การประเมินแบบจำลองการปกปิดข้อมูลและการใช้กลุ่มตัวจำแนกประเภท

นายพีรพงศ์ วาณิชยวิศาลสกุล

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2560

An Evaluation of Anonymized Models and Ensemble Classifiers

Mr. Peerapong Vanichayavisalsakul

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science Program in Computer Science
Department of Computer Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2017

หัวข้อวิทยานิพนธ์	การประเมินแบบจำลองการปกปิดข้อมูลและการใช้กลุ่มตัวจำแนกประเภท
โดย	นายพีรพงศ์ วาณิชยวิศาลสกุล
สาขาวิชา	วิทยาศาสตร์คอมพิวเตอร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	ผู้ช่วยศาสตราจารย์ ดร. เกริก ภิรมย์โสภา

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

.....คณบดีคณะวิศวกรรมศาสตร์

(รองศาสตราจารย์ ดร.สุพจน์ เตชวรสินสกุล)

คณะกรรมการสอบวิทยานิพนธ์

.....ประธานกรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.ณัฐวุฒิ หนูไพโรจน์)

.....อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

(ผู้ช่วยศาสตราจารย์ ดร. เกริก ภิรมย์โสภา)

.....กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.สุกรี สินธุภิญโญ)

.....กรรมการภายนอกมหาวิทยาลัย

(ดร.พงศ์ธวัช ชีพพิมลชัย)

พีรพงศ์ วาณิชยวิศาลสกุล : การประเมินแบบจำลองการปกปิดข้อมูลและการใช้กลุ่มตัวจำแนกประเภท (An Evaluation of Anonymized Models and Ensemble Classifiers) อ.ที่ปรึกษาวิทยานิพนธ์หลัก: ผศ. ดร. เกริก ภิรมย์โสภา, 52 หน้า.

งานวิจัยชิ้นนี้ประเมินและวิเคราะห์ประสิทธิภาพของโมเดลความปลอดภัยในการปกปิดข้อมูลและกลุ่มตัวจำแนกประเภทในการจำแนกประเภทชนิดต่างๆ ในปัจจุบันการทำเหมืองข้อมูลถูกใช้งานอย่างต่อเนื่องในจุดประสงค์ต่างๆเพื่อหาองค์ความรู้ การทำเหมืองข้อมูลจำเป็นที่จะต้องคำนึงถึงความเป็นส่วนตัวของข้อมูลที่ใช้สอนโดยที่จะต้องป้องกันไม่ให้สามารถระบุตัวตนข้อมูลในชุดข้อมูลนั้นได้ การปกปิดข้อมูลถูกคิดค้นขึ้นมาโดยมีจุดประสงค์ที่ต้องการจะลดอัตราความเสี่ยงจากการถูกระบุตัวตน อย่างไรก็ตามการปกปิดข้อมูลถูกใช้งานคุณภาพของชุดข้อมูลก็จะลดลง ดังนั้นเราจำเป็นที่จะต้องคำนึงถึงสมดุลระหว่างความปลอดภัยของข้อมูลในการถูกระบุตัวตนและคุณภาพของชุดข้อมูล จุดประสงค์ของงานวิจัยชิ้นนี้คือทำการประเมินผลกระทบของการจำแนกประเภทด้วยข้อมูลที่ถูกปกปิดและประเมินประสิทธิภาพของโมเดลความปลอดภัยและอัลกอริทึ่มกลุ่มตัวจำแนกประเภทต่างๆ มาตรวัดที่จะใช้ในการทดลองคือ ความแม่นยำของการจำแนกประเภท อัตราความเสี่ยงจากการถูกระบุตัวตน และ จำนวนข้อมูลที่ถูกลบ จากผลลัพธ์ที่ได้จากการทดลองสามารถสรุปได้ว่าความแม่นยำของการจำแนกประเภทระหว่างข้อมูลดั้งเดิมและข้อมูลที่ถูกปกปิดไม่ได้แตกต่างกันอย่างมีนัยยะสำคัญ

ภาควิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนิสิต .....

สาขาวิชา วิทยาศาสตร์คอมพิวเตอร์ ลายมือชื่อ อ.ที่ปรึกษาหลัก .....

ปีการศึกษา 2560

5970946421 : MAJOR COMPUTER SCIENCE
KEYWORDS: DATA ANONYMIZATION / DATA MINING / DE-IDENTIFICATION / ENSEMBLE CLASSIFIER / PRIVACY MODEL / PRIVACY-PRESERVING DATA MINING

PEERAPONG VANICHAYAVISALSAKUL: An Evaluation of Anonymized Models and Ensemble Classifiers. ADVISOR: ASST. PROF. DR.KRERK PIROMSOPA, 52 pp.

We evaluate the performance of privacy models and ensemble classification algorithms for data anonymization on classification. Data mining is continuously used in various purposes to extract knowledge. It is necessary for us to concern about privacy to prevent the result from disclosing identity of persons. Data anonymization has emerged with the objective of reducing re-identification risk. However, when data anonymization is applied, the data utility may decrease. Therefore, it is necessary to trade-off between privacy risks and the data utility. Our objectives in this research are to evaluate the effects of data classification with anonymized data and to evaluate the performance of various privacy models and ensemble classification algorithms. The measurement metrics in this experiment are accuracy, re-identification risk and suppressed records. Our experiments show that there is no significant difference between the accuracy of classification using original data and the accuracy of classification using anonymized data. In addition, the average accuracy of each algorithm is not significantly different.

Department: Computer Engineering Student's Signature .....
Field of Study: Computer Science Advisor's Signature .....
Academic Year: 2017

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยความอนุเคราะห์จากผู้ช่วยศาสตราจารย์ ดร.เกริก ภิรมย์โสภา อาจารย์ที่ปรึกษาวิทยานิพนธ์ ซึ่งอาจารย์ได้ให้คำปรึกษา แนะนำแนวทางการทำวิจัย และสนับสนุนให้งานวิจัยสำเร็จลุล่วง ผมขอขอบพระคุณอาจารย์มากๆครับ

อีกทั้งผมขอขอบพระคุณ ผู่ช่วยศาตราจารย์ ดร.ณัฐวุฒิ หนูไพโรจน์ ผู้ช่วยศาสตราจารย์ ดร.สุกรี สินธุภิญโญ และ ดร.พงศ์ธวัช ชีพพิมลชัย กรรมการสอบวิทยานิพนธ์ ที่กรุณาสละเวลาในการให้คำแนะนำและตรวจสอบในวิทยานิพนธ์ฉบับนี้ครับ

สารบัญ

หน้า

หน้า

หน้า

สารบัญตาราง

สารบัญภาพ

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

เทคโนโลยีในปัจจุบันมีความก้าวหน้าและพัฒนาไปอย่างรวดเร็ว ส่งผลให้มีการนำเทคโนโลยีต่างๆเข้ามาช่วยในการดำเนินชีวิตประจำวันของมนุษย์ ซึ่งการใช้งานจะทำให้เกิดข้อมูลจำนวนมากที่ถูกบันทึกลงในฐานข้อมูลของระบบ ทำให้ผู้ให้บริการในระบบต่างๆจะมีฐานข้อมูลขนาดใหญ่ โดยปัจจุบันการทำเหมืองข้อมูลจะสามารถทำให้เจ้าของข้อมูลค้นหาความรู้จากข้อมูลที่มีอยู่ได้ การทำเหมืองข้อมูลเป็นกระบวนการค้นหารูปแบบจากข้อมูลจำนวนมหาศาลโดยอัตโนมัติ ดังนั้นเจ้าของข้อมูลสามารถจะนำข้อมูลมาใช้งานเพื่อเกิดประโยชน์ต่างๆได้ เช่น การสร้างตัวจำแนกประเภทเพื่อทำนายว่าผู้ใช้งานเครือข่ายโทรศัพท์จะใช้งานเครือข่ายต่อหรือไม่ แต่ในกรณีที่เจ้าของข้อมูลไม่มีความสามารถพอที่จะทำเหมืองข้อมูลได้เอง หรือกรณีที่เจ้าของข้อมูลต้องการเผยแพร่ข้อมูลเพื่อเป็นความรู้หรือสนับสนุนในการทำงานวิจัย เจ้าของข้อมูลจำเป็นที่จะต้องเผยแพร่ข้อมูลให้แก่บุคคลอื่น ซึ่งการเผยแพร่ข้อมูลให้บุคคลอื่นในการทำเหมืองข้อมูลนั้นอาจทำให้ข้อมูลรั่วไหลและก่อให้เกิดความเสียหายได้ เนื่องจากข้อมูลส่วนตัวจำเป็นที่จะต้องเก็บเป็นความลับ โดยเฉพาะข้อมูลที่มีความอ่อนไหว เช่น ข้อมูลทางการแพทย์ ข้อมูลทางการเงิน เป็นต้น จึงเกิดคำถามว่าเราจะสามารถมั่นใจได้อย่างไรว่าถ้าเราเผยแพร่ข้อมูลไปแล้วข้อมูลพวกนั้นจะไม่สามารถระบุตัวตนได้

เราสามารถป้องกันข้อมูลให้ลดความเสี่ยงจากการถูกระบุตัวตนได้โดยการนำข้อมูลไปผ่านกระบวนการปกปิดข้อมูลก่อนที่จะเผยแพร่ข้อมูลให้กับบุคคลอื่น ขั้นตอนการปกปิดข้อมูลจะต้องทำการเลือกโมเดลความปลอดภัยในการใช้งาน ซึ่งปัจจุบันได้มีหลากหลายโมเดลความปลอดภัยถูกเสนอสำหรับการปกปิดข้อมูล แต่เราจะต้องตระหนักว่าการนำข้อมูลที่ถูกปกปิดไปทำเหมืองข้อมูลอาจจะทำให้ประสิทธิภาพของการทำเหมืองข้อมูลนั้นลดลง ดังนั้นเราจำเป็นที่จะต้องหาความสมดุลระหว่างความปลอดภัยของข้อมูลกับประสิทธิภาพการทำงานของเหมืองข้อมูล

การจำแนกประเภทเป็นการทำเหมืองข้อมูลชนิดหนึ่ง โดยการทำเหมืองข้อมูลประเภทนี้จะทำการหากฎเพื่อระบุประเภทของวัตถุจากคุณสมบัติของวัตถุ ซึ่งจะทำให้เราสามารถทำนายผลลัพธ์วัตถุใหม่ที่เข้ามาได้ ตัวจำแนกประเภทนั้นมีด้วยกันหลายชนิด การใช้กลุ่มการจำแนกประเภทคือหนึ่งในเทคนิคในการเพิ่มประสิทธิภาพของการทำเหมืองข้อมูล โดยเทคนิคการใช้กลุ่มตัวจำแนกประเภทจะสามารถทำให้ค่าความแปรปรวนและความเอนเอียงของตัวจำแนกประเภทลดลงเมื่อเทียบกับตัวจำแนกประเภทแบบเดี่ยว

จากปัญหาที่กล่าวข้างต้น การจำแนกประเภทโดยเรียนรู้จากข้อมูลที่ถูกปกปิดเพื่อรักษาความปลอดภัยเป็นเรื่องที่สำคัญ แต่จำเป็นที่จะต้องหาความสมดุลระหว่างความปลอดภัยของข้อมูลกับประสิทธิภาพการทำงานของเหมืองข้อมูล ดังนั้นงานวิจัยชิ้นนี้จึงทำการทดลองการทำเหมืองข้อมูลกับข้อมูลที่ถูกปกปิด โดยจะใช้โมเดลความปลอดภัยต่างๆในการปกปิดข้อมูล และอัลกอริทึ่มของกลุ่มตัวจำแนกประเภทต่างๆในการจำแนกประเภท จากนั้นนำผลการทดลองที่ได้มาสรุปผลว่าการทำเหมืองข้อมูลกับข้อมูลที่ถูกปกปิดยังมีประสิทธิภาพดีเพียงพอหรือไม่ โดยจะนำมาเปรียบเทียบกับประสิทธิภาพของการทำเหมืองข้อมูลโดยเรียนรู้จากข้อมูลดั้งเดิม และประเมินประสิทธิภาพของโมเดลความปลอดภัยและอัลกอริทึ่มของกลุ่มตัวจำแนกประเภทในแต่ละชนิด

1.2 วัตถุประสงค์ของงานวิจัย

งานวิจัยชิ้นนี้มีวัตถุประสงค์เพื่อที่จะประเมินประสิทธิภาพของการทำเหมืองข้อมูลกับข้อมูลที่ถูกปกปิด โดยจะทำการทดลองด้วยโมเดลความปลอดภัยต่างๆในการปกปิดข้อมูลและใช้อัลกอริทึ่มกลุ่มตัวจำแนกประเภทที่หลากหลายในการทำเหมืองข้อมูล จากนั้นนำผลลัพธ์ที่ได้มาสรุปผลว่าประสิทธิภาพของการทำเหมืองข้อมูลกับข้อมูลที่ถูกปกปิดนั้นลดลงอย่างมีนัยยะสำคัญหรือไม่และประเมินการทำงานของโมเดลความปลอดภัยและอัลกอริทึ่มกลุ่มตัวจำแนกประเภทในแต่ละชนิด

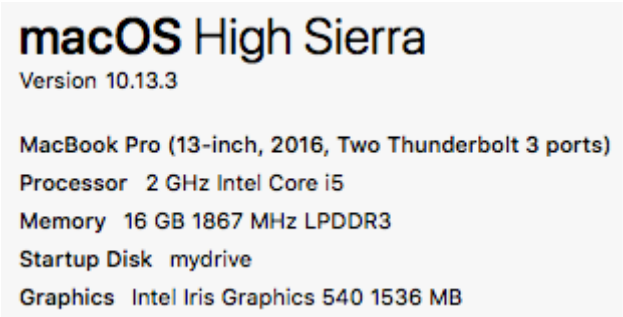
1.3 ขอบเขตงานวิจัย

1.3.1 ใช้ชุดข้อมูล 3 ชุด ในการทดลอง ได้แก่ ADULT, FARS, ATUS

1.3.2 ใช้อัลกอริทึ่มกลุ่มของตัวจำแนกประเภทในการสร้างตัวจำแนกประเภท

1.3.3 ใช้โปรแกรม ARX ในการปกปิดข้อมูล

1.3.4 ใช้เครื่องคอมพิวเตอร์ที่มีข้อมูลจำเพาะดังภาพที่ 1 ในการทดลอง



ภาพที่ 1 ข้อมูลจำเพาะของคอมพิวเตอร์ที่ใช้ในการทดลอง

1.4 ขั้นตอนการวิจัย

1.4.1 ศึกษาความรู้ ทฎษฎีที่เกี่ยวข้อง และงานวิจัยที่เกี่ยวข้อง

1.4.2 ออกแบบวิเคราะห์การทดลอง

1.4.3 ศึกษาเครื่องมือเพื่อใช้ในการทดลอง

1.4.4 ทำการทดลองตามแบบแผน

1.4.5 ตรวจสอบความถูกต้องของผลการทดลอง

1.4.6 ประเมินและสรุปผลการทดลอง

1.4.7 จัดทำวิทยานิพนธ์

1.5 ประโยชน์ที่คาดว่าจะได้รับ

จากผลการทดลองจะสามารถสรุปได้ว่าการใช้ตัวจำแนกประเภทโดยเรียนรู้จากข้อมูลที่ถูกปกปิดมีประสิทธิภาพที่ดีเพียงพอต่อการใช้งานหรือไม่ และสามารถเลือกใช้งานโมเดลความปลอดภัยในการปกปิดข้อมูลและอัลกอริทึ่มของกลุ่มตัวจำแนกประเภทได้อย่างเหมาะสมตามความต้องการในการใช้งาน

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

ทฤษฎีที่เกี่ยวข้องของการออกแบบและพัฒนางานวิจัยนี้ ประกอบไปด้วย การทำเหมืองข้อมูล กลุ่มของการจำแนกประเภท ประเภทของคุณสมบัติ และ การทำเหมืองข้อมูลโดยรักษาความเป็นส่วนตัวของข้อมูล

2.1.1 การทำเหมืองข้อมูล

การทำเหมืองข้อมูล (Data mining) คือกระบวนการที่กระทำกับข้อมูลจำนวนมากเพื่อค้นหารูปแบบและความสัมพันธ์ที่ซ่อนอยู่ในชุดข้อมูลนั้น ในปัจจุบันการทำเหมืองข้อมูลได้ถูกนำไปประยุกต์ใช้ในงานหลายประเภท ทั้งในด้านธุรกิจที่ช่วยในการตัดสินใจของผู้บริหาร ในด้านวิทยาศาสตร์และการแพทย์รวมทั้งในด้านเศรษฐกิจและสังคม

การทำเหมืองข้อมูล เปรียบเสมือนวิวัฒนาการหนึ่งในการจัดเก็บและตีความหมายข้อมูล จากเดิมที่มีการจัดเก็บข้อมูลอย่างง่ายๆ มาสู่การจัดเก็บในรูปแบบฐานข้อมูลที่สามารถดึงข้อมูลสารสนเทศมาใช้จนถึงการทำเหมืองข้อมูลที่สามารถค้นพบความรู้ที่ซ่อนอยู่ในข้อมูล ความรู้ที่ได้จากการทำเหมืองข้อมูลมีหลายรูปแบบ ได้แก่

2.1.1.1 กฎความสัมพันธ์

กฎความสัมพันธ์ (Association rule) คือการแสดงความสัมพันธ์ของเหตุการณ์หรือวัตถุ ที่เกิดขึ้นพร้อมกัน ตัวอย่างของการประยุกต์ใช้กฎเชื่อมโยง เช่น การวิเคราะห์ข้อมูลการขายสินค้า โดยเก็บข้อมูลจากระบบ ณ จุดขาย (POS) หรือร้านค้าออนไลน์ แล้วพิจารณาสินค้าที่ผู้ซื้อมักจะซื้อพร้อมกัน เช่น ถ้าพบว่าคนที่ซื้อเทปวิดีโอมักจะซื้อเทปกาวด้วย ร้านค้าก็อาจจะจัดร้านให้สินค้าสองอย่างอยู่ใกล้กัน เพื่อเพิ่มยอดขาย หรืออาจจะพบว่าหลังจากคนซื้อหนังสือ ก แล้ว มักจะซื้อหนังสือ ข ด้วย ก็สามารถนำความรู้นี้ไปแนะนำผู้ที่กำลังจะซื้อหนังสือ ก ได้

2.1.1.2 การจำแนกประเภทข้อมูล

การจำแนกประเภทข้อมูล (Data classification) คือการหากฎเพื่อระบุประเภทของวัตถุจากคุณสมบัติของวัตถุ เช่น หาความสัมพันธ์ระหว่างผลการตรวจร่างกายต่าง ๆ กับการเกิดโรค โดยใช้ข้อมูลผู้ป่วยและการวินิจฉัยของแพทย์ที่เก็บไว้ เพื่อนำมาช่วยวินิจฉัยโรคของผู้ป่วย หรือการวิจัย

ทางการแพทย์ ในทางธุรกิจจะใช้เพื่อดูคุณสมบัติของผู้ที่จะก่อหนี้ดีหรือหนี้เสีย เพื่อประกอบการพิจารณาการอนุมัติเงินกู้

2.1.1.3 การแบ่งกลุ่มข้อมูล

การแบ่งกลุ่มข้อมูล (Data clustering) คือการแบ่งข้อมูลที่มีลักษณะคล้ายกันออกเป็นกลุ่ม แบ่งกลุ่มผู้ป่วยที่เป็นโรคเดียวกันตามลักษณะอาการ เพื่อนำไปใช้ประโยชน์ในการวิเคราะห์หาสาเหตุของโรค โดยพิจารณาจากผู้ป่วยที่มีอาการคล้ายคลึงกัน

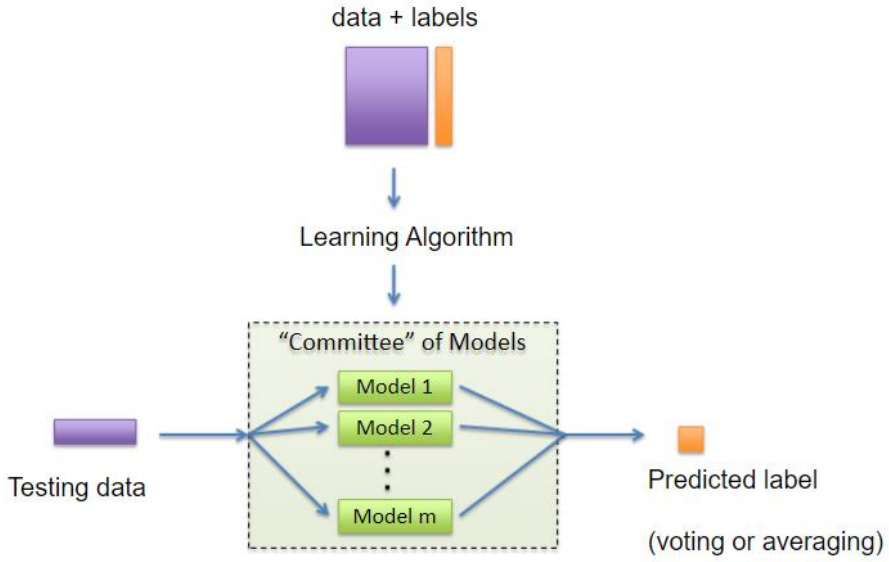
2.1.1.4 การสร้างมโนภาพ

การสร้างมโนภาพ (Visualization) คือการสร้างภาพคอมพิวเตอร์กราฟิกที่สามารถนำเสนอข้อมูลมากมายอย่างครบถ้วนแทนการใช้ข้อความนำเสนอข้อมูลที่มากมาย เราอาจพบข้อมูลที่ซ้อนเร้นเมื่อดูข้อมูลชุดนั้นด้วยจินตทัศน์

2.1.2 กลุ่มของการจำแนกประเภท

กลุ่มของการจำแนกประเภท (Ensemble classification) [1] คือการสร้างตัวจำแนกประเภทหลายๆตัวแล้วนำผลลัพธ์ที่ได้มาผ่านกระบวนการรวมและนำมาพิจารณาว่าผลลัพธ์อันไหนดีที่สุด โดยอาจใช้วิธีการโหวตเลือกคำตอบที่ตอบตรงกันมากที่สุด เป้าหมายของการใช้กลุ่มของการจำแนกประเภทคือการทำให้โมเดลมีความเหมาะสมกับลักษณะชุดข้อมูลต่างๆ (Generalizability) และมีความทนทาน (Robustness) เพราะการใช้กลุ่มของการจำแนกประเภทจะทำให้ค่าความแปรปรวนและความเอนเอียงของตัวจำแนกประเภทลดลง จึงทำให้ประสิทธิภาพการทำงานของตัวจำแนกประเภทจะดีกว่าการใช้ตัวจำแนกประเภทแบบเดี่ยว

การใช้กลุ่มของการจำแนกประเภท อาจจะไม่ได้ค่าความแม่นยำที่ดีที่สุดแต่การใช้กลุ่มของการจำแนกประเภทจะช่วยลดความเสี่ยงของการเกิดความเหมาะสมกับข้อมูลสอนมากเกินไป (overfitting) เพราะบางครั้งการใช้การจำแนกประเภทแบบเดี่ยวผลลัพธ์ที่ได้จะมีความแม่นยำกับข้อมูลสอนอย่างเดียว และจะได้ความแม่นยำน้อยเมื่อใช้กับข้อมูลชุดอื่นๆ การใช้กลุ่มของการจำแนกประเภทจะได้ผลลัพธ์ที่ดีกับข้อมูลขนาดใหญ่และข้อมูลที่มาจากหลายแหล่งข้อมูลมากกว่าการจำแนกประเภทแบบเดี่ยว



ภาพที่ 2 การทำงานโดยกลุ่มของการจำแนกประเภท [2]

การใช้กลุ่มของการจำแนกประเภทมีหลายอัลกอริทึมในการสร้าง โดยอัลกอริทึมที่จะใช้ในงานวิจัยชิ้นนี้มีดังต่อไปนี้

2.1.2.1 แบ้กกิ้ง

แบ้กกิ้ง (Bagging) [3] คือการสร้างตัวจำแนกประเภทหลายตัวโดยตัวจำแนกประเภทแต่ละตัวจะใช้ข้อมูลสอนที่แตกต่างกัน โดยข้อมูลสอนแต่ละชุดจะถูกสุ่มจากข้อมูลสอนตัวตั้งต้น และนำผลลัพธ์ที่ได้ในแต่ละตัวจำแนกประเภทมารวมกันและทำนายผลลัพธ์สุดท้าย วิธีนี้จะทำให้ลดความแปรปรวนและลดการเกิดปัญหาความเหมาะสมกับข้อมูลสอนมากเกินไป (overfitting)

2.1.2.2 โหวตติ้ง

โหวตติ้ง (Voting) [4] คือการผสมผสานของการเรียนรู้ด้วยอัลกอริที่มต่างๆ และการโหวตจากเสียงส่วนมาก (majority vote) หรือการโหวตจากความน่าจะเป็นของค่าเฉลี่ยจากการทำนาย (soft vote) เพื่อที่จะทำนายผลลัพธ์ โดยวิธีนี้จะใช้ตัวจำแนกประเภทหลายอัลกอริที่มมาใช้และนำผลลัพธ์ที่ได้มาโหวต เช่นการใช้ Decision Tree, K-Nearest Neighbors (K-NN) และ Neural Network เป็นต้น การใช้กลุ่มของการจำแนกประเภทในอัลกอริที่มนี้จะทำให้เกิดความสมดุลของข้อเสียในแต่ละตัวจำแนกประเภท

2.1.2.3 แรนด้อมฟอเรส

แรนด้อมฟอเรส (Random Forest) [5] คือการทำแผนภาพต้นไม้ (Decision Tree) หลายต้นโดยใช้ข้อมูลสอนที่ต่างกัน ข้อมูลสอนได้จากการสุ่มจากข้อมูลดั้งเดิม และใช้คุณสมบัติที่ต่างกันในการสอน

2.1.2.4 เอด้าบูส

เอด้าบูส (AdaBoost) [6] คือเทคนิคในกลุ่มบู้สติ้ง (boosting) โดยสร้างกลุ่มของตัวจำแนกประเภทจากอัลกอริทึ่มเดียวและชุดข้อมูลเดียวกัน แต่จะทำการวนลูปในการปรับค่าน้ำหนักเพื่อทำให้ accuracy ของโมเดลดีมากขึ้น เทคนิคในกลุ่ม บู้สติ้งจะได้รับความนิยมเพราะสามารถใช้ได้กับทุกอัลกอริทึ่ม

2.1.2.5 โรเตชั่นฟอเรส

โรเตชั่นฟอเรส (Rotation Forest) [7] คือการใช้กลุ่มของตัวจำแนกประเภทที่มีการเรียนรู้จากชุดข้อมูลสอนที่แตกต่างกัน โดยทำการสุ่มจากชุดข้อมูลสอนตัวตั้งต้น และนำการวิเคราะห์องค์ประกอบหลัก (Principle Component Analysis, PCA) มาใช้ในแต่ละชุดข้อมูล

2.1.3 ประเภทของคุณสมบัติ

คุณสมบัติของข้อมูลจะสามารถแบ่งประเภทได้ 4 ประเภทดังต่อไปนี้

2.1.3.1 Explicit Identifiers

Explicit Identifiers คือคุณสมบัติที่สามารถนำมาระบุตัวตนได้ เช่น ชื่อ นามสกุล เลขบัตรประจำตัว เป็นต้น

2.1.3.2 Quasi-identifiers

Quasi-identifiers คือกลุ่มของคุณสมบัติที่มีความสามารถเพียงพอที่จะระบุตัวตนได้เมื่อนำมารวมกันข้อมูลสาธารณะ เช่น รหัสไปรษณีย์ อายุ เป็นต้น

2.1.3.3 Sensitive Attributes

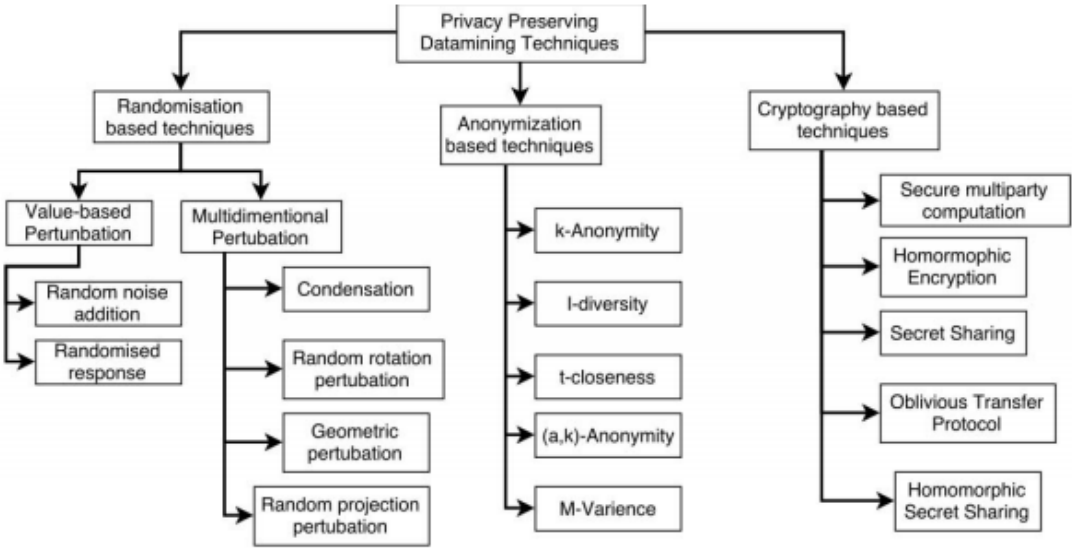
Sensitive Attributes คือกลุ่มของคุณสมบัติที่เป็นข้อมูลเฉพาะตัวของบุคคลที่มีความอ่อนไหว เช่น โรคประจำตัว เงินเดือน เป็นต้น

2.1.3.4 Non-sensitive Attributes

Non-sensitive Attributes คือกลุ่มของคุณสมบัติที่ไม่ได้มีความสำคัญอะไรถึงแม้ข้อมูลจะถูกเปิดเผย

2.1.4 การทำเหมืองข้อมูลโดยรักษาความเป็นส่วนตัวของข้อมูล

การทำเหมืองข้อมูลโดยรักษาความเป็นส่วนตัวของข้อมูล (Privacy-Preserving Data Mining) คือการทำให้ข้อมูลที่จะนำไปทำเหมืองข้อมูลไม่สามารถระบุตัวตนได้ โดยขั้นตอนในการรักษาความเป็นส่วนตัวสามารถทำได้หลายวิธีในการพยายามแปลงข้อมูลดั้งเดิมให้กลายเป็นข้อมูลที่ถูกปกปิด โดยสามารถแบ่งประเภทได้ดังนี้ [8]



ภาพที่ 3 เทคนิคการป้องกันความเป็นส่วนตัวของข้อมูล [8]

จากภาพที่ 3 จะเห็นได้ว่าสามารถแบ่งเทคนิคการทำเหมืองข้อมูลโดยรักษาความเป็นส่วนตัวได้สามชนิด เทคนิคแรกคือเทคนิคการสุ่ม (Randomization) วิธีการนี้จะนำค่ารบกวน (noise) มาทำกระบวนการบางอย่างกับข้อมูลเดิม ซึ่งจะทำให้ผลลัพธ์ที่ได้จะต่างไปจากเดิม จึงทำให้คนที่อ่านข้อมูลนี้ไม่สามารถรู้ได้ว่าข้อมูลรายการนี้เป็นของใคร เทคนิคที่สองคือการปกปิดข้อมูล (Anonymization) โดยรายละเอียดจะกล่าวในหัวข้อถัดไป เทคนิคที่สามคือการเข้ารหัส (Cryptography) เทคนิคนี้จะเป็นการนำข้อมูลเข้ารหัสก่อนที่จะนำไปใช้งาน ซึ่งแต่ละเทคนิคก็จะมีข้อดีข้อเสียที่ต่างกัน

2.1.5 การปกปิดข้อมูล (Anonymization or De-identification)

วิธีการนี้คือการทำให้ข้อมูลแต่ละรายการมีความเหมือนกับข้อมูลรายการอื่น สามารถทำได้โดยการทำข้อมูลให้มีความทั่วไปมากขึ้น (generalization) หรือการทำข้อมูลให้ไม่มีความเฉพาะเจาะจง (suppression) กับคุณสมบัติต่างๆที่เป็นคุณสมบัติที่จะสามารถนำไประบุตัวตนได้ (quasi-identifiers) จนกว่าจะไม่สามารถระบุได้ว่าข้อมูลเป็นของใคร เช่น อายุ 27 แปลงเป็น 25-30 กระบวนการนี้มีโมเดลของความปลอดภัยที่ได้รับความนิยมได้แก่ k-anonymity , l-diversity , t-closeness

2.1.5.1 k-anonymity

k-anonymity [9] คือโมเดลที่จะการันตีว่าทุกลักษณะของข้อมูลจะมีอย่างน้อย k-1 รายการ ซึ่งจะทำให้ผู้ที่ต้องการจะโจมตีไม่สามารถทำการระบุตัวตนของข้อมูลได้ (re-identification attack)

	Non-Sensitive			Sensitive
	Zip Code	Age	Nationality	Condition
1	13053	28	Russian	Heart Disease
2	13068	29	American	Heart Disease
3	13068	21	Japanese	Viral Infection
4	13053	23	American	Viral Infection
5	14853	50	Indian	Cancer
6	14853	55	Russian	Heart Disease
7	14850	47	American	Viral Infection
8	14850	49	American	Viral Infection
9	13053	31	American	Cancer
10	13053	37	Indian	Cancer
11	13068	36	Japanese	Cancer
12	13068	35	American	Cancer

	Non-Sensitive			Sensitive
	Zip Code	Age	Nationality	Condition
1	130**	< 30	*	Heart Disease
2	130**	< 30	*	Heart Disease
3	130**	< 30	*	Viral Infection
4	130**	< 30	*	Viral Infection
5	1485*	≥ 40	*	Cancer
6	1485*	≥ 40	*	Heart Disease
7	1485*	≥ 40	*	Viral Infection
8	1485*	≥ 40	*	Viral Infection
9	130**	3*	*	Cancer
10	130**	3*	*	Cancer
11	130**	3*	*	Cancer
12	130**	3*	*	Cancer

ภาพที่ 4 ตัวอย่างการใช้งาน k-anonymity [10]

ภาพที่ 4 คือการปกปิดข้อมูลโดยใช้โมเดล k-Anonymity โดย k=4 จะเห็นได้ว่าข้อมูลที่ถูกแปลงแล้วจะมี 4 รายการที่มีลักษณะข้อมูลที่เหมือนกัน

2.1.5.2 l-diversity

l-diversity [10] เป็นโมเดลที่ถูกพัฒนาจาก k-anonymity จึงเป็นโมเดลที่มีลักษณะคล้ายกัน แต่ l-diversity จะเพิ่มคุณสมบัติอีกข้อนึงโดยโมเดลจะการันตีว่าทุกๆกลุ่มที่มี quasi-identifier เหมือนกัน (equivalence class) จะต้องมีคุณลักษณะที่มีความอ่อนไหว (sensitive attributes) แตกต่างกันตามค่า l

	Non-Sensitive			Sensitive
	Zip Code	Age	Nationality	Condition
1	130**	< 30	*	Heart Disease
2	130**	< 30	*	Heart Disease
3	130**	< 30	*	Viral Infection
4	130**	< 30	*	Viral Infection
5	1485*	≥ 40	*	Cancer
6	1485*	≥ 40	*	Heart Disease
7	1485*	≥ 40	*	Viral Infection
8	1485*	≥ 40	*	Viral Infection
9	130**	3*	*	Cancer
10	130**	3*	*	Cancer
11	130**	3*	*	Cancer
12	130**	3*	*	Cancer

	Non-Sensitive			Sensitive
	Zip Code	Age	Nationality	Condition
1	1305*	≤ 40	*	Heart Disease
4	1305*	≤ 40	*	Viral Infection
9	1305*	≤ 40	*	Cancer
10	1305*	≤ 40	*	Cancer
5	1485*	> 40	*	Cancer
6	1485*	> 40	*	Heart Disease
7	1485*	> 40	*	Viral Infection
8	1485*	> 40	*	Viral Infection
2	1306*	≤ 40	*	Heart Disease
3	1306*	≤ 40	*	Viral Infection
11	1306*	≤ 40	*	Cancer
12	1306*	≤ 40	*	Cancer

ภาพที่ 5 ตัวอย่างการใช้งานระหว่าง k-anonymity และ l-diversity [10]

2.1.5.3 t-closeness

t-closeness [11] เป็นโมเดลที่ถูกพัฒนาจาก l-diversity โดยโมเดลจะมีคุณสมบัติที่การันตีว่าทุกๆกลุ่มที่มี quasi-identifier เหมือนกัน (equivalence class) จะต้องมีความต่างระหว่างการกระจายของคุณลักษณะที่มีความอ่อนไหว (sensitive attributes) ของกลุ่มนั้นและการกระจายของคุณลักษณะทั้งหมดไม่เกินค่า t (threshold)

	ZIP Code	Age	Salary	Disease
1	4767*	≤ 40	3K	gastric ulcer
3	4767*	≤ 40	5K	stomach cancer
8	4767*	≤ 40	9K	pneumonia
4	4790*	≥ 40	6K	gastritis
5	4790*	≥ 40	11K	flu
6	4790*	≥ 40	8K	bronchitis
2	4760*	≤ 40	4K	gastritis
7	4760*	≤ 40	7K	bronchitis
9	4760*	≤ 40	10K	stomach cancer

ภาพที่ 6 ตัวอย่างการใช้งาน t-closeness [11]

2.1.5.4 SafePub

SafePub เป็นการอัลกอริทึ่มในการปกปิดข้อมูลชนิดหนึ่ง อัลกอริทึ่มนี้จะสามารถทำให้ชุดข้อมูลมีความเหมือนกับการใช้งาน Differential privacy อัลกอริทึ่ม SafePub จะไม่ทำการแก้ไขชุดข้อมูลแต่จะทำการสร้างชุดข้อมูลจำลองขึ้นมาใหม่โดยที่มีการเพิ่ม noise ในชุดข้อมูลใหม่

2.1.5.5 population uniqueness

Population uniqueness เป็นโมเดลความปลอดภัยที่ทำการวัดความไม่เหมือนใครของข้อมูล (uniqueness) โดยนำไปเทียบกับข้อมูลของประชากรในประเทศหรือภูมิภาคต่างๆ โดยข้อมูลในชุดข้อมูลจะต้องมีความไม่เหมือนใครน้อยกว่าค่า threshold ที่กำหนด

2.1.5.6 k-map

k-map จะมีการทำงานที่แทบจะใกล้เคียงกับ k-anonymity โดยที่จะเหมาะกับการใช้งานกรณีที่ชุดข้อมูลมีขนาดเล็กจนทำให้ไม่สามารถปกปิดข้อมูลแล้วได้ผลลัพธ์ที่มีประสิทธิภาพ k-map จะมีการพิจารณาค่าของคุณสมบัติเพื่อที่จะลดการปกปิดข้อมูลให้น้อยที่สุด จะทำการปกปิดข้อมูลเฉพาะข้อมูลที่มีอัตราการเสี่ยงจากการถูกระบุตัวตน

2.2 งานวิจัยที่เกี่ยวข้อง

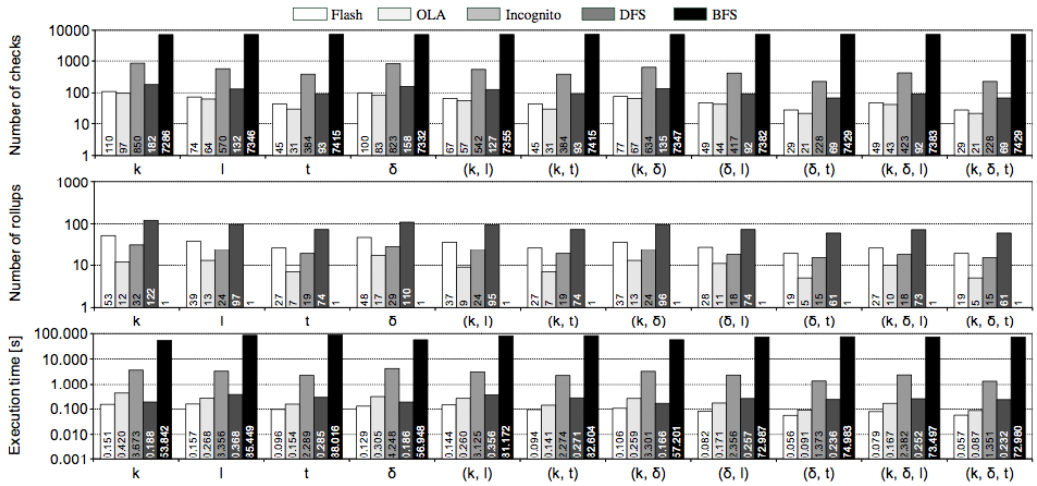
งานวิจัยที่สรุปและประเมินเกี่ยวกับการทำเหมืองข้อมูลโดยยังรักษาความปลอดภัยของข้อมูล (Privacy Preserving Data Mining) โดย K.Saranya et al. [12] ได้ทำการสรุปออกมาว่าเทคนิคการรักษาความปลอดภัยของข้อมูลแบ่งได้เป็นสามประเภทคือ Randomization method , Anonymization และ Distributed Privacy Preservation K.Saranya ได้ทำการอธิบายลักษณะของการทำงานในแต่ละประเภทและสรุปข้อดีและข้อเสียของแต่ละประเภท งานวิจัยของ Aobakwe Senosi et al. [8] ได้ทำการสรุปว่าสามารถแบ่งเทคนิคการรักษาความปลอดภัยของข้อมูลได้สามประเภทเช่นกัน แต่ได้ทำการเปลี่ยนประเภทจาก Distributed Privacy Preservation เป็น Cryptography โดยงานวิจัยของ Aobakwe Senosi ได้นำงานวิจัยที่เกี่ยวข้องกับการทำเหมืองข้อมูลโดยรักษาความปลอดภัยของข้อมูลต่างๆมาสรุปข้อดีข้อเสียของแต่ละเทคนิคและทำการสรุปได้ตามภาพที่ 7

Criteria / PPDM Technique	Computational Cost	Privacy Preser-vation	Accuracy of mining	Scalability
Cryptography	High[45], [44], [42], [9], [4], [33], [36]	High [42], [37], [4], [26], [46], [33], [36]	High [36], [9], [33], [4]	Low [4], [26]
Randomisation	Low [42], [4]	High [42], [4], [21], [9], [26]	High [42] Average [4], [21] Low	High [4]
anonymization	Low [4]	Average [24], [4], [26]	Average [4]	

ภาพที่ 7 การประเมินลักษณะของเทคนิคการรักษาความปลอดภัยของข้อมูล [8]

งานวิจัยชิ้นนี้เลือกใช้เทคนิคการรักษาความปลอดภัยแบบ Anonymization โดยได้มีงานวิจัยหลายงานได้เสนอโมเดลความปลอดภัย (Privacy Model) ในรูปแบบต่างๆ โมเดลความปลอดภัยที่ได้รับความนิยมมากมาจากงานวิจัยของ Latanya Sweeney [9] โดยได้เสนอโมเดลที่ชื่อว่า k-anonymity โมเดลความปลอดภัยนี้จะต้องการันตีว่าทุกลักษณะของข้อมูลจะมีอย่างน้อย k-1 รายการ [13] ซึ่งจะทำให้ลดความเสี่ยงจากการถูกระบุตัวตน (re-identification) โมเดลความปลอดภัยที่ได้รับความนิยมอีกหนึ่งตัวคือ l-diversity โมเดลความปลอดภัยนี้มาจากงานวิจัยของ Ashwin Machanavajjhala et al. [10] ซึ่งงานวิจัยของ Ashwin Machanavajjhala ได้บอกว่าโมเดลความปลอดภัย k-anonymity ยังมีจุดที่บอบบางและเป็นความผิดพลาดที่รุนแรง โดยได้ทำการแสดงการโจมตีสองรูปแบบคือ การโจมตีความเป็นแบบเดียวกัน (Homogeneity Attack) และ การโจมตีแบบมีความรู้เบื้องหลัง (Background Knowledge Attack) ซึ่งผลลัพธ์ในงานวิจัยชิ้นนี้สามารถสรุปได้ว่าโมเดลความปลอดภัย l- diversity มีประสิทธิภาพที่ดีกว่า k-anonymity หลังจากโมเดลความปลอดภัย l-diversity ได้ถูกเผยแพร่ งานวิจัยของ Ninghui Li et al. [11] มีการนำเสนอโมเดลความปลอดภัยที่มีชื่อว่า t-closeness โดยงานวิจัยชิ้นนี้ได้กล่าวว่าโมเดลความปลอดภัย l-diversity ยังมีข้อจำกัดอยู่ Ninghui Li ได้ทำการเสนอจุดด้อยของ l-diversity และกล่าวว่า t-closeness จะแก้ปัญหาเหล่านี้ได้

งานวิจัยของ Fabian Prasser et al. [14] ได้ทำการทดลองและวัดผลเพื่อเปรียบเทียบประสิทธิภาพของการทำงานในแต่ละโมเดลความปลอดภัยและอัลกอริทึ่มที่ใช้ในการปกปิดข้อมูล โดยมีโมเดลความปลอดภัยดังต่อไปนี้ k-anonymity , l-diversity , t-closeness , δ-Presence และมีอัลกอริทึ่มดังต่อไปนี้ BFS , DFS , Incognito , OLA , Flash ผลการทดลองของงานวิจัยสามารถวิเคราะห์ได้จากภาพที่ 8



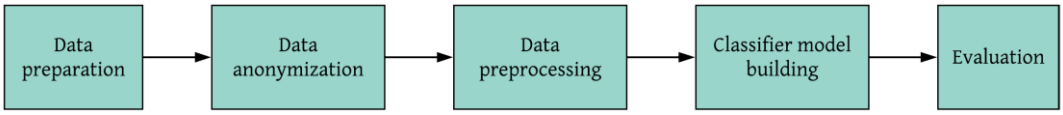
ภาพที่ 8 ผลลัพธ์จากการทดลอง [14]

งานวิจัยของ Ines Buratovic et al. [15] ได้ทำการทดลองว่าเมื่อนำข้อมูลที่ถูกปกปิด (Anonymized data) มาทำเหมืองข้อมูลจะให้ผลลัพธ์เป็นอย่างไร ผลลัพธ์ที่ได้จากงานวิจัยของ Ines Buratovic คือการทำเหมืองข้อมูลกับข้อมูลที่ถูกปกปิดจะมีความแม่นยำของตัวจำแนกประเภทลดลงแต่ไม่แตกต่างกันอย่างมีนัยยะสำคัญ ซึ่งสามารถสรุปได้เช่นเดียวกับงานวิจัยของ Hebert O. Silva et al. [16]

บทที่ 3
วิธีดำเนินการวิจัย

3.1 แนวคิดและวิธีวิจัย

ภาพรวมแนวคิดงานวิจัยของงานวิจัยชิ้นนี้จะสามารถแบ่งออกเป็นกระบวนการดังต่อไปนี้ การนำเข้าข้อมูล กระบวนการปกปิดข้อมูล การประมวลผลก่อน การสร้างโมเดล และการวัดผล ตามลำดับ โดยการทดลองจะทำแต่ละขั้นตอนตามลำดับดังภาพที่ 9



ภาพที่ 9 ขั้นตอนการดำเนินงานของการทดลอง

3.1.1 การนำเข้าข้อมูล

Dataset	#Quasi-Identifiers	#Records	File size (MB)
ADULT	8	30162	2.52
FARS	7	100937	7.19
ATUS	8	539253	84.03

ตารางที่ 1 รายละเอียดของชุดข้อมูล

ข้อมูลที่ใช้ในการทดลองมีทั้งหมด 3 ชุดข้อมูล ได้แก่ 1) Adult dataset [17] เป็นข้อมูลการสำรวจประชากรของประเทศสหรัฐอเมริกาในปี 1994 ซึ่งชุดข้อมูลนี้เป็น De Facto Standard problem และนิยมถูกใช้เพื่อเป็นเกณฑ์มาตรฐานในการปกปิดข้อมูล 2) FARS dataset (The Fatality Analysis Reporting System) เป็นข้อมูลสถิติการชนของรถทุกชนิดที่เกิดขึ้นบนท้องถนนภายในประเทศสหรัฐอเมริกา โดยชุดข้อมูลนี้มาจากหน่วยงาน NHTSA (National Highway Traffic Safety Administration) และ 3) ATUS dataset เป็นข้อมูลผลสำรวจการใช้เวลาของคนอเมริกา ชุดข้อมูลทั้งสามชุดนั้นเป็นปัญหาในการปกปิดข้อมูลที่เกิดขึ้นจริง (real-world anonymization problems) โดยรายละเอียดคุณสมบัติของแต่ละชุดข้อมูลจะแสดงในตารางที่ 2-4

Attributes	Type	Possible values
sex	String	Male, Female
age	Integer	Continuous
race	String	White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black
marital-status	String	Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse
education	String	Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool
native-country	String	United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad&Tobago, Peru, Hong, Holand-Netherlands
workclass	String	Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked.
occupation	String	Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces
salary-class	String	>50K, <=50K.

ตารางที่ 2 รายละเอียดของคุณสมบัติในชุดข้อมูล ADULT

Attributes	Type	Possible values
iage	Integer	Continuous
irace	String	#NV, All other races, American Indian, Asian Indian, Asian/Pacific Is, Black, Chinese, Filipino, Guamanian,

		Hawaiian, Japanese, Korean, Multiple races, Not a fatal(N/A), Other Asian, Other Indian, Samoan, Unknown, Vietnamese, White
ideathmon	String	January, February, March, April, May, June, July, August, September, October, November, December, Not applicable, Unknown
ideathday	Integer	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 99
isex	String	Male, Female, Unknown
ihispanic	String	#NV, Central/So Amer, Cuban, Mexican, No specified/Oth, Non-Hispanic, Not a fatal(N/A), Other/unk Hisp, Puerto Rican, Unknown
istatenum	String	Alabama, Alaska, Arizona, Arkansas, California, Colorado, Connecticut, Delaware, Dist of Columbia, Florida, Guam, Hawaii, Idaho, Illinois, Indiana, Iowa, Kansas, Kentucky, Louisiana, Maine, Maryland, Massachusetts, Michigan, Minnesota, Mississippi, Missouri, Montana, Nebraska, Nevada, New Hampshire, New Jersey, New Mexico, New York, North Carolina, North Dakota, Ohio, Oklahoma, Oregon, Pennsylvania, Rhode Island, South Carolina, South Dakota, Tennessee, Taxas, Utah, Vermont, Virginia, Washington, West Virginia, Wisconsin, Wyoming
iinjury	String	Died Prior to, Fatal Injury, Incapacitating, Inj (Sev Unk) , No, Non Incapacity, Possible, Unknown

ตารางที่ 3 รายละเอียดของคุณสมบัติในชุดข้อมูล FARS

Attributes	Type	Possible values
region	String	South, Midwest, West, Northeast
age	Integer	Continuous
sex	String	NIU (Not in universe), Female, Male

race	String	White only, Black only, Asian only, NIU (Not in universe), American Indian, Alaskan Native, White-American Indian, White-Black, White-Asian, Hawaiian Pacific Islander only, Asian or Pacific Islander, Black-American Indian, White-Hawaiian, White-Black-American Indian
marital_status	String	Married – spouse present, NIU (Not in universe), Never married, Divorced, Widowed, Seperated, Married – spouse absent
citizenship_status	String	Native, born in United States, Foreign born, not a U.S. citizen, Foreign born, U.S. citizen by naturalization, NIU (Not in universe), Native, born abroad of American parent or parents, Native, born in Puerto Rico or U.S. Outlying Area
birthplace	String	155 countries
highest_level_of_school_completed	String	NIU (Not in universe), High school graduate – diploma, Some college but no degree, Bachelor's degree (BA, AB, BS, etc.) Master's degree (MA, MS, MEng, Med, MSW, etc.), 10^{th} grade, 11^{th} grade, 9^{th} grade, Associate degree – occupational vocational, Associate degree – academic program, 7^{th} or 8^{th} grade, High school graduate – GED, 5^{th} or 6^{th} grade, 12^{th} grade – no diploma
Labor_force_status	String	NIU (Not in universe), Employed – at work, Not in labor force, Unemployed – looking, Employed – absent, Unemployed – on layoff

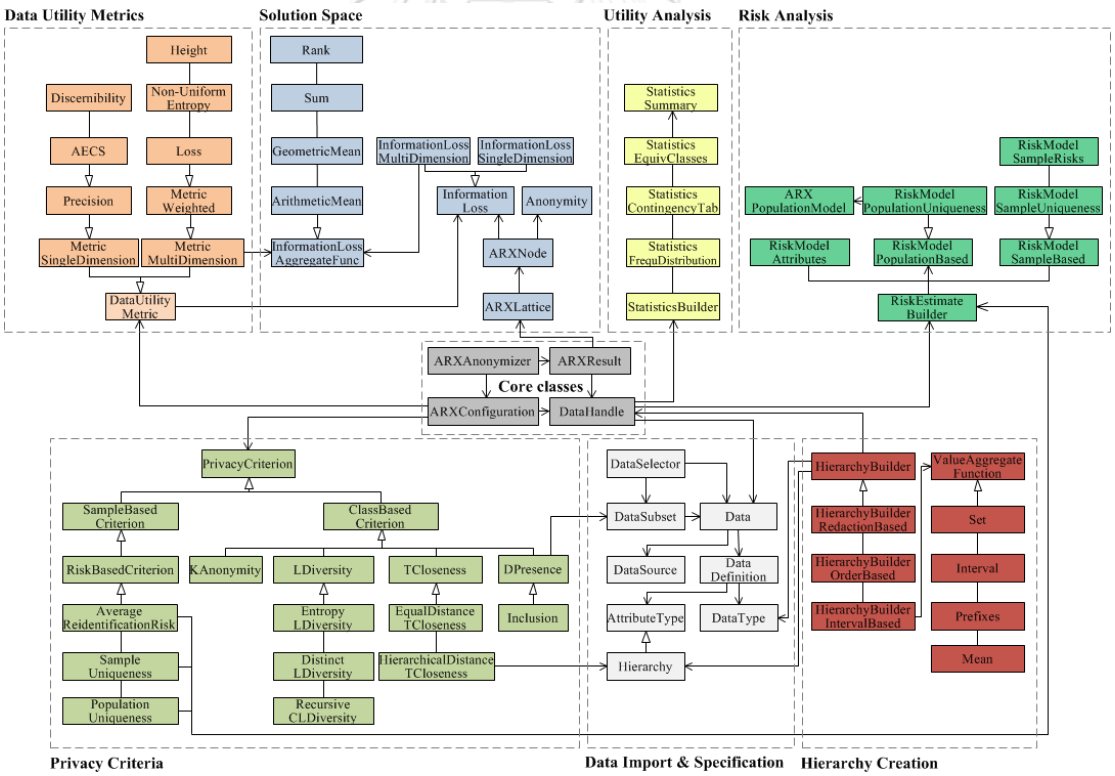
ตารางที่ 4 รายละเอียดของคุณสมบัติในชุดข้อมูล ATUS

3.1.2 การปกปิดข้อมูล

ในการป้องกันความเป็นส่วนตัวของข้อมูล (Anonymization or De-Identification) เราจำเป็นที่จะคำนึงถึงคุณภาพของข้อมูลให้มากที่สุดเท่าที่ทำได้ โดยงานวิจัยชิ้นนี้จะใช้ความเสี่ยงจากการถูกระบุตัวตน (Re-Identification risk) เป็นมาตรวัดความปลอดภัยของข้อมูล

งานวิจัยชิ้นนี้จะใช้โปรแกรม ARX [18] ในการปกปิดข้อมูล ARX เป็น open source ซอฟต์แวร์สำหรับการปกปิดข้อมูล โปรแกรมถูกพัฒนาด้วยภาษา Java ข้อดีของโปรแกรม ARX เมื่อเทียบกับโปรแกรมอื่นๆที่ใช้ในการปกปิดข้อมูลสามารถสรุปได้ดังนี้

- มีโมเดลความปลอดภัยที่หลากหลายให้เลือกใช้อย่างมีประสิทธิภาพ
- โปรแกรมได้ออกแบบ Graphic User Interface ให้สามารถใช้งานง่าย
- มี Programming Interface ให้สามารถใช้งานร่วมกับซอฟต์แวร์อื่นๆ
- มีเอกสารที่บอกวิธีการใช้และปัจจุบันยังมีการสนับสนุนจากผู้ที่พัฒนา
- เป็นซอฟต์แวร์ที่ open source



ภาพที่ 10 Unified Modeling Language (UML) diagram ของ ARX [18]

งานวิจัยชิ้นนี้จะทำการปกปิดข้อมูลด้วยโมเดลความปลอดภัยและค่าพารามิเตอร์ต่างๆ โดยจะแบ่งเป็นชุดข้อมูลในการทดลองทั้งหมด 17 ชุด ดังตารางที่ 5

Data version	Privacy Model	Parameters
original	-	-
k1	k-anonymity	k = 2
k2	k-anonymity	k = 5
k3	k-anonymity	k = 10
l1	l-diversity	l = 4 (Distinct-l)
l2	l-diversity	l = 4 (Shannon entropy)
l3	l-diversity	l = 4, c = 3 (Recursive)
t1	t-closeness	t = 0.2 (EMD with equal ground-distance)
t2	t-closeness	t = 0.2 (EMD with hierarchical ground-distance)
t3	t-closeness	t = 0.2 (EMD with ordered distance)
dp1	safepub	delta = 10^{-5}
dp2	safepub	delta = 10^{-6}
dp3	safepub	delta = 10^{-7}
pu1	uniqueness	threshold = 10^{-4}
pu2	uniqueness	threshold = 10^{-5}
pu3	uniqueness	threshold = 10^{-6}
km1	k-map	k = 2
km2	k-map	k = 5
km3	k-map	k = 10

ตารางที่ 5 รายละเอียดของชุดข้อมูลในการทดลอง

3.1.3 การประมวลผลก่อน

ก่อนที่จะนำข้อมูลไปทำการสร้างโมเดล จะต้องทำการเลือกคุณสมบัติที่ใช้ในการจำแนกประเภทและทำการเข้ารหัส (label encoding) ในทุกๆคุณสมบัติ เพื่อที่จะสามารถนำไปใช้สร้างตัวจำแนกประเภทด้วยโมเดลต่างๆได้และทำให้ใช้พื้นที่ในการเก็บข้อมูลน้อยลง

3.1.4 การสร้างโมเดล

ในขั้นตอนของการสอนโมเดลตัวจำแนกประเภทจะต้องทำการเลือกอัลกอริที่มในการสร้างตัวจำแนกประเภทและทำการเลือกคุณสมบัติที่ต้องการจะใช้ในการเรียนรู้ โดยจะต้องระบุว่าคุณสมบัติไหนเป็นคลาส (class label) ของการจำแนกประเภท ในการวิจัยชิ้นนี้จะทำการเลือกคุณสมบัติในการสอนตัวจำแนกประเภทดังตารางที่ 6 จะสรุปได้ว่าตัวจำแนกประเภทที่เรียนรู้จากชุดข้อมูล ADULT จะสามารถจำแนกประเภทข้อมูลในลักษณะต่างๆว่าจะมีเงินเดือนต่อปีอยู่ในระดับไหน (น้อยกว่า 50,000 หรือ มากกว่า 50,000) ตัวจำแนกประเภทที่เรียนรู้จากชุดข้อมูล FARS จะสามารถจำแนกประเภทข้อมูลในลักษณะต่างๆว่าจะมีการเจ็บป่วยแบบไหน ตัวจำแนกประเภทที่เรียนรู้จากชุดข้อมูล ATUS จะสามารถจำแนกประเภทข้อมูลในลักษณะต่างๆว่ามีสถานะแรงงานเป็นอย่างไร

Dataset	Attributes	Class label
ADULT	sex, age, race, marital-status, education, native-country, workclass, occupation	salary-class
FARS	iage, irace, ideathmon, ideathday, isex, ihispanic, istatenum	iinjury
ATUS	region, age, sex, race, marital_status, citizenship_status, birthplace, highest_level_of_school_completed	labor_force_status

ตารางที่ 6 การกำหนดคุณสมบัติในการสอนตัวจำแนกประเภท

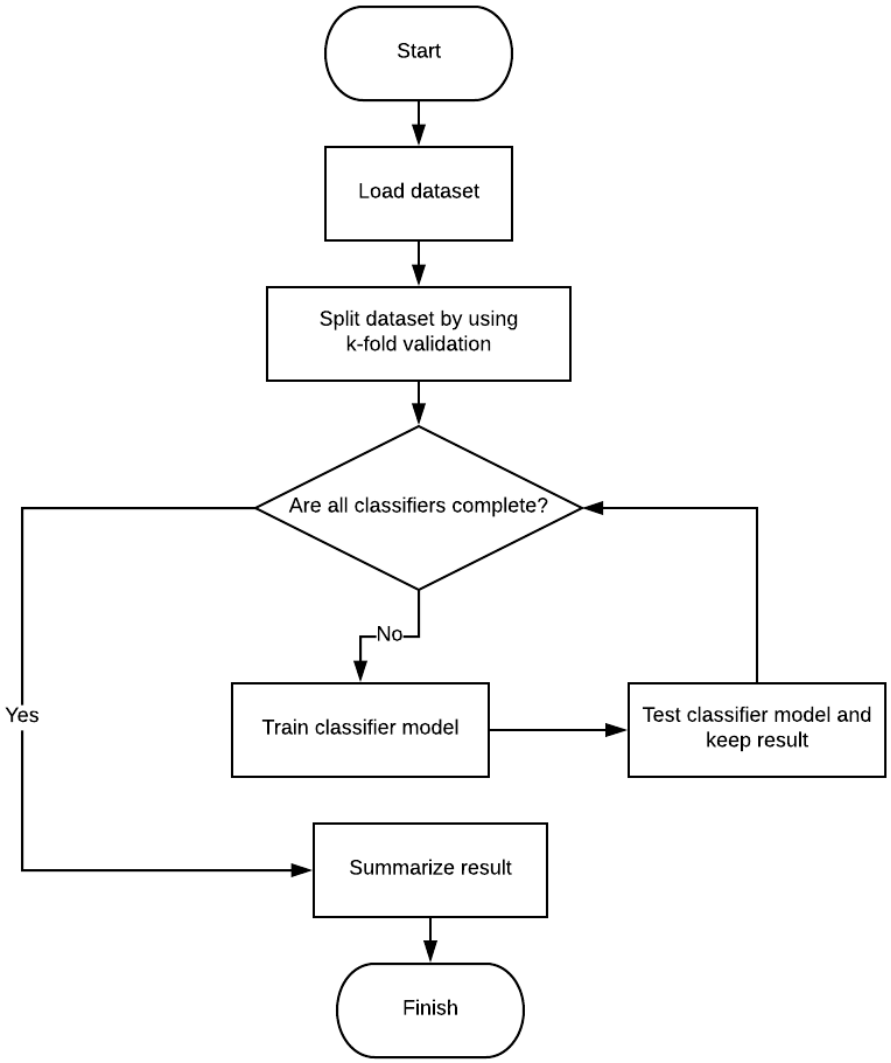
ในงานวิจัยชิ้นนี้จะเลือกใช้ Scikit-Learn library [19] มาใช้ในการสร้างโมเดลตัวจำแนกประเภท Scikit-Learn เป็น open source library สำหรับการเรียนรู้เครื่องที่ถูกพัฒนาด้วยภาษา Python โดย Scikit-Learn มีโมดูลให้เลือกใช้งานในกระบวนการที่เกี่ยวกับการเรียนรู้เครื่องต่างๆ เช่น การจำแนกประเภท (classification) การถดถอย (regression) การจับกลุ่ม (clustering) และการประมวลผลก่อน (data preprocessing) เป็นต้น อีกทั้งยังมีโมดูลการใช้งานกลุ่มตัวจำแนกประเภท (ensemble classifier) ให้เลือกใช้งาน เช่น Random forest , Bagging และ Voting model เป็นต้น

Scikit-Learn เป็น open source library ที่ถูกใช้ในการทำเหมืองข้อมูลอย่างแพร่หลาย เพราะ library ออกแบบมาให้ผู้ใช้งานสามารถใช้งานได้อย่างง่ายในการทำเหมืองข้อมูลและการวิเคราะห์ข้อมูล สามารถใช้ code ชุดเดียวกันในหลายๆการใช้งาน และสามารถใช้งานร่วมกับ library ที่หลากหลายเช่น NumPy, SciPy และ matplotlib เป็นต้น

```
# Decision tree
def decision_tree(x_train, y_train, x_test, y_test, p_table):
    start_time = timeit.default_timer()

    dtree = DecisionTreeClassifier()
    prediction_dt = dtree.fit(x_train,y_train).predict(x_test)

    p_add_row('decision tree', y_test, prediction_dt, start_time, p_table)


# Naive bayes
def naive_bayes(x_train, y_train, x_test, y_test, p_table):
    start_time = timeit .default_timer()

    gnb = GaussianNB()
    prediction_gnb = gnb.fit(x_train, y_train).predict(x_test)

    p_add_row('naive bayes', y_test, prediction_gnb, start_time, p_table)
```

ภาพที่ 11 ตัวอย่างการใช้ Scikit-Learn ในการสร้างตัวจำแนกประเภท

การทดลองจะใช้อัลกอริทึ่มกลุ่ม bagging, voting, random forest, adaboost และ rotation forest ในการทดสอบประสิทธิภาพความแม่นยำของการจำแนกประเภทกับข้อมูลที่ถูกปกปิด โดยขั้นตอนการสร้างโมเดลสามารถเขียนเป็น flow chart ได้ตามภาพที่ 12



ภาพที่ 12 flowchart การทำงานของการสร้างโมเดล

3.1.5 การวัดผล

การวัดผลในงานวิจัยชิ้นนี้จะมีมาตรวัดได้แก่ ความแม่นยำของตัวจำแนกประเภท อัตราความเสี่ยงจากการถูกระบุตัวตน และ จำนวนข้อมูลที่ถูกลบ

ความแม่นยำของตัวจำแนกประเภทจะทำการวัดโดยใช้ library ของ Scikit-Learn โดยจะใช้โมดูล model_selection ในการทำ K-fold Validation และใช้โมดูล metric ในการวัดผลความแม่นยำของตัวจำแนกประเภท

```
dataset/adult.csv
+----------------+----------+-----------------+
|     Model      | Accuracy |  Execute time   |
+----------------+----------+-----------------+
| random forest  |  0.7971  | 3.0814 seconds  |
|    bagging     |  0.8018  | 4.7456 seconds  |
|     voting     |  0.8239  | 9.0951 seconds  |
+----------------+----------+-----------------+
dataset/adult-k1.csv
+----------------+----------+-----------------+
|     Model      | Accuracy |  Execute time   |
+----------------+----------+-----------------+
| random forest  |  0.8196  |  1.2453 seconds |
|    bagging     |  0.7947  |  8.7440 seconds |
|     voting     |  0.8153  | 11.2228 seconds |
+----------------+----------+-----------------+
```

ภาพที่ 13 ตัวอย่างผลลัพธ์ความแม่นยำของโมเดลในการจำแนกประเภท

มาตรวัดที่ใช้ในการวัดความปลอดภัยของข้อมูลคือ อัตราความเสี่ยงจากการถูกระบุตัวตน (re-identification risk) โดยอัตราความเสี่ยงจากการถูกระบุตัวตนจะสามารถบอกความน่าจะเป็นที่ข้อมูลจำถูกระบุตัวตนสำเร็จได้ เราสามารถที่จะนำค่านี้ไว้ใช้ในการพิจารณาว่าชุดข้อมูลที่เราต้องการจะเผยแพร่มีความปลอดภัยของข้อมูลมากหรือน้อยเพียงใด

Re-identification risks | Population uniques | Quasi-identifiers

Measure	Value [%]
Lowest prosecutor risk	2.22222%
Records affected by lowest risk	0.14919%
Average prosecutor risk	64.65752%
Highest prosecutor risk	100%
Records affected by highest risk	51.42895%
Estimated prosecutor risk	100%
Estimated journalist risk	100%
Estimated marketer risk	64.65752%
Sample uniques	51.42895%
Population uniques	3.88962%
Population model	PITMAN

ภาพที่ 14 ตัวอย่างผลลัพธ์การวัดความปลอดภัยของข้อมูล

เนื่องจากการปกปิดข้อมูลจะต้องทำการเปลี่ยนแปลงข้อมูลเพื่อที่จะทำให้ชุดข้อมูลมีความปลอดภัยตามโมเดลความปลอดภัยที่เลือกใช้ ดังนั้นการปกปิดข้อมูลอาจจะทำให้บางข้อมูลในชุดข้อมูลจะต้องถูกลบออกไปในกรณีที่ข้อมูลนั้นไม่สามารถแปลงแล้วทำให้ชุดข้อมูลปลอดภัยตามโมเดลความปลอดภัย งานวิจัยชิ้นนี้จึงนำจำนวนของข้อมูลที่ถูกลบมาเป็นหนึ่งในมาตรวัด

เมื่อได้ผลลัพธ์ของการทดลองการจำแนกประเภทนำความแม่นยำของในจำแนกประเภทด้วยชุดข้อมูลดั้งเดิมและความแม่นยำในการจำแนกประเภทด้วยชุดข้อมูลที่ถูกปกปิดมาทำการทดสอบที (t-test) เพื่อที่จะนำมาสรุปว่าการใช้ชุดข้อมูลที่ถูกปกปิดในการจำแนกประเภททำให้ความแม่นยำของตัวจำแนกประเภทลดลงอย่างมีนัยยะสำคัญหรือไม่ จากนั้นนำผลลัพธ์ทั้งหมดที่ได้จากการทดลองมา

วิเคราะห์ว่าการปกปิดข้อมูลด้วยโมเดลความปลอดภัยชนิดใดและการจำแนกประเภทด้วยกลุ่มของตัวจำแนกประเภทชนิดใดที่เหมาะสมกับลักษณะการใช้งานตามความต้องการของเจ้าของข้อมูล

บทที่ 4
การออกแบบและการพัฒนาระบบ

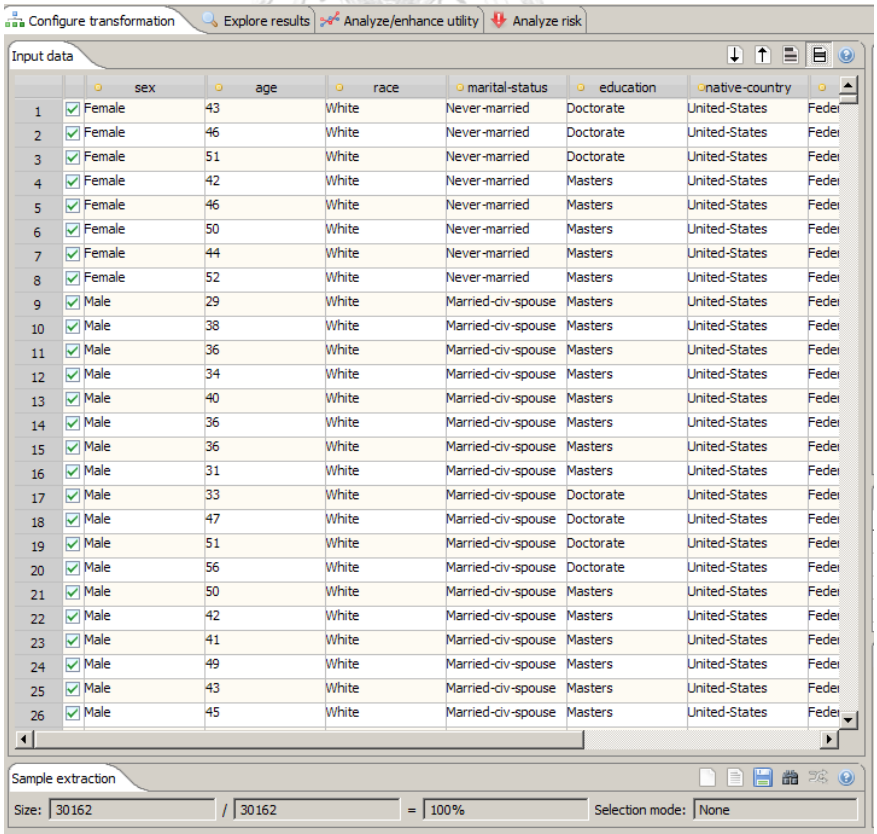
ขั้นตอนการดำเนินงานของการทดลองจะทำแต่ละขั้นตอนตามลำดับดังต่อไปนี้ กระบวนการปกปิดข้อมูล การจำแนกประเภท การแสดงผล และการวัดผล ตามลำดับ

4.1 การปกปิดข้อมูล

การทดลองในงานวิจัยชิ้นนี้จะทำการทดลองกับชุดข้อมูล ADULT, FARS และ ATUS โดยจะทำการปกปิดข้อมูลด้วยการใช้โมเดลความปลอดภัยและการตั้งค่าของโมเดลที่หลากหลาย ทำให้แต่ละชุดข้อมูลจะถูกแบ่งออกเป็น 17 ชุดย่อยตามตารางที่ 5 ซึ่งขั้นตอนของการปกปิดข้อมูลในแต่ละชุดข้อมูลด้วยโปรแกรม ARX มีดังต่อไปนี้

4.1.1 นำเข้าข้อมูลที่ต้องการ

เปิดโปรแกรม ARX เลือกเมนู File -> New project จากนั้น Import ข้อมูลโดยเลือกเมนู File -> Import data และเลือกชุดข้อมูลที่ต้องการ

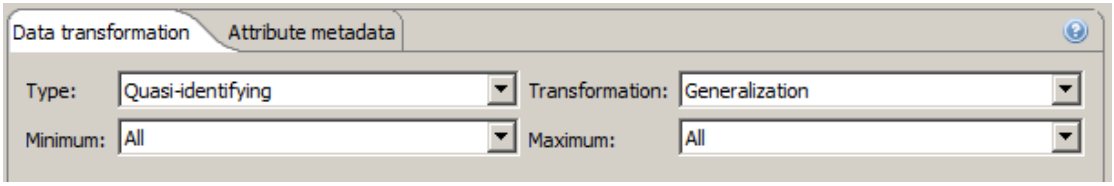


	sex	age	race	marital-status	education	native-country
1	Female	43	White	Never-married	Doctorate	United-States
2	Female	46	White	Never-married	Doctorate	United-States
3	Female	51	White	Never-married	Doctorate	United-States
4	Female	42	White	Never-married	Masters	United-States
5	Female	46	White	Never-married	Masters	United-States
6	Female	50	White	Never-married	Masters	United-States
7	Female	44	White	Never-married	Masters	United-States
8	Female	52	White	Never-married	Masters	United-States
9	Male	29	White	Married-civ-spouse	Masters	United-States
10	Male	38	White	Married-civ-spouse	Masters	United-States
11	Male	36	White	Married-civ-spouse	Masters	United-States
12	Male	34	White	Married-civ-spouse	Masters	United-States
13	Male	40	White	Married-civ-spouse	Masters	United-States
14	Male	36	White	Married-civ-spouse	Masters	United-States
15	Male	36	White	Married-civ-spouse	Masters	United-States
16	Male	31	White	Married-civ-spouse	Masters	United-States
17	Male	33	White	Married-civ-spouse	Doctorate	United-States
18	Male	47	White	Married-civ-spouse	Doctorate	United-States
19	Male	51	White	Married-civ-spouse	Doctorate	United-States
20	Male	56	White	Married-civ-spouse	Doctorate	United-States
21	Male	50	White	Married-civ-spouse	Masters	United-States
22	Male	42	White	Married-civ-spouse	Masters	United-States
23	Male	41	White	Married-civ-spouse	Masters	United-States
24	Male	49	White	Married-civ-spouse	Masters	United-States
25	Male	43	White	Married-civ-spouse	Masters	United-States
26	Male	45	White	Married-civ-spouse	Masters	United-States

ภาพที่ 15 การนำเข้าข้อมูล

4.1.2 เลือกประเภทของคุณสมบัติในแต่ละคุณสมบัติ

ทำการเลือกประเภทคุณสมบัติของแต่ละคุณสมบัติว่าเป็นประเภทไหน (Identifying, Quasi-identifying, Insensitive และ Sensitive) โดยคลิกที่แท็บ Data transformation



ภาพที่ 16 การเลือกประเภทของแต่ละคุณสมบัติ

4.1.3 สร้างการจัดระบบตามลำดับขั้นของแต่ละคุณสมบัติ

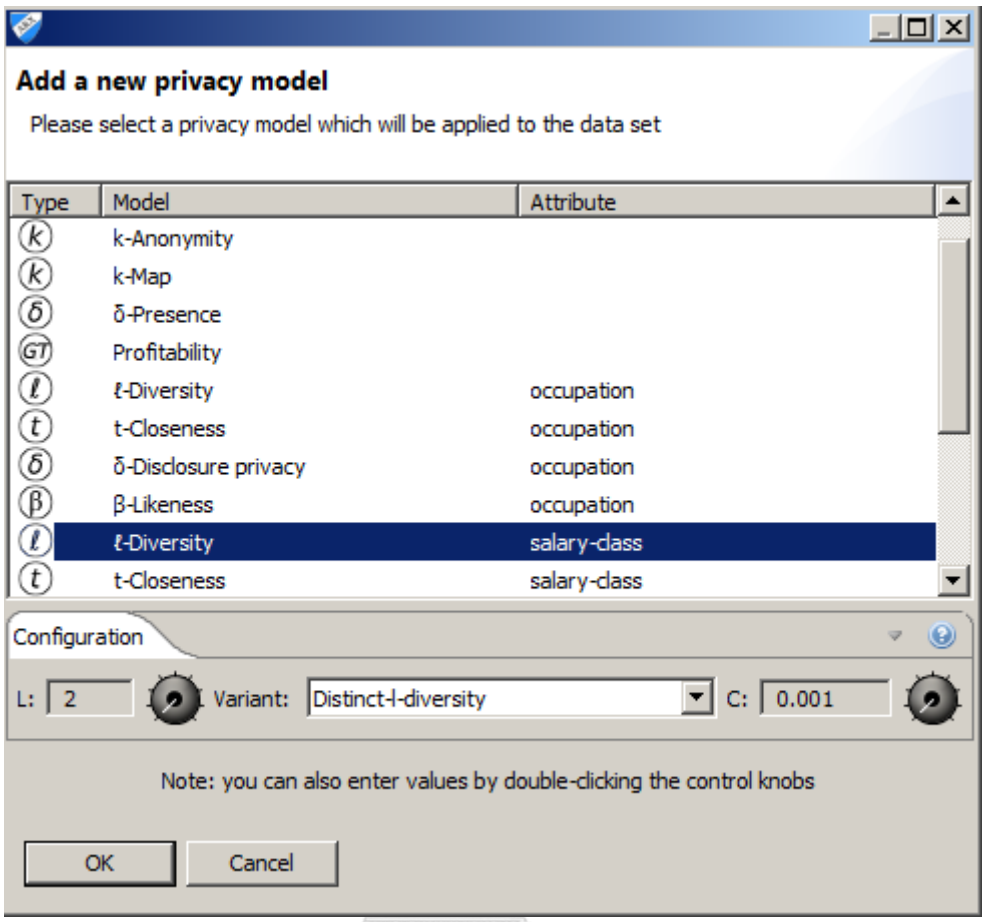
สร้างลำดับขั้นในการเปลี่ยนแปลงข้อมูล (hierarchy) ของคุณสมบัติโดยการเลือกเมนู Edit -> Create hierarchy และทำการกำหนดลำดับขั้นตามที่ต้องการ

Level-0	Level-1	Level-2	Level-3
Bachelors	Undergraduate	Higher education	*
Some-college	Undergraduate	Higher education	*
11th	High School	Secondary educa...	*
HS-grad	High School	Secondary educa...	*
Prof-school	Professional Educ...	Higher education	*
Assoc-acdm	Professional Educ...	Higher education	*
Assoc-voc	Professional Educ...	Higher education	*
9th	High School	Secondary educa...	*
7th-8th	High School	Secondary educa...	*
12th	High School	Secondary educa...	*
Masters	Graduate	Higher education	*
1st-4th	Primary School	Primary education	*
10th	High School	Secondary educa...	*
Doctorate	Graduate	Higher education	*
5th-6th	Primary School	Primary education	*
Preschool	Primary School	Primary education	*

ภาพที่ 17 ตัวอย่าง hierarchy ของคุณสมบัติ education

4.1.4 เลือกโมเดลความปลอดภัยที่ต้องการ

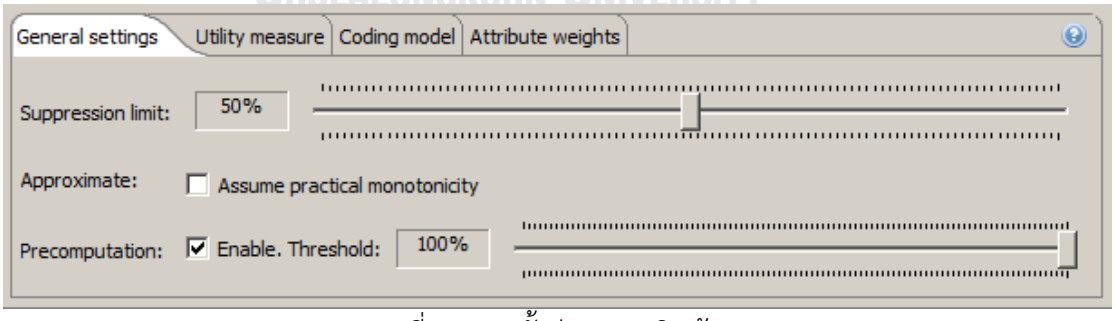
สามารถเลือกโมเดลความปลอดภัย (privacy model) ในการปกปิดข้อมูลโดยคลิกที่แท็บ Privacy models และคลิกปุ่ม Add privacy model จากนั้นเลือกโมเดลความปลอดภัยและตั้งค่าพารามิเตอร์ของโมเดลที่ต้องการใช้งาน



ภาพที่ 18 การเลือกใช้โมเดลความปลอดภัยในการปกปิดข้อมูล

4.1.5 การตั้งค่าการปกปิดข้อมูล

สามารถปรับแต่งตั้งค่าการปกปิดข้อมูลตามที่ต้องการโดยคลิกที่แท็บ General settings ในส่วนนี้จะสามารถปรับตั้งว่าจะให้การปกปิดข้อมูลมีลักษณะอย่างไร



ภาพที่ 19 การตั้งค่าการปกปิดข้อมูล

4.1.6 ทำการสั่งให้โปรแกรมทำงานเพื่อปกปิดข้อมูล

ทำการสั่งให้โปรแกรมทำงานเพื่อปกปิดข้อมูลโดยการเลือกเมนู Edit -> Anonymize

4.1.7 บันทึกข้อมูลที่ได้จากการปกปิดข้อมูล

บันทึกข้อมูลที่ได้จากการปกปิดข้อมูลโดยการเลือกเมนู File -> Export data

	sex	age	race	marital-status	education	native-country	workclass	occupation	salary-cl
1	Male	40-59	Asian-Pac-Islander	Married-civ-spouse	Higher education	United-States	Federal-gov	Prof-specialty	>50K
2	Male	40-59	Asian-Pac-Islander	Married-civ-spouse	Higher education	United-States	Federal-gov	Tech-support	>50K
3	Male	40-59	Asian-Pac-Islander	Married-civ-spouse	Higher education	United-States	Federal-gov	Exec-managerial	>50K
4	Male	40-59	Asian-Pac-Islander	Married-civ-spouse	Higher education	United-States	Federal-gov	Craft-repair	>50K
5	Male	40-59	Asian-Pac-Islander	Married-civ-spouse	Higher education	United-States	Federal-gov	Exec-managerial	>50K
6	Female	20-39	Black	Divorced	Higher education	United-States	Federal-gov	Handlers-cleaners	<=50K
7	Female	20-39	Black	Divorced	Higher education	United-States	Federal-gov	Exec-managerial	<=50K
8	Female	20-39	Black	Divorced	Higher education	United-States	Federal-gov	Adm-clerical	<=50K
9	Female	20-39	Black	Divorced	Higher education	United-States	Federal-gov	Exec-managerial	>50K
10	Female	20-39	Black	Divorced	Higher education	United-States	Federal-gov	Adm-clerical	<=50K
11	Female	20-39	Black	Divorced	Higher education	United-States	Federal-gov	Tech-support	<=50K
12	Female	20-39	Black	Divorced	Higher education	United-States	Federal-gov	Exec-managerial	<=50K
13	Female	20-39	Black	Divorced	Higher education	United-States	Federal-gov	Exec-managerial	<=50K
14	Female	20-39	Black	Divorced	Higher education	United-States	Federal-gov	Exec-managerial	<=50K
15	Female	20-39	Black	Divorced	Higher education	United-States	Federal-gov	Tech-support	<=50K
16	Female	20-39	Black	Divorced	Higher education	United-States	Federal-gov	Craft-repair	>50K
17	Female	20-39	Black	Never-married	Higher education	United-States	Federal-gov	Adm-clerical	<=50K
18	Female	20-39	Black	Never-married	Higher education	United-States	Federal-gov	Prof-specialty	<=50K
19	Female	20-39	Black	Never-married	Higher education	United-States	Federal-gov	Exec-managerial	<=50K
20	Female	20-39	Black	Never-married	Higher education	United-States	Federal-gov	Prof-specialty	<=50K
21	Female	20-39	Black	Never-married	Higher education	United-States	Federal-gov	Tech-support	<=50K
22	Female	20-39	Black	Never-married	Higher education	United-States	Federal-gov	Adm-clerical	<=50K
23	Female	20-39	Black	Never-married	Higher education	United-States	Federal-gov	Adm-clerical	<=50K
24	Female	20-39	Black	Never-married	Higher education	United-States	Federal-gov	Adm-clerical	<=50K
25	Female	20-39	Black	Never-married	Higher education	United-States	Federal-gov	Other-service	<=50K
26	Female	20-39	Black	Never-married	Higher education	United-States	Federal-gov	Adm-clerical	<=50K
27	Female	20-39	Black	Never-married	Higher education	United-States	Federal-gov	Adm-clerical	<=50K
28	Female	20-39	Black	Never-married	Higher education	United-States	Federal-gov	Adm-clerical	<=50K
29	Male	20-39	Black	Married-civ-spouse	Higher education	United-States	Federal-gov	Protective-serv	>50K

ภาพที่ 20 ตัวอย่างผลลัพธ์ชุดข้อมูลที่ถูกปกปิด

4.2 การจำแนกประเภท

หลังจากการปกปิดข้อมูลเราจะได้ชุดข้อมูล ADULT, FARS, ATUS ที่ถูกแบ่งออกเป็นชุดข้อมูลย่อย 17 ชุดตารางที่ 5 ทำให้มีชุดข้อมูลย่อยทั้งหมดในการทดสอบทั้งหมด 51 ชุด

4.2.1 การประมวลผลก่อน (Data preprocessing)

ทำการเข้ารหัสข้อมูลในทุกคุณสมบัติของทุกชุดข้อมูลย่อยในการทดลองเพื่อที่จะนำไปใช้ในการสร้างตัวจำแนกประเภท ซึ่งจะทำให้ใช้พื้นที่ในการเก็บข้อมูลน้อยลงเช่นเดียวกัน การเข้ารหัสข้อมูลจะใช้แบบ label encoding โดยใช้โมดูล preprocessing จาก library ของ Scikit-learn

```python
from sklearn import preprocessing

le = preprocessing.LabelEncoder()
i = 0


class MultiColumnLabelEncoder:
    def __init__(self, columns=None):
        self.columns = columns

    def fit(self, X, y=None):
        return self

    def transform(self, X):
        output = X.copy()
        if self.columns is not None:
            for col in self.columns:
                output[col] = LabelEncoder().fit_transform(output[col].astype(str))
        else:
            for col_name, col in output.iteritems():
                output[col_name] = LabelEncoder().fit_transform(col)
        return output

    def fit_transform(self, X, y=None):
        return self.fit(X, y).transform(X)
```

ภาพที่ 21 ตัวอย่าง code ของการเข้ารหัส

4.2.2 การทำ Cross-validation

การทดลองจะใช้ k-fold Cross-Validation ในการแบ่งข้อมูลในการสอนโมเดลตัวจำแนกประเภท โดยจะให้ค่า k = 5 ประโยชน์ของการทำ Cross-Validation คือจะทำให้ตัวจำแนกประเภทมีความเหมาะสมกับลักษณะทั่วไป (generalization) ซึ่งจะทำให้ลดโอกาสการเกิดปัญหา overfitting ได้ โดยการทำ k-fold Cross-Validation สามารถใช้งานจากโมดูล model_selection ใน Scikit-Learn

4.2.3 การสอนตัวจำแนกประเภท

ในขั้นตอนการสอนตัวจำแนกประเภทจะทำการเรียนรู้กับทุกชุดข้อมูลย่อยที่เป็นผลลัพธ์จากการปกปิดข้อมูลในขั้นตอนก่อนหน้า โดยจะทำการเลือกคุณสมบัติในการสอนตัวจำแนกประเภทตามตารางที่ 6 งานวิจัยชิ้นนี้จะทำการใช้อัลกอริทึ่มกลุ่มตัวจำแนกประเภท Random Forest, Bagging, AdaBoost, Voting และ Rotation Forest ซึ่งการใช้อัลกอริทึ่มของตัวจำแนกประเภทจำเป็นที่จะต้องตั้งค่าพารามิเตอร์ในการสอนตัวจำแนกประเภท การทดลองจะทำการตั้งค่าพารามิเตอร์ตามตารางที่ 7

Algorithm	Classifier parameters
Random Forest	n_estimators=50, max_depth=None, min_samples_split=2, random_state=0, bootstrap=False
Bagging	Decistion Tree, n_estimators=50, max_samples=0.5, max_features=0.5
AdaBoost	n_estimators=50
Voting	estimators=[‘random forest’, ’bagging’, ’adaboost’], voting=’hard’
Rotation Forest	random_state=1234

ตารางที่ 7 การตั้งค่าพารามิเตอร์ในการสอนตัวจำแนกประเภท

```
print('random forest start')
# random forest
start_time = timeit.default_timer()

rfc = RandomForestClassifier(n_estimators=50, max_depth=None, min_samples_split=2, random_state=0, bootstrap=False)
prediction_rf = rfc.fit(x_train, y_train).predict(x_test)

acc_rf = accuracy_score(y_test, prediction_rf)
precision_rf, recall_rf, f_score_rf, support_rf = precision_recall_fscore_support(y_test, prediction_rf, average='weighted')
time_rf = format_float(timeit.default_timer() - start_time)

print('bagging start')
# bagging
start_time = timeit.default_timer()

bagging = BaggingClassifier(DecisionTreeClassifier(), n_estimators=50, max_samples=0.5, max_features=0.5)
prediction_bg = bagging.fit(x_train, y_train).predict(x_test)

acc_bag = accuracy_score(y_test, prediction_bg)
precision_bag, recall_bag, f_score_bag, support_bag = precision_recall_fscore_support(y_test, prediction_bg, average='weighted')
time_bag = format_float(timeit.default_timer() - start_time)

print('adaboost start')
# adaboost
start_time = timeit.default_timer()

adaboost = AdaBoostClassifier(n_estimators=50)
prediction_ada = adaboost.fit(x_train, y_train).predict(x_test)

acc_ada = accuracy_score(y_test, prediction_ada)
precision_ada, recall_ada, f_score_ada, support_ada = precision_recall_fscore_support(y_test, prediction_ada, average='weighted')
time_ada = format_float(timeit.default_timer() - start_time)

print('voting start')
# voting
start_time = timeit.default_timer()

voting = VotingClassifier(estimators=[('clf1', rfc), ('clf2', bagging), ('clf3', adaboost)], voting='hard')
prediction_vt = voting.fit(x_train, y_train).predict(x_test)

acc_vt = accuracy_score(y_test, prediction_vt)
precision_vt, recall_vt, f_score_vt, support_vt = precision_recall_fscore_support(y_test, prediction_vt, average='weighted')
time_vt = format_float(timeit.default_timer() - start_time)

print('rotation forest start')
# rotation forest
start_time = timeit.default_timer()

rotation = RotationForestClassifier(random_state=1234)
prediction_rot = rotation.fit(x_train, y_train).predict(x_test)

acc_rof = accuracy_score(y_test, prediction_rot)
precision_rof, recall_rof, f_score_rof, support_rof = precision_recall_fscore_support(y_test, prediction_rot, average='weighted')
time_rof = format_float(timeit.default_timer() - start_time)
```

ภาพที่ 22 การสร้างตัวจำแนกประเภทด้วยอัลกอริทึ่มกลุ่มตัวจำแนกประเภท

บทที่ 5
การประเมินและการวัดผล

งานวิจัยชิ้นนีจะใช้มาตรวัดในการวัดประสิทธิภาพของการทำเหมืองข้อมูลด้วยข้อมูลที่ถูกปกปิดคือ ความแม่นยำของตัวจำแนกประเภท อัตราความเสี่ยงจากการถูกระบุตัวตน และ จำนวนข้อมูลที่ถูกลบ โดยที่งานวิจัยชิ้นนี้ต้องการที่จะศึกษาว่าประสิทธิภาพจากของการทำเหมืองข้อมูลด้วยข้อมูลที่ถูกปกปิดน้อยลงหรือไม่เมื่อเทียบกับการทำเหมืองข้อมูลด้วยข้อมูลดั้งเดิม และทำการประเมินประสิทธิภาพของโมเดลความปลอดภัยและอัลกอริทึ่มกลุ่มตัวจำแนกประเภทในแต่ละชนิดเพื่อให้สามารถเลือกใช้งานได้ตามความเหมาะสม

เนื่องจากในการวัดความแม่นยำของตัวจำแนกประเภท และอัตราความเสี่ยงจากการถูกระบุตัวตนจะวัดจากผลลัพธ์ที่ได้จากชุดข้อมูลย่อยต่างๆ ดังนั้นเราจำเป็นที่จะต้องทำการหาค่าเฉลี่ย ในงานวิจัยชิ้นนี้เราจะใช้ค่าเฉลี่ยเรขาคณิต (geometric mean) ในการหาค่าเฉลี่ย เนื่องจากค่าเฉลี่ยเรขาคณิตจะเหมาะสมกับการเป็นมาตรวัดในข้อมูลที่มีการเพิ่มแบบชี้กำลัง (exponential growth) และมีค่าการกระจายแบบเบ้ (skewed distribution) ทำให้ค่าเฉลี่ยเรขาคณิตจึงถูกทำไปใช้บ่อยในการวิเคราะห์การวัดผล

```
attributes = ['sex', 'age', 'race', 'marital-status', 'education', 'native-country', 'workclass', 'occupation', 'salary-class']
dropped_attributes = []
class_label_attribute = 'salary-class'
columns = {'attributes': attributes, 'dropped_attributes': dropped_attributes, 'class_label_attribute': class_label_attribute}
ADULT_result = list()

ADULT_result.append(my_classify('data/adult/adult-original.csv', columns, data_info_original))
ADULT_result.append(my_classify('data/adult/adult-k1.csv', columns, data_info_k1))
ADULT_result.append(my_classify('data/adult/adult-k2.csv', columns, data_info_k2))
ADULT_result.append(my_classify('data/adult/adult-k3.csv', columns, data_info_k3))
ADULT_result.append(my_classify('data/adult/adult-l1.csv', columns, data_info_l1))
ADULT_result.append(my_classify('data/adult/adult-l2.csv', columns, data_info_l2))
ADULT_result.append(my_classify('data/adult/adult-l3.csv', columns, data_info_l3))
ADULT_result.append(my_classify('data/adult/adult-t1.csv', columns, data_info_t1))
ADULT_result.append(my_classify('data/adult/adult-t2.csv', columns, data_info_t2))
ADULT_result.append(my_classify('data/adult/adult-t3.csv', columns, data_info_t3))
ADULT_result.append(my_classify('data/adult/adult-dp1.csv', columns, data_info_dp1))
ADULT_result.append(my_classify('data/adult/adult-dp2.csv', columns, data_info_dp2))
ADULT_result.append(my_classify('data/adult/adult-dp3.csv', columns, data_info_dp3))
ADULT_result.append(my_classify('data/adult/adult-pu1.csv', columns, data_info_pu1))
ADULT_result.append(my_classify('data/adult/adult-pu2.csv', columns, data_info_pu2))
ADULT_result.append(my_classify('data/adult/adult-pu3.csv', columns, data_info_pu3))
ADULT_result.append(my_classify('data/adult/adult-km1.csv', columns, data_info_km1))
ADULT_result.append(my_classify('data/adult/adult-km2.csv', columns, data_info_km2))
ADULT_result.append(my_classify('data/adult/adult-km3.csv', columns, data_info_km3))

write_json_to_file('result/adult/result.json', ADULT_result)
ADULT_summary = summarize_result(ADULT_result)
```

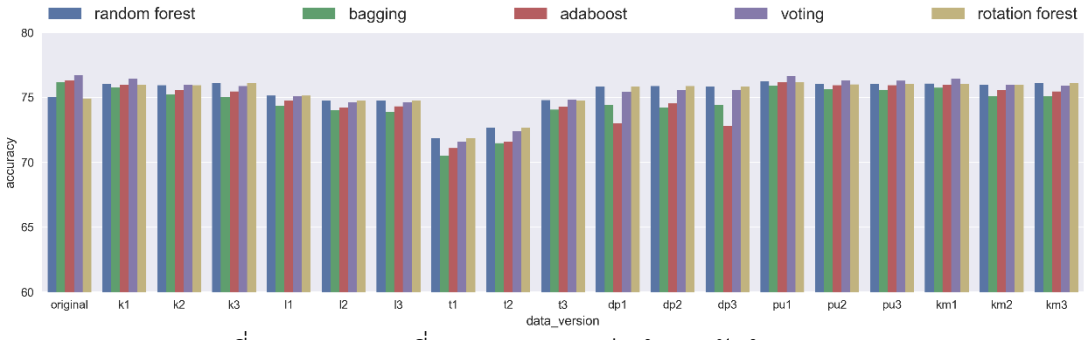
ภาพที่ 23 ตัวอย่าง code ในการสอนตัวจำแนกประเภทด้วยชุดข้อมูลย่อยต่างๆ

ความแม่นยำของตัวจำแนกประเภทจะถูกวัดในขั้นตอนการสอนตัวจำแนกประเภทและอัตราความเสี่ยงจากการถูกระบุตัวตนจะถูกวัดในขั้นตอนการปกปิดข้อมูล ภาพที่ 24 คือตัวอย่างผลลัพธ์ที่ได้จากการทดลองในแต่ละชุดข้อมูลย่อยซึ่งจะประกอบไปด้วย ความแม่นยำของตัวจำแนกประเภทและเวลาในการสอนในแต่ละอัลกอริทึ่ม อัตราความเสี่ยงจากการถูกระบุตัวตน และอัตราจำนวนข้อมูลที่ถูกลบ

```
{
  "index": 1,
  "file_name": "data/adult/adult-k1.csv",
  "classifiers_output": {
    "rf": {
      "accuracy_score": "0.8149",
      "precision": "0.8062",
      "recall": "0.8149",
      "f1_score": "0.8083",
      "execute_time": "3.1828"
    },
    "bg": {
      "accuracy_score": "0.8151",
      "precision": "0.8075",
      "recall": "0.8151",
      "f1_score": "0.7926",
      "execute_time": "1.4414"
    },
    "ada": {
      "accuracy_score": "0.8260",
      "precision": "0.8179",
      "recall": "0.8260",
      "f1_score": "0.8194",
      "execute_time": "1.7627"
    },
    "vt": {
      "accuracy_score": "0.8259",
      "precision": "0.8166",
      "recall": "0.8259",
      "f1_score": "0.8147",
      "execute_time": "6.6472"
    },
    "rof": {
      "accuracy_score": "0.8135",
      "precision": "0.8049",
      "recall": "0.8135",
      "f1_score": "0.8072",
      "execute_time": "2.3659"
    }
  },
  "data_info": {
    "privacy_level": {
      "record_risk": "0.0868",
      "highest_risk": "0.5000",
      "success_rate": "0.0582"
    },
    "suppressed_percent": "0.0816"
  }
},
```


ภาพที่ 24 ตัวอย่างผลลัพธ์ที่ได้จากการทดลอง

นำผลลัพธ์ความแม่นยำของตัวจำแนกประเภทในแต่ละอัลกอริทึ่มที่ได้จากชุดข้อมูลย่อยต่างๆ มาแสดงผลด้วยกราฟแท่ง (bar plot) เพื่อที่จะเห็นประสิทธิภาพของตัวจำแนกประเภทโดยรวมในแต่ละอัลกอริทึ่ม สามารถแสดงผลกราฟแท่งดังภาพที่ 25



ภาพที่ 25 bar plot ที่แสดงผลความแม่นยำของตัวจำแนกประเภท

Rank	Dataset			
	ADULT	FARS	ATUS	Average
1	vt (81.84%)	rf (63.19%)	vt (83.45%)	rf (75.32%)
2	ada (81.79%)	rof (63.19%)	ada (83.34%)	rof (75.30%)
3	rf (81.32%)	bg (62.57%)	rf (83.14%)	vt (75.29%)
4	rof (81.28%)	vt (62.49%)	rof (83.12%)	ada (74.57%)
5	bg (80.10%)	ada (60.83%)	bg (82.32%)	bg (74.57%)

ตารางที่ 8 ผลลัพธ์ความแม่นยำของการจำแนกประเภท

ตารางที่ 8 คือผลลัพธ์ความแม่นยำของการจำแนกประเภทของแต่ละอัลกอริทึ่มที่เฉลี่ยจากชุดข้อมูลย่อยในแต่ละชุดข้อมูล โดยที่

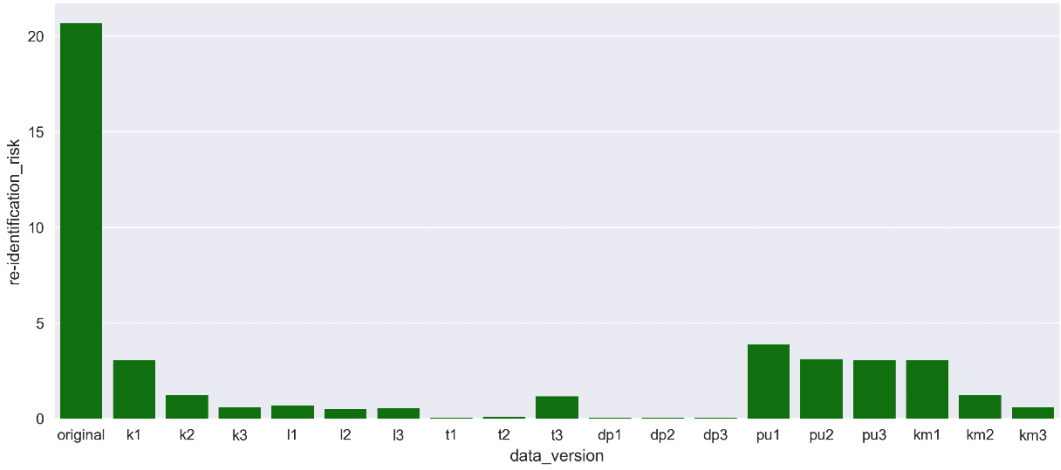
- rf คืออัลกอริทึ่ม Random Forest
- bg คืออัลกอริทึ่ม Bagging
- ada คืออัลกอริทึ่ม AdaBoost
- vt คืออัลกอริทึ่ม Voting
- rof คืออัลกอริทึ่ม Rotation Forest

จากตารางจะสามารถสรุปได้ว่าในชุดข้อมูล ADULT และ ATUS อัลกอริทึ่ม Voting จะให้ความแม่นยำในการจำแนกประเภทมากที่สุด ส่วนชุดข้อมูล FARS อัลกอริทึ่ม Random Forest จะให้ความแม่นยำในการจำแนกประเภทมากที่สุด โดยเมื่อนำผลลัพธ์ที่ได้จากทุกชุดข้อมูลมาหาค่าเฉลี่ยจะเห็นว่าประสิทธิภาพของแต่ละอัลกอริทึ่มแทบจะไม่ต่างกัน อัลกอริทึ่มที่ให้ความแม่นยำในการจำแนกประเภทมากที่สุดคือ Random Forest (75.32%) และอัลกอริทึ่มที่ให้ความแม่นยำในการจำแนกประเภทน้อยที่สุดคือ Bagging (74.57%)

Dataset	Data type	
	Original data	Anonymized data
ADULT	81.39%	81.28%
FARS	64.44%	62.49%
ATUS	83.15%	83.08%
Mean	76.33%	75.62%
S.D.	10.33%	11.40%
p-value	0.1854	

ตารางที่ 9 ผลลัพธ์การทดสอบที (t-test)

เพื่อที่จะทำการประเมินประสิทธิภาพของการจำแนกประเภทด้วยข้อมูลที่ถูกปกปิดกับประสิทธิภาพของการจำแนกประเภทด้วยข้อมูลดั้งเดิม งานวิจัยชิ้นนี้จะทำการวิเคราะห์จากผลลัพธ์การทดสอบที (t-test) ตารางที่ 9 คือผลลัพธ์ที่ได้จากการทดสอบที โดยมีค่าพี (p-value) เท่ากับ 0.1854 ซึ่งจะสามารถสรุปจากค่าพีที่ได้ว่าประสิทธิภาพของการจำแนกประเภทด้วยข้อมูลที่ถูกปกปิด (mean=75.62%, SD=11.41%) ลดลงจากประสิทธิภาพของการจำแนกประเภทด้วยข้อมูลดั้งเดิม (mean=76.33%, SD=10.33%) อย่างไม่มีนัยยะสำคัญ



ภาพที่ 26 bar plot ที่แสดงอัตราความเสี่ยงจากการถูกระบุตัวตน

ภาพที่ 26 คือ bar plot ที่ทำการแสดงอัตราความเสี่ยงจากการถูกระบุตัวตน โดยนำค่าที่ได้จากค่าเฉลี่ยอัตราความเสี่ยงจากการถูกระบุตัวตนของการปกปิดข้อมูลย่อยในแต่ละรูปแบบจากชุดข้อมูลทั้งสามชุด จะเห็นได้ว่าค่าเฉลี่ยอัตราความเสี่ยงจากการถูกระบุตัวตนของข้อมูลดั้งเดิมจะอยู่ที่ 20.68% และค่าเฉลี่ยอัตราความเสี่ยงจากการถูกระบุตัวตนของข้อมูลที่ถูกปกปิดจะอยู่ในช่วง 0.03% ถึง 3.90% โดยชุดข้อมูลย่อยที่ถูกปกปิดข้อมูลด้วยโมเดลความปลอดภัย t-closeness โดยมี

พารามิเตอร์ t = 0.2 (EMD with equal ground-distance) หรือชุดข้อมูลย่อย t1 จะให้อัตราความเสี่ยงจากการถูกระบุตัวตนน้อยที่สุด และชุดข้อมูลย่อยที่ถูกปกปิดข้อมูลด้วยโมเดลความปลอดภัย population uniqueness โดยมีพารามิเตอร์ threshold = 10^{-4} หรือชุดข้อมูลย่อย pu1 จะให้อัตราความเสี่ยงจากการถูกระบุตัวตนมากที่สุด

บทที่ 6
สรุปผลการวิจัย

ในบทนี้จะทำการสรุปผลลัพธ์และวิเคราะห์ผลลัพธ์ที่ได้จากการทดลองว่าประสิทธิภาพของการจำแนกประเภทด้วยข้อมูลที่ถูกปกปิดมีประสิทธิภาพเพียงพอที่จะใช้แทนข้อมูลดั้งเดิมหรือไม่ และประเมินประสิทธิภาพของโมเดลความปลอดภัยในการปกปิดข้อมูลและอัลกอริทึ่มกลุ่มตัวจำแนกประเภทที่ใช้กับข้อมูลที่ถูกปกปิด

Data version	Experiment result						
	Algorithm	Accuracy	Precision	Recall	F1 score	Risk	Suppressed
original	vt	76.70%	81.69%	82.68%	81.45%	20.68%	0%
k1	vt	76.44%	81.66%	82.59%	81.47%	3.06%	2.83%
k2	vt	75.95%	81.30%	82.26%	81.34%	1.23%	4.06%
k3	rf	76.10%	80.95%	81.69%	81.17%	0.58%	4.31%
l1	rf	75.13%	80.64%	81.35%	80.85%	0.69%	5.70%
l2	rf	74.75%	80.80%	81.51%	81.03%	0.51%	6.99%
l3	rof	74.74%	80.51%	81.21%	80.73%	0.55%	7.42%
t1	rf	71.83%	78.18%	78.88%	78.34%	0.03%	20.58%
t2	rf	72.63%	77.81%	78.38%	77.98%	0.10%	16.40%
t3	vt	74.81%	81.40%	82.35%	81.42%	1.17%	7.73%
dp1	rf	75.83%	81.14%	82.04%	81.27%	0.06%	4.97%
dp2	rf	75.87%	81.53%	82.38%	81.64%	0.06%	5.65%
dp3	rf	75.85%	81.81%	82.56%	81.97%	0.05%	6.71%
pu1	vt	76.64%	81.83%	82.74%	81.57%	3.90%	1.37%
pu2	vt	76.31%	81.59%	82.53%	81.40%	3.11%	2.78%
pu3	vt	76.27%	81.67%	82.60%	81.42%	3.06%	2.83%
km1	vt	76.46%	81.77%	82.70%	81.61%	3.06%	2.83%
km2	vt	75.96%	81.20%	82.18%	81.24%	1.23%	4.06%
km3	rf	76.09%	80.86%	81.58%	81.07%	0.58%	4.31%

ตารางที่ 10 ตารางสรุปผลการทดลอง

ตารางที่ 10 คือตารางที่ทำการสรุปผลลัพธ์ accuracy precision recall f1-score ของการจำแนกประเภท อัตราความเสี่ยงจากการถูกระบุตัวตน และ จำนวนข้อมูลที่ถูกลบ โดยทำการเฉลี่ยจากทั้งสามชุดข้อมูลที่ได้แบ่งเป็นชุดข้อมูลย่อยและทำการแสดงอัลกอริทึ่มกลุ่มตัวจำแนกประเภทที่ให้ความแม่นยำในการจำแนกประเภทมากที่สุด

จากผลการทดลองการจำแนกประเภทด้วยอัลกอริทึ่มกลุ่มตัวจำแนกประเภทในแต่ละอัลกอริทึ่มจะให้ค่าเฉลี่ยผลลัพธ์ที่แตกต่างกันเพียงแค่เล็กน้อย โดยผลการทดลองการจำแนกประเภทด้วยข้อมูลดั้งเดิม อัลกอริทึ่ม Voting จะให้ความแม่นยำในการจำแนกประเภทที่ดีที่สุด ส่วนการจำแนกประเภทด้วยข้อมูลที่ปกปิดจะสามารถสรุปได้ว่า

- อัลกอริทึ่ม Random Forest จะให้ความแม่นยำของการจำแนกประเภทมากที่สุดเมื่อปกปิดข้อมูลด้วยโมเดลความปลอดภัย t-closeness k-map และ SafePub
- อัลกอริทึ่ม Voting จะให้ความแม่นยำของการจำแนกประเภทมากที่สุดเมื่อปกปิดข้อมูลด้วยโมเดลความปลอดภัย k-anonymity และ population uniqueness
- อัลกอริทึ่ม Rotation Forest จะให้ความแม่นยำของการจำแนกประเภทมากที่สุดเมื่อปกปิดข้อมูลด้วยโมเดลความปลอดภัย l-diversity
- จากค่าเฉลี่ยจะสามารถสรุปได้ว่า อัลกอริทึ่ม Random Forest จะให้ความแม่นยำของการจำแนกประเภทมากที่สุดในการจำแนกประเภทด้วยข้อมูลที่ถูกปกปิด

Dataset	Algorithm type	
	Single Classifiers	Emsemble Classifiers
ADULT	76.99%	81.26%
FARS	61.59%	62.45%
ATUS	75.73%	83.07%
Mean	71.44%	75.59%
S.D.	8.55%	11.42%
p-value	0.0782	

ตารางที่ 11 ตารางการเปรียบเทียบประสิทธิภาพอัลกอริทึ่มในแต่ละชนิด

งานวิจัยชิ้นนี้เลือกใช้อัลกอริทึ่มกลุ่มตัวจำแนกประเภทในการทดลองด้วยเหตุผลที่ว่าอัลกอริทึ่มกลุ่มตัวจำแนกประเภทสามารถช่วยทำให้โมเดลที่สร้างมีความเหมาะสมกับลักษณะชุดข้อมูลต่างๆ (Generalizability) และมีความทนทาน (Robustness) เพราะการใช้กลุ่มของการจำแนกประเภทจะทำให้ค่าความแปรปรวนและความเอนเอียงของตัวจำแนกประเภทลดลง ตารางที่ 11 คือการเปรียบเทียบความแม่นยำในการจำแนกประเภทระหว่างการใช้กลุ่มตัวจำแนกประเภทและตัวจำแนกประเภทแบบเดี่ยว โดยกลุ่มของตัวจำแนกประเภทจะใช้อัลกอริทึ่ม Random Forest, Bagging, AdaBoost, Voting และ Rotation Forest ส่วนตัวจำแนกประเภทแบบเดียวจะใช้อัลกอริทึ่ม Naïve Bayes, Decision Tree และ Linear Regression ซึ่งผลลัพธ์ที่ได้คือความแม่นยำในการจำแนก

ประเภทโดยใช้อัลกอริทึ่มตัวจำแนกประเภทแบบเดี่ยวมีค่าเฉลี่ยอยู่ที่ 71.44% และความแม่นยำในการจำแนกประเภทโดยใช้อัลกอริทึ่มกลุ่มตัวจำแนกประเภทมีค่าเฉลี่ยอยู่ที่ 75.59% เมื่อพิจารณาค่าพี (p-value) จากการทดสอบที (t-test) ที่เท่ากับ 0.0782 จึงสามารถสรุปได้ว่าอัลกอริทึ่มกลุ่มตัวจำแนกประเภทมีความแม่นยำต่างจากอัลกอริทึ่มตัวจำแนกประเภทแบบเดี่ยวอย่างมีนัยยะสำคัญ (significant level) ที่ 0.01

การใช้โมเดลความปลอดภัยและการตั้งค่าการปกปิดข้อมูลที่ต่างกันในการปกปิดข้อมูลจะส่งผลให้ลักษณะของชุดข้อมูลที่ได้มีความแตกต่างกัน จากผลการทดลองจะเห็นได้ว่าโมเดลความปลอดภัย t-closeness จะให้อัตราความเสี่ยงจากการถูกระบุตัวตนน้อยที่สุด ทำให้ข้อมูลมีความปลอดภัยจากการถูกโจมตีด้วยการระบุตัวตนมากที่สุด แต่ชุดข้อมูลย่อยที่ได้ก็จะจำเป็นจะต้องถูกลบออกจากชุดข้อมูลมากที่สุดเช่นเดียวกัน จึงทำให้ความแม่นยำของการจำแนกประเภทก็น้อยที่สุด ดังนั้นการเลือกใช้งานโมเดลความปลอดภัยในการปกปิดข้อมูลควรที่จะเลือกใช้งานตามความต้องการของเจ้าของข้อมูล เช่นถ้าเจ้าของข้อมูลต้องการที่จะรักษาความปลอดภัยของข้อมูลให้ได้มากที่สุด เราก็ควรที่จะใช้โมเดลความปลอดภัย t-closeness หรือกรณีที่เจ้าของข้อมูลต้องการที่จะให้ข้อมูลมีประสิทธิภาพมากที่สุด เราก็ควรจะใช้โมเดลความปลอดภัย k-anonymity หรือ population uniqueness เป็นต้น

งานวิจัยชิ้นนี้ได้ทำการทดสอบที (t-test) ระหว่างความแม่นยำของการจำแนกประเภทด้วยข้อมูลดั้งเดิมและความแม่นยำของการจำแนกประเภทด้วยข้อมูลที่ถูกปกปิด ผลลัพธ์ค่าพีที่ได้จากการทดสอบทีเท่ากับ 0.1854 จึงสามารถสรุปได้ว่าประสิทธิภาพการจำแนกประเภทด้วยข้อมูลที่ถูกปกปิดลดลงจากการจำแนกประเภทด้วยข้อมูลดั้งเดิมอย่างไม่มีนัยยะสำคัญ

งานวิจัยชิ้นนี้จึงแนะนำให้เจ้าของข้อมูลทำการปกปิดข้อมูลก่อนที่จะเผยแพร่ให้บุคคลอื่นใช้งาน เพราะการทำเพียงแค่นำคุณสมบัติที่สามารถจะระบุตัวตนบุคคล (Explicit identifiers) ออกจากชุดข้อมูลไม่สามารถทำให้ลดความเสี่ยงจากการถูกระบุตัวตนได้เพียงพอ และเมื่อข้อมูลบุคคลสามารถถูกระบุตัวตนได้อาจจะทำให้เกิดความเสียหายที่ประเมินค่าไม่ได้ ดังนั้นเราควรที่จะให้ความสำคัญกับการปกปิดข้อมูล

รายการอ้างอิง

1. Zhang, C. and Y. Ma, *Ensemble Machine Learning: Methods and Applications*. 2012: Springer Publishing Company, Incorporated
2. *http://slideplayer.com/slide/9261331/*.
3. Breiman, L., *Bagging predictors.* Mach. Learn., 1996. **24**: p. 123-140.
4. Bauer, E. and R. Kohavi, *An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants.* Mach. Learn., 1999. **36**: p. 105-139.
5. Ho, T.K., *The random subspace method for constructing decision forests.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 1998. **20**: p. 832-844.
6. Freund, Y. and R.E. Schapire, *Experiments with a new boosting algorithm*, in *Proceedings of the Thirteenth International Conference on International Conference on Machine Learning*. 1996: Bari, Italy. p. 148-156.
7. Rodriguez, J.J., L.I. Kuncheva, and C.J. Alonso, *Rotation Forest: A New Classifier Ensemble Method.* IEEE Trans. Pattern Anal. Mach. Intell., 2006. **28**: p. 1619-1630.
8. Senosi, A. and G. Sibiya, *Classification and Evaluation of Privacy Preserving Data Mining: A Review*, in *IEEE Africon 2017 Proceedings*. 2017: Cape Town. p. 849-855.
9. Sweeney, L., *k-ANONYMITY: A MODEL FOR PROTECTING PRIVACY.* International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, 2002: p. 557-570.
10. Machanavajjhala, A., et al., *L-diversity: Privacy beyond k-anonymity.* ACM Trans. Knowl. Discov. Data, 2007. **1**: p. 3.
11. Li, N., T. Li, and S. Venkatasubramanian, *t-Closeness: Privacy Beyond k-Anonymity and l-Diversity*, in *t-Closeness: Privacy Beyond k-Anonymity and l-Diversity*. 2007: Istanbul. p. 106-115.

12. K.Saranya, K.Premalatha, and S.S.Rajasekar, *A Survey on Privacy Preserving Data Mining*, in *IEEE SPONSORED 2ND INTERNATIONAL CONFERENCE ON ELECTRONICS AND COMMUNICATION SYSTEM (ICECS 2015)*. 2015: Coimbatore. p. 1740-1744.
13. Samarati, P. and L. Sweeney, *Generalizing data to provide anonymity when disclosing information (abstract)*, in *Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*. 1998: Seattle, Washington, USA. p. 188.
14. Prasser, F., F. Kohlmayer, and K.A. Kuhn, *A Benchmark of Globally-Optimal Anonymization Methods for Biomedical Data*, in *IEEE 27th International Symposium on Computer-Based Medical Systems*. 2014. p. 66-71.
15. Buratović, I., M. Miličević, and K. Žubrinić, *Effects of Data Anonymization on the Data Mining Results*, in *Proceedings of the 35th International Convention MIPRO*. 2012: Opatija, Croatia. p. 1619-1623.
16. Inan, A., M. Kantarcioglu, and E. Bertino, *Using Anonymized Data for Classification*, in *Proceedings of the 2009 IEEE International Conference on Data Engineering*. 2009. p. 429-440.
17. Kohavi, R., *Scaling up the accuracy of Naive-Bayes classifiers: a decision-tree hybrid*, in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. 1996: Portland, Oregon. p. 202-207.
18. Prasser, F., et al., *ARX--A Comprehensive Tool for Anonymizing Biomedical Data.* AMIA ... Annual Symposium proceedings / AMIA Symposium., 2014.
19. Pedregosa, F., et al., *Scikit-learn: Machine Learning in Python.* J. Mach. Learn. Res., 2011. **12**: p. 2825-2830.

ภาคผนวก

ภาคผนวก ก

code ที่ใช้ในการสอนตัวจำแนกประเภท

CORE.PY

```
import pandas as pd
import numpy as np
import timeit
import copy
import csv
import json

from sklearn.ensemble import RandomForestClassifier, BaggingClassifier, VotingClassifier, AdaBoostClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import KFold
from sklearn import preprocessing
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import accuracy_score, precision_recall_fscore_support
from rotation_forest.rotation_forest import RotationForestClassifier
from prettytable import PrettyTable

i = 0

class MultiColumnLabelEncoder:
    def __init__(self, columns=None):
        self.columns = columns

    def fit(self, X, y=None):
        return self

    def transform(self, X):
        output = X.copy()
        if self.columns is not None:
            for col in self.columns:
                output[col] = LabelEncoder().fit_transform(output[col].astype(str))
        else:
            for col_name, col in output.iteritems():
                output[col_name] = LabelEncoder().fit_transform(col)
        return output
```

```
    def fit_transform(self, X, y=None):
        return self.fit(X, y).transform(X)

def format_float(number):
    return "{0:.4f}".format(number)

def get_data_version_from_file_name(file_name):
    return file_name[file_name.rfind('-')+1:file_name.find('.')]

def get_privacy_criterion_from_file_name(file_name):
    return file_name[file_name.rfind('-')+1:file_name.find('.')-1]

def get_data_set_name_from_file_name(file_name):
    return file_name[file_name.find('/')+1:file_name.rfind('/')]

def convert_to_np_array(df):
    return df.reset_index().values

def convert_to_data_frame(np_array, column_set):
    return pd.DataFrame(data=np_array[0:, 1:], index=np_array[0:, 0], columns=column_set)

def my_classify(file_name, columns, data_info):
    print(file_name)

    column_set = copy.deepcopy(columns)
    attributes = column_set['attributes']
    dropped_attributes = column_set['dropped_attributes']
    class_label_attribute = column_set['class_label_attribute']

    df = pd.read_csv(file_name, sep=';')

    for dropped_attribute in dropped_attributes:
        attributes.remove(dropped_attribute)
        df.drop(dropped_attribute, axis=1, inplace=True)

    enc_df = MultiColumnLabelEncoder(columns=attributes).fit_transform(df)
    data = convert_to_np_array(enc_df)

    k_of_fold = 5
    kf = KFold(n_splits=k_of_fold, random_state=None, shuffle=True)
```

```
    kf.get_n_splits(data)

    sum_random_forest_acc, sum_bagging_acc, sum_adaboost_acc, sum_voting_acc, sum_rotation_forest_acc = 0
    sum_random_forest_pre, sum_bagging_pre, sum_adaboost_pre, sum_voting_pre, sum_rotation_forest_pre = 0
    sum_random_forest_rec, sum_bagging_rec, sum_adaboost_rec, sum_voting_rec, sum_rotation_forest_rec = 0
    sum_random_forest_f1, sum_bagging_f1, sum_adaboost_f1, sum_voting_f1, sum_rotation_forest_f1 = 0
    sum_random_forest_time, sum_bagging_time, sum_adaboost_time, sum_voting_time,
sum_rotation_forest_time = 0

    for train, test in kf.split(data):
        train_data = convert_to_data_frame(np.array(data)[train], attributes)
        test_data = convert_to_data_frame(np.array(data)[test], attributes)

        x_train = train_data.drop([class_label_attribute], axis=1)
        y_train = train_data[class_label_attribute]
        x_test = test_data.drop([class_label_attribute], axis=1)
        y_test = test_data[class_label_attribute]

        print('random forest start')
        # random forest
        start_time = timeit.default_timer()

        rfc = RandomForestClassifier(n_estimators=50, max_depth=None, min_samples_split=2, random_state=0,
bootstrap=False)
        prediction_rf = rfc.fit(x_train, y_train).predict(x_test)

        acc_rf = accuracy_score(y_test, prediction_rf)
        precision_rf, recall_rf, f_score_rf, support_rf = precision_recall_fscore_support(y_test, prediction_rf,
average='weighted')
        time_rf = format_float(timeit.default_timer() - start_time)

        print('bagging start')
        # bagging
        start_time = timeit.default_timer()

        bagging = BaggingClassifier(DecisionTreeClassifier(), n_estimators=50, max_samples=0.5, max_features=0.5)
        prediction_bg = bagging.fit(x_train, y_train).predict(x_test)

        acc_bag = accuracy_score(y_test, prediction_bg)
        precision_bag, recall_bag, f_score_bag, support_bag = precision_recall_fscore_support(y_test, prediction_bg,
average='weighted')
```

```
        time_bag = format_float(timeit.default_timer() - start_time)

        print('adaboost start')
        # adaboost
        start_time = timeit.default_timer()

        adaboost = AdaBoostClassifier(n_estimators=50)
        prediction_ada = adaboost.fit(x_train, y_train).predict(x_test)

        acc_ada = accuracy_score(y_test, prediction_ada)
        precision_ada, recall_ada, f_score_ada, support_ada = precision_recall_fscore_support(y_test,
prediction_ada, average='weighted')
        time_ada = format_float(timeit.default_timer() - start_time)

        print('voting start')
        # voting
        start_time = timeit.default_timer()

        voting = VotingClassifier(estimators=[('clf1', rfc), ('clf2', bagging), ('clf3', adaboost)], voting='hard')
        prediction_vt = voting.fit(x_train, y_train).predict(x_test)

        acc_vt = accuracy_score(y_test, prediction_vt)
        precision_vt, recall_vt, f_score_vt, support_vt = precision_recall_fscore_support(y_test, prediction_vt,
average='weighted')
        time_vt = format_float(timeit.default_timer() - start_time)

        print('rotation forest start')
        # rotation forest
        start_time = timeit.default_timer()

        rotation = RotationForestClassifier(random_state=1234)
        prediction_rot = rotation.fit(x_train, y_train).predict(x_test)

        acc_rof = accuracy_score(y_test, prediction_rot)
        precision_rof, recall_rof, f_score_rof, support_rof = precision_recall_fscore_support(y_test, prediction_rot,
average='weighted')
        time_rof = format_float(timeit.default_timer() - start_time)

        sum_random_forest_acc = sum_random_forest_acc + float(acc_rf)
        sum_bagging_acc = sum_bagging_acc + float(acc_bag)
        sum_adaboost_acc = sum_adaboost_acc + float(acc_ada)
```

```
    sum_voting_acc = sum_voting_acc + float(acc_vt)
    sum_rotation_forest_acc = sum_rotation_forest_acc + float(acc_rof)

    sum_random_forest_pre = sum_random_forest_pre + float(precision_rf)
    sum_bagging_pre = sum_bagging_pre + float(precision_bag)
    sum_adaboost_pre = sum_adaboost_pre + float(precision_ada)
    sum_voting_pre = sum_voting_pre + float(precision_vt)
    sum_rotation_forest_pre = sum_rotation_forest_pre + float(precision_rof)

    sum_random_forest_rec = sum_random_forest_rec + float(recall_rf)
    sum_bagging_rec = sum_bagging_rec + float(recall_bag)
    sum_adaboost_rec = sum_adaboost_rec + float(recall_ada)
    sum_voting_rec = sum_voting_rec + float(recall_vt)
    sum_rotation_forest_rec = sum_rotation_forest_rec + float(recall_rof)

    sum_random_forest_f1 = sum_random_forest_f1 + float(f_score_rf)
    sum_bagging_f1 = sum_bagging_f1 + float(f_score_bag)
    sum_adaboost_f1 = sum_adaboost_f1 + float(f_score_ada)
    sum_voting_f1 = sum_voting_f1 + float(f_score_vt)
    sum_rotation_forest_f1 = sum_rotation_forest_f1 + float(f_score_rof)

    sum_random_forest_time = sum_random_forest_time + float(time_rf)
    sum_bagging_time = sum_bagging_time + float(time_bag)
    sum_adaboost_time = sum_adaboost_time + float(time_ada)
    sum_voting_time = sum_voting_time + float(time_vt)
    sum_rotation_forest_time = sum_rotation_forest_time + float(time_rof)

accuracy_of_random_forest = format_float(sum_random_forest_acc / k_of_fold)
accuracy_of_bagging = format_float(sum_bagging_acc / k_of_fold)
accuracy_of_adaboost = format_float(sum_adaboost_acc / k_of_fold)
accuracy_of_voting = format_float(sum_voting_acc / k_of_fold)
accuracy_of_rotation_forest = format_float(sum_rotation_forest_acc / k_of_fold)

precision_of_random_forest = format_float(sum_random_forest_pre / k_of_fold)
precision_of_bagging = format_float(sum_bagging_pre / k_of_fold)
precision_of_adaboost = format_float(sum_adaboost_pre / k_of_fold)
precision_of_voting = format_float(sum_voting_pre / k_of_fold)
precision_of_rotation_forest = format_float(sum_rotation_forest_pre / k_of_fold)

recall_of_random_forest = format_float(sum_random_forest_rec / k_of_fold)
recall_of_bagging = format_float(sum_bagging_rec / k_of_fold)
```

```
    recall_of_adaboost = format_float(sum_adaboost_rec / k_of_fold)
    recall_of_voting = format_float(sum_voting_rec / k_of_fold)
    recall_of_rotation_forest = format_float(sum_rotation_forest_rec / k_of_fold)

    f_score_of_random_forest = format_float(sum_random_forest_f1 / k_of_fold)
    f_score_of_bagging = format_float(sum_bagging_f1 / k_of_fold)
    f_score_of_adaboost = format_float(sum_adaboost_f1 / k_of_fold)
    f_score_of_voting = format_float(sum_voting_f1 / k_of_fold)
    f_score_of_rotation_forest = format_float(sum_rotation_forest_f1 / k_of_fold)

    time_of_random_forest = format_float(sum_random_forest_time)
    time_of_bagging = format_float(sum_bagging_time)
    time_of_adaboost = format_float(sum_adaboost_time)
    time_of_voting = format_float(sum_voting_time)
    time_of_rotation_forest = format_float(sum_rotation_forest_time)

    j_rf = {'accuracy_score': accuracy_of_random_forest, 'precision': precision_of_random_forest, 'recall':
recall_of_random_forest, 'f1_score': f_score_of_random_forest, 'execute_time': time_of_random_forest}
    j_bg = {'accuracy_score': accuracy_of_bagging, 'precision': precision_of_bagging, 'recall': recall_of_bagging,
'f1_score': f_score_of_bagging, 'execute_time': time_of_bagging}
    j_ada = {'accuracy_score': accuracy_of_adaboost, 'precision': precision_of_adaboost, 'recall': recall_of_adaboost,
'f1_score': f_score_of_adaboost, 'execute_time': time_of_adaboost}
    j_vt = {'accuracy_score': accuracy_of_voting, 'precision': precision_of_voting, 'recall': recall_of_voting, 'f1_score':
f_score_of_voting, 'execute_time': time_of_voting}
    j_rof = {'accuracy_score': accuracy_of_rotation_forest, 'precision': precision_of_rotation_forest, 'recall':
recall_of_rotation_forest, 'f1_score': f_score_of_rotation_forest, 'execute_time': time_of_rotation_forest}

    global i

    j_classifiers_output = dict()
    j_classifiers_output['rf'] = j_rf
    j_classifiers_output['bg'] = j_bg
    j_classifiers_output['ada'] = j_ada
    j_classifiers_output['vt'] = j_vt
    j_classifiers_output['rof'] = j_rof

    j_result = dict()
    j_result['index'] = i
    j_result['file_name'] = file_name
    j_result['classifiers_output'] = j_classifiers_output
```

```
    data_info['privacy_level']['record_risk'] = format_float(data_info['privacy_level']['record_risk'])
    data_info['privacy_level']['highest_risk'] = format_float(data_info['privacy_level']['highest_risk'])
    data_info['privacy_level']['success_rate'] = format_float(data_info['privacy_level']['success_rate'])
    data_info['suppressed_percent'] = format_float(data_info['suppressed_percent'])
    j_result['data_info'] = data_info

    i = i + 1

    return j_result

def summarize_result(result):
    s_results = list()

    for item in result:
        s_result = dict()

        classifiers_output = item['classifiers_output']

        max_accuracy = 0
        max_execute_time = 0
        max_model_name = ''

        for key in classifiers_output.keys():
            classifier_output = classifiers_output[key]

            if float(classifier_output['accuracy_score']) > float(max_accuracy):
                max_accuracy = classifier_output['accuracy_score']
                max_execute_time = classifier_output['execute_time']
                max_model_name = key

        s_result['file_name'] = item['file_name']
        s_result['model'] = max_model_name
        s_result['accuracy'] = max_accuracy
        s_result['execute_time'] = max_execute_time
        s_result['success_rate'] = item['data_info']['privacy_level']['success_rate']
        s_result['suppressed_percent'] = item['data_info']['suppressed_percent']

        s_results.append(s_result)

    return s_results
```

```
def write_json_to_file(file_name, json_string):
    with open(file_name, 'w') as outfile:
        json.dump(json_string, outfile)
```

TEST.PY

```
from core import my_classify, summarize_result, write_json_to_file


# for ADULT data-set

from data_info.data_info_adult import *

attributes = ['sex', 'age', 'race', 'marital-status', 'education', 'native-country', 'workclass', 'occupation', 'salary-class']
dropped_attributes = []
class_label_attribute = 'salary-class'
columns = {'attributes': attributes, 'dropped_attributes': dropped_attributes, 'class_label_attribute':
class_label_attribute}
ADULT_result = list()

ADULT_result.append(my_classify('data/adult/adult-original.csv', columns, data_info_original))
ADULT_result.append(my_classify('data/adult/adult-k1.csv', columns, data_info_k1))
ADULT_result.append(my_classify('data/adult/adult-k2.csv', columns, data_info_k2))
ADULT_result.append(my_classify('data/adult/adult-k3.csv', columns, data_info_k3))
ADULT_result.append(my_classify('data/adult/adult-l1.csv', columns, data_info_l1))
ADULT_result.append(my_classify('data/adult/adult-l2.csv', columns, data_info_l2))
ADULT_result.append(my_classify('data/adult/adult-l3.csv', columns, data_info_l3))
ADULT_result.append(my_classify('data/adult/adult-t1.csv', columns, data_info_t1))
ADULT_result.append(my_classify('data/adult/adult-t2.csv', columns, data_info_t2))
ADULT_result.append(my_classify('data/adult/adult-t3.csv', columns, data_info_t3))
ADULT_result.append(my_classify('data/adult/adult-dp1.csv', columns, data_info_dp1))
ADULT_result.append(my_classify('data/adult/adult-dp2.csv', columns, data_info_dp2))
ADULT_result.append(my_classify('data/adult/adult-dp3.csv', columns, data_info_dp3))
ADULT_result.append(my_classify('data/adult/adult-pu1.csv', columns, data_info_pu1))
ADULT_result.append(my_classify('data/adult/adult-pu2.csv', columns, data_info_pu2))
ADULT_result.append(my_classify('data/adult/adult-pu3.csv', columns, data_info_pu3))
ADULT_result.append(my_classify('data/adult/adult-km1.csv', columns, data_info_km1))
ADULT_result.append(my_classify('data/adult/adult-km2.csv', columns, data_info_km2))
ADULT_result.append(my_classify('data/adult/adult-km3.csv', columns, data_info_km3))

write_json_to_file('result/adult/result.json', ADULT_result)
ADULT_summary = summarize_result(ADULT_result)
```

```
print('done')

# for ATUS data

from data_info.data_info_atus_full import *

attributes = ['region', 'age', 'sex', 'race', 'marital_status', 'citizenship_status', 'birthplace',
'highest_level_of_school_completed', 'labor_force_status']
dropped_attributes = []
class_label_attribute = 'labor_force_status'
columns = {'attributes': attributes, 'dropped_attributes': dropped_attributes, 'class_label_attribute':
class_label_attribute}
ATUS_result = list()

ATUS_result.append(my_classify('data/atus-full/atus-original.csv', columns, data_info_original))
ATUS_result.append(my_classify('data/atus-full/atus-k1.csv', columns, data_info_k1))
ATUS_result.append(my_classify('data/atus-full/atus-k2.csv', columns, data_info_k2))
ATUS_result.append(my_classify('data/atus-full/atus-k3.csv', columns, data_info_k3))
ATUS_result.append(my_classify('data/atus-full/atus-l1.csv', columns, data_info_l1))
ATUS_result.append(my_classify('data/atus-full/atus-l2.csv', columns, data_info_l2))
ATUS_result.append(my_classify('data/atus-full/atus-l3.csv', columns, data_info_l3))
ATUS_result.append(my_classify('data/atus-full/atus-t1.csv', columns, data_info_t1))
ATUS_result.append(my_classify('data/atus-full/atus-t2.csv', columns, data_info_t2))
ATUS_result.append(my_classify('data/atus-full/atus-t3.csv', columns, data_info_t3))
ATUS_result.append(my_classify('data/atus-full/atus-dp1.csv', columns, data_info_dp1))
ATUS_result.append(my_classify('data/atus-full/atus-dp2.csv', columns, data_info_dp2))
ATUS_result.append(my_classify('data/atus-full/atus-dp3.csv', columns, data_info_dp3))
ATUS_result.append(my_classify('data/atus-full/atus-pu1.csv', columns, data_info_pu1))
ATUS_result.append(my_classify('data/atus-full/atus-pu2.csv', columns, data_info_pu2))
ATUS_result.append(my_classify('data/atus-full/atus-pu3.csv', columns, data_info_pu3))
ATUS_result.append(my_classify('data/atus-full/atus-km1.csv', columns, data_info_km1))
ATUS_result.append(my_classify('data/atus-full/atus-km2.csv', columns, data_info_km2))
ATUS_result.append(my_classify('data/atus-full/atus-km3.csv', columns, data_info_km3))

write_json_to_file('result/atus-full/result.json', ATUS_result)
ATUS_summary = summarize_result(ATUS_result)

print('done')

# for FARS data
```

```
from data_info.data_info_fars import *

attributes = ['iage', 'irace', 'ideathmon', 'ideathday', 'isex', 'ihispanic', 'istatenum', 'iinjury']
dropped_attributes = []
class_label_attribute = 'iinjury'
columns = {'attributes': attributes, 'dropped_attributes': dropped_attributes, 'class_label_attribute':
class_label_attribute}

FARS_result = list()

FARS_result.append(my_classify('data/fars/fars-original.csv', columns, data_info_original))
FARS_result.append(my_classify('data/fars/fars-k1.csv', columns, data_info_k1))
FARS_result.append(my_classify('data/fars/fars-k2.csv', columns, data_info_k2))
FARS_result.append(my_classify('data/fars/fars-k3.csv', columns, data_info_k3))
FARS_result.append(my_classify('data/fars/fars-l1.csv', columns, data_info_l1))
FARS_result.append(my_classify('data/fars/fars-l2.csv', columns, data_info_l2))
FARS_result.append(my_classify('data/fars/fars-l3.csv', columns, data_info_l3))
FARS_result.append(my_classify('data/fars/fars-t1.csv', columns, data_info_t1))
FARS_result.append(my_classify('data/fars/fars-t2.csv', columns, data_info_t2))
FARS_result.append(my_classify('data/fars/fars-t3.csv', columns, data_info_t3))
FARS_result.append(my_classify('data/fars/fars-dp1.csv', columns, data_info_dp1))
FARS_result.append(my_classify('data/fars/fars-dp2.csv', columns, data_info_dp2))
FARS_result.append(my_classify('data/fars/fars-dp3.csv', columns, data_info_dp3))
FARS_result.append(my_classify('data/fars/fars-pu1.csv', columns, data_info_pu1))
FARS_result.append(my_classify('data/fars/fars-pu2.csv', columns, data_info_pu2))
FARS_result.append(my_classify('data/fars/fars-pu3.csv', columns, data_info_pu3))
FARS_result.append(my_classify('data/fars/fars-km1.csv', columns, data_info_km1))
FARS_result.append(my_classify('data/fars/fars-km2.csv', columns, data_info_km2))
FARS_result.append(my_classify('data/fars/fars-km3.csv', columns, data_info_km3))

write_json_to_file('result/fars/result.json', FARS_result)
FARS_summary = summarize_result(FARS_result)

print('done')
```

ประวัติผู้เขียนวิทยานิพนธ์

นาย พีรพงศ์ วาณิชยวิศาลสกุล เกิดเมื่อวันที่ 6 กุมภาพันธ์ 2535 ที่จังหวัดกรุงเทพมหานคร สำเร็จการศึกษาจากมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี และเข้าศึกษาต่อที่ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2559