเส้นตรง และกำหนดให้จุดทุกจุด $X = (x_1, x_2)^T$ จะได้สมการของเส้นแบ่ง ดังสมการที่

$$h(x) = w^T x + b = w_1 x_1 + w_2 x_2 + b = 0$$

เมื่อพิจารณาจุด 2 จุดบนเส้นแบ่ง ได้แก่จุด $A = (A_1, A_2)$ และจุด $B = (B_1, B_2)$ และเวกเตอร์ถ่วงน้ำหนักหมายถึงความชันของเส้นแบ่ง สามารถคำนวณได้จากสมการที่ 2

$$weigth\ vector = -\frac{w_1}{w_2} = -\frac{(B_2 - A_2)}{(B_1 - A_1)} \quad (2)$$

และสามารถคำนวณหาขนาดความกว้างของขอบได้ดังสมการที่ 3

$$margin = \frac{2}{||w||} \quad (3)$$

โดยที่ขนาดของ w สามารถคำนวณได้จากสมการ 4

$$||w|| = \sqrt{w_1^2 + w_2^2} \quad (4)$$

ในบางข้อมูลการใช้วิธีการจำแนกประเภทข้อมูลด้วยอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนแบบใช้เคอร์เนลเส้นตรง (Linear Kernel) นั้นไม่สามารถที่จะจำแนกข้อมูลได้ จึงมีการพัฒนาเคอร์เนล (Chistianini and Shawe-Taylor, 2000; Muller et al, 2001) เพื่อใช้ร่วมกับอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนสำหรับจำแนกประเภทข้อมูลที่ไม่สามารถจำแนกด้วยเส้นตรงได้ โดยเคอร์เนลต่าง ๆ จะเป็นการประยุกต์ใช้สมการเพื่อสร้างเส้นแบ่งในรูปแบบอื่น ๆ ที่ไม่ใช่เส้นตรง

สำหรับการจำแนกคลาสของข้อมูลด้วยอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีน จะประกอบไปด้วยพารามิเตอร์ต่าง ๆ มากมาย เพื่อนำไปตั้งค่าให้อัลกอริทึมให้มีความสามารถในการจำแนกประเภทข้อมูลได้มีประสิทธิภาพมากยิ่งขึ้น โดยพารามิเตอร์ที่นิยมปรับค่าเพื่อเพิ่มประสิทธิภาพอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีน ได้แก่ พารามิเตอร์ C, พารามิเตอร์ Epsilon และพารามิเตอร์ Gamma (สำหรับจำแนกด้วยเคอร์เนลเรเดียลเบสิสฟังก์ชัน) โดยที่

- พารามิเตอร์ C ทำหน้าที่เป็นตัวควบคุมค่าใช้จ่ายในการจำแนกข้อมูลผิดคลาสในชุดข้อมูลฝึกสอน เนื่องจากอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนจะทำการหาไฮเปอร์เพลนที่มีความกว้างของขอบระหว่างแต่ละคลาสมากที่สุด โดยจะเกิดการแลกเปลี่ยนระหว่างการหาขอบที่มีระยะห่างน้อยแต่ไม่มีความผิดพลาดในการจำแนกข้อมูลเลย หรืออาจจะมีการจำแนกข้อมูลผิดพลาดเพียงเล็กน้อย กับการหาขอบที่มีระยะห่างของขอบกว้างมากแต่แลกกับการจำแนกข้อมูลผิดพลาดมากขึ้น โดยหากค่าพารามิเตอร์ C มีค่าน้อยจะทำให้ความกว้างของขอบมีขนาดใหญ่ ทำให้โมเดลมีความยืดหยุ่นต่อข้อมูลในอนาคตสูง และหากค่าพารามิเตอร์ C มีค่ามากจะส่งผลให้ความกว้างของขอบมีขนาดเล็ก อาจจะทำให้เกิดปัญหา Over Fitting ได้

- พารามิเตอร์ Epsilon ทำหน้าที่ควบคุมการเพิ่มประสิทธิภาพในการจำแนกข้อมูล โดยจะช่วยในการลู่เข้าหาค่าไฮเปอร์แพลนที่ดีที่สุดได้ดียิ่งขึ้น เช่น สำหรับการทำงานในแต่ละรอบเพื่อจำแนกข้อมูล ค่าใช้จ่าย (Cost) หรือที่เรียกว่า Loss จะมีการคำนวณค่าใช้จ่ายจากการจำแนกผิดคลาสเกิดขึ้น สำหรับการทำงานในรอบถัดไป ไฮเปอร์แพลนจะมีการปรับปรุงตามข้อผิดพลาดที่พบในรอบที่ผ่านมา จนกระทั่งได้ไฮเปอร์เพลนที่มีระยะห่างตามที่กำหนด หรือดีที่สุด การกำหนดค่าพารามิเตอร์ Epsilon จะช่วยให้ลู่เข้าหาคำตอบที่ทำให้ได้ไฮเปอร์เพลนที่ดีที่สุดได้เร็วขึ้น

- พารามิเตอร์ Gamma เป็นพารามิเตอร์สำหรับ Kernel Radial Basis Function (RBF Kernel เป็น Kernel ที่ใช้จำแนกข้อมูลที่มีลักษณะไม่เชิงเส้น) สำหรับการจำแนกในสองมิติที่ไม่สามารถแบ่งแยกข้อมูลออกเป็นสองคลาสได้ โดยจะใช้การแก้ปัญหาไม่เชิงเส้น (Non-Linear) เข้ามาจำแนกข้อมูลออกเป็นสองคลาส โดยพารามิเตอร์ Gamma จะทำหน้าที่ควบคุมความโค้งของเส้นไฮเปอร์แพลน โดยหากค่าพารามิเตอร์ Gamma มีค่าน้อยส่งผลให้ความโค้งของเส้นแบ่งมีความโค้งน้อย แต่หากค่าพารามิเตอร์ Gamma มีค่าสูงจะส่งผลให้ความโค้งของเส้นแบ่งมีความโค้งงอมากขึ้นทำให้สามารถสร้างเส้นแบ่งที่มีความโค้ง ความหยักเพื่อแบ่งแยกประเภทข้อมูลได้ดีขึ้น

3. 3 เกณฑ์สำหรับการประเมินประสิทธิภาพการจำแนกข้อมูล

สำหรับการประเมินประสิทธิภาพการจำแนกข้อมูลจะใช้เกณฑ์สำหรับประเมินประสิทธิภาพซึ่งได้มาจากการเปรียบเทียบการจำแนกคลาสที่ได้จากการทำนายมาเปรียบเทียบกับคลาสที่แท้จริงของข้อมูล โดยแสดงผลที่เป็นไปได้จากการทำนายในลักษณะของเมตริกซ์วัดประสิทธิภาพ (Powers, 2011) แสดงดังตารางที่ 1

ตารางที่ 1 เมตริกซ์วัดประสิทธิภาพสำหรับจำแนกข้อมูลสองคลาส

	Positive Class Prediction	Negative Class Prediction
Actual Positive Class	True Positive Class	False Negative Class
Actual Negative Class	False Positive Class	True Negative Class

จากตารางที่ 1 แถวของเมตริกซ์จะแสดงจำนวนของข้อมูลจริงของแต่ละคลาส และคอลัมน์ของเมตริกซ์จะแสดงจำนวนที่ทำนายได้ของแต่ละคลาส แบ่งออกเป็น 4 กรณี ดังนี้

กรณีที่ 1: True Positive Class หมายถึง จำนวนข้อมูลที่อยู่ในคลาส Positive แล้วโมเดลสามารถทำนายได้ถูกต้องว่าข้อมูลนั้นอยู่ในคลาส Positive

กรณีที่ 2: False Negative Class หมายถึง จำนวนข้อมูลที่อยู่ในคลาส Negative แต่โมเดลทำนายผิดพลาด โดยทำนายว่าข้อมูลนั้นอยู่ในคลาส Positive

กรณีที่ 3: False Positive Class หมายถึง จำนวนข้อมูลที่อยู่ในคลาส Positive แต่โมเดลทำนายผิดพลาด โดยทำนายว่าข้อมูลนั้นอยู่ในคลาส Negative

กรณีที่ 4: True Negative Class หมายถึง จำนวนข้อมูลที่อยู่ในคลาส Negative แล้วโมเดลสามารถทำนายได้ถูกต้องว่าข้อมูลนั้นอยู่ในคลาส Negative

ค่าความแม่นยำในการจำแนก (Accuracy)

มาตรวัดความแม่นเป็นการประเมินประสิทธิภาพการจำแนกโดยรวมของทุกคลาสของโมเดล แสดงดังสมการที่ 5

$$Accuracy = \frac{(\text{True Positive Class} + \text{True Negative Class})}{(Total\ data)} \quad (5)$$

ค่าความเที่ยง (Precision)

มาตรวัดความเที่ยง เป็นการประเมินความแม่นยำในการทำนายข้อมูลที่อยู่ในคลาส Positive โดยคำนวณจากจำนวนข้อมูลที่ทำนายเป็นคลาส Positive ได้ถูกต้อง เทียบกับจำนวนข้อมูลที่ถูกทำนายเป็นคลาส Positive ทั้งหมด แสดงดังสมการที่ 6

$$Precision = \frac{(\text{True Positive Class})}{(\text{True Positive Class} + \text{False Positive Class})} \quad (6)$$

ค่าระลึก หรือค่าความไว (Recall / Sensitivity)

มาตรวัดค่าระลึกหรือค่าความไว เป็นการประเมินความแม่นยำในการทำนายข้อมูลที่อยู่ในคลาส Positive ว่าสามารถทำนายได้ถูกต้องแม่นยำเพียงใด โดยคำนวณจากจำนวนข้อมูลที่ทำนายเป็นคลาส Positive ได้ถูกต้อง เทียบกับจำนวนข้อมูลจริงของคลาส Positive ทั้งหมด แสดงดังสมการที่ 7

$$Sensitivity = Recall = \frac{(\text{True Positive Class})}{(\text{True Positive Class} + \text{False Negative Class})} \quad (7)$$

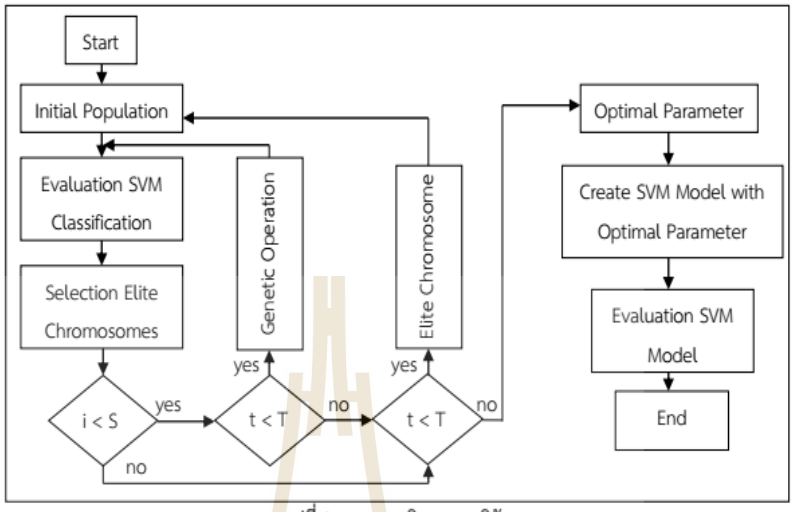
ค่าการวัดเอฟ (F-Measure)

การวัดเอฟ เป็นการประเมินความแม่นยำของการจำแนกคลาสส่วนน้อยโดยดูจากผลเฉลี่ยของ Precision และ Recall แสดงดังสมการที่ 8

$$F - measure = \frac{(2 \cdot \text{Precision} \cdot \text{Recall})}{(\text{Precision} + Recall)} \quad (8)$$

4. วิธีการวิจัย

ในงานวิจัยนี้ผู้วิจัยได้ออกแบบกระบวนการ และขั้นตอนดำเนินงานวิจัยการจำแนกชนิดของป่าด้วยซัพพอร์ตเวกเตอร์แมชชีนและขั้นตอนวิธีเชิงพันธุกรรมที่มีการเริ่มต้นใหม่ แสดงดังรูปที่ 3



รูปที่ 3 กรอบแนวคิดของการวิจัย

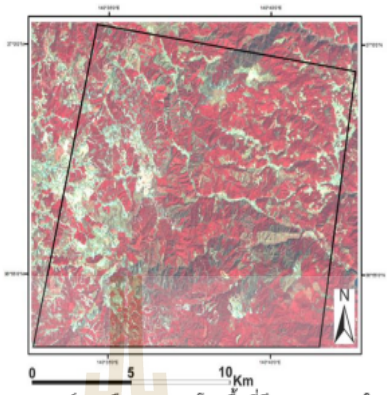
จากรูปที่ 3 สามารถอธิบายกรอบแนวคิดของการวิจัยได้ดังนี้ ในงานวิจัยนี้ใช้การเข้ารหัสโครโมโซมด้วยการใช้การเข้ารหัสแบบค่าจริง (Real Values) โดยทำการสุ่มสร้างประชากรเริ่มต้นตามจำนวนที่กำหนด หลังจากนั้นจะประเมินค่าความเหมาะสมของแต่ละโครโมโซมด้วยการใช้ค่าความถูกต้องในการจำแนก (Accuracy) จากการใช้อัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนร่วมกับข้อมูลฝึกสอนมาใช้ในการจำแนกเพื่อประเมินค่าความเหมาะสม หลังจากนั้นจะเลือกโครโมโซมระดับหัวกะทิจำนวน k ตัว (โครโมโซมที่มีค่าความเหมาะสมสูงที่สุดเป็นจำนวน k อันดับ) แล้วทำการดำเนินการทางสายพันธุ์จนกระทั่งได้ประชากรรุ่นใหม่ หากประชากรรุ่นใหม่มีค่าความเหมาะสมที่ด้อยกว่าประชากรรุ่นเก่าติดต่อกันเป็นจำนวน S รอบ ให้ทำการเริ่มต้นกระบวนการขั้นตอนวิธีเชิงพันธุกรรมใหม่ด้วยการนำประชากรระดับหัวกะทิจำนวน k ตัวไปสร้างเป็นประชากรเริ่มต้นด้วย จนกระทั่งครบรอบการทำงานที่กำหนด (T) หลังจากนั้นจะนำพารามิเตอร์ที่เหมาะสมที่สุดไปสร้างโมเดลการจำแนกข้อมูลไม่สมดุลด้วยอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีน หลังจากนั้นทดสอบโมเดลด้วยการใช้ข้อมูลทดสอบ และประเมินประสิทธิภาพด้วย ค่าความแม่นยำในการจำแนก ค่าความเที่ยง ค่าระลึกหรือค่าความไว และค่าการวัด F

4.1 ข้อมูลที่ใช้ในการวิจัย

ข้อมูลที่ใช้ในการวิจัยจะใช้ข้อมูลภาพถ่ายพื้นที่ป่าในประเทศญี่ปุ่นจากดาวเทียมแอสเตอร์ของบริเวณพื้นที่เมือง Ibaraki แสดงดังรูปที่ 4 ซึ่งได้รับมาจากฐานข้อมูล UCI Machine Learning Repository (UCI Dataset, 2015) โดยข้อมูลมีจำนวนทั้งหมด 523 ข้อมูล ประกอบไปด้วย 28 แอททริบิวต์ มีจำนวนคลาสทั้งหมด 4 คลาสได้แก่ คลาส s (Sugi Forest) คลาส h (Hinoki forest) คลาส d (Mixed deciduous forest) คลาส o (Other non-forest land) โดยแบ่งเป็นชุดข้อมูลฝึกสอน (Training Set) จำนวน 198 ข้อมูล และข้อมูลสำหรับทดสอบ (Testing Set) จำนวน 325 ข้อมูล

4.2 ขั้นตอนวิธีเชิงพันธุกรรมที่มีการเริ่มต้นใหม่

สำหรับขั้นตอนวิธีเชิงพันธุกรรมที่มีการเริ่มต้นใหม่ ผู้วิจัยได้ปรับปรุงมาจากขั้นตอนวิธีเชิงพันธุกรรมอย่างง่าย



รูปที่ 4 ภาพถ่ายจากดาวเทียมแอสเตอร์ของเมือง Ibaraki โดยพื้นที่ศึกษาแสดงภายในกรอบสีดำ (Johnson, 2012)

ขั้นตอนวิธีเชิงพันธุกรรมที่มีการเริ่มต้นใหม่

ข้อมูลเข้า : พารามิเตอร์ C, พารามิเตอร์ Epsilon, พารามิเตอร์ Gamma, จำนวนรุ่นของประชากรที่ต้องการสร้าง T, จำนวนรุ่นของประชากรที่ไม่ดีขึ้นติดกัน S

ผลลัพธ์ : พารามิเตอร์ C, พารามิเตอร์ Epsilon, พารามิเตอร์ Gamma ที่มีความเหมาะสมที่สุด

วิธีการ :

1. เข้ารหัสโครโมโซม
2. สุ่มสร้างประชากรเริ่มต้นจำนวน p
3. ประเมินความเหมาะสมของแต่ละโครโมโซมด้วยการใช้ค่าความถูกต้องในการจำแนกจากอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีน
4. สำหรับรอบที่ $t \leq T$
   1) สำหรับจำนวนรุ่นประชากรรุ่นใหม่ด้อยกว่าประชากรรุ่นเก่า $i < S$
      1) ดำเนินการทางขั้นตอนวิธีเชิงพันธุกรรม (การคัดเลือก การสลับสายพันธุ์ การกลายพันธุ์)
      2) แทนที่ประชากร
      3) ประเมินค่าความเหมาะสมของแต่ละโครโมโซม
   2) สำหรับจำนวนรุ่นประชากรรุ่นใหม่ด้อยกว่าประชากรรุ่นเก่า $i \geq S$
      1) สุ่มสร้างประชากรเริ่มต้นใหม่ โดยใช้โครโมโซมที่ดีที่สุดจากขั้นตอนวิธีเชิงพันธุกรรมเป็นประชากรเริ่มต้นด้วย
      2) ประเมินค่าความเหมาะสมของแต่ละโครโมโซม
5. ประเมินค่าความเหมาะสมของแต่ละโครโมโซมเพื่อคัดเลือกโครโมโซมเหมาะสมที่สุด

รูปที่ 5 หลักการทำงานของขั้นตอนวิธีเชิงพันธุกรรมที่มีการเริ่มต้นใหม่

โดยได้ดัดแปลงด้วยการเพิ่มเงื่อนไขในการสร้างประชากรเริ่มต้นใหม่หากประชากรรุ่นถัดไปมีความเหมาะสมเทียบเท่ากับประชากรรุ่นเก่า หรือด้อยกว่าประชากรรุ่นเก่า และยังไม่ถึงจุดสิ้นสุดในการทำงาน จะทำการสุ่มสร้างประชากร

เริ่มต้นใหม่โดยใช้ประชากรระดับหัวกะทิเป็นจำนวน k อันดับ (ประชากรที่มีค่าความเหมาะสมสูงที่สุดเป็นจำนวน k อันดับ) แสดงดังรูปที่ 5

5. ผลการวิจัย

สำหรับการหาค่าพารามิเตอร์ที่เหมาะสมด้วยการใช้ขั้นตอนวิธีเชิงพันธุกรรมอย่างง่าย และขั้นตอนวิธีเชิงพันธุกรรมที่มีการเริ่มต้นใหม่จะต้องมีการกำหนดขอบเขตในการค้นหาคำตอบ กำหนดจำนวนรอบในการทำงาน กำหนดจำนวนประชากรที่กำหนด กำหนดค่าความน่าจะเป็นในการสลับสายพันธุ์ และความน่าจะเป็นในการกลายพันธุ์ โดยการกำหนดค่าพารามิเตอร์สำหรับขั้นตอนวิธีเชิงพันธุกรรมอย่างง่าย และขั้นตอนวิธีเชิงพันธุกรรมที่มีการเริ่มต้นใหม่ แสดงดังตารางที่ 2

ตารางที่ 2 รายละเอียดพารามิเตอร์สำหรับขั้นตอนวิธีเชิงพันธุกรรมอย่างง่าย และขั้นตอนวิธีเชิงพันธุกรรมที่มีการเริ่มต้นใหม่

Probability of Crossover	Probability of Mutation	Pop Size	Iteration	C	Gamma	Epsilon	Restart GA
0.8	0.01	100	100	10^{-4} - 10^{2}	10^{-3} - 10	10^{-2} - 10	2

5.1 การจำแนกชนิดของป่าด้วยซัพพอร์ตเวกเตอร์แมชชีน

การสร้างโมเดลการจำแนกชนิดของป่าด้วยซัพพอร์ตเวกเตอร์แมชชีน จะใช้ข้อมูลสำหรับฝึกสอนร่วมกับอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนด้วยการใช้ค่าพารามิเตอร์เริ่มต้นของอัลกอริทึม และประเมินประสิทธิภาพของโมเดลด้วยการใช้ชุดข้อมูลทดสอบ โดยผลการจำแนกสามารถแสดงในรูปแบบของเมตริกซ์วัดประสิทธิภาพได้ดังตารางที่ 3

ตารางที่ 3 เมตริกซ์วัดประสิทธิภาพการจำแนกชนิดของป่าด้วยซัพพอร์ตเวกเตอร์แมชชีน

		Prediction			
		d	H	o	s
Actual Class	d	84	2	9	10
	h	0	33	0	5
	o	9	0	36	1
	s	5	12	0	119

5.2 การจำแนกชนิดของป่าด้วยซัพพอร์ตเวกเตอร์แมชชีนร่วมกับขั้นตอนวิธีเชิงพันธุกรรมอย่างง่าย

การสร้างโมเดลการจำแนกชนิดของป่าด้วยซัพพอร์ตเวกเตอร์แมชชีนร่วมกับขั้นตอนวิธีเชิงพันธุกรรมอย่างง่าย จะประยุกต์ใช้ขั้นตอนวิธีเชิงพันธุกรรมอย่างง่ายเข้ามาร่วมในการหาค่าพารามิเตอร์ที่เหมาะสมสำหรับอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีน โดยกำหนดจำนวนรุ่นประชากรที่สร้างใหม่เป็นจำนวน 100 รุ่น จำนวนประชากรในแต่ละรุ่นมีจำนวน 100 ประชากร ค่าพารามิเตอร์ C จะมีค่าอยู่ระหว่าง 10^{-4} ถึง 10^{2} สำหรับค่าพารามิเตอร์ Gamma จะมีค่าอยู่ระหว่าง 10^{-3} ถึง 10 และค่าพารามิเตอร์ Epsilon จะมีค่าอยู่ระหว่าง 10^{-2} ถึง 10 และกำหนดความน่าจะเป็นในการสลับสายพันธุ์อยู่ที่ 0.8 (หากค่าความน่าจะเป็นที่สุ่มการสลับสายพันธุ์สุ่มได้น้อยกว่า 0.8 จะไม่เกิดการสลับสายพันธุ์) และกำหนดค่าความน่าจะเป็นในการกลายพันธุ์อยู่ที่ 0.01 (หากค่าความน่าจะเป็นที่สุ่มการ

กลายพันธุ์สุ่มได้น้อยกว่า 0.01 จะไม่เกิดการกลายพันธุ์) โดยทำการหาค่าพารามิเตอร์ที่เหมาะสมด้วยขั้นตอนวิธีเชิงพันธุกรรมอย่างง่ายทั้งหมด 10 ครั้ง หลังจากนั้นจะสร้างโมเดลการจำแนกชนิดของป่าด้วยอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนโดยใช้พารามิเตอร์ที่ได้จากขั้นตอนวิธีเชิงพันธุกรรมอย่างง่ายร่วมกับข้อมูลสำหรับฝึกสอน และประเมินประสิทธิภาพของโมเดลด้วยการใช้ชุดข้อมูลทดสอบ โดยผลการจำแนกในกรณีที่แย่ที่สุดโดยได้รับค่าพารามิเตอร์จากขั้นตอนวิธีเชิงพันธุกรรมอย่างง่าย พารามิเตอร์ C = 61.86062 พารามิเตอร์ Gamma = 0.965378 และพารามิเตอร์ Epsilon = 1.007055 แสดงในรูปแบบของเมตริกซ์วัดประสิทธิภาพได้ดังตารางที่ 4 และ ผลการจำแนกในกรณีที่ดีที่สุดโดยได้รับค่าพารามิเตอร์จากขั้นตอนวิธีเชิงพันธุกรรมอย่างง่าย พารามิเตอร์ C = 37.73213 พารามิเตอร์ Gamma = 0.096948 และพารามิเตอร์ Epsilon = 0.075855 แสดงในรูปแบบของเมตริกซ์วัดประสิทธิภาพได้ดังตารางที่ 5

ตารางที่ 4 เมตริกซ์วัดประสิทธิภาพการจำแนกชนิดของป่าด้วยซัพพอร์ตเวกเตอร์แมชชีนร่วมกับขั้นตอนวิธีเชิงพันธุกรรมอย่างง่าย (กรณีแย่ที่สุด)

		Prediction			
		d	H	o	s
Actual Class	d	51	0	1	53
	h	0	17	0	21
	o	3	0	2	41
	s	2	4	0	130

ตารางที่ 5 เมตริกซ์วัดประสิทธิภาพการจำแนกชนิดของป่าด้วยซัพพอร์ตเวกเตอร์แมชชีนร่วมกับขั้นตอนวิธีเชิงพันธุกรรมอย่างง่าย (กรณีดีที่สุด)

		Prediction			
		d	H	o	s
Actual Class	d	89	0	7	6
	h	2	32	0	12
	o	8	0	37	0
	s	6	6	2	118

5.3 การจำแนกชนิดของป่าด้วยซัพพอร์ตเวกเตอร์แมชชีนร่วมกับขั้นตอนวิธีเชิงพันธุกรรมที่มีการเริ่มต้นใหม่

การสร้างโมเดลการจำแนกชนิดของป่าด้วยซัพพอร์ตเวกเตอร์แมชชีนร่วมกับขั้นตอนวิธีเชิงพันธุกรรมที่มีการเริ่มต้นใหม่ กำหนดจำนวนรุ่นประชากรทั้งหมดเป็นจำนวน 100 รุ่น จำนวนประชากรมีจำนวน 100 ประชากร ค่าพารามิเตอร์ C จะมีค่าอยู่ระหว่าง 10^{-4} ถึง 10^{2} ค่าพารามิเตอร์ Gamma จะมีค่าอยู่ระหว่าง 10^{-3} ถึง 10 และค่าพารามิเตอร์ Epsilon จะมีค่าอยู่ระหว่าง 10^{-2} ถึง 10 โดยกำหนดจำนวนประชากรรุ่นใหม่ที่มีค่าความเหมาะสมเท่าประชากรรุ่นเก่า หรือมีค่าความเหมาะสมด้อยกว่ารุ่นก่อนหน้าติดต่อกันเป็นจำนวน 2 รอบให้ทำการสุ่มสร้างประชากรเริ่มต้นใหม่ด้วยการใช้ประชากรระดับหัวกะทิเป็นประชากรเริ่มต้นด้วยเป็นจำนวน 10 ประชากร และกำหนดความน่าจะเป็นในการสลับสายพันธุ์อยู่ที่ 0.8 และกำหนดค่าความน่าจะเป็นในการกลายพันธุ์อยู่ที่ 0.01 โดยทำการหาค่าพารามิเตอร์ที่เหมาะสมด้วยขั้นตอนวิธีเชิงพันธุกรรมที่มีการเริ่มต้นใหม่ทั้งหมด 10 ครั้ง หลังจากนั้นจะสร้างโมเดลการจำแนกชนิดของป่าด้วยอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนโดยใช้พารามิเตอร์ที่ได้จากขั้นตอนวิธีเชิงพันธุกรรมที่มีการเริ่มต้นใหม่ร่วมกับข้อมูลสำหรับฝึกสอน และประเมินประสิทธิภาพของโมเดลด้วยการใช้ชุดข้อมูลทดสอบ โดยผล

การจำแนกจากพารามิเตอร์ที่ได้ในแต่ละรอบให้ผลการจำแนกในรูปแบบของเมตริกซ์วัดประสิทธิภาพในรูปแบบเดียวกันทั้ง 10 รอบ โดยที่ค่าพารามิเตอร์ที่ได้รับจะแตกต่างกันไป เช่น พารามิเตอร์ C = 45.80654 พารามิเตอร์ Gamma = 0.196194 พารามิเตอร์ Epsilon = 1.656478 และ พารามิเตอร์ C = 62.71552 พารามิเตอร์ Gamma = 0.204721 พารามิเตอร์ Epsilon = 1.734214 จะให้ผลการจำแนกในรูปแบบของเมตริกซ์วัดประสิทธิภาพในรูปแบบเดียวกัน แสดงเมตริกซ์วัดประสิทธิภาพได้ดังตารางที่ 6

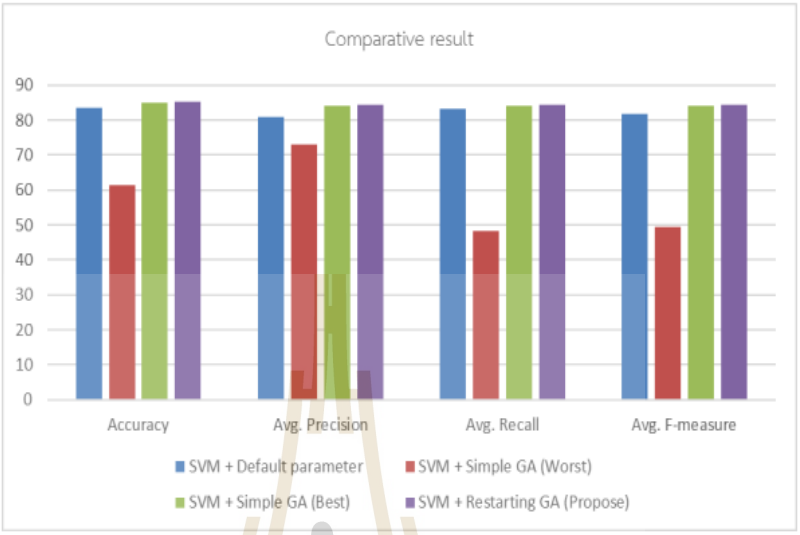
ตารางที่ 6 เมตริกซ์วัดประสิทธิภาพการจำแนกชนิดของป่าด้วยซัพพอร์ตเวกเตอร์แมชชีนร่วมกับขั้นตอนวิธีเชิงพันธุกรรมที่มีการเริ่มต้นใหม่

		Prediction			
		d	H	o	s
Actual Class	d	85	0	6	3
	h	2	33	0	11
	o	9	0	37	0
	S	9	5	3	122

โดยการประเมินประสิทธิภาพการจำแนกในรูปแบบค่าความแม่นยำในการจำแนก ค่าความเที่ยง ค่าระลึกหรือค่าความไว และค่าการวัด F ของทั้งสามวิธีการแสดงดังตารางที่ 7 และแสดงภาพแผนภูมิเปรียบเทียบผลการจำแนกของแต่ละเทคนิคโดยใช้ค่าความเฉลี่ยของแต่ละมาตรวัดดังรูปที่ 6

ตารางที่ 7 เปรียบเทียบประสิทธิภาพการจำแนก

		เทคนิคในการจำแนก			
		SVM + Default Parameter	SVM + Simple GA (Worst)	SVM + Simple GA (Best)	SVM + Restarting GA
เกณฑ์วัดประสิทธิภาพ	Accuracy	83.69 %	61.54 %	84.92 %	**85.23 %**
	Precision Class d	85.71 %	**91.07 %**	84.76 %	80.95 %
	Precision Class h	70.21 %	80.95 %	84.21 %	**86.84 %**
	Precision Class o	80.00 %	66.67 %	**80.44 %**	**80.44 %**
	Precision Class s	88.15 %	53.06 %	86.77 %	**89.71 %**
	Recall Class d	80.00 %	48.57 %	**84.76 %**	83.81 %
	Recall Class h	**86.84 %**	44.74 %	84.21 %	**86.84 %**
	Recall Class o	78.26 %	4.35 %	**80.43 %**	**80.43 %**
	Recall Class s	87.50 %	**95.59 %**	86.76 %	86.76 %
	F-measure class d	82.76 %	63.35 %	**84.76 %**	82.36 %
	F-measure class h	77.65 %	57.63 %	84.21 %	**86.84 %**
	F-measure class o	79.12 %	8.17 %	**80.44 %**	**80.44 %**
	F-measure class s	87.82 %	68.24 %	86.76 %	**88.21 %**



รูปที่ 6 เปรียบเทียบผลการจำแนกของแต่ละเทคนิค

6. วิจารณ์ผลและสรุปผล

เมื่อพิจารณาประสิทธิภาพการจำแนกจากตารางที่ 7 จะเห็นได้ว่าเมื่อพิจารณาที่ค่าความแม่นยำในการจำแนกพบว่าวิธีการที่นำเสนอมีค่าความแม่นยำในการจำแนกสูงกว่าวิธีการอื่น ๆ โดยมีค่าความแม่นยำในการจำแนกอยู่ที่ 85.23% เมื่อพิจารณาที่ค่าระลึกหรือค่าความไว พบว่ามีค่าระลึกหรือค่าความไวอยู่ในเกณฑ์ระดับกลางถึงดีที่สุดเมื่อเทียบกับวิธีการอื่น ๆ ซึ่งหมายความว่าวิธีการที่นำเสนอสามารถทำนายข้อมูลแต่ละคลาสได้ดีเมื่อเทียบกับข้อมูลจริงทั้งหมดในแต่ละคลาส หรือเมื่อพิจารณาที่ค่าความเที่ยงพบว่ามีค่าความเที่ยงในแต่ละคลาสสูงกว่าวิธีการอื่น ๆ เป็นส่วนมาก ตัวอย่างเช่นค่าความเที่ยงของคลาส s ที่มีค่าความเที่ยงอยู่ที่ 89.71% สำหรับการพิจารณาที่ค่าการวัด F พบว่าวิธีการที่นำเสนอมีค่าการวัด F สูงกว่าวิธีการอื่น ๆ เป็นส่วนมากเช่นกัน ตัวอย่างเช่นค่าการวัด F ของคลาส s ที่มีค่าการวัด F อยู่ที่ 88.21% โดยเมื่อพิจารณาภาพรวมของประสิทธิภาพในการจำแนก พบว่าการใช้ขั้นตอนวิธีเชิงพันธุกรรมที่มีการเริ่มต้นใหม่มาใช้ในการหาค่าพารามิเตอร์ที่เหมาะสมสำหรับอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนมีประสิทธิภาพสูงกว่าการใช้วิธีการอื่นที่นำมาเปรียบเทียบ

งานวิจัยนี้นำเสนอเทคนิคในการหาค่าพารามิเตอร์สำหรับการจำแนกด้วยอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนด้วยการใช้เทคนิคขั้นตอนวิธีเชิงพันธุกรรมที่มีการเริ่มต้นใหม่เพื่อแก้ปัญหาการสุ่มสร้างประชากรเริ่มต้น (พารามิเตอร์เริ่มต้น) ไม่ครอบคลุมกับช่วงของคำตอบที่ต้องการ เนื่องจากการใช้ขั้นตอนวิธีเชิงพันธุกรรมอย่างง่ายพบปัญหาไม่สามารถสร้างประชากรรุ่นใหม่ (พารามิเตอร์ชุดใหม่) ที่เหมาะสมกว่าประชากรรุ่นเก่า (พารามิเตอร์ชุดก่อน) ทำให้คำตอบที่ได้นั้นไม่ใช่คำตอบที่ดีที่สุด โดยพบว่าการใช้วิธีการที่นำเสนอช่วยแก้ปัญหาการสุ่มสร้างประชากรเริ่มต้นไม่ครอบคลุมกับช่วงของคำตอบที่ต้องการ ซึ่งพารามิเตอร์ที่ได้จากการใช้วิธีที่นำเสนอในแต่ละชุดจะให้ประสิทธิภาพในการจำแนกประเภทข้อมูลเท่ากัน ในขณะที่ขั้นตอนวิธีเชิงพันธุกรรมแบบง่ายจะพบปัญหาดังกล่าวจึงต้องทำการหาค่าพารามิเตอร์มากกว่าหนึ่งชุด เพื่อเพิ่มโอกาสในการสุ่มสร้างประชากรเริ่มต้นให้ครอบคลุมกับช่วงของคำตอบ ส่วนการใช้พารามิเตอร์เริ่มต้นของอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนจะให้ประสิทธิภาพที่ด้อยกว่าวิธีการปรับค่าพารามิเตอร์

7. ข้อเสนอแนะ

ในงานวิจัยนี้ได้เสนอวิธีการปรับค่าพารามิเตอร์สำหรับการจำแนกด้วยอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนเพียงวิธีเดียวเท่านั้น โดยการประยุกต์ใช้ขั้นตอนวิธีเชิงพันธุกรรมที่มีการเริ่มต้นใหม่เข้ามาช่วยในการหาค่าพารามิเตอร์ที่เหมาะสม และใช้ในการแก้ปัญหาการสุ่มสร้างประชากรเริ่มต้นไม่ครอบคลุม ซึ่งในปัจจุบันมีเทคนิคและวิธีการในการหาค่าพารามิเตอร์ หรือวิธีการหาคำตอบที่เหมาะสมที่สุดวิธีอื่น เช่นการใช้ Particle Swarm Intelligence เพื่อหาค่าพารามิเตอร์ที่เหมาะสมเพื่อเพิ่มประสิทธิภาพในการจำแนกซึ่งสามารถนำเทคนิคการเริ่มต้นใหม่ไปประยุกต์ใช้งานได้

เอกสารอ้างอิง

Cateni, S., Colla, V., and Vannucci, M. (2014). A method for resampling imbalanced datasets in binary classification tasks for real-world problems. **Neurocomputing**, 135:32-41.

Chistianini, N., and Shawe-Taylor J. (2000). **An Introduction to Support Vector Machines, and Other Kernel-based Learning Methods**. Cambridge University Press.

Cortes C, and Vapnik V. (1995). Support vector network. **Machine Learning**, 20(3):273–297.

Han, J., and Kamber, M. (2006). **Data mining: Concepts and Techniques**. Morgan Kaufmann.

Holland, H. (1975). **Adaptation in Natural and Artificial Systems**. Ann Arbor: The University of Michigan Press, Michigan.

Jamshidi, M., Ghaedi, M., Dashtian, K., Hajati, S., and Bazrafshan, A. (2015). Ultrasound-assisted removal of Al 3+ ions and Alizarin red S by activated carbon engrafted with Ag nanoparticles: central composite design and genetic algorithm optimization. **RSC Advances**, 5(73):59522-59532.

Johnson, B., Tateishi, R., and Kobayashi, T. (2012). Remote sensing of fractional green vegetation cover using spatially-interpolated endmembers. **Remote Sensing**, 4(9):2619-2634.

Liao, J. J., Shih, C. H., Chen, T. F., and Hsu, M. F. (2014). An ensemble-based model for two class imbalanced financial problem. **Economic Modelling**, 37:175-183.

Muller, KR., Mika, S., Ratsch, G., Tsuda, K., and Scholkopf, B. (2001). An Introduction to kernel-based learning algorithms. **IEEE Transactions on Neural Networks**, 12(2):199–222.

Powers, D. M. (2011). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. **Journal of Machine Learning Technologies**, 2(1):37-63.

Shiff, S., Swissa, M., and Zlochiver, S. (2016). A Genetic Algorithm Optimization Method for Mapping Non-Conducting Atrial Regions: A Theoretical Feasibility Study. **Cardiovascular Engineering and Technology**, 7(1):87-101.

UCI Dataset. (2015). Forest Type Mapping Data Set [On-line]. Access on August 2016, Available: https://archive.ics.uci.edu/ml/datasets/Forest+type+mapping

Yin, F., Mao, H., and Hua, L. (2011). A hybrid of back propagation neural network and genetic algorithm for optimization of injection molding process parameters. **Materials & Design**, 32(6):3457-3464.

Zheng, H., Kong, L. X., and Nahavandi, S. (2002). Automatic inspection of metallic surface defects using genetic algorithms. **Journal of Materials Processing Technology**, 125:427-433.

Parameter Optimization for Mammogram Image Classification with Support Vector Machine

Keerachart Suksut, Ratiporn Chanklan, Nuntawut Kaoungku, Kedkard Chaiyakhan,
Nittaya Kerdprasop, Kittisak Kerdprasop

***Abstract*— Breast cancer is the malignant tumor occurred mostly in women. Even though breast cancer can be fatal, the patient's survival rate could be as high as 90% if it is detected at the early stage of development. Mammography, ultrasound, and magnetic resonance imaging are examples of screening test for breast cancer. However, to precisely and correctly interpret these images, the medical expertise of radiologists is essential. At present with the matured machine learning techniques, computerized methods can be applied to assist tumor diagnosis, such as the classification between benign and malignant types of tumor. We present in this paper the image-preprocessing and the optimized parametric techniques to help improving accuracy of benign-malignant classification from mammogram images. For the image-preprocessing, we used median filter for noise reduction and gamma correction for image brightness adjustment. We also used region growing technique to find the region of interest, then we extracted three groups of potentially discriminative features: texture feature, shape feature, and intensity histogram feature. After the image-preprocessing, we performed parameter optimization using genetic algorithm prior to the classification done by support vector machine. The results showed that with the appropriate feature selection and the optimal parameter adjustment, the support vector machine can improve its accuracy from 89.47% into 92.98% for mammogram image classification.**

***Index Terms*— Parameter Optimization, Genetic Algorithm, Mammogram Images Classification, Support Vector Machine.**

Manuscript received September 26, 2016; revised January 16, 2017. This work was supported in part by grant from Suranaree University of Technology through the funding of Data Engineering Research Unit.

K. Suksut is a doctoral student with the School of Computer Engineering, Institute of Engineering, Suranaree University of Technology, NakhonRatchasima, Thailand (corresponding author: phone: +66879619062; e-mail: mikaiterng@gmail.com).

R. Chanklan is a doctoral student with the School of Computer Engineering, Institute of Engineering, Suranaree University of Technology, NakhonRatchasima, Thailand (e-mail: arc_angle@hotmail.com).

N. Kaoungku is with the School of Computer Engineering, Suranaree University of Technology, Muang, Nakhon Ratchasima, Thailand (e-mail: nittaya.k@gmail.com).

K. Chaiyakhan is with the Computer Engineering Department, Rajamangala University of Technology Isan, Muang, Nakhon Ratchasima, Thailand (e-mail: kedkarnc@hotmail.com).

N. Kerdprasop is with the School of Computer Engineering, Suranaree University of Technology, Muang, Nakhon Ratchasima, Thailand (e-mail: nittaya.k@gmail.com).

K. Kerdprasop is with the School of Computer Engineering, Suranaree Universiyof Technology, Muang, Nakhon Ratchasima, Thailand (e-mail: kittisakThailand@gmail.com).

I. Introduction

Among diagnosed cancers in women, breast cancer is the most prominent type and it can be deadly. Usually, early tumor diagnosis can improve survival rate and help the preparation for appropriate treatment. Breast cancer detection can be done through the ultrasound screening [1], magnetic resonance imaging [2], and mammography [3]. The background knowledge for screening cancerous cases is that for the benign (or non-harmful) cases, tumor shapes are regularly round and smooth. On the contrary, for the malignant (or harmful) breast cancer cases, tumors tend to demonstrate irregular and undulated shapes [4].

During the last years, many researchers used mammogram images for breast cancer diagnosis. However, the mammogram images always have noise. The effect of noise is that it can blur some important parts in the images (some points or pixels in images that are normal tissue might look like tumor).

Currently, there are many techniques for de-noise (remove noise) such as image enhancement [5], image segmentation [6], and image feature extraction [7]. It can improve the accuracy for classifying between benign and malignant tumors.

At present, there are many efficient automatic techniques for classification such as decision tree learning, artificial neural network, support vector machine, and many more. Among the existing techniques, support vector machine is generally the most accurate one. If we apply techniques for de-noising and then adopt support vector machine algorithm with the optimized parameters for classification, it can intuitively improve performance of mammogram image classification.

In this paper, we thus propose parameter optimization for support vector machine to classify mammogram image. The goal of this research is to improve the breast cancer classification performance. We apply genetic algorithm for parameter optimization (parameters C, epsilon, and gamma to be used in the support vector machine). We pre-process the images by de-noising with the median filter technique, adjusting image intensity with the gamma correction technique, then finding the region of interest to choose only the potential area for cancerous cell detection with region growing technique, and finally performing feature extraction to contain texture feature, shape feature, and intensity histogram.

II. Materials and Methods

A. Median Filter

The intuitive idea of median filter is that some pixels in the image may contain noise and this noise can be detected through its extreme value that does not get along with the surrounding pixels. The median filter method [8] to handle noisy pixel is thus to create a small window frame for normalizing a specific pixel value within that frame (in this work, we set the size of a window to be 3x3 pixels). During the filtration process, a small window is moved along the pixel grid within the image. At each position of a window frame, all the pixel values (i.e., nine values for our 3x3 frame) within the frame are sorted. The median pixel value is then used to replace the existing pixel value. Example of a median filter process is illustrated in fig 1.

B. Gamma Correction

Gamma correction [9] can enhance the contrast of the image. It has value between 0 to 1, where 0 means darkness (black color) and 1 means the brightness (white color). Given the parameter γ as the encoding or decoding value, we can compute the value of gamma correction with the formula given in equation (1).

$$Corrected = 255 * \left(\frac{Image}{255}\right)^{\left(\frac{1}{\gamma}\right)} \quad (1)$$

Note that if γ > 1, it is called a decoding gamma in which the shadow in that image will be set darker. For γ < 1, it is called an encoding gamma and will be used to make the dark region to be lighter.

C. Region Growing

Region growing [10] is applied to choose only specific are of interest by merging surrounding areas with similar intensity. The process starts by setting the seed point, which is normally the middle point (or middle pixel) in the image and then compare the intensity value of that point with the intensity values of the neighbor pixels. If the values are in the same class, we then increase the size of the region.

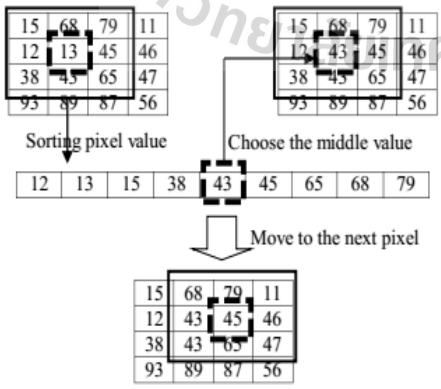


Fig 1. Demonstration of the median filter process

When the growth of one region stops, we then select another seed pixel outside the area previously processed.

D. Texture Feature

Texture feature [11] can help to identify the object in the image. Texture in the image can describe the physical properties (such as shape, curve) and can help to split different objects in an image. We can find texture feature with Grey Level Co-occurrence Matrix (GLCM).

E. Intensity Histogram Feature

Intensity histogram feature is used for describing the properties of the image. In this work, we consider four statistical features that can be obtained from the histogram. These statistics are mean, variance, skewness and kurtosis.

Mean is an average intensity level. Variance is the variation of intensities around the mean. Skewness is the indicator whether the histogram is symmetric, and kurtosis is a measure of whether the data are peak.

Given that G be the image gray scale level and P be the probability level of gray scale, the mean (μ), variance (σ^2), skewness (s), and kurtosis (k) can be computed with formulas given in equations (2) to (5), respectively.

$$\mu = \sum_{i=1}^{G-1} iP(i) \quad (2)$$

$$\sigma^2 = \sum_{i=1}^{G-1} (i-\mu)^2 P(i) \quad (3)$$

$$s = \sigma^{-3} \sum_{i=1}^{G-1} (i-\mu)^3 P(i) \quad (4)$$

$$k = \sigma^4 \sum_{i=1}^{G-1} (i-\mu)^4 P(i) \quad (5)$$

F. Shape Feature

Shape feature [12] can help to identify the object in the image by using shape of object within the image. Shape can differentiate between benign and malignant cases because benign tumors have smooth shapes and regularly round but malignant breast tumors tend to demonstrate irregular and undulated shapes. So, we can classify the object in image by compute the distance between center point in tumor and its edge. For a number of computed distances, if the values do not change or there is only a few change, we can predict that that image is a benign tumor. But if the distance values show much fluctuation, we can predict that the image is malignant tumor.

G. Genetic Algorithm

Genetic algorithm [13-14] is an algorithm to find the solution with adaptive heuristic search based on the evolutionary characteristic of nature. Genetic algorithm combines the concept of random search space and compares the randomly selected solutions based on some fitness function, and then selects the better solution. The simple genetic algorithm is shown in fig 2.

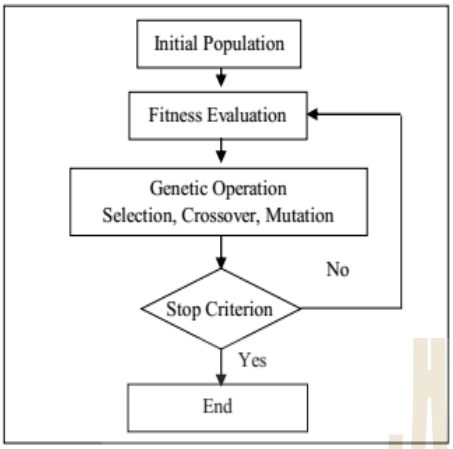


Fig 2. Flowchart for simple genetic algorithm.

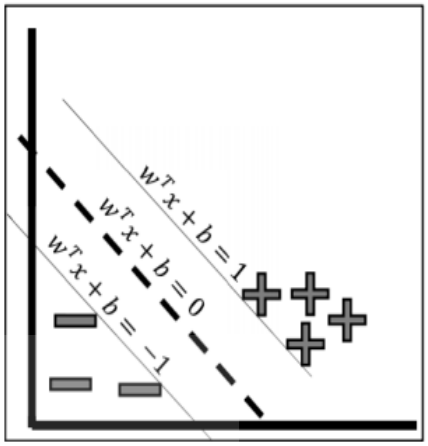


Fig 3. Optimal hyperplane

From fig 2, we can describe genetic algorithm with 5 main steps. Step 1 is setting the initial population; it is normally a random selection. Step 2 is defining the fitness function; it is used for evaluating the fitness of each population or chromosome. Step 3 is applying the genetic operation; the operation can be either selecting the chromosome or population with random selection, crossing over two parent chromosomes to create better offspring, or mutating a chromosome with randomly selected point. Step 4 is replacing individual in the population; it is the replacement of the old chromosome (parent chromosome or parent population) with the new generation. Step 5 is checking for stop criterion; it is a check point for whether to end the process such as stop the process when it has created the new generation over 3 generations.

H. Support Vector Machine

Support Vector Machine (SVM) [15] is a machine learning algorithm for classifying different classes of objects. SVM has been widely applied to many fields. SVM is a supervised learning machine in that it requires a class attribute for guiding the learning process to build a model that can classify objects with mixing classes correctly. The main concept of SVM is the generation of the optimal hyperplane that can separate the objects such that objects with the same class form themselves as a group, whereas objects in different classes should be in a different group. The hyperplane is called an optimal one if such plane can separate classes with the most distance between each class. Fig 3 shows an optimal hyperplane with a dashed line and the two classes in the figure are positive (represented as 1) and negative (-1). To use the hyperplane as a model to classify objects, the formula given in equation (6) can be applied.

$$w^T x + b \geq 1, when\ y_i = +1 \quad (6)$$
$$w^T x + b \leq 1, when\ y_i = -1$$

where
x is data vector,
w is weight vector,
b is bias, and
y is a class.

To apply support vector machine for the classification task, users have to set three important parameters (C, epsilon, and gamma). Parameter C is to control the cost for miss-classification. This parameter is used to control the influence of each individual support vector (i.e., the data points on the borderlines which are up and below the optimal hyperplane in fig 3). Setting the C parameter involves trading error penalty for stability. Parameter epsilon is used to fit the training data. It controls the width of the epsilon-insensitive zone. The value of epsilon can affect the number of support vectors that are used to find the optimal hyperplane. Parameter gamma is the kernel parameter of the Gaussian radial basis function.

The small gamma implies that the learned model will have the large margin; the hyperplane has large distance between two class borderlines and more flexibility in data classification. The large gamma means that learned model will have small margin; the hyperplane has small distance between two class borderlines and thus no flexible in new data classification (may cause overfitting).

III. Proposed Work

In the proposed work, we have designed the process of parameter optimization with genetic algorithm for mammogram image classification with the support vector machine as shown in fig 4.

From fig 4. We can describe our proposed framework as follows. For pre-processing images, we used median filter method for de-nosing, the output from this process is clearer image without noise. After that, we use gamma correction to enhance contrast of the image, the output from this step is sharp image such that the tumor area has lighter intensity and density than the original image. For segmentation process, we use region of interest technique for choosing only region of interest. The output of this process is the smaller image than the original one. A small size means the reduction in dimension to contain only discriminative regions. For feature extraction process, we extract feature

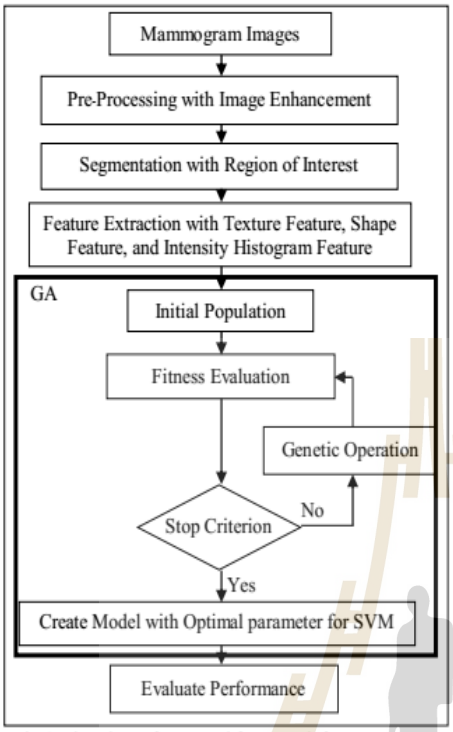


Fig 4. Flowchart of proposed framework for mammogram image classification.

with three techniques (texture feature, shape feature, and intensity histogram feature). The output of this process is the data that extract properties of images (shape, texture, mean, variance, etc.). Then we split the data from previous process into 2 parts. The first part (70% of all data) has been used to find parameter C, epsilon, and gamma with genetic algorithm. This first part of data is also used to create a classification model with support vector machine. The second part (30% of all data) has been used for performance evaluation of the learned model.

In genetic algorithm process, we define parameter control for genetic algorithm as follows:

Population size = 100
Iteration (number of generation) = 100
Probability of crossover = 0.8
Probability of mutation = 0.01
C in the range: $10^{-4} \leq C \leq 10$
Epsilon in the range: $10^{-2} \leq \text{epsilon} \leq 2$
Gamma in the range: $10^{-3} \leq \text{gamma} \leq 3$

Fitness function = $\text{Accuracy} = \frac{TP+TN}{N}$

where

TP is number of true predicted benign cases,
TN is number of true predicted malignant cases, and
N is number of all data that are used to test model.

The output of genetic algorithm is the three parameters that are optimal ones for SVM. After that, we use the optimal parameter to create model with SVM. Finally, we evaluate performance model to assess its accuracy by using the test data. We finally compare the SVM performance with different set of input features.

IV. Experimental Results

For experimentation, we use data set from the Digital Database for Screening Mammography (DDSM) with 190 images (benign 80 images, malignant 110 images) and split data into two parts with 133 images (70% of all data) used for creating a model and finding optimized parameters; we call this data set as "training set". We use 57 images (30% of all data) for evaluating the performance of classification model; we call this data set as "testing set". This work has been implemented with MATLAB and RStudio. We run our experiments on a core i3/3.50 GHZ computer with 12 GB of RAM. The details of data after extracting features by using texture feature, shape feature, and intensity histogram are shown in Table 1.

In the classification process, we also compare between different sets of input features that used as input to the support vector machine. We test different combinations of texture feature, shape feature, intensity histogram feature, and the optimized parameter with genetic algorithm for support vector machine. The accuracies of SVM after applying different combinations of input features are shown in Table 2.

From table 2, it can be seen that the adjusted optimal parameters for support vector machine combined with techniques to extract only important features including texture feature, shape feature, and intensity histogram altogether can improve the performance for mammogram

Table 1. Detail of data set

Feature Extraction Techniques	# Training set	# Testing set	# Features
Texture + Shape + Intensity Histogram	133	57	21
Shape + Intensity Histogram	133	57	6
Texture + Shape	133	57	17
Texture + Intensity Histogram	133	57	20

Table 2. Classification results

Feature Extraction Techniques	Accuracy
Texture + Intensity Histogram	81.58%
Texture + Shape	85.26%
Shape + Intensity Histogram	87.37%
Texture + Shape + Intensity Histogram	89.47%
Texture + Shape + Intensity Histogram + Optimized Parameter for SVM with Genetic Algorithm	**92.98%**

image classification from the 81.58% accuracy level at 81.58% up to the 92.98%. The classification by SVM using only the extracted features (i.e., the texture feature, shape feature, and intensity histogram) can obtain the highest accuracy at 89.47%. The experimental results show that with an extra steps of optimal parameter adjustment through genetic algorithm, the support vector machine shows an improve performance (from 89.47% to 92.98%) for classification mammogram images.

V. Conclusion

Breast cancer is the major type of dangerous tumors mostly occurred in women and causes numerous deaths in the developing countries. Early detection of malignant breast cancer cases is, more or less, expected to help the appropriate preparation for successful treatment. Breast cancer can be screened with ultrasound imaging, magnetic resonance, or mammogram imaging.

In this work, we propose a framework for automatic classification of malignant breast cancer, the harmful one, from the benign cases, the non-harmful. According to our framework of breast cancer classification with mammogram image, the first step is the noise removal from the mammogram image and the image intensity enhancement. Median filter and gamma correction are the two techniques to de-noise and to enhance contrast of the image, respectively. Region growing technique is then applied to select only area or region of interest. In our work, it is the image regions that are anticipated to contain tumor cells.

We then apply image feature extraction to obtain only important features suitable for the subsequent classification model learning step. The prominent features are texture feature, shape feature, and intensity histogram containing statistical information including mean, variance, skewness, and kurtosis. Another important step in our framework is the application of the genetic algorithm to find the optimal parameters (cost, epsilon, and gamma) for training the support vector machine. The experimental results show that the parameter optimization through genetic algorithm technique can obviously improve the SVM performance for mammogram image classification; it is better than using the default parameters.

References

[1] X. Shi, H.D. Cheng, L. Hu, W. Ju, and J. Tian, "Detection and classification of masses in breast ultrasound images," Digital Signal Processing, vol. 20, no. 1, pp.824-836, 2010.

[2] M.J. Collins, J. Hoffmeister, and S.W. Worrell, "Computer-aided detection and diagnosis of breast cancer," Seminars in Ultrasound, CT and MRI, vol. 27, no. 4, pp.351-355, 2006.

[3] A. Oliver, X. Llado, E. Perez, J. Pont, E. Denton, J. Freixenet, and J. Marti, "A statistical approach for breast density segmentation," Journal of Digital Imaging, vol. 23, no. 5, pp.527-537, 2010.

[4] H. Lee, and Y. Chen, "Image based computer aided diagnosis system for cancer detection," Expert Systems with Applications, vol. 42, no. 1, pp.5356-5365, 2015.

[5] R. Berannek, W. Jakubowski, A. Mazurczak, M. Postolski, and W. Wiazel, "Contrast enhanced evaluation of the solid lesions in the breast-own experience," European Journal of Ultrasound, vol. 7, no. 1, pp. S13, 1998.

[6] R. Szeliski, "Computer Vision Algorithms and Applications," Springer, 2010.

[7] K. Chaiyakhan, N. Kerdprasop, K. Kerdprasop, "Feature selection techniques for breast cancer image classification with support vector machine," Proceedings of the 24th International Multi Conference of Engineers and Computer Scientists (IMECS2016), Hong Kong, pp.237-232, March 2016.

[8] T. Chen, K. K. Ma, and L. H. Chen. "Tri-state median filter for image denoising. Image Processing," IEEE Transactions on, vol. 8, no. 12, pp.1834-1838. 1999.

[9] H. Farid, "Blind inverse gamma correction," Image Processing, IEEE Transactions on, vol. 10, no. 10, pp.1428-1433, 2001.

[10] R. Rouhi, M. Jafari, S. Kasaei, and P. Keshavarzian, "Benign and malignant breast tumors classification based on region growing and CNN segmentation," Expert Systems with Applications, vol. 42, no. 1, pp.990-1002, 2015.

[11] A. V. Alvarenga, W. C. A. Pereira, A. F. C. Infantosi, and C. M. Azevedo, "Complexity curve and grey level co-occurrence matrix in the texture evaluation of breast tumor on ultrasound images," Medical Physics, vol. 34, no. 2, pp.379-387, 2007.

[12] W. C. Pereira, A.V. Alvarenga, A. F. Infantosi, L. Macrini, and C. E. Pedreira, "A non-linear morphometric feature selection approach for breast tumor contour from ultrasonic images," Computer in Biology and Medicine, vol. 40, 2010.

[13] H. Holland, "Adaptation in Natural and Artificial Systems," Ann Arbor: the University of Michigan Press, Michigan, 1975.

[14] R. A. C. Yang, Z. Zhou, L. Wang, and Y. Pan, "Comparison of Different Optimization Methods with Support Vector Machine for Blast Furnace Multi-Fault Classification," IFAC-Papers Online, vol. 48, no. 21, pp.1204-1209, 2015.

[15] C. Cortes, and V. Vapnik, "Support-vector networks," Machine learning, vol. 20, no. 3, pp. 273-297, 1995.

Support Vector Machine with Restarting Genetic Algorithm for Classifying Imbalanced Data

Keerachart Suksut, Kittisak Kerdprasop and Nittaya Kerdprasop

***Abstract*—Algorithms for data classification are normally at their high performance when the dataset has good balance in which the number of data instances in each class is approximately equal. But when the dataset is imbalanced, the classification model tends to bias toward the majority class. The goal of imbalanced data classification is how to improve the performance of a model to better recognize data from minority class, especially when minority is more interesting than the majority data. In this research, we propose technique for balancing data with hybrid resampling techniques and then perform parameter optimization with restarting genetic algorithm. The optimized parameters are for support vector machine to induce efficient model for recognizing data in minority class, whereas maintaining overall accuracy. The experimental results show that the proposed technique has high performance than others.**

***Index Terms*— Imbalanced Data, Restarting Genetic Algorithm, Support Vector Machine**

I. Introduction

Currently, data mining has been applying to many fields. The concept of data mining is to find the knowledge from the stored information and database. Knowledge can be a pattern or relationship that is hidden in the data. The knowledge extraction can be done with mathematical method, statistics or other computational methods [1]. There are many types of data mining such as data classification, association rule mining, clustering, forecasting, and other analysis tasks.

Techniques in the data classification include artificial neural network (ANN), decision tree, naïve Bayes, support vector machine (SVM), and many more. The concept of ANN is simulating computer to resemble the human brain, which can learn as a human learns. The idea of decision tree induction for data classification is to partition data into subsets using tree as a data structure to store data subsets. The nodes in a tree represent data attributes used for partitioning data into subsets and the leaf nodes are classes of data. The concept of naïve Bayes is to use the probability to classify the data. Main concept of SVM is creating the hyperplane for separating data with high distance between groups of data. SVM has recently gained popularity due to its overall high performance on classifying both balanced and imbalanced data [2], [3]. However, recognition rate over minority class is still low.

To improve the algorithm on classifying minority, some techniques to properly adjust learning parameters have been proposed. For instance, Yin et al. [4], Jamshidi et al. [5] and Shiff et al. [6] applied genetic algorithm to learn optimal parameter values. But the problem of genetic algorithm is that sometime the algorithm cannot find the best parameter due to improper setting of a random initial value. In addition, most classification algorithms work effectively when the data is balanced. In this research, we thus propose techniques for balancing data and then optimizing parameters with restarting genetic algorithm for the subsequent application of SVM learning algorithm.

Manuscript received February 15, 2017; revised March 20, 2017. This work was supported by grant from Suranaree University of Technology through the funding of Knowledge and Data Engineering Research Units.

K. Suksut is a doctoral student with the School of Computer Engineering, Suranaree University of Technology (SUT), 111 University Avenue, Muang, Nakhon Ratchasima 30000, Thailand. (corresponding author: +66879619062; e-mail: mikaiterng@gmail.com).

K. Kerdprasop is an associate professor with the School of Computer Engineering and head of Knowledge Engineering Research Unit, SUT. (e-mail: kerdpras@sut.ac.th).

N. Kerdprasop is an associate professor with the School of Computer Engineering and head of Data Engineering Research Unit, SUT. (e-mail: nittaya@sut.ac.th)

II. Background Theories

A. Data Sampling

Data sampling is a pre-processing step of classification to balance amount of data in each class. The two major sampling approaches to balance data are under sampling and over sampling. Under sampling is a technique of down sampling that reduce the amount of data in the majority class to be in the same proportion as the number of data in the minority class [7]. The basic idea is shown in fig. 1.

Over sampling, on the contrary, is the up-sampling technique in the sense that data in the minority class is increased to be in the same amount of data in other classes. Sampling data from minority class can be either the repeated selection of data from the minority class, or the generation of data points based on some criteria.

SMOTE technique [8] applies the later scheme by creating a synthetic data by measuring the distance from the sample data to the nearest data point and then randomly create new data. The new data are created within the distance computed as in equation (1):

$$N_p = O_p + (Rand[0,1] * dist(x, y, \dots, z)) \qquad (1)$$

where Np is the new data of minority class,
Op is the old data point in minority class used as the reference point for computing neighbor distance,
Rand[0,1] is random number between 0 to 1,
dist(x,y,…,z) is the distance between default data and neighbors.

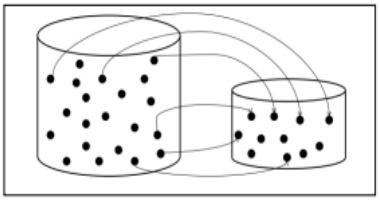


Fig. 1. Under sampling data

B. Genetic Algorithm

Genetic algorithm is the search for optimal answer by using imitation of natural evolution such that the one who is stronger has more chance to survive than those who are weaker and the stronger one can inherit strength to their children. John Holland [9] introduced this concept of genetic algorithm in 1975. After that, it has been successfully applied to many applications. The draft computation steps of genetic algorithm are shown in fig. 2.

Firstly, the initial population has to be randomly created. Random number of the population equals to the number of the population size. After that, the fitness value of each population is computed for selecting the best population to be used as the chromosomes to inherit as genetic material. Then, genetic operation process such as crossover and mutation will be applied to mutate chromosome for hopefully being stronger. The new generation of population that is stronger than the old one will replace the old population. The process iterates until it converges to the stopping criterion.

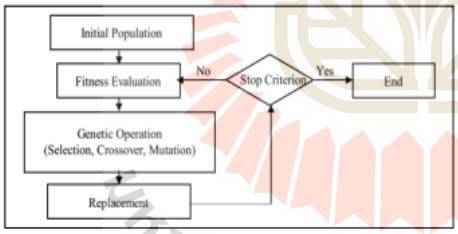


Fig. 2. Simple genetic algorithm

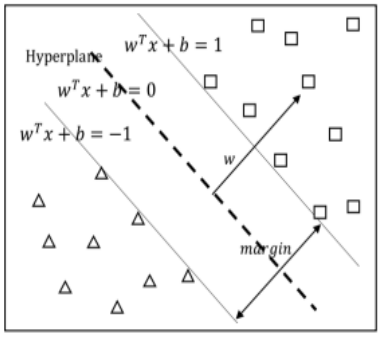


Fig. 3. Optimal hyperplane for support vector machine

C. Support Vector Machine

Support vector machine, or SVM, [10] is an algorithm for classifying data by creating a hyperplane to separate data with different classes. Optimal hyperplane for SVM is the line or plane that has maximum margin between the plane and the nearest data points on each side of the plane. This concept is shown in fig. 3.

The hyperplane will split the data having different classes apart from each others with the maximum distance between data from each class. The weight vector is used for determining the direction and inclination of the hyperplane. Weight vector is perpendicular to the hyperplane and the data with classes 1 and -1 can be separated according to the equation (2):

$$\boldsymbol{w^T x + b \geq 1, when\ y_i = +1}$$
$$\boldsymbol{w^T x + b \leq 1, when y_i = -1} \quad (2)$$

where w is weight vector, and b is bias.

Weight vector is the line perpendicular to the hyperplane and bias will determine the distance between the hyperplane and origin. Consider two dimensional data $X = (x_1, x_2)^T$, the equation of linear hyperplane is:

$$\boldsymbol{h(x) = w^T x + b = w_1 x_1 + w_2 x_2 + b = 0} \quad (3)$$

Given two data points on hyperplane $A = (A_1, A_2)$ and $B = (B_1, B_2)$, the equation for compute the weight vector is:

$$\boldsymbol{weight\ vector = -\frac{w_1}{w_2} = -\frac{(B_2 - A_2)}{(B_1 - A_1)}} \quad (4)$$

The margin can be computed with equation (5) and the size of weight vector is computed as in (6):

$$\boldsymbol{margin = \frac{2}{||w||}} \quad (5)$$

$$\boldsymbol{||w|| = \sqrt{w_1^2 + w_2^2}} \quad (6)$$

D. Adaboost (Adaptive Boosting)

Adaboost algorithm is the application of boosting technique [11] to increase classification performance. The main concept (shown in fig. 4) is a combination of weak learners with adjusted higher weight for data that are wrongly classified. Then create new learner from miss-classified data until receiving strong learner with high predictive performance. There is an extension of Adaboost called RUSBoost in which under sampling technique has been applied before classifying data with Adaboost.

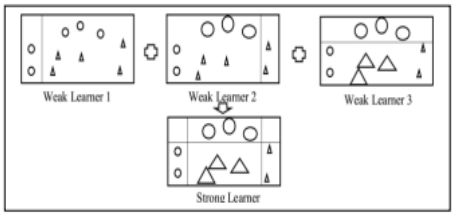


Fig. 4. Adaboost algorithm

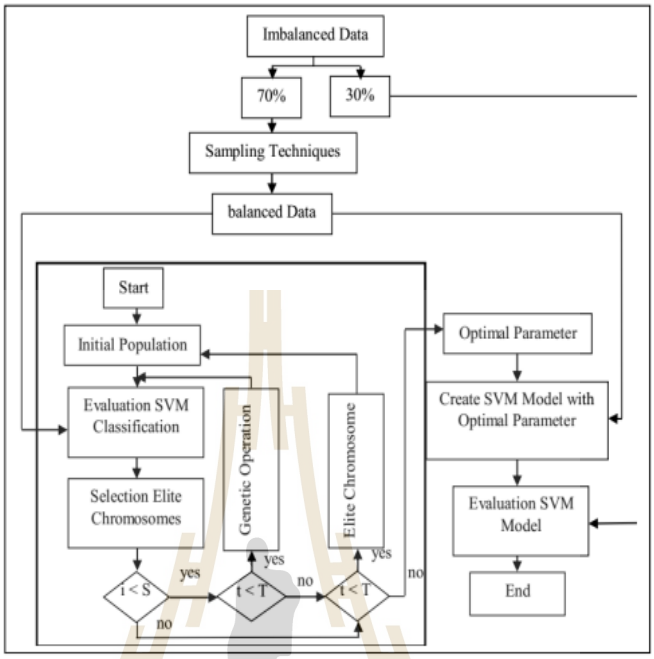


Fig. 5. Research framework

E. *Classification Performance Evaluation*

To evaluate performance of classification model on recognizing majority and minority classes of imbalanced data, we use four measurements: Accuracy, Precision, Recall and F-measure. The computation of these metrics is based on the values in confusion matrix as shown in table 1.

TABLE I: CONFUSION MATRIX FOR TWO CLASS CLASSIFICATION

		Predicted Data	
		Positive	Negative
Actual Data	Positive	TP	FN
	Negative	FP	TN

Rows in the matrix are number of actual data for each class and columns are number of predicted data for each class. The acronyms TP, FP, FN, TN are possible outcomes of prediction made by the classification model. Suppose the data are either of class positive or negative, the outcome of prediction can be one of the following 4 cases:

Case 1: TP is the number of actual data from positive class and the model can correctly predict that data to be in a positive class.

Case 2: FN is the number of actual data from positive class but the model predict that the data incorrectly as in a negative class.

Case 3: FP is the number of actual data from negative class but the model incorrectly predict that data to be in a positive class.

Case 4: TN is the number of actual data from negative class and the model can correctly predict that data to be in a negative class.

Accuracy is a measure for overall performance of the classification model, and the computation is as shown in equation (7):

$$Accuracy = \frac{(TP + TN)}{(TP+FN+FP+TN)} \quad (7)$$

Precision is the proportion of predicted positive class to the real positive class, computed as in equation (8):

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

Recall or Sensitivity is the ration of data that are predicted as positive to the number of all positive data, computed as in equation (9):

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

F-measure is a measure that taking into account both precision and recall. The computation of F-measure is as shown in equation (10):

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

III. MATERIALS AND METHODS

The design and implementation of our work to deal with imbalanced data classification are as shown in fig. 5. Firstly, we split data into 2 subsets, 70% of them is training set and the remaining 30% is testing set. We preprocess training set

with random under sampling to reduce number of data in the majority class and synthetically generate data in the minority class with SMOTE technique. We then find the optimal parameter for the subsequent classification process by introducing restarting genetic algorithm. For the Chromosome encoding, we use real-value encoding and random initial population until obtaining the specified population size. The fitness value of each chromosome is evaluated based on the accuracy from classifying data with support vector machine by using training set and parameter from each chromosome. After that, we select elite chromosomes, which are the top k chromosomes with highest fitness values, and applying the genetic operation to obtain new population.

If the new generation is less powerful than the old population, repeat the process by replacing initial population with elite chromosome and proceed until the stopping criterion has been met. After completion, create model with optimal parameters for support vector machine and evaluate model with testing set. Then, compute performance with accuracy, precision, recall and F-measure metrics.

Restarting genetic algorithm in this research is the addition of condition to re-create the initial population when the new generation has fitness value less than the old population and the stopping criterion has not been met. The steps in restarting genetic algorithm are shown in fig. 6.

Restarting Genetic Algorithm
Input : c, epsilon, gamma,
number of generation T,
number of worst generations S
Output : optimal of c, epsilon, gamma
Method:
1. chromosome encoding
2. initial population p
3. evaluate fitness value of each chromosome
4. for t ≤ T
4.1) for new generation having fitness value less than the old population, i < S
4.1.1) genetic operation (selection, crossover, mutation)
4.1.2) replacement
4.1.3) fitness evaluation (select elite chromosome)
4.2) for new generation having fitness value less than the old population, i ≥ S
4.2.1) re-create initial population with elite chromosome
4.2.2) fitness evaluation
5. select the best chromosome

Fig. 6. Restarting genetic algorithm

IV. Experimental Results

A. Dataset

In this research, we use 2 datasets. One is a real dataset, another is synthetic dataset. Details of data are as follows.

Synthetic dataset contains 700 data records with 16 attributes. The majority class comprises of 600 records, whereas the minority class has 100 records.

The real dataset is Asthma data [12] containing 677 data records with 16 attributes. The majority class contains 570 records, but the minority class has only 128 records.

B. Parameter Setup

The setting of parameters c, epsilon, gamma, number of iteration, population size, probability of crossover, probability of mutation, and number of worst generations for restarting genetic algorithm are summarized in table II.

TABLE II: PARAMETER DETAIL FOR RESTARTING GENETIC ALGORITHM

Cost	$10^{-4} - 10^{-2}$	Prob. of crossover	0.8
Gamma	$10^{-3} - 10$	Prob. of mutation	0.01
Epsilon	$10^{-2} - 10$	Iteration	100
Population size	100	Restart GA	2

C. Results

For evaluate performance of classification model, we use the accuracy, precision, recall, and F-measure metrics. We compare the classification performance of our proposed method against the powerful algorithms that have been widely used to learn model from imbalanced data. These standard algorithms are support vector machine (with default parameters), Adaboost, and RUSBoost. The comparative results on synthetic dataset are shown in table III.

TABLE III: COMPARATIVE PERFORMANCE OF SYNTHETIC DATASET

	SVM	Adaboost	RUSBoost	Propose
Accuracy	**88.00**	87.50	78.50	85.00
Precision	**100.00**	88.89	32.56	47.92
Recall	14.29	25.00	50.00	**82.14**
F-measure	25.01	39.03	39.44	**60.53**

From table III, when considering overall accuracy for classifying imbalanced data, we found that SVM using default parameters has highest accuracy at 88.00%, whereas Adaboost is the second best accurate model at 87.50% prediction correctness. Our proposed method is the third at 85.00% correctness, and RUSTBoost is the worst with 78.50% correctness.

When considering precision value, SVM show the best performance at 100%. Adaboost comes second at 88.89% of precision on predicting minority class. Our proposed technique is the third one (47.92%) and RUSBoost are the worst (32.56%).

For the recall measurement on minority class recognition, we found that our proposed technique performs the best at recall rate 82.14%. The second best recall model is RUSBoost (50.00%), whereas Adaboost is the third one (25.00%) and SVM is the worst (14.29%) in terms of minority class recognition. To consider both precision and recall with the F-measure metric, our proposed method is the best (60.53%). RUSBoost is the second best one (39.44%), followed by Adaboost (39.03%) and SVM (25.01%).

The results of asthma dataset are shown in table IV.

TABLE IV: COMPARATIVE PERFORMANCE OF ASTHMA DATASET

	SVM	Adaboost	RUSBoost	Propose
Accuracy	**79.52**	78.10	66.67	70.00
Precision	**38.89**	38.71	35.78	37.76
Recall	17.95	30.77	**100.00**	94.87
F-measure	24.56	34.29	52.70	**54.02**

For the real asthma dataset, SVM is also the best model in terms of overall accuracy (79.25%) and precision (38.89%) on predicting class. The second best one is Adaboost model (accuracy = 78.10% and precision = 38.71%). Our proposed model is the third one (accuracy = 70%, precision = 37.76%). The worst model is RUSBoost (accuracy = 66.67%, precision = 35.78%).

But when considering only recall rate, RUSBoost is the best model on recognition the minority class of asthma dataset. Out proposed model performs the second best at 94.87% of recognition rate. The Adaboost and SVM models are very poor on recalling data in the minority class with the recognition rate at 30.77% and 17.95%, respectively.

To evaluate with the F-measure, our proposed model is the best one (54.02%), and the RUSBoost model is the second (52.70). Both Adaboost and SVM show poor performance at 34.29% and 24.56%, respectively.

It's can be seen that when considering only accuracy and precision, SVM shows higher performance than other techniques. But when considering about recall performance and F-measure, which is the compromising of both recall and precision metrics, our proposed technique performs better than others.

V. Conclusion

The major problem on building a model to classify data that distribution among classes is uneven is that the model built from traditional method tends to bias toward majority class in such a way that the model is most likely to guess the class of all new data as the majority one. This tendency of a model is not harmful when the main model measurement of interest is overall predictive accuracy. But when data in minority class is the class of concern, traditional method is not powerful enough to catch the minority cases.

To improve the algorithm on classify imbalanced data to better recognizing the minority data that are normally overshadowed by the majority class, we propose a novel method that firstly balancing data by using random under sampling data in majority class, as well as creating synthetic the data to increase the amount in the minority class with SMOTE technique. We then propose to use restarting genetic algorithm to find the optimal parameters for support vector machine. The experimental results show that support vector machine (with default parameters) performs better than other techniques in terms of accuracy and precision, but shows poor performance when evaluated with recall and F-measure. When high recall of data in minority class and F-measure are the main measurements of interest, our proposed method has been experimentally proven better than the traditional support vector machine.

References

[1] J. Han, and M. Kamber, Data mining: Concepts and Techniques, Morgan Kaufmann, 2006.

[2] J. J. Liao, C. H. Shih, T. F. Chen, and M. F. Hsu, "An ensemble-based model for two class imbalanced financial problem," Economic Modelling, vol. 37, pp. 175-183, 2014.

[3] S. Cateni, V. Colla, and M. Vannucci, "A method for resampling imbalanced datasets in binary classification tasks for real-world problems," Neurocomputing, vol. 135, pp. 32-41, 2014.

[4] F. Yin, H. Mao, and L. Hua, "A hybrid of back propagation neural network and genetic algorithm for optimization of injection molding process parameters," Materials & Design, vol. 32, no. 6, pp. 3457-3464, 2011.

[5] M. Jamshidi, M. Ghaedi, K. Dashtian, S. Hajati, and A. Bazrafshan, "Ultrasound-assisted removal of Al 3+ ions and Alizarin Red S by activated carbon engrafted with Ag nanoparticles: central composite design and genetic algorithm optimization," RSC Advances, vol. 5, no. 73, pp. 59522-59532, 2015.

[6] S. Shiff, M. Swissa, and S. Zlochiver, "A Genetic Algorithm Optimization Method for Mapping Non-Conducting Atrial Regions: A Theoretical Feasibility Study," Cardiovascular Engineering and Technology, vol. 7, no. 1, pp. 87-101, 2016.

[7] K. Chomboon, "Classification technique for minority class on imbalanced dataset with data partitioning method," A thesis submitted in partial fulfillment of the requirements for the degree of doctor of philosophy in computer engineering, Suranaree university of technology, 2016.

[8] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W.P. Kegelmeyer, "SMOTE: synthetic minority over- sampling technique," Journal of Artificial Intelligence Research, vol. 16, pp. 321–357, 2002.

[9] H. Holland, "Adaptation in Natural and Artificial Systems," Ann Arbor: The University of Michigan Press, Michigan, 1975.

[10] C. Cortes, and V. Vapnik, "Support vector network," Machine Learning, vol. 20, no. 3, pp. 273–297, 1995.

[11] Y. Freund, and R. E. Schapire, "Experiments with a new boosting algorithm," Proceedings 13thInternational Conference on Machine Learning, vol. 96, pp. 148-156, 1996.

[12] P. Teerarassamee, "The methodology to find appropriate k for k-nearest neighbor classification with medical datasets," A thesis submitted in partial fulfillment of the requirements for the degree of master of engineering in computer engineering, Suranaree university of technology, 2015.

K. Suksut is currently a doctoral student with the School of Computer Engineering, Suranaree University of Technology, Thailand. He received his bachelor degree in Computer Engineering from Suranaree University of Technology, Thailand, in 2011, master degree in Computer Engineering from Suranaree University of Technology, Thailand, in 2013. His current research of interest includes data mining, genetic algorithm, and imbalanced data classification.

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

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

ประวัติผู้เขียน

นายกีระชาติ สุขสุทธิ์ เกิดเมื่อวันที่ 12 สิงหาคม พ.ศ. 2533 ที่ จังหวัดนครราชสีมา เริ่มเข้าศึกษาระดับชั้นอนุบาล 1 ถึงชั้นประถมศึกษาปีที่ 6 ที่โรงเรียนประชารัฐสามัคคี อำเภอสูงเนิน จังหวัดนครราชสีมา จากนั้นได้เข้าศึกษาต่อในระดับมัธยมศึกษาตอนต้นและตอนปลาย ที่โรงเรียนสูงเนิน อำเภอสูงเนิน จังหวัดนครราชสีมา ปีการศึกษา 2551 ได้เข้าศึกษาต่อระดับปริญญาตรีในสาขาวิชาวิศวกรรมคอมพิวเตอร์ สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี และสำเร็จการศึกษาเมื่อปี พ.ศ. 2554 ภายหลังสำเร็จการศึกษาในระดับปริญญาตรี ได้เข้าทำงานเป็นลูกจ้างชั่วคราวในบริษัทการบินไทย เมื่อปี พ.ศ. 2554 หลังจากนั้นได้เข้าศึกษาในระดับปริญญาโท สาขาวิชาวิศวกรรมคอมพิวเตอร์ สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี ในปี 2556 และสำเร็จการศึกษาในปี 2557 และในปีเดียวกันได้เข้าศึกษาต่อในระดับปริญญาเอกในสาขาวิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

ในระหว่างการศึกษาได้รับความอนุเคราะห์อย่างดียิ่งจากอาจารย์ประจำวิชา Database Systems ได้รับความไว้วางใจให้เป็นผู้ช่วยสอนปฏิบัติการ ได้รับการตีพิมพ์เผยแพร่บทความวิชาการซึ่งรายละเอียดสามารถดูได้ที่ภาคผนวก ข