การประเมินการส่งผ่านข้อมูลส่วนตัวของแอปพลิเคชันโดยวิเคราะห์จากนโยบายความเป็นส่วนตัว

นายเมธัส นาคเสนีย์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมซอฟต์แวร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2562

A MACHINE-LEARNING BASED APPROACH FOR EVALUATING PERSONALLY IDENTIFIABLE INFORMATION TRANSMISSION FROM ONLINE PRIVACY POLICIES

Mr. Methus Narksenee

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Software Engineering
Department of Computer Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2019

หัวข้อวิทยานิพนธ์	การประเมินการส่งผ่านข้อมูลส่วนตัวของแอปพลิเคชันโดยวิเคราะห์จากนโยบายความเป็นส่วนตัว
โดย	นายเมธัส นาคเสนีย์
สาขาวิชา	วิศวกรรมซอฟต์แวร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	อาจารย์ ดร.กุลวดี ศรีพานิชกุลชัย

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร.สุพจน์ เตชวรสินสกุล)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(อาจารย์ ดร.ดวงดาว วิชาดากุล)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(อาจารย์ ดร.กุลวดี ศรีพานิชกุลชัย)

..... กรรมการ
(อาจารย์ ดร.เอกพล ช่วงสุวนิช)

..... กรรมการภายนอกมหาวิทยาลัย
(รองศาสตราจารย์ ดร.สุดสงวน งามสุริยโรจน์)

เมธัส นาคเสนีย์ : การประเมินการส่งผ่านข้อมูลส่วนตัวของแอปพลิเคชันโดยวิเคราะห์จากนโยบายความเป็นส่วนตัว. (A MACHINE-LEARNING BASED APPROACH FOR EVALUATING PERSONALLY IDENTIFIABLE INFORMATION TRANSMISSION FROM ONLINE PRIVACY POLICIES) อ.ที่ปรึกษาหลัก : อ. ดร.กุลวดี ศรีพานิชกุลชัย

โมไบล์แอปพลิเคชันในปัจจุบันได้ขอเข้าถึงข้อมูลของผู้ใช้บริการเพื่อที่จะนำข้อมูลเหล่านี้ไปพัฒนาการให้บริการ เช่น ข้อมูลส่วนตัว อีเมล ซึ่งการนำข้อมูลเหล่านี้ไปใช้มีทั้งจุดประสงค์ในการใช้ข้อมูลในทางที่ดีและไม่ดี จึงเป็นเรื่องที่ผู้ใช้บริการควรตระหนักถึง ทั้งนี้ผู้ใช้บริการสามารถตรวจสอบรายละเอียดการนำข้อมูลไปใช้จากแหล่งที่สามารถเข้าถึงได้ง่าย ได้แก่ นโยบายความเป็นส่วนตัว แต่เนื่องจากนโยบายความเป็นส่วนตัวมีข้อความที่ยาวและทำความเข้าใจได้ยาก ผู้ใช้บริการอาจพลาดส่วนสำคัญจากการอ่านนโยบายความเป็นส่วนตัวได้ ดังนั้นวิทยานิพนธ์นี้จึงได้ตั้งข้อสมมุติฐานเพื่อทำการพิสูจน์สมมุติฐานว่าการประเมินการส่งผ่านข้อมูลส่วนตัวของแอปพลิเคชันสามารถวิเคราะห์ได้จากข้อความในนโยบายความเป็นส่วนตัวหรือไม่ โดยการใช้การเรียนรู้ด้วยเครื่องเข้ามาช่วยเพื่อที่จะประเมินการส่งผ่านของข้อมูลส่วนตัวแทนการอ่านจากนโยบายความเป็นส่วนตัว

สาขาวิชา วิศวกรรมซอฟต์แวร์ ลายมือชื่อนิสิต .....

ปีการศึกษา 2562 ลายมือชื่อ อ.ที่ปรึกษาหลัก .....

6070956121 : MAJOR SOFTWARE ENGINEERING

KEYWORD: personally identifiable information - privacy policy - machine learning - text classification - natural language processing

Methus Narksenee : A MACHINE-LEARNING BASED APPROACH FOR EVALUATING PERSONALLY IDENTIFIABLE INFORMATION TRANSMISSION FROM ONLINE PRIVACY POLICIES. Advisor: Kunwadee Sripanidkulchai, Ph.D.

Mobile applications frequently request private information from users, supposedly to improve their services and applications. The collected data, such as personally identifiable information, raises users' concerns since some applications actually have malicious intentions to leak personal data. Privacy policies are an important resource as they are the sole source of information users can use to determine how applications plan to collect and use their data that is easily accessible prior to downloading and using the application. However, users tend to ignore or gloss over privacy policies as they are often written in complicated hard-to-understand language. Thus, users may miss crucial privacy-related information after reading such documents. In this thesis, we experimentally determine how much we can trust an application's privacy policy by looking at the language used in more than 8,000 privacy policies and compare them to what applications actually do. We classify whether or not applications transmit privacy-related information using machine learning with three classifiers, support vector machines (SVM), k-nearest neighbors (KNN), and logistic regression (LR).

Field of Study: Software Engineering Student's Signature .....

Academic Year: 2019 Advisor's Signature .....

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยความอนุเคราะห์อย่างดียิ่งของอาจารย์ดร. กุลวดี ศรีพานิชกุลชัยอาจารย์ที่ปรึกษาวิทยานิพนธ์ที่กรุณาเสียสละเวลาในการให้คำปรึกษาช่วยตรวจสอบให้คำแนะนำแนวทางวิจัยคอยสนับสนุนและให้กำลังใจ ขอขอบพระคุณคุณคณาจารย์ทุกท่านในภาควิชาวิศวกรรมศาสตร์จุฬาลงกรณ์มหาวิทยาลัยที่ได้ให้ความรู้ทางวิชาการคำแนะนำและกำลังใจที่มีค่าอย่างยิ่งต่อผู้วิจัยรวมถึงบุคลากรทุกท่านในภาควิชาวิศวกรรมศาสตร์จุฬาลงกรณ์มหาวิทยาลัยที่ให้คำแนะนำและคอยช่วยเหลือในระหว่างที่ผู้วิจัยกำลังศึกษา สุดท้ายขอขอบพระคุณคุณแม่คุณพ่อครอบครัวและเพื่อน ๆ ที่สนับสนุนให้ความช่วยเหลือให้คำแนะนำและให้กำลังใจแก่ผู้วิจัยเสมอมา

เมธัส นาคเสนีย์

สารบัญ

หน้า

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ความก้าวหน้าของสมาร์ทโฟน (Smartphone) และความหลากหลายของโมไบล์แอปพลิเคชันในหลายปีที่ผ่านมาได้มีส่วนสำคัญทำให้สมาร์ทโฟนได้เป็นส่วนหนึ่งของการใช้ชีวิตประจำวันของผู้ใช้งานหลายพันล้านคน โดยเฉพาะระบบปฏิบัติการแอนดรอยด์ที่มีผู้ใช้งานกว่า 88% ของตลาดสมาร์ทโฟนทั้งหมด [1] เนื่องด้วยโมไบล์แอปพลิเคชันมีการขอเข้าถึงและใช้งานข้อมูลของผู้ใช้งานเพื่อต้องการปรับปรุงการบริการตามจุดประสงค์ของผู้พัฒนา การเก็บข้อมูลส่วนตัวของผู้ใช้งานทำให้ผู้ใช้งานมีความกังวลต่อการนำข้อมูลไปใช้มากขึ้น โดยส่วนของการชี้แจงการใช้ข้อมูลได้ถูกระบุในนโยบายความเป็นส่วนตัวที่ได้มีกฎหมายออกมาควบคุม [2] แต่ผู้ใช้งานสมาร์ทโฟนส่วนใหญ่ได้เพิกเฉยต่อการอ่านนโยบายความเป็นส่วนตัวเนื่องด้วย มีข้อความที่ยาว ยากต่อการทำความเข้าใจ และไม่ได้ตระหนักถึงความเสี่ยงของภัยคุกคามที่จะเกิดขึ้นในการนำข้อมูลส่วนตัวไปใช้

การทำความเข้าใจเอกสารนโยบายความเป็นส่วนตัวที่มีข้อความที่ยาวและชี้แจงในหลายหมวดหมู่ของการปฏิบัติต่อข้อมูลเป็นเรื่องที่ได้รับความสนใจ มีงานวิจัยอื่นที่เกี่ยวข้องที่ได้ศึกษาภาษาธรรมชาติ (Natural Language) ของนโยบายความเป็นส่วนตัว และนำการเรียนรู้ของเครื่อง (Machine Learning) มาช่วยวิเคราะห์นโยบายความเป็นส่วนตัว เพื่อจัดกลุ่มให้อยู่ในรูปแบบของหัวข้อการปฏิบัติต่อข้อมูลของผู้ใช้งานเพื่อให้มีความโปร่งใสมากขึ้น และมีความชัดเจนในการชี้แจงรายละเอียดตามหมวดหมู่ย่อยของนโยบายความเป็นส่วนตัว [3] แต่งานเหล่านั้นยังไม่ได้ครอบคลุมถึงการประเมินการส่งผ่านของข้อมูลส่วนตัวโดยวิเคราะห์จากนโยบายความเป็นส่วนตัวเปรียบเทียบกับการใช้งานแอปพลิเคชันจริง

ในส่วนของการวิเคราะห์ประเมินความเสี่ยง ได้มีงานวิจัยที่วิเคราะห์ความเสี่ยงจากตัวแพคเกจ (Package) ที่รวบรวมไฟล์ในการติดตั้งบนระบบปฏิบัติการแอนดรอยด์ (Android) โดยใช้วิธีการตรวจสอบโค้ดแบบสถิตย์ (Static Code Analysis) [4] [5] เพื่อแยกฟังก์ชันของแหล่งที่มาของข้อมูล (Source) และฟังก์ชันของแหล่งเก็บข้อมูลเพื่อเตรียมส่งผ่านข้อมูล (Sink) ซึ่งจาก 2 คำนิยามนี้สามารถคาดการณ์ได้ว่ามีการส่งผ่านของข้อมูลสำคัญ (Sensitive Information) หรือไม่ นอกจากนี้ยังมีอีกวิธีหนึ่งคือ การวิเคราะห์เชิงพลวัต (Dynamic Analysis) [6] เป็นวิธีในการตรวจสอบพฤติกรรมที่ไม่พึงประสงค์โดยตรวจสอบจากการทดสอบการใช้แอปพลิเคชันเพื่อดูการส่งผ่านของข้อมูลว่าเกิดขึ้นจริงหรือไม่ งานวิจัยนี้ไม่ได้กล่าวถึงนโยบายความเป็นส่วนตัว ซึ่งเป็นส่วนที่ผู้ใช้งานสามารถเข้าถึงได้ง่ายที่สุดจาก Google Play Store [7]

งานวิทยานิพนธ์นี้จึงนำเสนอแนวคิดและเครื่องมือสำหรับการประเมินการส่งผ่านข้อมูลของแอปพลิเคชันและชี้แจงถึงการปฏิบัติต่อข้อมูลในทิศทางที่ผู้ใช้งานสามารถเข้าใจได้ง่าย ซึ่งแนวคิดของงานวิทยานิพนธ์นี้อยู่บนสมมติฐานว่างานวิทยานิพนธ์สามารถประเมินการส่งผ่านของข้อมูลจากนโยบายความเป็นส่วนตัวได้หรือไม่ จึงได้นำการเรียนรู้ด้วยเครื่องเข้ามาประยุกต์ใช้กับการประมวลผลทางภาษา (Natural Language Processing) และเปรียบเทียบความแม่นยำของแบบจำลองในการเรียนรู้ 3 แบบจำลองด้วยกัน

1.2 วัตถุประสงค์ของงานวิจัย

1.2.1 เพื่อที่จะช่วยผู้ใช้งานแอปพลิเคชันประเมินการเข้าถึงข้อมูลและการใช้ข้อมูลส่วนบุคคลของแอปพลิเคชันก่อนตัดสินใจใช้แอปพลิเคชัน

1.2.2 เพื่อที่จะชี้แจงประเภทของการส่งผ่านข้อมูลที่แอปพลิเคชันได้ระบุในนโยบายความเป็นส่วนตัวไปในทิศทางที่ผู้ใช้งานเข้าใจได้

1.2.3 เพื่อเปรียบเทียบความแม่นยำของแบบจำลองในการประเมินการส่งผ่านข้อมูลส่วนตัวจากนโยบายความเป็นส่วนตัว

1.3 ขอบเขตงานวิจัย

1.3.1 ภาษาของนโยบายความเป็นส่วนตัวที่ใช้ในการวิเคราะห์ประเมินความเสี่ยงจะต้องเขียนด้วยภาษาอังกฤษ

1.3.2 การตรวจสอบหรือวิเคราะห์เพื่อประเมินความเสี่ยงของแอปพลิเคชันจะต้องเป็นแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์เท่านั้น

1.3.3 แอปพลิเคชันที่ใช้ในการศึกษาและตรวจสอบมีความหลากหลายมาจาก 45 ประเภทแอปพลิเคชันใน Google Play Store

1.4 ขั้นตอนการดำเนินงาน

1.4.1 ศึกษาข้อมูลเอกสารและงานวิจัยที่เกี่ยวข้องกับพฤติกรรมที่ไม่พึงประสงค์ในการใช้ข้อมูลส่วนตัว

1.4.2 ศึกษาข้อมูลเอกสารและงานวิจัยที่เกี่ยวข้องกับวิธีประเมินความเสี่ยงต่อพฤติกรรมที่ไม่พึงประสงค์ในการใช้ข้อมูลส่วนตัว

1.4.3 ศึกษาการใช้งานของเครื่องมือประเมินความเสี่ยงของพฤติกรรมที่ไม่พึงประสงค์ในการใช้แอปพลิเคชัน

1.4.4 ศึกษาค้นคว้าโมเดลที่เหมาะสมในการใช้ประเมินความเสี่ยงของนโยบายความเป็นส่วนตัว

1.4.5 ศึกษาวิธีประเมินผลการตรวจสอบความเสี่ยงและปรับปรุงแบบโมเดลสำหรับนโยบายความเป็นส่วนตัว

1.4.6 เก็บข้อมูลนโยบายความเป็นส่วนตัวจาก Google Play

1.4.7 วิเคราะห์แอปพลิเคชันเพื่อประเมินความเสี่ยงต่อพฤติกรรมที่ไม่พึงประสงค์

1.4.8 ประเมินผลการตรวจสอบความเสี่ยงและปรับปรุงแบบโมเดล

1.4.9 สรุปผลการวิจัย

1.4.10 เรียบเรียงและจัดทำบทความวิชาการ

1.4.11 จัดทำรายงานวิทยานิพนธ์

ตารางที่ 1. ภาพแสดงถึงแผนขั้นตอนการดำเนินงาน

ขั้นตอนที่	ก.ย. -61	ต.ค.- 61	พ.ย.- 61	ธ.ค. -61	ม.ค.- 62	ก.พ.- 62	มี.ค.- 62	เม.ย. -62	พ.ค.- 62	ก.ย. -62	ต.ค.- 62	พ.ย.- 62	ธ.ค. -62
1.4.1													
1.4.2													
1.4.3													
1.4.4													
1.4.5													
1.4.6													
1.4.7													
1.4.8													
1.4.9													
1.4.10													
1.4.11													

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1.5.1 สามารถประเมินการส่งผ่านข้อมูลจากนโยบายความเป็นส่วนตัวได้เบื้องต้น

1.5.2 สามารถนำหลักการของการวิเคราะห์ภาษาธรรมชาติไปประยุกต์ใช้กับเอกสารนโยบายความเป็นส่วนตัวได้

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 กระบวนการประมวลผลทางภาษา (Natural Language Processing)

กระบวนการเป็นสาขาหนึ่งของเทคโนโลยีปัญญาประดิษฐ์ (Artificial Intelligence) [8] ที่ศึกษาปัญหาในการประมวลผลและใช้งานภาษาธรรมชาติ โดยต้องการให้คอมพิวเตอร์สามารถเข้าใจภาษาของมนุษย์ได้ ซึ่งภาษามีความซับซ้อนกว่ารหัสของคอมพิวเตอร์ กระบวนการเรียนรู้ของการประมวลผลทางภาษา ได้แบ่งออกเป็น 5 ประเภทดังนี้

(1) การวิเคราะห์ทางองค์ประกอบ (Morphological Level) คือ การวิเคราะห์ในระดับหน่วยของคำ สามารถแยกเป็นกลุ่มคำที่มีความหมายได้

(2) การวิเคราะห์ทางไวยากรณ์ (Syntactic Level) คือ การชี้ระบุหน้าที่ให้กับคำแต่ละคำที่ถูกแบ่งแล้วว่ามีความสัมพันธ์กันอย่างไรในระดับประโยค เช่น ประธาน กริยา กรรม

(3) การวิเคราะห์ทางความหมาย (Semantic Level) คือ การวิเคราะห์ความหมายของคำในความสัมพันธ์ของโครงสร้างทางไวยากรณ์ หรือโครงสร้างระดับประโยค

(4) บูรณาการทางวจนิพนธ์ (Discourse Level) คือ การพิจารณาความหมายของประโยคจาก ประโยคข้างเคียง เนื่องจากการตีความหมายอาจจะต้องตีความจากคำหรือประโยคก่อนหน้า

(5) การวิเคราะห์ทางปฏิบัติ (Pragmatic Level) คือ ขั้นตอนเข้าใจความหมายของคำและประโยคอ้างอิงจากสถานการณ์หรือฐานความรู้เดิม ซึ่งอาจไม่ได้ระบุอยู่ในเนื้อหานั้น ๆ เพื่อให้สามารถตีความได้ใกล้เคียงกับมนุษย์ที่สามารถเชื่อมโยงข้อมูลใหม่เข้ากับความรู้เดิมได้ตลอดเวลา

เมื่อคอมพิวเตอร์สามารถเข้าใจภาษาของมนุษย์ได้ จึงสามารถต่อยอดนำไปใช้เป็นนวัตกรรมที่มีประโยชน์ได้ เช่น แชทบอท (Chat bot) ระบบค้นหา (Search Engine) ระบบแนะนำสินค้าหรือบริการต่างๆ (Recommendation System) การแบ่งประเภทของเอกสาร (Document Classification) เป็นต้น โดยงานวิทยานิพนธ์นี้ต้องการศึกษากระบวนการประมวลผลทางภาษาเพื่อที่จะทำให้คอมพิวเตอร์เข้าใจโครงสร้างและ

ความหมายของเอกสารนโยบายความเป็นส่วนตัวเพื่อหาความเป็นไปได้ของคำหรือประโยคที่ส่งผลต่อความเสี่ยงต่อการปฏิบัติต่อข้อมูลในทางที่ไม่พึงประสงค์ของแอปพลิเคชัน

2.1.2 การเรียนรู้ของเครื่อง (Machine Learning)

การเรียนรู้ของเครื่อง [9] เป็นสาขาหนึ่งของปัญญาประดิษฐ์ที่พัฒนามาจากการศึกษาการรู้จำแบบซึ่งเกี่ยวข้องกับการศึกษาและการสร้างอัลกอริทึมที่สามารถเรียนรู้ข้อมูลและทำนายข้อมูลได้ อัลกอริทึมนั้นจะทำงานโดยอาศัยโมเดลที่สร้างมาจากชุดข้อมูลตัวอย่างขาเข้าเพื่อเรียนรู้จากชุดข้อมูลของตัวอย่างขาเข้า การเรียนรู้ของเครื่องสามารถแบ่งโดยกว้าง ๆ ได้เป็น 5 ประเภท ตามลักษณะการใช้ข้อมูล ได้ดังนี้

(1) การเรียนรู้แบบมีผู้สอน (Supervised Learning) คือ ข้อมูลตัวอย่างและผลลัพธ์ที่ถูกป้อนเข้าสู่คอมพิวเตอร์ เป้าหมายคือการสร้างกฎทั่วไปที่สามารถเชื่อมโยงข้อมูลขาเข้ากับขาออกได้

(2) การเรียนรู้แบบไม่มีผู้สอน (Unsupervised Learning) คือ ไม่มีการทำฉลาก (label) ใดๆ และให้คอมพิวเตอร์หาโครงสร้างของข้อมูลขาเข้าเองได้

(3) การเรียนรู้แบบเสริมกำลัง (Reinforcement Learning) คือ คอมพิวเตอร์มีปฏิสัมพันธ์กับสิ่งแวดล้อมที่เปลี่ยนไปตลอดเวลาโดยคอมพิวเตอร์จะต้องทำงานบางอย่าง (เช่น ขับรถ) โดยที่ไม่มี “ผู้สอน” คอยบอกอย่างจริงจังว่าวิธีการที่ทำอยู่นั้นเข้าใกล้เป้าหมายแล้วหรือไม่ ตัวอย่างเช่น การเรียนรู้เพื่อเล่นเกม

(4) การเรียนรู้แบบกึ่งมีผู้สอน (Semi Supervised Learning) เป็นการเรียนรู้อีกแบบหนึ่งที่ระหว่างการเรียนรู้แบบมีผู้สอนกับการเรียนรู้แบบไม่มีผู้สอน โดยที่ “ผู้สอน” จะไม่สอนอย่างสมบูรณ์ นั่นคือบางข้อมูลในเซ็ตการสอนนั้นขาดข้อมูลขาออก

(5) การเรียนวิธีการเรียน (Learning to learn, Meta-learning) เป็นวิธีที่จะเรียนวิธีการเรียนรู้ของตนเอง โดยปรับปรุงพารามิเตอร์ (Parameter) ที่เป็นข้อสมมติฐานที่อัลกอริทึมใช้ในการเรียนรู้จากประสบการณ์ที่ผ่านมา

นอกจากนี้การเรียนรู้ของเครื่องยังสามารถแบ่งประเภทของงานได้ตามข้อมูลขาออกจากระบบที่เครื่องจักรได้เรียนรู้แล้ว เป็นหลายประเภทดังนี้

(1) การแบ่งประเภทข้อมูล (Classification) คือ ข้อมูลขาออกถูกแบ่งออกเป็นหลายประเภท (class)

(2) การวิเคราะห์การถดถอย (Regression) ใช้หลักการเดียวกับการแบ่งประเภทข้อมูล แต่ข้อมูลขาออกเป็นลักษณะต่อเนื่องมากกว่าเป็นประเภทแยกกัน

(3) การแบ่งกลุ่มข้อมูล (Clustering) คือ การแบ่งข้อมูลขาออกเป็นกลุ่มๆ โดยปกติแล้วมักเป็นการเรียนรู้แบบไม่มีผู้สอน

(4) การประเมินความหนาแน่น (Density Estimation) เป็นการหาการกระจายของข้อมูลในมิติบางมิติ

(5) การลดขนาดของมิติ (Dimensionality Reduction) เป็นการเชื่อมโยงข้อมูลหลายมิติไปสู่ปรภูมิที่มิติต่ำกว่า

งานวิทยานิพนธ์นี้ได้นำการเรียนรู้ด้วยเครื่องแบบมีผู้สอนมาช่วยแบ่งประเภทของข้อมูล 2 ประเภท คือ แอปพลิเคชันที่มีการส่งผ่านของข้อมูลส่วนตัวและแอปพลิเคชันที่ไม่มีการส่งผ่านข้อมูลส่วนตัว โดยผลของข้อมูลขาออกจะขึ้นอยู่กับข้อมูลขาเข้าคือเอกสารนโยบายความเป็นส่วนตัว โดยมีกระบวนการนำไปใช้ของการเรียนรู้ด้วยเครื่อง ได้เลือกแบบจำลองเพื่อประเมินผล 3 แบบจำลอง คือ การเรียนรู้ด้วยซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine) การวิเคราะห์ถดถอยโลจิสติก (Logistic Regression) และ การเรียนรู้แบบเทียบข้อมูลที่ใกล้เคียงที่สุด (K-Nearest Neighbor) ทั้ง 3 โมเดลนี้เป็นการเรียนรู้แบบมีผู้สอน

2.1.2.1 การเรียนรู้ด้วยซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine)

การเรียนรู้ด้วยซัพพอร์ตเวกเตอร์แมชชีน [9] เป็นอัลกอริทึมในการคัดแยกที่มีการนำมาใช้กันอย่างกว้างขวางในด้านการประมวลผลภาพดิจิตัล หลักการของ SVM คือการให้ข้อมูลขาเข้าที่ใช้ฝึกเป็นเวคเตอร์ในสเปซ n มิติ จากนั้นทำการสร้างไฮเปอร์เพลน (Hyperplane) ที่จะแยกกลุ่มของเวคเตอร์ข้อมูลขาเข้าเป็นประเภทต่าง ๆ ในกรณีที่เป็น 2 มิติ และ 3 มิติ ไฮเปอร์เพลน คือเส้นตรงและระนาบตามลำดับ ข้อเด่นของ SVM จะทำการเก็บแมพ (Map) เวคเตอร์ในสเปซอินพุทให้เข้าสู่ Feature Space โดยใช้ฟังก์ชั่นหรือเรียกว่าเคอร์นัล (kernel) ชนิดต่างๆ เช่น โพลีโนเมียล (Polynomial) เรเดียล (Radial) เป็นต้น โดยเส้นที่แบ่งระหว่างคลาสของข้อมูลจะถูกแบ่งโดยให้มีความห่างของข้อมูลในแต่ละคลาสมากที่สุดและเพื่อให้เกิดการทำนายผลผิดพลาดน้อยที่สุด

2.1.2.2 การวิเคราะห์ถดถอยโลจิสติก (Logistic Regression)

การวิเคราะห์ถดถอยโลจิสติก [9] เป็นเทคนิคการวิเคราะห์สถิติเชิงคุณภาพ (Qualitative Statistical Techniques) เพื่อศึกษาความสัมพันธ์ระหว่างตัวแปรตาม (Dependent Variable) และตัวแปรอิสระ (Independent Variable) แล้วนำสมการถดถอยที่ได้ไปประมาณคาตัวแปรตาม ซึ่งเทคนิคนี้ถูกพัฒนามาจาก Linear Regression โดยฟังก์ชันซิกมอยด์ (Sigmoid function) ได้ถูกนำมาใช้แปลงค่าตัวแปรตามเพื่อให้ค่าในอยู่ในช่วง [0 - 1] เพื่อใช้แก้ปัญหาการแบ่งประเภทของข้อมูล (Binary Classification)

2.1.2.3 การเรียนรู้แบบเทียบข้อมูลที่ใกล้เคียงที่สุด (K-Nearest Neighbor)

การเรียนรู้เทียบข้อมูลที่ใกล้เคียงที่สุด [9] เป็นเทคนิคใช้ในการแบ่งคลาสของข้อมูล (Label Class) ซึ่งโมเดลอื่นๆจะอาศัยเงื่อนไขในการแบ่งประเภทของข้อมูล แต่สำหรับข้อมูลที่ไม่สามารถแบ่งด้วยเงื่อนไขได้ การเทียบข้อมูลกับชุดข้อมูลก่อนหน้าที่ใกล้เคียงมากที่สุด แล้วยึดคลาสของข้อมูลนั้นเป็นคำตอบจะสามารถแก้ไขปัญหานี้ได้ โดยการทำงานของอัลกอริทึมนี้จะหาระยะทางระหว่างข้อมูลเป็นกลุ่ม ถ้ากลุ่มไหนใกล้เคียงกับข้อมูลประเภทนั้นถูกจัดประเภทอยู่ในกลุ่มนั้น

งานวิทยานิพนธ์นี้ได้เลือกการใช้การเรียนรู้ด้วยเครื่องเข้ามาช่วยประเมินความเสี่ยง จาก 3 แบบจำลองนี้ เนื่องจากเป็นแบบจำลองที่มีประสิทธิภาพในการเรียนรู้จากจำนวนของข้อมูลน้อยและแบบจำลองที่ถูกสร้างขึ้นมาเหมาะกับปัญหาการแบ่งประเภทของข้อมูล

2.1.3 การเตรียมข้อมูลประเภทเอกสารก่อนการนำไปใช้ในการเรียนรู้ด้วยเครื่อง

การเตรียมข้อมูลประเภทเอกสารก่อนนำไปใช้พัฒนาโมเดลมี 6 ขั้นตอนหลัก คือ การทำความสะอาดข้อมูล (Data Cleansing) การตัดคำให้อยู่ในรูปคำที่มีความหมาย (Tokenization) การนอร์มอลไลซ์คำ (Word Normalization) การแปลงภาษาธรรมชาติให้อยู่ในรูปของเวกเตอร์คำ (Text Data Vectorization) การทำสมดุลของข้อมูล (Balancing Data) และ การแบ่งข้อมูลในการประเมินผล (Cross Validation)

2.1.3.1 การทำความสะอาดข้อมูลประเภทเอกสาร (Data Cleansing)

ในขั้นตอนนี้จะตัดข้อมูลที่ไม่จำเป็นออกจากเอกสารเพื่อที่จะได้ข้อมูลประโยคหรือคำที่ถูกต้อง มีความหมาย และไม่ซ้ำซ้อน เช่น ข้อมูลซ้ำจากแอปพลิเคชันเดียวกันที่จัดอยู่ในหลายประเภทการใช้งานใน Google Play ออกจากชุดข้อมูลเอกสาร และ การตัดคำจำพวกคำหยุด (Stop words) เครื่องหมายอะพอสทรอฟี (Apostrophe) เครื่องหมายวรรคตอน (Punctuation)

2.1.3.2 การตัดคำให้อยู่ในรูปคำที่มีความหมาย (Tokenization)

ในขั้นตอนนี้จะตัดคำจากประโยคในทั้งเอกสารให้อยู่ในรูปของจำนวนคำ เช่น กำหนดให้การตัดคำอยู่ในรูปของคำ 1 คำ (Unigrams) ผลลัพธ์จะได้กลุ่มของคำ 1 คำ ใน เอกสารนั้นๆติดต่อกันเพื่อสะดวกต่อการนำไปวิเคราะห์และใช้งานต่อ เช่น การนับถุงคำ (Bag of Word Representation)

2.1.3.3 การนอร์มอลไลซ์คำ (Word Normalization)

ในขั้นตอนนี้เป็นการแปลงคำที่อยู่ในหลายมิติแต่ความหมายของคำเหมือนให้อยู่ในรูปของมิติเดียว เพื่อลดคำซ้ำซ้อน และ ลดการกระจายตัวของข้อมูลมากเกินไป (Sparse Matrix) ซึ่งสามารถทำได้โดยการลดรูปคำ (Stemming & Lemmatization) ทำเพื่อให้คำอยู่ในรูปพื้นฐานของคำๆนั้น โดยมีความแตกต่างกันที่วิธีการลดรูป การลดรูปแบบ Stemming จะตัดส่วนท้ายของคำออกเพื่อให้เหลือแค่รากของคำนั้นๆ ในขณะที่ Lemmatization คือการแปลงคำต่างๆ ให้อยู่ในรูปพื้นฐานของคำ

2.1.3.4 การแปลงภาษาธรรมชาติให้อยู่ในรูปของเวกเตอร์คำ (Text Data Vectorization)

ในขั้นตอนนี้เป็นการแปลงคำจากข้อความหรือเอกสารให้อยู่ในรูปของเวกเตอร์ที่สามารถสื่อความหมายได้ เช่น จำนวนของคำที่เกิดขึ้นบ่อย (Term Frequency) โดยวิทยานิพนธ์นี้ได้นำเทคนิคการแปลงภาษาธรรมชาติให้อยู่ในรูปเวกเตอร์ในมาใช้ 2 ประเภท คือ การแปลงค่าน้ำหนักความถี่ของคำที่เกิดขึ้นบนเอกสารทั้งหมด (Term

Frequency-Inverse Document Frequency) และ การแปลงประโยคให้อยู่ในรูปเวกเตอร์โดยดูจากบริบทรอบข้าง (Doc2vec)

(1) Term Frequency - Inverted Document Frequency (TF-IDF) [10] เป็นวิธีการให้ค่าน้ำหนักคำที่มีความสำคัญในตัวแทนของเอกสาร (Document) โดยค่าน้ำหนักเกิดขึ้นจากผลคูณระหว่างสองค่า คือ TF (Term Frequency) ความถี่ของคำหรือวลี (Term) ที่เกิดขึ้นในแต่ละเอกสาร กับ IDF (Inverse Document Frequency) ส่วนกลับความถี่ของคำเดียวกันที่เกิดขึ้นในหลายเอกสาร โดยมีวิธีการนับหลักๆ 2 วิธี คือ การนับแบบปกติ (Raw Count) และ การนับปรับค่าโดยวิธีการนอร์มัลไลเซชันแบบลอการิทึม (Log Normalization)

ตัวอย่างการนับแบบปกติ

สมมติว่า ฟังก์ชันของการนับคำหรือวลี ในเอกสารคือ $f(term', document)$ หมายถึงความถี่ของคำหรือวลีนั้นที่จะเกิดขึ้นบนเอกสารนั้นๆ เช่น ถ้าเอกสารตัวอย่างมีคำดังต่อไปนี้

$$\text{“}car\ bird\ fish\ car\text{”}$$ ในเอกสารชุดแรก $(document_1)$

หมายความว่า

$$f(car, document_1) = 2$$
$$f(bird, document_1) = 1$$
$$f(fish, document_1) = 1$$

โดยการนับแบบปกติ ของ $tf(term', document)$ มีค่าเท่ากับ

$$f(term, document) / \sum_{term' \in document} f(term', document)$$

เมื่อ $\sum_{term' \in document} f(term', document)$ คือ ผลรวมของความถี่ที่เกิดขึ้นของคำทั้งหมดในเอกสารชุดนั้น ซึ่งในเชิงของความถี่ของคำหรือวลี เราสามารถเพิ่มความยาวของคำหรือวลีได้ เช่น เพิ่มเป็นความถี่ของคำสองคำ (bi-gram) ติดกัน หรือ สามคำติดต่อกัน (Tri-gram)

ถ้าค่าความถี่ของแต่ละคำมีช่วงที่แตกต่างกันมาก สามารถใช้การนับแบบ Log-scale ได้เพื่อทำการปรับค่าไม่ให้แตกต่างกันมาก (Normalize) โดยใช้สูตร

$$LogTF(term, document) = 1 + Log(f(term, document))$$

ในส่วนของ Inverse Document Frequency ใช้วัดความสำคัญของคำโดยอิงจากเอกสารทั้งหมด โดยสูตร คือ

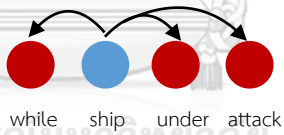
$$IDF(term, Documents) = log \frac{N}{df(term)}$$

โดย N คือ จำนวนของเอกสารทั้งหมด , $df(term)$ คือ จำนวนเอกสารที่มีกลุ่มคำหรือวลี term

เมื่อได้ค่าทั้งสองค่าแล้ว สามารถหาค่าน้ำหนักของ TF-IDF จากผลคูณทั้งสองได้ ซึ่งค่าน้ำหนักสามารถบอกถึงความสำคัญได้ เช่น ค่ายิ่งมากยิ่งเป็นคำที่มีความสำคัญกับชุดเอกสารนโยบายความเป็นส่วนตัวชุดนั้น [10]

(2) Doc2vec (Distributed Representations of Sentences and Documents) [11] เป็นวิธีการทำการแปลงคำที่ถูกพัฒนาวิธีการมาจาก Word2Vec [12] โดยข้อมูลขาเข้าให้แทนที่คำด้วยประโยคก่อนหน้าแทนแล้วทำการเลื่อนคำรอบข้างประโยคนั้น โดย Word2Vec เป็นวิธีการแปลงคำ (Word Embedding) ให้อยู่ในรูปแบบของเวกเตอร์โดยจะไม่เหมือนกับการเข้ารหัสข้อมูลแบบไบนารี่ (One-hot Encoding) ที่เป็นเวกเตอร์แทนที่คำ ๆ นั้นด้วยไบนารี่ (Binary) 0 หรือ 1 ข้อเสียคือถ้ามีคำหรือเอกสารจำนวนมาก จะทำให้เวกเตอร์นั้นมีการกระจายตัวของเลขศูนย์มากในเวกเตอร์ (Sparse Vector) ซึ่ง Word2Vec จะไม่มีปัญหานี้เพราะจะแทนที่คำหรือวลีจากบริบท (Context) รอบ ๆ ข้างแบ่งได้ออกเป็น 2 ประเภท คือ

(2.1) Skip-Gram เป็นการทำนายความน่าจะเป็น (Probability) ของคำรอบข้างโดยมีข้อมูลขาเข้า 1 คำ เพื่อหาคำถัดไปหลายคำ ตามรูปที่ 1



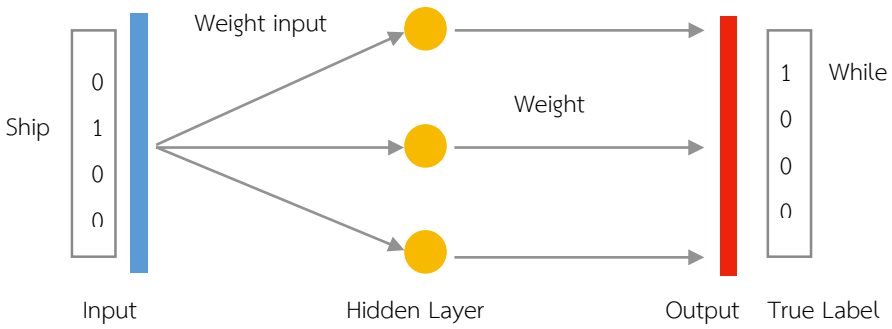
รูปที่ 1 ภาพแสดงความสัมพันธ์ของข้อมูลขาเข้ากับข้อมูลขาออก

โดยการแพร่ของข้อมูลจะเลื่อนตามตัวอย่างรูปที่ 2 กำหนดให้ หน้าต่างของข้อมูล (Window Size) มีค่า 2

(Ship, while) (Ship, under) (Ship, attack)

รูปที่ 2 ภาพแสดงลำดับการใส่ข้อมูลขาเข้าเทียบข้อมูลขาออกเพื่อเรียนรู้

โดยการเทรนข้อมูลจะเป็นการกำหนดหน้าต่างแล้วทำการเลื่อน เพื่อทำการใช้โครงสร้างประสาทเทียมแบบแพร่ไปข้างหน้า (Feed-forward Propagation) ตามรูปที่ 3 โดย สีฟ้า หมายถึง ข้อมูลขาเข้า สีแดง หมายถึง ข้อมูลขาออกที่ถูกเลื่อนไปทางขวา ตามรูปที่ 2



รูปที่ 3 ภาพโครงร่างประสาทเทียมในการเรียนรู้ของข้อมูลขาเข้า

ข้อมูลขาเข้าจะถูกแทนด้วยการเข้ารหัสข้อมูลแบบไบนารี่โดยการซ่อนของชั้น (Hidden Layer) จะหาได้จากการคูณเมทริกซ์ (Matrix) ของชั้นของข้อมูลขาเข้า (Input Layer) กับค่าน้ำหนักของข้อมูลขาเข้าตามสูตรนี้

$$H = W_{INPUT}(X^T)$$

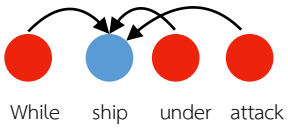
โดย H คือ เมทริกซ์การซ่อนของชั้น

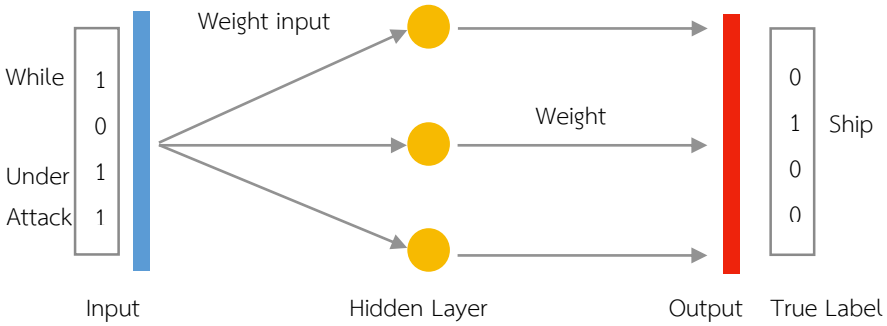
W_{INPUT} คือ ค่าน้ำหนักของข้อมูลขาเข้า

X คือ เวกเตอร์ของข้อมูลขาเข้า

ชั้นของข้อมูลขาออกสามารถหาจากการคูณเมทริกซ์ของการซ่อนของชั้น ด้วยค่าน้ำหนักของข้อมูลขาออก เมื่อได้ค่ามาแล้ว สามารถนำไปเทียบกับค่าจริง (True Label) เพื่อทำการปรับค่าน้ำหนักในแต่ละชั้นโครงประสาทเทียมได้ โดยการทำการใช้โครงสร้างประสาทเทียมแบบการส่งค่าย้อนกลับ (Back Propagation)

(2.2) Continuous Bag-of-Words (CBOW) เป็นการทำนายความน่าจะเป็นของคำจากคำรอบข้างโดยมีข้อมูลขาเข้าเข้ามาหลายคำขึ้นอยู่กับหน้าต่างที่กำหนด เช่น กำหนดให้ หน้าต่างของข้อมูล (Window Size) มีค่า 4 ตามรูปที่ 4





รูปที่ 4 ภาพแสดงความสัมพันธ์ของข้อมูลขาเข้ากับข้อมูลขาออก

โดยงานวิทยานิพนธ์นี้จะเลือกการเตรียมข้อมูลโดยการแปลงข้อมูลให้อยู่ในรูปเวกเตอร์ ที่ได้กำหนดข้างต้น เพื่อศึกษาเปรียบเทียบความแม่นยำของการให้ค่าน้ำหนักของคำด้วยความถี่ของคำและการให้ค่าน้ำหนักของคำจากบริบทรอบข้าง (Doc2vec)

2.1.3.5 การทำสมดุลของข้อมูล (Balancing Data)

ขั้นตอนนี้จะทำเมื่อข้อมูลระหว่างคลาสแต่ละคลาสมีจำนวนข้อมูลห่างกันมากเกินไป โดยเทคนิคการทำสมดุลของข้อมูลแบ่งได้เป็น 2 ประเภท คือ การเพิ่มข้อมูลของคลาสให้เท่ากัน (Over-Sampling Data) และ การลดข้อมูลของคลาสให้เท่ากัน (Under-Sampling Data) [13]

(1) การเพิ่มข้อมูลของคลาสให้เท่ากัน (Over-Sampling Data)

คือ การสุ่มเพิ่มข้อมูลกลุ่มน้อย (Minority Class) ให้เท่ากับข้อมูลกลุ่มที่มีมาก (Majority Class) โดยในงานวิทยานิพนธ์นี้ใช้ 2 ประเภท คือ การสุ่มแบบปกติ (Random Over-Sampling Data) และ การสังเคราะห์ข้อมูลใหม่โดยอิงจากกลุ่มข้อมูลชุดเดิม (Smote Over-Sampling Data) งานวิทยานิพนธ์นี้ได้ใช้วิธีนี้เพิ่มคลาสข้อมูลของการส่งผ่านข้อมูล

(2) การลดข้อมูลของคลาสให้เท่ากัน (Under-Sampling Data)

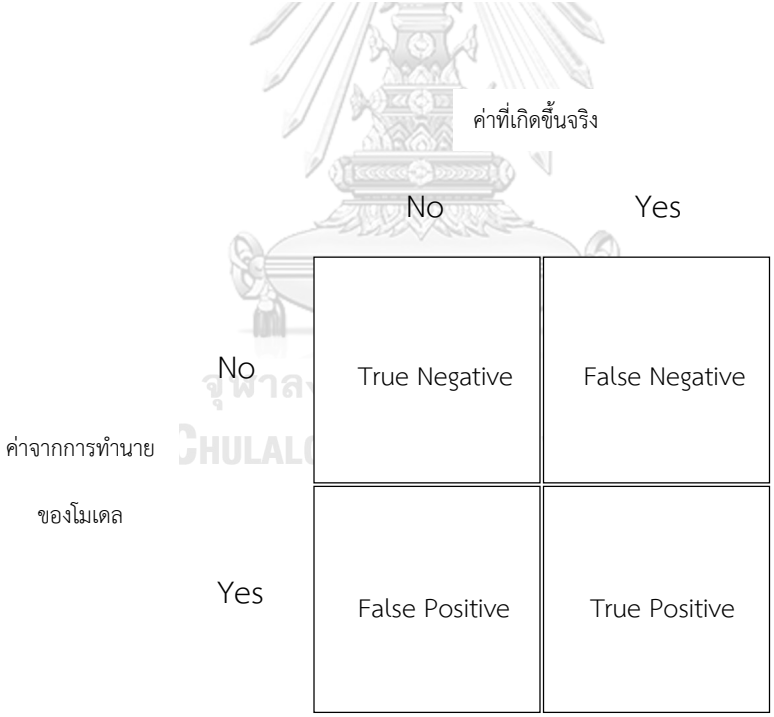
คือ การสุ่มลบข้อมูลกลุ่มที่มีชุดข้อมูลมาก ให้เท่ากับกลุ่มข้อมูลที่มีน้อย เพื่อให้ประสิทธิผลในการประเมินโมเดลมีประสิทธิภาพไม่เข้ากับชุดข้อมูลใดชุดข้อมูลหนึ่งมากเกินไป ข้อเสียคือ ถ้ากลุ่มชุดข้อมูลมีน้อยอยู่แล้วการลดข้อมูลจะทำให้ไม่เหลือข้อมูลในการเทรนโมเดล และเสี่ยงต่อการตัดข้อมูลสำคัญออกไป ในงานวิทยานิพนธ์นี้ไม่ได้ใช้วิธีนี้ เนื่องจากคลาสข้อมูลของการส่งผ่านข้อมูลมีน้อย

2.1.3.6 การออกแบบการทดลองโดยการแบ่งข้อมูลในการประเมินผล (Cross Validation)

ในขั้นตอนนี้จะทำการแบ่งชุดข้อมูลการเรียนรู้ออกเป็นส่วน ๆ ตามที่กำหนดและเหมาะสมกับชุดข้อมูลนั้น ๆ [14] เพื่อที่จะใช้นำบางส่วนของชุดข้อมูลเป็นข้อมูลตรวจสอบผลจากโมเดลที่ได้สร้างขึ้นมาจากกลุ่มชุดข้อมูลเรียนรู้ (Training Data) การแบ่งข้อมูลเป็นส่วน ๆ เรียกว่า K-Fold Cross Validation ค่า k สามารถกำหนดได้ว่าจะแบ่งชุดข้อมูลหลักออกเป็นกี่กลุ่ม ซึ่งการแบ่งข้อมูลเป็นชุด ๆ และประเมินผลหลาย ๆ ชุดข้อมูลสามารถช่วยลดการประเมินผลดีเกินไปสำหรับบางกลุ่มของข้อมูล (Overfitting Model) โดยงานวิทยานิพนธ์นี้ได้เลือกทำ K-Fold Cross Validation ในการกระจายชุดข้อมูลเรียนรู้และทดสอบเพื่อประเมินประสิทธิภาพของแบบจำลอง

2.1.4 การประเมินประสิทธิภาพของโมเดลด้วยเมทริกซ์ความสับสน (Confusion Matrix)

วิธีในการประเมินผลความแม่นยำมีได้หลายวิธีด้วยกัน ในงานวิทยานิพนธ์นี้ใช้การวัดผลของโมเดล 4 ประเภทจาก Confusion Matrix [15] คือ การประเมินผลลัพธ์จากการทำนายของโมเดล โดยเทียบกับผลลัพธ์ที่เกิดขึ้นจริง ทั้งนี้ทำให้เกิดการเปรียบเทียบโดยยกตัวอย่างให้คลาสของการทำนายมี 2 คลาสด้วยกัน คือ No และ Yes ตามรูปที่ 6. โดยแกน X เป็นตัวแทนของค่าที่เกิดขึ้นจริง และ แกน Y เป็นตัวแทนของค่าจากการทำนายของโมเดล



รูปที่ 6. ตัวอย่างตาราง Confusion Matrix 2 คลาสข้อมูล

ค่าต่าง ๆ ใน Confusion Matrix อธิบายได้ดังนี้

True Positive (TP) หมายถึง ตัวแบบทำนายว่า YES และ คลาสเป้าหมายคือ YES

False Negative (FN) หมายถึง ตัวแบบทำนายว่า NO และ คลาสเป้าหมายคือ YES

True Negative (TN) หมายถึง ตัวแบบทำนายว่า NO และ คลาสเป้าหมายคือ NO

False Positive (FP) หมายถึง ตัวแบบทำนายว่า YES และ คลาสเป้าหมายคือ NO

จาก Confusion Matrix สามารถประเมินผลได้ 4 แบบดังนี้

1) การหาความแม่นยำ (Accuracy) ในการทำนายผล หาได้จาก

$$(TP + TN) / (TP+TN+FP+FN)$$

2) Precision (Positive Predictive Value)

คือ ค่าของตัวแบบที่ทำนายได้ถูกต้อง คำนวณจากจำนวนข้อมูลที่ทำนายถูกในคลาสนั้นหารด้วยจำนวนข้อมูลทั้งหมดที่การทำนายให้ผลลัพธ์เดียวกันในคลาสนั้น หาได้จาก

$$\text{Precision ของคลาสเป้าหมาย YES} = TP / (TP + FP)$$
$$\text{Precision ของคลาสเป้าหมาย NO} = TN / (TN + FN)$$

3) Recall

คือค่าจากการทำนายด้วยตัวแบบที่ตรงกับความเป็นจริง มีค่าเท่ากับ TP Rate คำนวณจากจำนวนข้อมูลที่ทำนายถูกในคลาสนั้นหารด้วยจำนวนข้อมูลทั้งหมดในคลาสนั้น หาได้จาก

$$\text{Recall ของคลาสเป้าหมาย YES} = TP / (TP + FN)$$
$$\text{Recall ของคลาสเป้าหมาย NO} = TN / (FP + TN)$$

4) F-Measure

คือ ค่าที่เกิดจากการเปรียบเทียบระหว่างค่า Precision และ ค่า Recall ของแต่ละคลาสเป้าหมาย

$$\text{F-Measure ของคลาสเป้าหมาย YES} = (2 * \text{Precision (YES)} * \text{Recall (YES)}) / (\text{Precision (YES)} + \text{Recall (YES)})$$
$$\text{F-Measure ของคลาสเป้าหมาย NO} = (2 * \text{Precision (NO)} * \text{Recall (NO)}) / (\text{Precision (NO)} + \text{Recall (NO)})$$

ในงานวิทยานิพนธ์นี้เมื่อได้ผลการประเมินโมเดลจากทั้ง 4 ค่านี้ จะสามารถเปรียบเทียบความแม่นยำการประเมินการส่งผ่านข้อมูลของแอปพลิเคชันจากข้อความในเอกสารนโยบายความเป็นส่วนตัวได้

2.2 งานวิจัยที่เกี่ยวข้อง

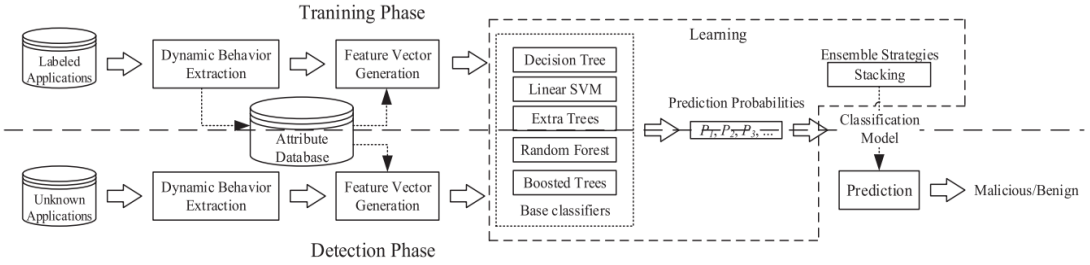
2.2.1. งานวิจัย “FlowDroid: Precise Context, Flow, Field, Object-sensitive and Lifecycle-aware Taint Analysis for Android Apps”

งานวิจัยนี้ [16] ได้เห็นถึงปัญหาของผู้ใช้สมาร์ทโฟนที่ใช้งานแอปพลิเคชันที่มีการขอเข้าถึงข้อมูลสำคัญของผู้ใช้ อาจจะเกิดจากแอปพลิเคชันไม่ได้ระมัดระวังในการออกแบบที่ดี หรือ มีความตั้งใจหาทางเก็บข้อมูลสำคัญเพื่อเป็นผลประโยชน์ของแอปพลิเคชัน ทั้งนี้งานวิจัยได้นำเสนอวิธีการตรวจสอบรหัสต้นฉบับโดยตรง (Static Taint Analysis) ในการตรวจจับข้อมูลที่มีการส่งผ่าน ได้นิยามฟังก์ชันของแหล่งที่มาของข้อมูล (Source) คือ ฟังก์ชันของข้อมูลที่มีการถูกอ่านหรือถูกเขียนโดยให้ค่าไม่คงที่ (Non-constant Variable) ลงบนรหัสแอปพลิเคชัน (Application Code) ฟังก์ชันของแหล่งเก็บข้อมูลเพื่อเตรียมส่งผ่านข้อมูล (Sink) คือ ฟังก์ชันที่เก็บข้อมูลของค่าไม่คงที่อย่างน้อยหนึ่งค่าจากแอปพลิเคชันโค้ดไว้ในพารามิเตอร์ (Parameter) โดยขั้นตอนจะเป็นการรันรหัสแอปพลิเคชันให้ครอบคลุมฟังก์ชันทั้งหมดตามวัฏจักรของแอนดรอยด์ (Android Lifecycle) เพื่อจับคู่ฟังก์ชันของแหล่งที่มาของข้อมูลที่ถูกนำค่ามาใช้ในฟังก์ชันของแหล่งเก็บข้อมูลเพื่อเตรียมส่งผ่านข้อมูล จะถูกตรวจพบว่ามีการส่งผ่านของข้อมูลเกิดขึ้น

งานวิจัยนี้มีประโยชน์ต่องานวิทยานิพนธ์นี้ โดยสามารถใช้ประเมินความเสี่ยงได้จากแพคเกจไฟล์ของแอปพลิเคชันได้ เพื่อหาข้อมูลที่แอปพลิเคชันมีการส่งผ่าน แต่เนื่องจากเป็นการวิเคราะห์ตัวแพคเกจของแอปพลิเคชันใช้เวลานานในการประเมิน งานวิทยานิพนธ์นี้จึงเสนอการวิเคราะห์นโยบายความเป็นส่วนตัวซึ่งผู้ใช้งานเข้าถึงได้ง่ายที่สุดและสามารถตรวจสอบความเสี่ยงเบื้องต้นได้รวดเร็วกว่าการวิเคราะห์ตัวแพคเกจไฟล์แอนดรอยด์

2.2.2 งานวิจัย “A Novel Dynamic Android Malware Detection System with Ensemble Learning”

งานวิจัยนี้ [17] เสนอวิธีการตรวจจับมัลแวร์ (Malware) ในแอปพลิเคชันโดยใช้การวิเคราะห์เชิงพลวัต กล่าวคือศึกษาพฤติกรรมของแอปพลิเคชันโดยการรันใช้งานแอปพลิเคชันโดยตรง และได้ศึกษาพฤติกรรมของแอปพลิเคชัน และแบ่งการกระทำของแอปพลิเคชันเป็น 10 ประเภทด้วยกัน คือ Cryptographic operation , Network operation, File Operation, Dexclass load, Information leaks, Sent SMS, Phone Calls, Service Start, Receiver Action, System Call โดยพฤติกรรมเหล่านี้สามารถแปลงให้อยู่ในรูปแบบเวกเตอร์ของโครงสร้างในการคัดเลือก และนำไปประยุกต์ใช้กับโมเดลการเรียนรู้ด้วยเครื่องเพื่อทำนายหามัลแวร์ในแอปพลิเคชันได้ งานวิจัยนี้ได้เปรียบเทียบหลายโมเดลด้วยกัน เช่น Linear SVM , Naive Bayes , KNN, Decision Tree รูปที่ 7 เป็นกระบวนการทั้งหมดในการตรวจสอบแอปพลิเคชัน



รูปที่ 7 ภาพกระบวนการวิเคราะห์มัลแวร์ของงานวิจัย [19]

งานวิจัยนี้เป็นกรณีศึกษาสำหรับการใช้การเรียนรู้ด้วยเครื่องมาใช้กับบริบทของการตรวจจับมัลแวร์บนแอปพลิเคชัน แต่การวิเคราะห์เชิงพลวัตดังกล่าว ได้นำเครื่องมือที่ชื่อ MonkeyRunner เข้ามาช่วยซึ่งเป็นการควบคุมการใช้งานหน้าจอแอปพลิเคชันเพื่อศึกษาพฤติกรรมโดยเครื่องมือตัวนี้ยังไม่ครอบคลุมทุกฟังก์ชันการทำงานบนแอปพลิเคชัน เนื่องจากเป็นการใช้งานแบบสุ่ม โดยวิทยานิพนธ์นี้จะนำการเรียนรู้ด้วยเครื่องเข้ามาช่วยวิเคราะห์ความเสี่ยงจากนโยบายความเป็นส่วนตัวโดยไม่จำเป็นต้องใช้การวิเคราะห์เชิงพลวัต

2.2.3 งานวิจัย “Towards Automatic Classification of Privacy Policy Text”

งานวิจัยนี้ [3] ใช้การเรียนรู้ด้วยเครื่องเข้ามาช่วยวิเคราะห์นโยบายความเป็นส่วนตัวเพื่อจัดประเภทของการปฏิบัติที่มีต่อข้อมูลตามที่นโยบายความเป็นส่วนตัวได้ชี้แจงไว้ โดยการปฏิบัติแต่ละประเภทมีจุดประสงค์ในการใช้ข้อมูลแตกต่างกัน แบ่งออกเป็น 9 ประเภทคือ First Party Collection/Use, Third Party Sharing/Collection, User Choice/Control, User-Access, Data Retention, Data Security, Policy Change, Do not Track, International & Specific Audiences โดยการเทรนโมเดลจะใช้ผู้เชี่ยวชาญหลายคนเพื่อวิเคราะห์รูปแบบของนโยบายความเป็นส่วนตัว (Privacy Annotation) แล้วใช้การเรียนรู้ด้วยเครื่องนำไปเรียนรู้ต่อ ได้ใช้การคำนวณค่าน้ำหนักเบื้องต้นจาก TF-IDF ก่อนเข้าแบบจำลองประเภท ซัพพอร์ตเวกเตอร์แมชชีน การวิเคราะห์ถดถอยโลจิสติก และคอนโวลูชันโครงข่ายประสาทเทียม แล้ววัดเพื่อประเมินผลของข้อมูลทั้งหมด และได้มีการนำไปใช้จริงบนเว็บไซต์ [18]

งานวิจัยนี้ไม่ได้ให้ความสำคัญกับความเสี่ยงต่อพฤติกรรมที่ไม่พึงประสงค์ของแอปพลิเคชัน งานวิทยานิพนธ์นี้ได้นำการจัดหมวดหมู่ของการปฏิบัติต่อข้อมูลในนโยบายความเป็นส่วนตัวจากงานวิจัยนี้มาช่วยคัดกรองประโยคและย่อหน้าที่ไม่ได้พูดถึงเกี่ยวกับข้อมูลส่วนตัวและตัดข้อความไม่เกี่ยวข้องออก เพื่อลดปริมาณข้อมูลที่ต้องใช้ในการเรียนร์ตามรายละเอียดในบทถัดไป

บทที่ 3
ขั้นตอนการประเมินการส่งผ่านข้อมูลของแอปพลิเคชันจากนโยบายความเป็นส่วนตัว

ในงานวิทยานิพนธ์นี้ให้ความสำคัญกับการเข้าถึงข้อมูลส่วนตัวของผู้ใช้โมไบล์แอปพลิเคชัน เนื่องจากเป็นข้อมูลที่สามารถระบุตัวตนของผู้ใช้งานได้ จึงมีความจำเป็นที่ผู้ใช้งานควรจะทราบถึงทิศทางการใช้ข้อมูลของแอปพลิเคชัน ซึ่งนโยบายของ Google Play เอกสารนโยบายความเป็นส่วนตัวของแอปพลิเคชันที่ถูกระบุไว้บนเว็บไซต์ ได้อธิบายถึงรายละเอียดการปฏิบัติต่อข้อมูลของผู้ใช้บริการ ซึ่งพฤติกรรมการใช้ข้อมูลจริงควรตรงกับสิ่งที่เขียนในข้อความของนโยบายส่วนตัว งานวิทยานิพนธ์นี้จึงได้ตั้งข้อสมมติฐาน และพิสูจน์ว่าผู้ใช้งานสามารถประเมินการส่งผ่านข้อมูลส่วนตัวเบื้องต้นจากข้อความในนโยบายความเป็นส่วนตัวได้หรือไม่ โดยออกแบบการทดลองและสร้างเครื่องมือเพื่อประเมินการส่งผ่านข้อมูลของแอปพลิเคชัน การพัฒนาเครื่องมือเพื่อที่จะประเมินการส่งผ่านข้อมูลของแอปพลิเคชันนี้ ได้นำการเรียนรู้ด้วยเครื่องเข้ามาช่วยเพื่อที่จะศึกษาพฤติกรรมการใช้ข้อมูลของแอปพลิเคชันและการใช้ภาษาจากนโยบายความเป็นส่วนตัว โดยแบ่งเป็น 4 ขั้นตอนหลัก คือ การเก็บข้อมูลเพื่อใช้ในการประเมินการส่งผ่านข้อมูลส่วนตัวของแอปพลิเคชัน การเตรียมข้อมูลเพื่อใช้ในการประเมินการส่งผ่านข้อมูลประเภทเอกสาร การเตรียมข้อมูลเพื่อใช้ในการประเมินการส่งผ่านข้อมูลประเภทไฟล์แอปพลิเคชัน และการพัฒนาแบบจำลองในการประเมินการส่งผ่านข้อมูลส่วนตัว ดังนี้

3.1 การเก็บข้อมูลเพื่อใช้ในการประเมินการส่งผ่านข้อมูลของแอปพลิเคชัน

การรวบรวมข้อมูลสำหรับการสร้างแบบจำลองในการประเมินความเสี่ยงได้แบ่งการเก็บข้อมูลเป็น 2 ส่วน คือ การเก็บข้อมูลประเภทเอกสารนโยบายความเป็นส่วนตัว และการเก็บข้อมูลประเภทไฟล์แอปพลิเคชัน โดยเอกสารนโยบายความเป็นส่วนตัวควรช่วยให้ผู้ใช้เข้าใจถึงประเภทของข้อมูลที่แอปพลิเคชันขอเข้าถึง รวมถึงเหตุผลในการเข้าถึงและการนำข้อมูลไปใช้ ซึ่งสิ่งที่เขียนในเอกสารควรจะต้องตรงกับสิ่งที่แอปพลิเคชันทำ วิทยานิพนธ์นี้ได้ทำการเก็บเอกสารนโยบายความเป็นส่วนตัวและไฟล์แอปพลิเคชันทั้งหมด 8533 แอปพลิเคชันจาก 45 ประเภทในเว็บไซต์ Google Play Store เพื่อเตรียมข้อมูลสำหรับการวิเคราะห์การประมวลผลทางภาษาธรรมชาติสำหรับเอกสารและเตรียมวิเคราะห์พฤติกรรมการส่งผ่านข้อมูลสำหรับไฟล์แอปพลิเคชัน ดังตารางที่ 2 ได้แจกแจงจำนวนไฟล์แอปพลิเคชันที่ได้เก็บข้อมูลแยกตามหมวดหมู่

ตารางที่ 2 ข้อมูลจำนวนไฟล์แอปพลิเคชันที่แยกตามหมวดหมู่จาก Google Play Store

ประเภทของแอปพลิเคชัน	จำนวนไฟล์แอปพลิเคชันและเอกสารนโยบายความเป็นส่วนตัว
GAME_EDUCATIONAL	359
GAME_CASINO	296
GAME_WORD	294
GAME_SPORTS	289
GAME_ARCADE	272
GAME_SIMULATION	269
GAME_PUZZLE	256
GAME_CARD	253
BEAUTY	251
ENTERTAINMENT	249
GAME_ROLE_PLAYING	241
GAME_BOARD	237
LIFESTYLE	237
GAME_ADVENTURE	237
HEALTH_AND_FITNESS	235
GAME_TRIVIA	234
FAMILY_ACTION	230
EDUCATION	223
TRAVEL_AND_LOCAL	221
MUSIC_AND_AUDIO	220
COMMUNICATION	211

ประเภทของแอปพลิเคชัน	จำนวนไฟล์แอปพลิเคชันและเอกสารนโยบายความเป็นส่วนตัว
ART_AND_DESIGN	203
FAMILY_BRAINGAMES	199
AUTO_AND_VEHICLES	192
MEDICAL	185
FINANCE	178
MAPS_AND_NAVIGATION	177
GAME_MUSIC	174
FAMILY_EDUCATION	173
FOOD_AND_DRINK	172
COMICS	168
EVENTS	156
DATING	151
FAMILY_CREATE	143
HOUSE_AND_HOME	139
FAMILY_MUSICVIDEO	130
NEWS_AND_MAGAZINES	126
GAME_CASUAL	108
PARENTING	96
LIBRARIES_AND_DEMO	87
BUSINESS	72
BOOKS_AND_REFERENCE	71
VIDEO_PLAYERS	70

ประเภทของแอปพลิเคชัน	จำนวนไฟล์แอปพลิเคชันและเอกสารนโยบายความเป็นส่วนตัว
FAMILY	38
GAME_RACING	11
TOTAL	8533

3.2 การเตรียมข้อมูลเพื่อใช้ในการประเมินการส่งผ่านข้อมูลสำหรับเอกสารนโยบายความเป็นส่วนตัว

ข้อความในเอกสารนโยบายความเป็นส่วนตัวมีใจความ 2 ประเภท คือ ประโยคที่ได้อธิบายเกี่ยวกับการใช้ข้อมูลของผู้ใช้ และประโยคที่ไม่ได้เกี่ยวข้องกับการอธิบายเกี่ยวกับการใช้ข้อมูลของผู้ใช้ โดยงานวิทยานิพนธ์นี้ให้ความสำคัญกับการใช้ข้อมูลของผู้ใช้เป็นหลัก วิธีการในการเตรียมข้อมูลการปฏิบัติต่อข้อมูลส่วนตัวในเอกสารแบ่งออกเป็น 3 ขั้นตอนหลัก คือ การคัดกรองข้อมูลที่มีส่วนเกี่ยวข้องกับการปฏิบัติต่อข้อมูลในเอกสารนโยบายความเป็นส่วนตัว การประมวลผลข้อมูลเบื้องต้นของนโยบายความเป็นส่วนตัว และการแปลงเวกเตอร์คำของเอกสารนโยบายความเป็นส่วนตัว

3.2.1 การคัดกรองข้อมูลที่มีส่วนเกี่ยวข้องกับการปฏิบัติต่อข้อมูลในเอกสารนโยบายความเป็นส่วนตัว

เอกสารนโยบายความเป็นส่วนตัวมีข้อความที่ยาวและมีบางส่วนในเอกสารไม่ได้อธิบายถึงการปฏิบัติต่อข้อมูลของผู้ใช้บริการ เช่น “We may response to your comments or questions, or contact you if needed while processing a product or service” [19] ซึ่งรูปความเป็นการแสดงความรับผิดชอบในการตอบคำถามในการใช้บริการต่าง ๆ ไม่ได้เกี่ยวข้องกับการปฏิบัติต่อข้อมูล งานวิทยานิพนธ์นี้ได้เห็นถึงความสำคัญของการอธิบายในการปฏิบัติต่อข้อมูลเท่านั้นจึงต้องคัดกรองใจความของเอกสารให้เหลือแต่ส่วนที่กล่าวถึงการปฏิบัติต่อข้อมูลเป็นหลัก โดยใช้ประโยชน์จากงานวิจัย [3] ที่ได้ทำคำอธิบายประกอบ (Annotation) ข้อความในนโยบายความเป็นส่วนตัว และได้แบ่งประโยคออกเป็น 10 ประเภท ดังนี้

1. First Party Collection/Use: การอธิบายถึงวิธีการและเหตุผลในการเก็บข้อมูลของผู้ใช้บริการ
2. Third Party Sharing/Collection: การอธิบายถึงการแชร์ข้อมูลให้กับผู้ให้บริการอื่น ๆ
3. User Choice/Control: การอธิบายถึงการควบคุมสิทธิของผู้ใช้งานในการอนุญาตให้ใช้ข้อมูลของผู้ใช้งาน
4. User Access, Edit, & Deletion: การอธิบายถึงการเข้าถึงข้อมูล และการแก้ไขข้อมูลของผู้ใช้งาน
5. Data Retention: การอธิบายถึงระยะเวลาที่ข้อมูลผู้ใช้งานถูกเก็บไว้
6. Data Security: การชี้แจงถึงวิธีการปกป้องความปลอดภัยของข้อมูลผู้ใช้งาน
7. Policy Change: การชี้แจงถึงวิธีการเปลี่ยนแปลงของนโยบายความเป็นส่วนตัว
8. Do Not Track: การชี้แจงถึงการติดตามข้อมูลผู้ใช้งานและการโฆษณา
9. International & Specific Audiences: การอธิบายถึงการใช้งานข้อมูลของผู้ใช้งานเฉพาะกลุ่ม
10. Other: การอธิบายเพิ่มเติมเกี่ยวกับข้อมูลที่ไม่ได้เกี่ยวข้องกับหมวดอื่น ๆ ข้างต้น

จากประเภทของการปฏิบัติต่อข้อมูลข้างต้น ประเภทที่ 1 - 9 จะเป็นการกล่าวถึงการปฏิบัติต่อข้อมูลของผู้ใช้และประเภทที่ 10 จะเป็นการกล่าวเพิ่มเติมถึงสิ่งที่ไม่ได้เกี่ยวข้องกับการปฏิบัติต่อข้อมูลของผู้ใช้ งานวิทยานิพนธ์นี้จึงใช้ประโยชน์จากคลังข้อมูล OPP-115 [19] ในการตัดประเภทของประโยคในเอกสารนโยบายความเป็นส่วนตัวที่มีส่วนเกี่ยวข้องกับประเภทที่ 10 ออกจากเอกสาร โดยมีขั้นตอน 2 วิธีดังนี้

1. การเรียนรู้ด้วยแบบจำลองจากคลังข้อมูล OPP-115 เพื่อแยกประเภทข้อความ 2. การนำแบบจำลองไปใช้ตัดประโยคที่ไม่ได้เกี่ยวข้องกับการปฏิบัติต่อข้อมูลจากนโยบายความเป็นส่วนตัว

(1) การเรียนรู้ด้วยแบบจำลองจากคลังข้อมูล OPP-115 เพื่อแยกประเภทข้อความ

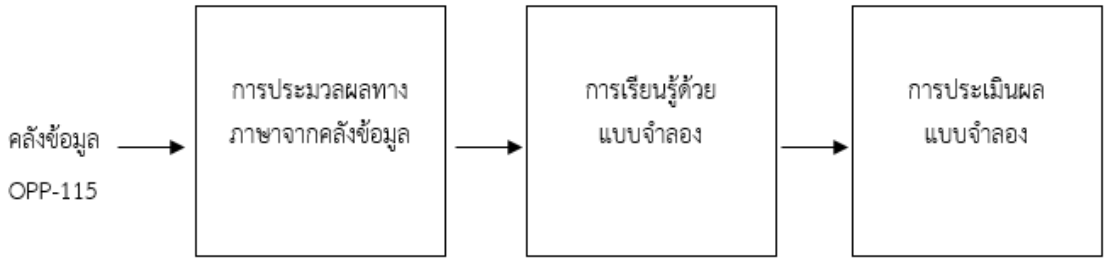
เนื่องด้วยงานวิจัย [3] ได้ตั้งใจที่จะแจ้งผู้ใช้งานอินเทอร์เน็ต (Internet) ให้ทราบถึงการปฏิบัติต่อข้อมูลของผู้ใช้บริการในการอ่านและจับใจความจากนโยบายความเป็นส่วนตัวได้ง่ายขึ้น จึงได้ทำคลังข้อมูล OPP-115 ซึ่งเป็นข้อมูลเกี่ยวกับคำอธิบายประกอบที่แบ่งประเภทประโยคนโยบายความเป็นส่วนตัว เพื่อใช้ในการเรียนรู้ด้วยแบบจำลองในการแบ่งประเภทประโยคจำนวนมากและสามารถนำไปใช้งานได้จริง จับใจความสำคัญของประโยคในเอกสารนโยบายความเป็นส่วนตัว [18] ซึ่งงานวิจัยนี้ได้เผยแพร่คลังข้อมูลให้สามารถนำไปใช้ประโยชน์ได้ ในคลังข้อมูล OPP-115 มีจำนวนเอกสารนโยบายความเป็นส่วนตัวที่ดึงมาจาก 115 เว็บไซต์ (Website) เป็นไฟล์ประเภท CSV ในตารางที่ 3 นี้แสดงถึงตัวอย่างข้อมูลของคลังข้อมูล OPP-115

ตารางที่ 3 ตัวอย่างจากไฟล์ CSV จากคลังข้อมูล OPP-115

Policy ID	Segment ID	Category name	Attribute-Value Pairs (Represented as JSON)
3635	0	Other	"SelectedText": "At the Atlantic Monthly Group, Inc. (\"The Atlantic\"), we want you to enjoy and benefit from our websites and online services secure in the knowledge that we have implemented fair information practices designed to protect your privacy. Our privacy policy is applicable to The Atlantic, and The Atlantics affiliates and subsidiaries whose websites, mobile applications and other online services are directly linked (the Sites). The privacy policy describes the kinds of information we may gather during your visit to these Sites, how we use your information, when we might disclose your personally identifiable information, and how you can manage your information."
3635	2	Policy Change	"SelectedText": "Your continued use of our Sites following the posting of changes to these terms will mean you accept those changes."

1. Policy ID คือ หมายเลขเอกสารนโยบายความเป็นส่วนตัวในคลังข้อมูล
2. Segment ID คือ ลำดับประโยคในเอกสารนโยบายความเป็นส่วนตัวของฉบับนั้น ๆ
3. Category name คือ ประเภทของประโยคในเอกสารนโยบายความเป็นส่วนตัว
4. attribute-value pairs (represented as JSON) คือ ข้อความของประโยคในเอกสารนโยบายความเป็นส่วนตัว

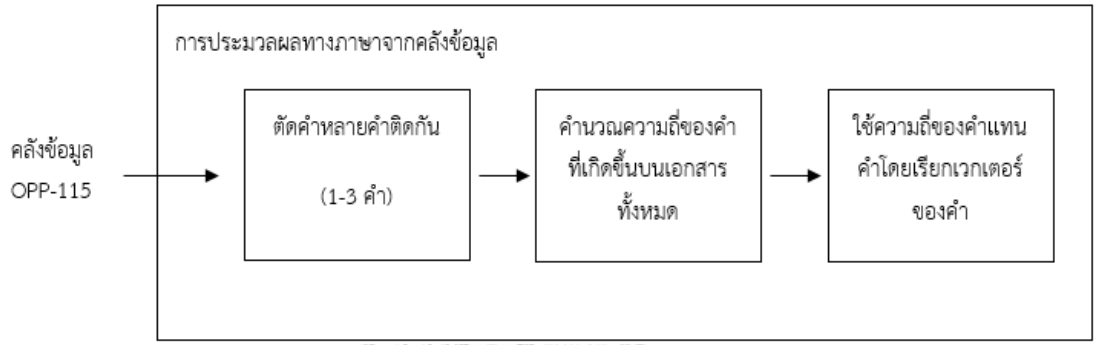
เช่น จากไฟล์ CSV ตัวอย่างแถวที่ 2 Policy ID หมายเลข 3635 แสดงถึงเอกสารนโยบายความเป็นส่วนตัวฉบับที่ 3635 จากคลังข้อมูล OPP-115 โดยมีส่วนของประโยคที่ 2 ในเอกสารฉบับนี้ มีข้อความว่า “Your continued use of our Sites following the posting of changes to these terms will mean you accept those changes.” ประโยคนี้ตรงกับประเภทการใช้ข้อมูลของผู้ใช้งานประเภทของ “Policy Change” ชี้แจงถึงวิธีการเปลี่ยนแปลงของนโยบายความเป็นส่วนตัว ซึ่งประเภทของประโยคในเอกสารนโยบายความเป็นส่วนตัวและข้อความของประโยคในเอกสารนโยบายความเป็นส่วนตัว ทั้ง 2 คอลัมน์นี้เป็นส่วนสำคัญในการแบ่งประเภทของประโยคในเอกสารนโยบายความเป็นส่วนตัว งานวิทยานิพนธ์จึงได้ใช้ประโยชน์จาก 2 คอลัมน์นี้โดยนำข้อมูลไปเรียนรู้ด้วยแบบจำลอง เพื่อนำไปวิเคราะห์ในการตัดประโยคจากนโยบายความเป็นส่วนตัว โดยกระบวนการนำคลังข้อมูลไปใช้แบ่งออกเป็น 3 งานย่อยคือ 1.การประมวลผลทางภาษาจากคลังข้อมูล 2. การเรียนรู้ด้วยแบบจำลอง 3. การประเมินผลในการนำคลังข้อมูล OPP-115 มาใช้แบ่งและตัดประเภทข้อความที่ไม่เกี่ยวข้องการปฏิบัติต่อข้อมูลจากนโยบายความเป็นส่วนตัว รูปที่ 8 แสดงถึงภาพรวมของกระบวนการนำคลังข้อมูลไปใช้ตัดประเภทของประโยคที่ไม่ได้เกี่ยวข้องกับการปฏิบัติต่อข้อมูล



รูปที่ 8 ภาพรวมกระบวนการนำคลังข้อมูลไปใช้ตัดประโยคจากนโยบายความเป็นส่วนตัว

(1.1) การประมวลผลทางภาษาจากคลังข้อมูล

การประมวลผลทางภาษาจากคลังข้อมูลเป็นกระบวนการแปลงคำในเอกสารให้อยู่ในรูปของเวกเตอร์ของคำเพื่อหาความหมายจากประโยคของเอกสารนโยบายความเป็นส่วนตัวในคลังข้อมูล OPP-115 ทั้งหมด 115 เว็บไซต์ ให้คอมพิวเตอร์สามารถเข้าใจและเรียนรู้ได้ด้วยแบบจำลอง โดยใช้การแปลงค่าน้ำหนักความถี่ของคำที่เกิดขึ้นบนเอกสารทั้งหมด (Term frequency-inverse document frequency) กำหนดการแปลงคำในเอกสารให้เป็นการตัดคำหลายคำติดกันในช่วง 1-3 คำ ดังรูปที่ 9 ที่แสดงถึงขั้นตอนการแปลงคำในเอกสารนโยบายความเป็นส่วนตัว



รูปที่ 9 ขั้นตอนการแปลงคำในเอกสารเพื่อสร้างเวกเตอร์ตัวแทนของเอกสาร

เอกสารนโยบายความเป็นส่วนตัวในคลังข้อมูล OPP-115 ถูกแปลงข้อความโดยตัดคำในช่วง 1-3 คำติดกัน เช่น

‘purposes’ ตัวแทนคำ 1 คำจากเอกสาร
‘personal information’ ตัวแทนคำ 2 คำจากเอกสาร
‘personally identifiable information’ ตัวแทนคำ 3 คำจากเอกสาร

เมื่อได้จับกลุ่มคำในเอกสารแล้ว สามารถนำกลุ่มคำเหล่านี้ไปคำนวนหาความถี่ที่ปรากฏของกลุ่มคำในเอกสารได้ เพื่อที่จะแปลงค่าความถี่ของคำที่เกิดขึ้นบนเอกสารทั้งหมด (Term frequency-inverse document frequency) วิธีการแปลงเวกเตอร์นี้ใช้ตามงานวิจัย [3] โดยกำหนดพารามิเตอร์ (parameter) ของการแปลงข้อมูลให้อยู่ในรูปเวกเตอร์ที่เหมาะสมที่สุด ดังตารางที่ 4 นี้ ซึ่งมีผลต่อประสิทธิภาพในการคัดกรองข้อมูล และตัวอย่างค่าความถี่ของคำในเอกสารจากการแปลงเวกเตอร์ ดังตารางที่ 5

ตารางที่ 4 พารามิเตอร์ของการแปลงค่าความถี่ของคำที่เกิดขึ้นบนเอกสารทั้งหมด

จำนวนคำที่ต้องการแปลงในรูปของเวกเตอร์ (Max Features)	จำนวนความถี่ที่น้อยที่สุด (Min Frequency %)	จำนวนความถี่ที่มากที่สุด (Max Frequency %)	จำนวนคำติดกัน (N-grams)
30,000 คำ	0.25%	50%	1-3 คำ

ตารางที่ 5 ตัวอย่างค่าความถี่ของคำในเอกสาร

คำในแต่ละประโยคของเอกสาร (Term)	ค่าน้ำหนักความถี่ของคำ (Weight)
news	0.30
com	0.195
collect	0.139
process	0.235
report	0.336
selected	0.13

(1.2) การเรียนรู้ด้วยแบบจำลองจากคลังข้อมูล

เพื่อให้คอมพิวเตอร์สามารถเข้าใจตัวแทนของเอกสารที่ได้แปลงค่าความถี่มาจะต้องใช้ข้อมูลขาเข้า 2 อย่างในการเรียนรู้ด้วยแบบจำลอง คือ ตัวแทนเวกเตอร์ของเอกสาร (TF-IDF Weight) และประเภทของตัวแทนเอกสารนั้น (Labeled Documents) ดังตารางที่ 6 แสดงตัวอย่างข้อมูลขาเข้าเพื่อนำไปเรียนรู้ด้วยแบบจำลอง ประกอบด้วย 2 อย่าง ในงานวิทยานิพนธ์สนใจการตัดประโยคประเภทที่ 10 หรือ “Other” ออกจากเอกสาร โดยในคอลัมน์ประเภทของ “Other” ค่า 1 แทนถึงประโยคในเอกสารนโยบายความเป็นส่วนตัวที่มีส่วนเกี่ยวข้องกับประเภทนี้ และ ค่า 0 แทนถึงประโยคในเอกสารนโยบายความเป็นส่วนตัวที่ไม่มีส่วนเกี่ยวข้องกับประเภทนี้

ตารางที่ 6 ตัวอย่างประเภทของการปฏิบัติต่อข้อมูลประเภท “Other” ในแต่ละลำดับประโยคของเอกสาร

ลำดับเวกเตอร์ประโยคในเอกสาร	ประเภทของการปฏิบัติต่อข้อมูลประเภท “Other”
กลุ่มเวกเตอร์ที่ 1	1
กลุ่มเวกเตอร์ที่ 2	0
กลุ่มเวกเตอร์ที่ 3	0
กลุ่มเวกเตอร์ที่ 4	0

เมื่อได้เวกเตอร์ข้อมูลและประเภทของการปฏิบัติต่อข้อมูลจึงนำข้อมูลเข้ากระบวนการเรียนรู้ด้วยแบบจำลอง ดังรูปที่ 10 ได้เลือกใช้แบบจำลองการวิเคราะห์ถดถอยโลจิสติก (Logistic Regression) ในการเรียนรู้ โดยอ้างอิงจากงานวิจัย [3] ที่มีค่า F1 -Score มากที่สุดในการทำนายของประโยคในเอกสารนโยบายความเป็นส่วนตัวและนำไปใช้ตัดนโยบายความเป็นส่วนตัวจากข้อมูลที่เก็บมาในหัวข้อ 3.1 ของวิทยานิพนธ์

รูปที่ 10 กระบวนการเรียนรู้ด้วยแบบจำลองเพื่อคัดกรองประโยคในเอกสารนโยบายความเป็นส่วนตัว

(1.3) การประเมินผลในการนำคลังข้อมูล OPP-115 มาใช้ตัดประโยคจากนโยบายความเป็นส่วนตัว

การนำแบบจำลองมาใช้ในการคัดกรองข้อมูลในเอกสารเป็นสิ่งสำคัญสำหรับการประเมินการส่งผ่านข้อมูลของแอปพลิเคชัน ดังนั้นแบบจำลองไม่ควรที่จะตัดประโยคที่เกี่ยวข้องกับการปฏิบัติต่อข้อมูลออกเพราะอาจทำให้สูญเสียข้อมูลที่สำคัญได้ ในการประเมินในงานวิทยานิพนธ์นี้ให้ความสำคัญกับการวัดผลของความแม่นยำในการทำนายคลาส ‘Others’ (Precision) โดยแบ่งข้อมูลเพื่อเรียนรู้และทดสอบแบบ 5-Fold Cross Validation ดังตารางที่ 7 เปรียบเทียบค่า Precision-Recall จาก 2 คลาส คือ คลาสที่เกี่ยวข้องกับ ‘Others’ (Positive Class) และไม่เกี่ยวข้องกับ ‘Others’ (Negative Class)

ตารางที่ 7 เปรียบเทียบค่า Precision-Recall ระหว่าง 2 คลาส

Class	Precision
Negative	0.93
Positive	0.85

ในการตัดประโยคที่เกี่ยวข้องกับประเภท “Others” ค่า Precision สูงหมายความว่า มีโอกาสน้อยที่แบบจำลองจะตัดประโยคที่มีความสำคัญออกไปด้วย ดังนั้นการนำแบบจำลองนี้ไปใช้จึงเป็นประโยชน์ต่อวิทยานิพนธ์นี้

(2) การนำแบบจำลองไปใช้ตัดประโยคที่ไม่ได้เกี่ยวข้องกับการปฏิบัติต่อข้อมูลจากนโยบายความเป็นส่วนตัว

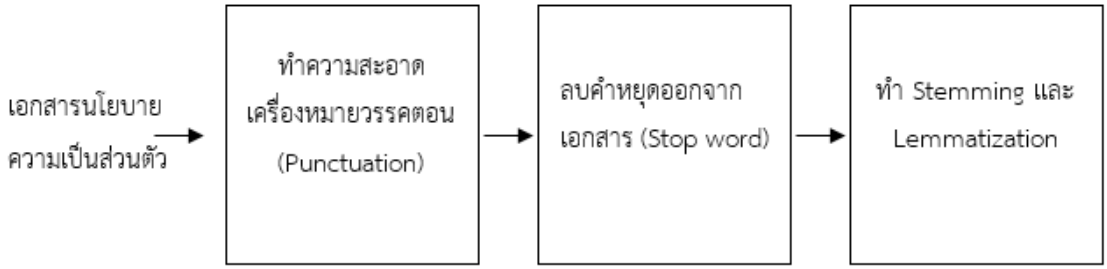
ตารางที่ 8 แสดงตัวอย่างการเปรียบเทียบความยาวของเอกสารนโยบายความเป็นส่วนตัวก่อนคัดกรองและหลังคัดกรองข้อมูลที่ไม่เกี่ยวข้องกับการปฏิบัติต่อข้อมูลด้วยแบบจำลองและหาค่าเฉลี่ยความยาวบนเอกสารทั้งหมด 8533 เอกสาร ซึ่งลดลงเฉลี่ยต่อเอกสารจาก 11914 อักษร เหลือ 5388 อักษร

ตารางที่ 8 ตัวอย่างการเปรียบเทียบความยาวของเอกสารนโยบายความเป็นส่วนตัวก่อนและหลังจากนำแบบจำลองไปใช้ตัดประโยค

ชื่อแอปพลิเคชัน	ความยาวตัวอักษรในเอกสารก่อนคัดกรองข้อมูล	ความยาวตัวอักษรหลังคัดกรองข้อมูล
jp.ne.ibis.ibispaintx.app	5989	2408
com.canva.editor	27391	17333
com.vblast.flipaclip	14042	7792
com.wallsstudio.bnk48	2922	1414
com.eyewind.paperone	11596	6964
app.over.editor	37260	14940
air.com.KalromSystems.SandDrawLite	1266	571
เฉลี่ยทั้งหมด 8533 แอปพลิเคชัน	11914	5388

3.2.2 การประมวลผลข้อมูลเบื้องต้นของนโยบายความเป็นส่วนตัว

เมื่อได้คัดกรองข้อมูลที่สำคัญจากนโยบายความเป็นส่วนตัวแล้ว ในกระบวนการประมวลผลข้อมูลเบื้องต้นจะเป็นการทำความสะอาดข้อมูลเพิ่มเติม ดังรูปที่ 11 แสดงถึงขั้นตอนทำความสะอาดข้อมูลเพื่อได้เอกสารที่พร้อมสำหรับเรียนรู้ด้วยแบบจำลอง



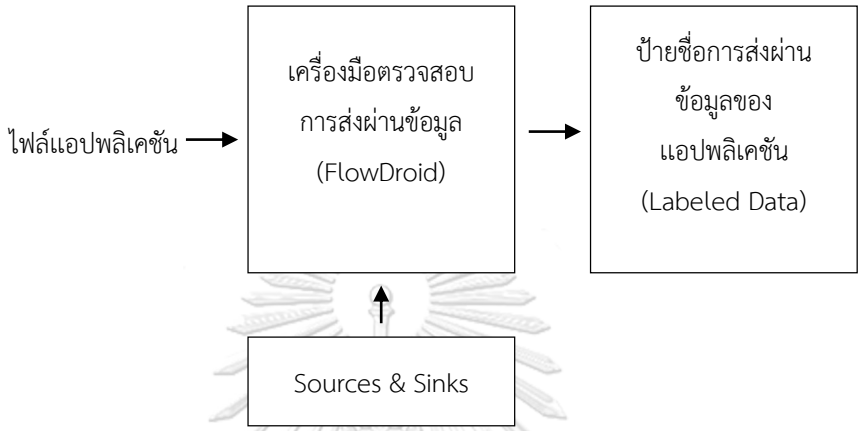
รูปที่ 11 ขั้นตอนการทำความสะอาดข้อความในเอกสารนโยบายความเป็นส่วนตัว

3.2.3 การแปลงเวกเตอร์คำของเอกสารนโยบายความเป็นส่วนตัว

งานวิทยานิพนธ์ได้เลือกวิธีการแปลงเวกเตอร์ของคำก่อนนำเข้าแบบจำลองด้วยกัน 2 วิธีดังนี้ วิธีที่ 1 การคำนวนค่าความถี่ของคำ (Term Frequency: TF) และการคำนวณค่าส่วนกลับความถี่ของคำ IDF (Inverse Document Frequency: IDF) วิธีที่ 2 การแปลงข้อมูลพารากราฟเป็นเวกเตอร์ (doc2vec) โดยการปรับพารามิเตอร์ของวิธีแปลงเวกเตอร์ของคำ งานวิทยานิพนธ์ได้ปรับพารามิเตอร์ดังนี้ 1. จำนวนคำที่ต้องการแปลงในรูปของเวกเตอร์ (Max Features) 2. จำนวนเปอร์เซ็นต์ความถี่ของคำที่เกิดขึ้นน้อยที่สุด (Min Frequency) 3.จำนวนเปอร์เซ็นต์ความถี่ของคำที่เกิดขึ้นมากที่สุด (Max Frequency) 4. จำนวนคำติดกัน (N-grams) 5. จำนวนรอบการเรียนรู้ (Epochs) 6. อัตราการเรียนรู้ (Learning rate) งานวิทยานิพนธ์ได้ปรับและเรียนรู้การพารามิเตอร์เหล่านี้เพื่อได้ผลดีและมีประสิทธิภาพให้ผลของ Recall มีค่ามากที่สุด พารามิเตอร์ดังกล่าวถูกแสดงผลใน ภาคผนวก ก ดังตารางที่ 17 และ 18

3.3 การเตรียมข้อมูลเพื่อใช้ในการประเมินการส่งผ่านข้อมูลประเภทไฟล์แอปพลิเคชัน

เพื่อที่จะศึกษาพฤติกรรมในการส่งผ่านข้อมูลส่วนตัวของแอปพลิเคชัน ได้แบ่งการเตรียมข้อมูลไฟล์แอปพลิเคชันเป็น 2 ขั้นตอน คือ 1. การตรวจสอบการส่งผ่านข้อมูลของแอปพลิเคชันด้วยเครื่องมือ FlowDroid 2. การทำป้ายชื่อการส่งผ่านข้อมูลของแอปพลิเคชัน ดังรูปที่ 12



รูปที่ 12. ขั้นตอนเตรียมข้อมูลเพื่อใช้ในการประเมินการส่งผ่านข้อมูลประเภทไฟล์แอปพลิเคชัน

ไฟล์แอปพลิเคชันจำนวน 8533 แอปพลิเคชันถูกนำไปตรวจสอบการส่งผ่านข้อมูลส่วนตัวด้วยเครื่องมือ FlowDroid โดยตัวเครื่องมือนี้จะค้นหาลำดับการใช้งานของฟังก์ชัน (Control Flow Graph) ในแอปพลิเคชันนั้น ไฟล์ข้อมูลขาเข้าที่จำเป็นต่อการใช้งาน FlowDroid คือ ไฟล์แอปพลิเคชัน และไฟล์แยกประเภทของฟังชันก์ในการส่งผ่านข้อมูล (Source & Sink) เพื่อให้ได้ป้ายชื่อการส่งผ่านข้อมูลของแอปพลิเคชันที่ได้ตรวจสอบ ตัวอย่างไฟล์แยกประเภทของฟังก์ชันในการส่งผ่านข้อมูลถูกแสดงในตารางที่ 9 และ 10

ตารางที่ 9 ตัวอย่างรายละเอียดไฟล์แยกประเภทของฟังก์ชันในการส่งผ่านข้อมูลและคำอธิบายการทำงานของฟังก์ชัน (Source-Sink File)

ฟังก์ชันแอนดรอยด์ (Android API)	แพคเกจแอนดรอยด์ (Package Android)	เวอรชันแอนดรอยด์ (Android Version)	ประเภทการส่งผ่านข้อมูล (Label)	การทำงานของฟังก์ชัน
<android.location.Location: double getLatitude()	android.location	7.1	LOCATION_INFORMATION	Get the latitude
<android.location.Location: double getLongitude()>	android.location	7.1	LOCATION_INFORMATION	Get the longitude
<android.location.LocationManager: android.location.Location getLastKnownLocation(java.lang.String)>	android.location	7.1	LOCATION_INFORMATION	Returns a Location indicating the data from the last known location fix obtained from the given provider.
<android.telephony.gsm.GsmCellLocation: int getCid()>	android.telephony	7.1	LOCATION_INFORMATION	Location Area Code is a unique number of current location area

ฟังก์ชันแอนดรอยด์ (Android API)	แพคเกจแอนดรอยด์ (Package Android)	เวอรชันแอนดรอยด์ (Android Version)	ประเภทการส่งผ่านข้อมูล (Label)	การทำงานของฟังก์ชัน
<android.telephony.gsm.GsmCellLocation: int getLac()>	android.telephony	7.1	LOCATION_INFORMATION	Location Area Code is a unique number of current location area
<android.telephony.TelephonyManager: java.lang.String getDeviceId()>	android.telephony	7.1	UNIQUE_IDENTIFIER	Returns the unique device ID, for example, the IMEI for GSM and the MEID or ESN for CDMA phones. Return null if device ID is not available.
<android.telephony.TelephonyManager: java.lang.String getSubscriberId()>	android.telephony	7.1	UNIQUE_IDENTIFIER	Returns the unique subscriber ID, for example, the IMSI for a GSM

ฟังก์ชันแอนดรอยด์ (Android API)	แพคเกจแอนดรอยด์ (Package Android)	เวอรชันแอนดรอยด์ (Android Version)	ประเภทการส่งผ่านข้อมูล (Label)	การทำงานของฟังก์ชัน
				phone. Return null if it is unavailable.
<android.telephony.TelephonyManager: java.lang.String getSimSerialNumber()>	android.telephony	7.1	UNIQUE_IDENTIFIER	Returns the serial number of the SIM, if applicable.
<android.telephony.TelephonyManager: java.lang.String getLine1Number()>	android.telephony	7.1	UNIQUE_IDENTIFIER	Returns the phone number string for line 1, for example, the MSISDN for a GSM phone. Return null if it is unavailable.

ฟังก์ชันแอนดรอยด์ (Android API)	แพคเกจแอนดรอยด์ (Package Android)	เวอรชันแอนดรอยด์ (Android Version)	ประเภทการส่งผ่านข้อมูล (Label)	การทำงานของฟังก์ชัน
<android.telephony.TelephonyManager: java.lang.String setLine1NumberForDisplay()>	android.telephony	7.1	UNIQUE_IDENTIFIER	Set the line 1 phone number string and its alpha tag for the current ICCID for display purpose only, for example, displayed in Phone Status.
<android.telephony.TelephonyManager: java.lang.String getImei()>	android.telephony	7.1	UNIQUE_IDENTIFIER	Returns the IMEI (International Mobile Equipment Identity).
<android.telephony.TelephonyManager: java.lang.String getDeviceId()>	android.telephony	7.1	UNIQUE_IDENTIFIER	Returns IMEI for GSM
<android.net.wifi.WifiInfo: java.lang.String getMacAddress()>	android.net	7.1	UNIQUE_IDENTIFIER	getMacAddress

ฟังก์ชันแอนดรอยด์ (Android API)	แพคเกจแอนดรอยด์ (Package Android)	เวอรชันแอนดรอยด์ (Android Version)	ประเภทการส่งผ่านข้อมูล (Label)	การทำงานของฟังก์ชัน
<android.net.wifi.WifiInfo: java.lang.String getSSID()>	android.net	7.1	UNIQUE_IDENTIFIER	Returns the service set identifier (SSID)
<android.bluetooth.BluetoothAdapter: java.lang.String getAddress()>	android.bluetooth	7.1	UNIQUE_IDENTIFIER	Returns the hardware address
<java.util.Locale: java.lang.String getCountry()>	java.util.Locale	7.1	USER_IDENTIFIERS	Returns the country/region code for this locale
<android.accounts.AccountManager: android.accounts.Account[] getAccounts()>	android.accounts	7.1	USER_IDENTIFIERS	Lists all accounts visible to the caller regardless of type
ฟังก์ชันอื่น ๆ ที่ไม่เกี่ยวข้องกับการส่งผ่านข้อมูล (ภาคผนวก ข)		7.1	OTHERS	

จากตารางที่ 9 แสดงถึงความสัมพันธ์ของฟังก์ชันการทำงานของแอนดรอยด์ กับ ประเภทการส่งผ่านข้อมูลส่วนตัวของแอปพลิเคชัน ซึ่งมาจากไฟล์แยกประเภทของฟังชันก์ในการส่งผ่านข้อมูล (Source-Sink File) [20] ประเภทการส่งผ่านข้อมูลที่ใช้เป็น Label มาจากคำอธิบายการทำงานของฟังก์ชันจากแพคเกจในเว็บไซต์ของแอนดรอยด์ [21] โดยฟังก์ชันที่เกี่ยวข้องกับการส่งผ่านข้อมูลอื่น ๆ “OTHERS” จะถูกอ้างอิงในภาคผนวก ข

โดยประเภทของการส่งผ่านข้อมูลส่วนตัวในงานวิทยานิพนธ์นี้ได้แบ่งออกเป็น 4 ประเภท

1.) หมายเลขเครื่อง (Unique Device Identifier) หมายถึง หมายเลขที่สามารถระบุถึงตัวเครื่องได้ เช่น International Mobile Equipment Identity (IMEI), an international mobile subscriber identity (IMSI), MAC address, Integrated Circuit Card Identifier (ICCID)
2.) ข้อมูลของผู้ใช้งาน (User Identifier) หมายถึง ข้อมูลสำคัญที่สามารถชี้ตัวถึงผู้ใช้งานหรือผู้ใช้บริการได้ เช่น ชื่อแอคเคาต์ (Account) อีเมล (Email) รหัสภูมิภาค (Region Code)
3.) ตำแหน่งที่อยู่ของผู้ใช้งาน (Location Information) หมายถึง ข้อมูลที่สามารถบอกถึงตำแหน่งที่อยู่ของผู้ใช้งานในขณะนั้น เช่น latitude, longitude, area code
4.) ข้อมูลประเภทอื่น ๆ (Others) หมายถึง ข้อมูลที่ไม่มีส่วนเกี่ยวข้องกับข้อมูลส่วนตัว เช่น รหัสไปรษณีย์

ตารางที่ 10 ตัวอย่างประเภทการส่งผ่านข้อมูลส่วนตัวของแต่ละแอปพลิเคชันที่ตรวจพบจากเครื่องมือ FlowDroid

ชื่อแอปพลิเคชัน	ประเภทของการส่งผ่านข้อมูล
jp.ne.ibis.ibispaintx.app	OTHERS
com.eyewind.paperone	OTHERS,UNIQUE_IDENTIFIER
com.landoncope.games.toddlersingandplay.christmas	LOCATION_INFORMATION
com.creative.colorfit.mandala.coloring.book	UNIQUE_IDENTIFIER, LOCATION_INFORMATION

3.4 การพัฒนาแบบจำลองในการประเมินการส่งผ่านข้อมูลส่วนตัวของแอปพลิเคชัน

การเตรียมข้อมูลในขั้นตอนที่ผ่านมาได้ข้อมูลหลัก 2 ส่วนคือ 1. เวกเตอร์ของตัวแทนเอกสารนโยบายความเป็นส่วนตัวที่สามารถพร้อมใช้ในการเรียนรู้ด้วยแบบจำลอง 2. พฤติกรรมการส่งผ่านข้อมูลส่วนตัวของแอปพลิเคชัน 4 ประเภท

ในการเลือกแบบจำลองในการเรียนรู้ในงานวิทยานิพนธ์นี้ เปรียบเทียบ 3 แบบจำลองด้วยกันในการแยกประเภทชนิดป้ายชื่อข้อมูล (Classification) คือ การเรียนรู้ด้วยซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine) การวิเคราะห์ถดถอยโลจิสติก (Logistic Regression) และการเรียนรู้แบบเทียบข้อมูลที่ใกล้เคียงที่สุด (K-Nearest Neighbor) ก่อนการพัฒนาด้วยแบบจำลอง เมื่อสำรวจข้อมูลเวกเตอร์ของตัวแทนเอกสารนโยบายความเป็นส่วนตัวและป้ายประเภทของการส่งผ่านข้อมูล พบว่าจำนวนข้อมูลในแต่ละคลาสระหว่างแอปพลิเคชันที่ไม่มีการส่งผ่านข้อมูลส่วนตัวกับแอปพลิเคชันที่มีการส่งผ่านข้อมูลส่วนตัวเกิดความไม่สมดุลของข้อมูล (Imbalance Data) ดังตารางที่ 11

ตารางที่ 11 จำนวนข้อมูลของแต่ละคลาสของแอปพลิเคชัน

ประเภทของป้ายชื่อข้อมูล	จำนวนข้อมูลของคลาสไม่มีการส่งผ่านข้อมูล	จำนวนข้อมูลของคลาสมีการส่งผ่านข้อมูล	จำนวนข้อมูลทั้งหมด
USER_IDENTIFIER	7454	1079	8533
UNIQUE_IDENTIFIER	7803	730	8533
LOCATION_INFORMATION	7878	655	8533
OTHER	5292	3241	8533

วิทยานิพนธ์นี้จึงได้พิจารณาใช้เทคนิคก่อนการพัฒนาโมเดล 2 ประเภท คือ การเพิ่มข้อมูลและการลดข้อมูล สำหรับการลดข้อมูลจะเป็นการลดขนาดข้อมูลของจำนวนข้อมูลที่มีมากกว่า เช่น จากตารางที่ 11 คลาสข้อมูล USER_IDENTIFIER มีจำนวนแอปพลิเคชันที่ไม่มีการส่งผ่านข้อมูลประเภทนี้มากกว่า เทคนิคนี้จะตัดข้อมูลบางส่วนให้จำนวนข้อมูลใกล้เคียงกับจำนวนข้อมูลของแอปพลิเคชันที่มีการส่งผ่านข้อมูล ปัญหาทำให้เกิดข้อมูลในการนำเข้าไปเรียนรู้ด้วยแบบจำลองจะน้อยและไม่สามารถเพิ่มความแม่นยำในการประเมินการส่งผ่านข้อมูลจากนโยบายความเป็นส่วนตัวได้ ดังนั้นการเพิ่มข้อมูลในส่วนของจำนวนแอปพลิเคชันที่มีการส่งผ่านข้อมูลเป็นประโยชน์มากกว่า

ในงานวิทยานิพนธ์นี้ โดยเทคนิคการเพิ่มข้อมูลได้เลือกมา 2 วิธี คือ การเพิ่มข้อมูลแบบสุ่มและการเพิ่มข้อมูลด้วยการสังเคราะห์ (Smote) แบบแรกการเพิ่มข้อมูลแบบสุ่ม คือ การสุ่มเลือกข้อมูลแล้วทำซ้ำข้อมูลชุดนั้นเพิ่มขึ้นมาอีกชุดเพื่อให้เท่ากับจำนวนข้อมูลที่มีมากกว่า แบบที่สองคือการเพิ่มข้อมูลด้วยการสังเคราะห์ คือ การสร้างข้อมูลที่มีความคล้ายกับข้อมูลเดิมทำขึ้นมาอีกชุดเพื่อให้เท่ากับจำนวนข้อมูลที่มีมากกว่าหลังจากใช้เทคนิคทำให้เกิดความสมดุลแล้ว การเข้าไปเรียนรู้ด้วยโมเดลจำเป็นต้องแบ่งข้อมูลออกเป็นสองส่วน คือ ข้อมูลสำหรับเรียนรู้ (Training Data) และข้อมูลสำหรับทดสอบเพื่อประเมินผล (Test Data) ในงานวิทยานิพนธ์นี้ไม่ได้แบ่งข้อมูลทดสอบสำหรับการปรับพารามิเตอร์ (Validation Test) แต่ได้เฉลี่ยประสิทธิภาพของโมเดลด้วย วิธีการแบ่งข้อมูลออกเป็นหลายชุดข้อมูล (K-Fold Cross-Validation) โดยแต่ละชุดข้อมูลจะมีข้อมูลทดสอบและชุดเรียนรู้ข้อมูลที่ต่างกัน ดังตารางที่ 12

ตารางที่ 12 การแบ่งข้อมูลในการเรียนรู้และการทดสอบประสิทธิภาพแบบจำลอง

5-Fold	จำนวนข้อมูลในแต่ละชุดสำหรับเรียนรู้	จำนวนข้อมูลในแต่ละชุดสำหรับทดสอบ
Fold-1	6827	1706
Fold-2	6827	1706
Fold-3	6827	1706
Fold-4	6827	1706
Fold-5	6827	1706

โดยเทคนิคการเพิ่มข้อมูลจะทำหลังจากการแบ่งข้อมูลเนื่องจากชุดข้อมูลทดสอบจะไม่มีข้อมูลที่ซ้ำในการประเมินและการแบ่งข้อมูลจะวนทำจนครบ N ครั้ง ในตารางข้างต้นกำหนด N = 5 หมายความว่ามีชุดเรียนรู้ข้อมูล 4 ชุด และชุดทดสอบ 1 ชุด ในการเรียนรู้แบบจำลอง

บทที่ 4
สรุปผลวิจัยและข้อเสนอแนะ

วิทยานิพนธ์นี้ได้นำเสนอวิธีการประเมินความเสี่ยงในการส่งผ่านข้อมูล โดยตั้งคำถามว่าสามารถประเมินการส่งผ่านข้อมูลส่วนตัวของแอปพลิเคชันจากนโยบายความเป็นส่วนตัวได้หรือไม่ ในบทนี้จะสรุปผลการทดลองเพื่อตอบข้อสมมติฐานรวมถึงอธิบายการแก้ปัญหาในเรื่องข้อจำกัดของข้อมูลโดยแบ่งออกเป็น 3 ส่วน คือ 1. การเลือกเครื่องมือวิเคราะห์พฤติกรรมการส่งผ่านของข้อมูล 2. ผลประเมินการส่งผ่านข้อมูลของแอปพลิเคชัน 3.สรุปผลวิจัยและข้อเสนอแนะ

4.1 การเลือกเครื่องมือวิเคราะห์พฤติกรรมการส่งผ่านของข้อมูล

เนื่องจากข้อเท็จจริงของการส่งผ่านข้อมูลของแอปพลิเคชันไม่ได้มีแหล่งข้อมูลที่สามารถใช้งานได้โดยตรง งานวิทยานิพนธ์นี้จึงเลือกเครื่องมือการวิเคราะห์การส่งผ่านข้อมูลโดยใช้วิธีการตรวจสอบโค้ดแบบสถิตย์ (Static Code Analysis) ด้วยเครื่องมือชื่อ FlowDroid เพื่อประเมินความเสี่ยง เครื่องมือ FlowDroid มีความแม่นยำที่สนใจแค่ส่วนที่ทำนาย (Precision) ถึง 86 เปอร์เซ็นต์และความแม่นยำที่สนใจในส่วนของความเป็นจริง (Recall) ถึง 93 เปอร์เซ็นต์ จากการทดสอบบนชุดข้อมูล Droidbench [22] เครื่องมืออื่นเช่น AppAudit [23] ได้ผสานประโยชน์ของวิธีการตรวจสอบโค้ดแบบสถิตย์ (Static Code Analysis) ที่สามารถครอบคลุมการรันของโค้ด (Code Coverage Path) กับการวิเคราะห์เชิงพลวัต (Dynamic Analysis) ซึ่งสามารถตรวจจับได้ว่ามีการส่งผ่านข้อมูลเกิดขึ้นจริง เข้าด้วยกันสามารถลดผลบวกลวง (False Positive) ที่จะเกิดขึ้นได้โดยได้ตรวจสอบกับกลุ่มมัลแวร์ (Malware) ได้แม่นยำถึง 99.3% ซึ่งมากกว่าเครื่องมือ FlowDroid แต่เนื่องจากเครื่องมือ FlowDroid สามารถปรับนิยามของการส่งผ่านข้อมูลได้ งานวิทยานิพนธ์นี้จึงเลือกใช้เครื่องมือ FlowDroid และจำกัดขอบเขตแค่การส่งผ่านข้อมูลส่วนตัวของผู้ใช้งาน (Personally Identifiable Information)

4.2 ผลประเมินการส่งผ่านข้อมูลของแอปพลิเคชัน

งานวิทยานิพนธ์ได้แสดงการเปรียบเทียบระหว่างการประเมินประสิทธิภาพของแบบจำลอง 3 โมเดล คือ การเรียนรู้ด้วยซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine) การวิเคราะห์เชิงถดถอยโลจิสติก (Logistic Regression) และ การเรียนรู้แบบเทียบข้อมูลที่ใกล้เคียงที่สุด (K-Nearest Neighbor) รวมถึงการใช้เทคนิคการเพิ่มข้อมูล 2 วิธี และการแปลงเวกเตอร์ 2 วิธี ด้วยเมทริกซ์ความแม่นยำที่สนใจแค่ส่วนที่ทำนาย (Precision) ความแม่นยำที่สนใจในส่วนของความเป็นจริง (Recall) และความแม่นยำ (Accuracy) บนชุดข้อมูลที่ผ่านการคัดกรองส่วนที่สำคัญของเอกสารโดยใช้คลังข้อมูล OPP-115 (Clean Documents) และไม่ผ่านการคัดกรองของข้อมูลส่วนสำคัญของเอกสาร (UnClean Documents) ตามตารางที่ 13 – 16

1. ป้ายชื่อข้อมูลประเภท “USER_IDENTIFIER”

ตารางที่ 13 เปรียบเทียบการนำข้อมูลในแต่ละชุดมาเรียนรู้ด้วยแบบจำลองโดยประเภทของป้ายชื่อข้อมูล “USER_IDENTIFIER”

เฉลี่ยชุดข้อมูลสำหรับสอนและทดสอบ	Logistic Regression			Support Vector Machine			K-Nearest Neighbors		
	Precision	Recall	Accuracy	Precision	Recall	Accuracy	Precision	Recall	Accuracy
ข้อมูลชุด (Clean-Random Oversampling-TF-IDF-USER-IDENTIFIER)									
Avg 5-folds CV	36.8	60.8	81.8	38.8	61.6	82.8	32.6	**68**	78.2
ข้อมูลชุด (UnClean-Random Oversampling-TF-IDF-USER-IDENTIFIER)									
Avg 5-folds CV	34.2	59.6	82.2	36.6	60.8	83.4	31.8	65	80
ข้อมูลชุด (Clean-SMOTE Oversampling-TF-IDF-USER-IDENTIFIER)									
Avg 5-folds CV	33.2	59.4	79.6	36	62.4	81	29	63.6	75.8
ข้อมูลชุด (UnClean-SMOTE Oversampling-TF-IDF-USER-IDENTIFIER)									
Avg 5-folds CV	31.2	57	80.4	33.4	60.8	81.8	26.2	60.6	76
ข้อมูลชุด (Clean-Random Oversampling-doc2vec-USER-IDENTIFIER)									
Avg 5-folds CV	30.4	60.4	77.4	32.6	60.2	79	44.8	59.4	85.4

ผลการประเมินบนป้ายชื่อข้อมูลประเภท “USER_IDENTIFIER” พบว่า การเรียนรู้แบบเทียบข้อมูลที่ใกล้เคียงที่สุด มีค่าความแม่นยำที่สนใจในส่วนของความเป็นจริง 68 เปอร์เซ็นต์ ซึ่งการทำการคัดกรองข้อมูล - เพิ่มข้อมูลแบบสุ่ม - แปลงค่าความถี่ของคำที่เกิดขึ้นบนเอกสารทั้งหมด มีค่าความแม่นยำที่สนใจในส่วนของความเป็นจริงสูงที่สุดเมื่อเปรียบเทียบกับเทคนิคอื่น ๆ ยกตัวอย่างในบรรดา 100 แอปพลิเคชัน ทำนายการส่งผ่านข้อมูลเกิดขึ้นจริงได้ 68 แอปพลิเคชัน สำหรับรายละเอียดการแบ่งข้อมูลวัดผลในแต่ละชุดอยู่ใน ภาคผนวก ค ตารางที่ 19 – 23

2. ป้ายชื่อข้อมูลประเภท “UNIQUE_IDENTIFIER”

ตารางที่ 14 เปรียบเทียบการนำข้อมูลในแต่ละชุดมาเรียนรู้ด้วยแบบจำลองโดยประเภทของป้ายชื่อข้อมูล “UNIQUE_IDENTIFIER”

เฉลี่ยชุดข้อมูลสำหรับสอนและทดสอบ	Logistic Regression			Support Vector Machine			K-Nearest Neighbors		
	Precision	Recall	Accuracy	Precision	Recall	Accuracy	Precision	Recall	Accuracy
ข้อมูลชุด (Clean-Random Oversampling-TF-IDF-UNIQUE-IDENTIFIER)									
Avg 5-folds CV	34.4	64	86.4	38.2	64.8	80.4	25.2	**76.2**	77.8
ข้อมูลชุด (UnClean-Random Oversampling-TF-IDF-UNIQUE-IDENTIFIER)									
Avg 5-folds CV	29.8	56.2	82.6	30.8	58	83	23.4	68.6	74.8
ข้อมูลชุด (Clean-SMOTE Oversampling-TF-IDF-UNIQUE-IDENTIFIER)									
Avg 5-folds CV	31.2	61.4	85	34.2	66.2	86.2	24.2	68.2	79
ข้อมูลชุด (UnClean-SMOTE Oversampling-TF-IDF-UNIQUE-IDENTIFIER)									
Avg 5-folds CV	27.8	55.2	81.6	29.4	59.6	82.4	22.6	63.2	75.4
ข้อมูลชุด (Clean-Random Oversampling-doc2vec-UNIQUE-IDENTIFIER)									
Avg 5-folds CV	33.4	67	85.6	34.6	66	86.4	46.6	61.2	90.4

ผลการประเมินบนป้ายชื่อข้อมูลประเภท “UNIQUE_IDENTIFIER” พบว่า การเรียนรู้แบบเทียบข้อมูลที่ใกล้เคียงที่สุด มีค่าความแม่นยำที่สนใจในส่วนของความเป็นจริง 76 เปอร์เซ็นต์ ซึ่งการทำการคัดกรองข้อมูล – เพิ่มข้อมูลแบบสุ่ม - แปลงค่าความถี่ของคำที่เกิดขึ้นบนเอกสารทั้งหมด มีค่าความแม่นยำที่สนใจในส่วนของความเป็นจริงสูงที่สุดเมื่อเปรียบเทียบกับเทคนิคอื่น ๆ ยกตัวอย่างในบรรดา 100 แอปพลิเคชัน ทำนายการส่งผ่านข้อมูลเกิดขึ้นจริงได้ 76 แอปพลิเคชัน สำหรับรายละเอียดการแบ่งข้อมูลวัดผลในแต่ละชุดอยู่ใน ภาคผนวก ค ตารางที่ 24 - 28

3. ป้ายชื่อข้อมูลประเภท “LOCATION_INFORMATION”

ตารางที่ 15 เปรียบเทียบการนำข้อมูลในแต่ละชุดมาเรียนรู้ด้วยแบบจำลองโดยประเภทของป้ายชื่อข้อมูล “LOCATION_INFORMATION”

เฉลี่ยชุดข้อมูลสำหรับสอนและทดสอบ	Logistic Regression			Support Vector Machine			K-Nearest Neighbors		
	Precision	Recall	Accuracy	Precision	Recall	Accuracy	Precision	Recall	Accuracy
ข้อมูลชุด (Clean-Random Oversampling-TF-IDF-LOCATION_INFORMATION)									
Avg 5-folds CV	19.2	45.2	81.2	20.8	45.2	82.6	15.6	**63**	71
ข้อมูลชุด (UnClean-Random Oversampling-TF-IDF-LOCATION_INFORMATION)									
Avg 5-folds CV	20	47.8	80.6	21.8	47.6	82.2	18.6	55.6	77.2
ข้อมูลชุด (Clean-SMOTE Oversampling-TF-IDF-LOCATION_INFORMATION)									
Avg 5-folds CV	19.2	45.6	80.8	20.8	47.2	82	17.8	53.4	77.2
ข้อมูลชุด (UnClean-SMOTE Oversampling-TF-IDF-LOCATION_INFORMATION)									
Avg 5-folds CV	18.8	46.6	79.8	20.2	48.4	81	17.6	53	76.6
ข้อมูลชุด (Clean-Random Oversampling-doc2vec-LOCATION_INFORMATION)									
Avg 5-folds CV	15.4	44	78.4	17.6	46	80.4	25.2	41.4	87.2

ผลการประเมินบนป้ายชื่อข้อมูลประเภท “LOCATION_INFORMATION” พบว่า การเรียนรู้แบบเทียบข้อมูลที่ใกล้เคียงที่สุด มีค่าความแม่นยำที่สนใจในส่วนของความเป็นจริง 63 เปอร์เซ็นต์ ซึ่งการทำการคัดกรองข้อมูล - เพิ่มข้อมูลแบบสุ่ม - แปลงค่าความถี่ของคำที่เกิดขึ้นบนเอกสารทั้งหมด มีค่าความแม่นยำที่สนใจในส่วนของความเป็นจริงสูงที่สุดเมื่อเปรียบเทียบกับเทคนิคอื่น ๆ ยกตัวอย่างในบรรดา 100 แอปพลิเคชัน ทำนายการส่งผ่านข้อมูลเกิดขึ้นจริงได้ 63 แอปพลิเคชัน สำหรับรายละเอียดการแบ่งข้อมูลวัดผลในแต่ละชุดอยู่ใน ภาคผนวก ค ตารางที่ 29 – 33

4. ป้ายชื่อข้อมูลประเภท “OTHERS”

ตารางที่ 16 เปรียบเทียบการนำข้อมูลในแต่ละชุดมาเรียนรู้ด้วยแบบจำลองโดยประเภทของป้ายชื่อข้อมูล “OTHERS”

เฉลี่ยชุดข้อมูลสำหรับสอนและทดสอบ	Logistic Regression			Support Vector Machine			K-Nearest Neighbors		
	Precision	Recall	Accuracy	Precision	Recall	Accuracy	Precision	Recall	Accuracy
ข้อมูลชุด (Clean-Random Oversampling-TF-IDF-OTHER)									
Avg 5-folds CV	72.4	**82.8**	70	74.8	75.4	69.6	73	78.4	68.4
ข้อมูลชุด (UnClean-Random Oversampling-TF-IDF-OTHER)									
Avg 5-folds CV	71	79.2	68.2	73.4	73.2	68.4	72	74	67
ข้อมูลชุด (UnClean-SMOTE Oversampling-TF-IDF-OTHER)									
Avg 5-folds CV	75.2	67	67.2	76.4	67.6	68.2	78	65.2	68.8
ข้อมูลชุด (Clean-SMOTE Oversampling-TF-IDF-OTHER)									
Avg 5-folds CV	76.8	69	68	78.2	69	68.8	78.6	64.2	66.6
ข้อมูลชุด (Clean-Random Oversampling-doc2vec-OTHER)									
Avg 5-folds CV	70.8	80.6	67.4	71.8	80	68.4	77.4	72.2	69.6

ผลการประเมินบนป้ายชื่อข้อมูลประเภท “OTHERS” พบว่า การวิเคราะห์เชิงถดถอยโลจิสติก (Logistic Regression) มีค่าความแม่นยำที่สนใจในส่วนของความเป็นจริง 82 เปอร์เซ็นต์ ซึ่งการทำการคัดกรองข้อมูล - เพิ่มข้อมูลแบบสุ่ม - แปลงค่าความถี่ของคำที่เกิดขึ้นบนเอกสารทั้งหมด มีค่าความแม่นยำที่สนใจในส่วนของความเป็นจริงสูงที่สุดเมื่อเปรียบเทียบกับเทคนิคอื่น ๆ ยกตัวอย่างในบรรดา 100 แอปพลิเคชัน ทำนายการส่งผ่านข้อมูลเกิดขึ้นจริงได้ 82 แอปพลิเคชัน สำหรับรายละเอียดการแบ่งข้อมูลวัดผลในแต่ละชุดอยู่ใน ภาคผนวก ค ในตารางที่ 34 – 38

เนื่องจากการที่มีการคัดกรองข้อมูลและเพิ่มข้อมูลแบบสุ่มให้ผลดีกว่าในทุกการทดลอง งานวิทยานิพนธ์จึงได้ประเมินแบบจำลองโดยใช้เทคนิคแปลงพารากราฟเอกสารเป็นเวกเตอร์แค่ 1 กลุ่มชุดการทดลองโดยมีการคัดกรองข้อมูลและเพิ่มข้อมูลแบบสุ่มในตารางที่ 13 - 16

4.3 สรุปผลวิจัยและข้อเสนอแนะ

จากผลการทดลองในตารางการเปรียบเทียบการเรียนรู้ป้ายชื่อประเภทการส่งผ่านข้อมูลทั้ง 4 ประเภท พบว่า ค่าความแม่นยำ (Accuracy) มีค่าสูง เพราะประเมินผลจากแอปพลิเคชันที่มีการส่งผ่านข้อมูลและไม่มีการส่งผ่านข้อมูล ด้วยข้อมูลในการเรียนรู้ของแอปพลิเคชันที่ไม่มีการส่งผ่านข้อมูลมีมากกว่า ทำให้แบบจำลองทำนายผลได้ดีบนคลาสนี้ โดยเฉลี่ยจากทั้ง 2 คลาสทำให้ค่าความแม่นยำมีค่าสูง

สำหรับค่าความแม่นยำที่สนใจในส่วนของความเป็นจริงมีค่าสูง (Recall) แสดงว่าแบบจำลองสามารถทำนายว่ามีการส่งผ่านข้อมูลเกิดขึ้นจริงเป็นส่วนใหญ่ ซึ่งสามารถนำไปประเมินการส่งผ่านข้อมูลได้จริงเบื้องต้นจากนโยบายความเป็นส่วนตัว สำหรับป้ายชื่อข้อมูลการส่งผ่านข้อมูลส่วนตัว USER_IDENTIFIER UNIQUE_IDENTIFIER และ LOCATION_INFORMATION การเรียนรู้แบบเทียบข้อมูลที่ใกล้เคียงที่สุด ได้รับผลความแม่นยำที่สนใจในส่วนของความเป็นจริงสูงที่สุด ส่วนป้ายชื่อข้อมูลการส่งผ่านข้อมูลส่วนตัว “OTHERS” การวิเคราะห์เชิงถดถอยโลจิสติก ได้รับผลความแม่นยำที่สนใจในส่วนของความเป็นจริงสูงที่สุด

สำหรับความแม่นยำที่สนใจแค่ส่วนที่ทำนาย (Precision) จากผลการทดลองทั้งหมดที่ได้ทดสอบแล้วได้ค่า Precision ต่ำ เกิดจากการที่แบบจำลองได้ทำนายว่ามีการส่งผ่านข้อมูลส่วนตัวจากการตรวจสอบข้อความในเอกสารนโยบายความเป็นส่วนตัว ทั้งที่ในความจริงไม่มีการส่งผ่านข้อมูลของแอปพลิเคชันเกิดขึ้น ดังนั้นการที่มีค่าความแม่นยำที่สนใจในส่วนของความเป็นจริงมีค่าสูง (Recall) และความแม่นยำที่สนใจแค่ส่วนที่ทำนาย (Precision) มีค่าต่ำ ทำให้แบบจำลองทำนายการส่งผ่านข้อมูลที่เกิดขึ้นจริงได้ถูกต้อง แต่จะทำนายผลว่ามีการส่งผ่านข้อมูลเกินความเป็นจริงเมื่อแอปพลิเคชันไม่ได้มีการส่งผ่านข้อมูล การนำแบบจำลองไปใช้ในการประเมินจะทำนายว่ามีการส่งผ่านข้อมูลเยอะกว่าความเป็นจริง

ทั้งนี้การพัฒนางานวิจัยต่อไปในอนาคตอาจพิจารณาเทคนิคอื่นเพิ่มเติม เช่น การเพิ่มปัจจัยอื่น ๆ ที่น่าจะมีผลต่อการส่งผ่านข้อมูล การเพิ่มข้อมูลการเรียนรู้แทนการสังเคราะห์ข้อมูลเพิ่ม เนื่องจากป้ายชื่อ “Other” ที่มีข้อมูลของทั้ง 2 คลาสไม่ห่างกันมาก ส่งผลให้ Precision และ Recall สูงกว่าป้ายชื่ออื่น ๆ ดังนั้นการเพิ่มข้อมูลจึงเป็นอีกหนึ่งวิธีที่สามารถเพิ่มค่าความแม่นยำได้มากกว่าการสังเคราะห์ข้อมูลเพิ่ม รวมไปถึงเทคนิคการปรับพารามิเตอร์ของแบบจำลอง เนื่องในงานวิทยานิพนธ์นี้ใช้ค่าเริ่มต้นของแบบจำลอง การใช้แบบจำลองอื่น ๆ ในการประเมินผล และการปรับสัดส่วนของความน่าจะเป็นที่ทำนายว่ามีการส่งผ่านข้อมูลหรือไม่มีการส่งผ่านข้อมูลของแอปพลิเคชัน โดยสร้างกราฟ ROC Curve เพื่อดูว่าจุดตัดไหนให้ผลของ True Positive Rate มีค่าสูงสุด โดยทำให้เกิด False Positive Rate มีค่าน้อยที่สุด เทคนิคที่กล่าวมาสามารถเพิ่มค่าความแม่นยำที่สนใจแค่ส่วนที่ทำนาย (Precision) และความแม่นยำที่สนใจในส่วนของความเป็นจริง (Recall) ได้

ภาคผนวก ก

ตารางที่ 17 และ 18 แสดงถึงการปรับพารามิเตอร์ที่เหมาะสมเพื่อแปลงเอกสารนโยบายความเป็นส่วนตัวให้อยู่ในรูปของเวกเตอร์เพื่อให้คอมพิวเตอร์หรือแบบจำลองสามารถเข้าใจตัวแทนของเอกสารได้

ตารางที่ 17 พารามิเตอร์ของการแปลงค่าความถี่ของคำที่เกิดขึ้นบนเอกสารทั้งหมด (TF-IDF)

จำนวนคำที่ต้องการแปลงในรูปของเวกเตอร์ (Max Features)	จำนวนความถี่ที่น้อยที่สุด (Min Frequency %)	จำนวนความถี่ที่มากที่สุด (Max Frequency %)	จำนวนคำติดกัน (N-grams)
30,000 คำ	20%	80%	2-3 คำ

ตารางที่ 18 พารามิเตอร์ของการแปลงข้อมูลพารากราฟเป็นเวกเตอร์ (Doc2vec)

จำนวนรอบการรัน (Epoch)	ขนาดเวกเตอร์ (Vector Size)	ค่าการเรียนรู้ (Learning rate)	ค่าพารามิเตอร์อื่น ๆ
10	3000	0.0025	ค่าเริ่มต้น (Default)

ภาคผนวก ข

<org.apache.http.HttpResponse: org.apache.http.HttpEntity getEntity()> -> _SOURCE_ (NO_CATEGORY)
<org.apache.http.util.EntityUtils: java.lang.String toString(org.apache.http.HttpEntity)> -> _SOURCE_ (NO_CATEGORY)
<org.apache.http.HttpResponse: org.apache.http.StatusLine getStatusLine()> -> _SOURCE_ (NO_CATEGORY)

<android.location.Location: double getLatitude()> -> _SOURCE_ (LOCATION_INFORMATION)
<android.location.Location: double getLongitude()> -> _SOURCE_ (LOCATION_INFORMATION)
<android.location.LocationManager: android.location.Location getLastKnownLocation(java.lang.String)> -> _SOURCE_ (LOCATION_INFORMATION)

<android.telephony.TelephonyManager: java.lang.String getDeviceId()>
android.permission.READ_PHONE_STATE -> _SOURCE_ (UNIQUE_IDENTIFIER)
<android.telephony.TelephonyManager: java.lang.String getSubscriberId()>
android.permission.READ_PHONE_STATE -> _SOURCE_ (UNIQUE_IDENTIFIER)
<android.telephony.TelephonyManager: java.lang.String getSimSerialNumber()>
android.permission.READ_PHONE_STATE -> _SOURCE_ (UNIQUE_IDENTIFIER)
<android.telephony.TelephonyManager: java.lang.String getLine1Number()>
android.permission.READ_PHONE_STATE -> _SOURCE_ (UNIQUE_IDENTIFIER)
<android.telephony.TelephonyManager: java.lang.String setLine1NumberForDisplay()>
android.permission.READ_PHONE_STATE -> _SOURCE_ (UNIQUE_IDENTIFIER)
<android.telephony.TelephonyManager: java.lang.String getImei()>
android.permission.READ_PHONE_STATE -> _SOURCE_ (UNIQUE_IDENTIFIER)
<android.telephony.TelephonyManager: java.lang.String getIccAuthentication()>
android.permission.READ_PHONE_STATE -> _SOURCE_ (UNIQUE_IDENTIFIER)
<android.telephony.TelephonyManager: java.lang.String getDeviceId()>
android.permission.READ_PRIVILEGED_PHONE_STATE -> _SOURCE_ (UNIQUE_IDENTIFIER)
<android.telephony.TelephonyManager: java.lang.String getSubscriberId()>
android.permission.READ_PRIVILEGED_PHONE_STATE -> _SOURCE_ (UNIQUE_IDENTIFIER)
<android.telephony.TelephonyManager: java.lang.String getSimSerialNumber()>
android.permission.READ_PRIVILEGED_PHONE_STATE -> _SOURCE_ (UNIQUE_IDENTIFIER)

<android.telephony.TelephonyManager: java.lang.String getLine1Number()>
android.permission.READ_PRIVILEGED_PHONE_STATE -> _SOURCE_ (UNIQUE_IDENTIFIER)
<android.telephony.TelephonyManager: java.lang.String setLine1NumberForDisplay()>
android.permission.READ_PRIVILEGED_PHONE_STATE -> _SOURCE_ (UNIQUE_IDENTIFIER)
<android.telephony.TelephonyManager: java.lang.String getImei()>
android.permission.READ_PRIVILEGED_PHONE_STATE -> _SOURCE_ (UNIQUE_IDENTIFIER)
<android.telephony.TelephonyManager: java.lang.String getIccAuthentication()>
android.permission.READ_PRIVILEGED_PHONE_STATE -> _SOURCE_ (UNIQUE_IDENTIFIER)

<java.net.URLConnection: void connect()> -> _SINK_ (NO_CATEGORY)
<java.net.URLConnection: java.io.InputStream getInputStream()> -> _BOTH_ (NO_CATEGORY)
<java.net.URLConnection: java.io.OutputStream getOutputStream()> -> _SINK_ (NO_CATEGORY)

<java.net.URL: java.io.InputStream openStream()> -> _BOTH_ (NO_CATEGORY)
<java.net.URL: java.lang.Object getContent()> -> _BOTH_ (NO_CATEGORY)
<java.net.URL: java.lang.Object getContent(java.lang.Class[])> -> _BOTH_ (NO_CATEGORY)

<java.net.URL: void set(java.lang.String,java.lang.String,int,java.lang.String,java.lang.String)> ->
SINK (NO_CATEGORY)
<java.net.URL: void
set(java.lang.String,java.lang.String,int,java.lang.String,java.lang.String,java.lang.String,java.lang.String,j
ava.lang.String)> -> _SINK_ (NO_CATEGORY)

<org.apache.http.HttpResponse: org.apache.http.HttpEntity getEntity()> -> _SOURCE_ (OTHERS)

%Covered by the EasyTaintWrapper given that the HttpEntity is tainted
%<org.apache.http.util.EntityUtils: java.lang.String toString(org.apache.http.HttpEntity)> ->
SOURCE (OTHERS)
%<org.apache.http.util.EntityUtils: java.lang.String
toString(org.apache.http.HttpEntity,java.lang.String)> -> _SOURCE_ (OTHERS)
%<org.apache.http.util.EntityUtils: byte[] toByteArray(org.apache.http.HttpEntity)> -> _SOURCE_
(OTHERS)
%<org.apache.http.util.EntityUtils: java.lang.String getContentCharSet(org.apache.http.HttpEntity)>
-> _SOURCE_ (OTHERS)

%add Activity.getIntent() as source instead of the next methods to avoid duplicate results.
%<android.content.Intent: java.lang.String getAction()> -> _SOURCE_ (NO_CATEGORY)
%<android.content.Intent: boolean[] getBooleanArrayExtra(java.lang.String)> -> _SOURCE_ (NO_CATEGORY)
%<android.content.Intent: boolean getBooleanExtra(java.lang.String, boolean)> -> _SOURCE_ (NO_CATEGORY)
%<android.content.Intent: android.os.Bundle getBundleExtra(java.lang.String)> -> _SOURCE_ (NO_CATEGORY)
%<android.content.Intent: byte[] getByteArrayExtra(java.lang.String)> -> _SOURCE_ (NO_CATEGORY)
%<android.content.Intent: byte getByteExtra(java.lang.String, byte)> -> _SOURCE_ (NO_CATEGORY)
%<android.content.Intent: java.util.Set getCategories()> -> _SOURCE_ (NO_CATEGORY)
%<android.content.Intent: char[] getCharArrayExtra(java.lang.String)> -> _SOURCE_ (NO_CATEGORY)
%<android.content.Intent: char getCharExtra(java.lang.String, char)> -> _SOURCE_ (NO_CATEGORY)
%<android.content.Intent: java.lang.CharSequence[] getCharSequenceArrayExtra(java.lang.String)> -> _SOURCE_ (NO_CATEGORY)
%<android.content.Intent: java.util.ArrayList getCharSequenceArrayListExtra(java.lang.String)> -> _SOURCE_ (NO_CATEGORY)
%<android.content.Intent: java.lang.CharSequence getCharSequenceExtra(java.lang.String)> -> _SOURCE_ (NO_CATEGORY)
%<android.content.Intent: android.content.ClipData getClipData()> -> _SOURCE_ (NO_CATEGORY)
%<android.content.Intent: android.content.ComponentName getComponent()> -> _SOURCE_ (NO_CATEGORY)
%<android.content.Intent: android.net.Uri getData()> -> _SOURCE_ (NO_CATEGORY)
%<android.content.Intent: java.lang.String getDataString()> -> _SOURCE_ (NO_CATEGORY)
%<android.content.Intent: double[] getDoubleArrayExtra(java.lang.String)> -> _SOURCE_ (NO_CATEGORY)
%<android.content.Intent: double getDoubleExtra(java.lang.String, double)> -> _SOURCE_ (NO_CATEGORY)
%<android.content.Intent: android.os.Bundle getExtras()> -> _SOURCE_ (NO_CATEGORY)
%<android.content.Intent: int getFlags()> -> _SOURCE_ (NO_CATEGORY)
%<android.content.Intent: float[] getFloatArrayExtra(java.lang.String)> -> _SOURCE_ (NO_CATEGORY)

%<android.content.Intent: float getFloatExtra(java.lang.String, float)> -> _SOURCE_ (NO_CATEGORY)

%<android.content.Intent: int[] getIntArrayExtra(java.lang.String)> -> _SOURCE_ (NO_CATEGORY)

%<android.content.Intent: java.util.ArrayList getIntegerArrayListExtra(java.lang.String)> -> _SOURCE_ (NO_CATEGORY)

%<android.content.Intent: android.content.Intent getIntent(java.lang.String)> -> _SOURCE_ (NO_CATEGORY)

%<android.content.Intent: android.content.Intent getIntentOld(java.lang.String)> -> _SOURCE_ (NO_CATEGORY)

%<android.content.Intent: int getIntExtra(java.lang.String, int)> -> _SOURCE_ (NO_CATEGORY)

%<android.content.Intent: long[] getLongArrayExtra(java.lang.String)> -> _SOURCE_ (NO_CATEGORY)

%<android.content.Intent: long getLongExtra(java.lang.String, long)> -> _SOURCE_ (NO_CATEGORY)

%<android.content.Intent: java.lang.String getPackage()> -> _SOURCE_ (NO_CATEGORY)

%<android.content.Intent: android.os.Parcelable[] getParcelableArrayExtra(java.lang.String)> -> _SOURCE_ (NO_CATEGORY)

%<android.content.Intent: java.util.ArrayList getParcelableArrayListExtra(java.lang.String)> -> _SOURCE_ (NO_CATEGORY)

%<android.content.Intent: android.os.Parcelable getParcelableExtra(java.lang.String)> -> _SOURCE_ (NO_CATEGORY)

%<android.content.Intent: java.lang.String getScheme()> -> _SOURCE_ (NO_CATEGORY)

%<android.content.Intent: android.content.Intent getSelector()> -> _SOURCE_ (NO_CATEGORY)

%<android.content.Intent: java.io.Serializable getSerializableExtra(java.lang.String)> -> _SOURCE_ (NO_CATEGORY)

%<android.content.Intent: short[] getShortArrayExtra(java.lang.String)> -> _SOURCE_ (NO_CATEGORY)

%<android.content.Intent: short getShortExtra(java.lang.String, short)> -> _SOURCE_ (NO_CATEGORY)

%<android.content.Intent: android.graphics.Rect getSourceBounds()> -> _SOURCE_ (NO_CATEGORY)

%<android.content.Intent: java.lang.String[] getStringArrayExtra(java.lang.String)> -> _SOURCE_ (NO_CATEGORY)

%<android.content.Intent: java.util.ArrayList getStringArrayListExtra(java.lang.String)> -> _SOURCE_ (NO_CATEGORY)

%<android.content.Intent: java.lang.String getStringExtra(java.lang.String)> -> _SOURCE_ (NO_CATEGORY)
%<android.content.Intent: java.lang.String getType()> -> _SOURCE_ (NO_CATEGORY)

%<android.content.Intent: void <init>()> -> _SOURCE_ (NO_CATEGORY)
%<android.content.Intent: void <init>(android.content.Intent)> -> _SOURCE_ (NO_CATEGORY)
%<android.content.Intent: void <init>(java.lang.String)> -> _SOURCE_ (NO_CATEGORY)
%<android.content.Intent: void <init>(java.lang.String,android.net.Uri)> -> _SOURCE_ (NO_CATEGORY)
%<android.content.Intent: void <init>(android.content.Context,java.lang.Class)> -> _SOURCE_ (NO_CATEGORY)
%<android.content.Intent: void <init>(java.lang.String,android.net.Uri,android.content.Context,java.lang.Class)> -> _SOURCE_ (NO_CATEGORY)

%bundle sources
 %do not consider them as sources, because we have the callback parameters from
 %which the apps obtain the bundles as sources anyway
%<android.os.Bundle: java.lang.Object get(java.lang.String)> -> _SOURCE_ (NO_CATEGORY)
%<android.os.Bundle: boolean getBoolean(java.lang.String)> -> _SOURCE_ (NO_CATEGORY)
%<android.os.Bundle: boolean getBoolean(java.lang.String,boolean)> -> _SOURCE_ (NO_CATEGORY)
%<android.os.Bundle: boolean[] getBooleanArray(java.lang.String)> -> _SOURCE_ (NO_CATEGORY)
%<android.os.Bundle: android.os.Bundle getBundle(java.lang.String)> -> _SOURCE_ (NO_CATEGORY)
%<android.os.Bundle: byte getByte(java.lang.String)> -> _SOURCE_ (NO_CATEGORY)
%<android.os.Bundle: java.lang.Byte getByte(java.lang.String,byte)> -> _SOURCE_ (NO_CATEGORY)
%<android.os.Bundle: byte[] getByteArray(java.lang.String)> -> _SOURCE_ (NO_CATEGORY)
%<android.os.Bundle: char getChar(java.lang.String)> -> _SOURCE_ (NO_CATEGORY)
%<android.os.Bundle: char getChar(java.lang.String,char)> -> _SOURCE_ (NO_CATEGORY)
%<android.os.Bundle: char[] getCharArray(java.lang.String)> -> _SOURCE_ (NO_CATEGORY)
%<android.os.Bundle: java.lang.CharSequence getCharSequence(java.lang.String,java.lang.CharSequence)> -> _SOURCE_ (NO_CATEGORY)
%<android.os.Bundle: java.lang.CharSequence getCharSequence(java.lang.String)> -> _SOURCE_ (NO_CATEGORY)

%<android.os.Bundle: java.lang.CharSequence[] getCharSequenceArray(java.lang.String)> -> _SOURCE_ (NO_CATEGORY)

%<android.os.Bundle: java.util.ArrayList getCharSequenceArrayList(java.lang.String)> -> _SOURCE_ (NO_CATEGORY)

%<android.os.Bundle: java.lang.ClassLoader getClassLoader()> -> _SOURCE_ (NO_CATEGORY)

%<android.os.Bundle: double getDouble(java.lang.String)> -> _SOURCE_ (NO_CATEGORY)

%<android.os.Bundle: double getDouble(java.lang.String,double)> -> _SOURCE_ (NO_CATEGORY)

%<android.os.Bundle: double[] getDoubleArray(java.lang.String)> -> _SOURCE_ (NO_CATEGORY)

%<android.os.Bundle: float getFloat(java.lang.String)> -> _SOURCE_ (NO_CATEGORY)

%<android.os.Bundle: float getFloat(java.lang.String,float)> -> _SOURCE_ (NO_CATEGORY)

%<android.os.Bundle: float[] getFloatArray(java.lang.String)> -> _SOURCE_ (NO_CATEGORY)

%<android.os.Bundle: int getInt(java.lang.String,int)> -> _SOURCE_ (NO_CATEGORY)

%<android.os.Bundle: int getInt(java.lang.String)> -> _SOURCE_ (NO_CATEGORY)

%<android.os.Bundle: int[] getIntArray(java.lang.String)> -> _SOURCE_ (NO_CATEGORY)

%<android.os.Bundle: java.util.ArrayList getIntegerArrayList(java.lang.String)> -> _SOURCE_ (NO_CATEGORY)

%<android.os.Bundle: long getLong(java.lang.String)> -> _SOURCE_ (NO_CATEGORY)

%<android.os.Bundle: long getLong(java.lang.String,long)> -> _SOURCE_ (NO_CATEGORY)

%<android.os.Bundle: long[] getLongArray(java.lang.String)> -> _SOURCE_ (NO_CATEGORY)

%<android.os.Bundle: android.os.Parcelable getParcelable(java.lang.String)> -> _SOURCE_ (NO_CATEGORY)

%<android.os.Bundle: android.os.Parcelable[] getParcelableArray(java.lang.String)> -> _SOURCE_ (NO_CATEGORY)

%<android.os.Bundle: java.util.ArrayList getParcelableArrayList(java.lang.String)> -> _SOURCE_ (NO_CATEGORY)

%<android.os.Bundle: java.io.Serializable getSerializable(java.lang.String)> -> _SOURCE_ (NO_CATEGORY)

%<android.os.Bundle: short getShort(java.lang.String)> -> _SOURCE_ (NO_CATEGORY)

%<android.os.Bundle: short getShort(java.lang.String,short)> -> _SOURCE_ (NO_CATEGORY)

%<android.os.Bundle: short[] getShortArray(java.lang.String)> -> _SOURCE_ (NO_CATEGORY)

%<android.os.Bundle: android.util.SparseArray getSparseParcelableArray(java.lang.String)> -> _SOURCE_ (NO_CATEGORY)

%<android.os.Bundle: java.lang.String getString(java.lang.String)> -> _SOURCE_ (NO_CATEGORY)

%<android.os.Bundle: java.util.ArrayList getStringArrayList(java.lang.String key)> -> _SOURCE_ (NO_CATEGORY)

%bundle sinks
<android.os.Bundle: void putBinder(java.lang.String,android.os.IBinder)> -> _SINK_ (NO_CATEGORY)
<android.os.Bundle: void putBoolean(java.lang.String,boolean)> -> _SINK_ (NO_CATEGORY)
<android.os.Bundle: void putBooleanArray(java.lang.String,boolean[])> -> _SINK_ (NO_CATEGORY)
<android.os.Bundle: void putBundle(java.lang.String,android.os.Bundle)> -> _SINK_
(NO_CATEGORY)
<android.os.Bundle: void putByte(java.lang.String,byte)> -> _SINK_ (NO_CATEGORY)
<android.os.Bundle: void putByteArray(java.lang.String,byte[])> -> _SINK_ (NO_CATEGORY)
<android.os.Bundle: void putChar(java.lang.String,char)> -> _SINK_ (NO_CATEGORY)
<android.os.Bundle: void putCharArray(java.lang.String,char[])> -> _SINK_ (NO_CATEGORY)
<android.os.Bundle: void putCharSequence(java.lang.String,java.lang.CharSequence)> -> _SINK_
(NO_CATEGORY)
<android.os.Bundle: void putCharSequenceArray(java.lang.String,java.lang.CharSequence[])> ->
SINK (NO_CATEGORY)
<android.os.Bundle: void putCharSequenceArrayList(java.lang.String,java.util.ArrayList)> -> _SINK_
(NO_CATEGORY)
<android.os.Bundle: void putDouble(java.lang.String,double)> -> _SINK_ (NO_CATEGORY)
<android.os.Bundle: void putDoubleArray(java.lang.String,double[])> -> _SINK_ (NO_CATEGORY)
<android.os.Bundle: void putFloat(java.lang.String,float)> -> _SINK_ (NO_CATEGORY)
<android.os.Bundle: void putFloatArray(java.lang.String,float[])> -> _SINK_ (NO_CATEGORY)
<android.os.Bundle: void putInt(java.lang.String,int)> -> _SINK_ (NO_CATEGORY)
<android.os.Bundle: void putIntArray(java.lang.String,int[])> -> _SINK_ (NO_CATEGORY)
<android.os.Bundle: void putIntegerArrayList(java.lang.String,java.util.ArrayList)> -> _SINK_
(NO_CATEGORY)
<android.os.Bundle: void putLong(java.lang.String,long)> -> _SINK_ (NO_CATEGORY)
<android.os.Bundle: void putLongArray(java.lang.String,long[])> -> _SINK_ (NO_CATEGORY)
<android.os.Bundle: void putParcelable(java.lang.String,android.os.Parcelable)> -> _SINK_
(NO_CATEGORY)
<android.os.Bundle: void putParcelableArray(java.lang.String,android.os.Parcelable[])> -> _SINK_
(NO_CATEGORY)
<android.os.Bundle: void putParcelableArrayList(java.lang.String,java.util.ArrayList)> -> _SINK_
(NO_CATEGORY)
<android.os.Bundle: void putSerializable(java.lang.String,java.io.Serializable)> -> _SINK_
(NO_CATEGORY)

<android.os.Bundle: void putShort(java.lang.String,short)> -> _SINK_ (NO_CATEGORY)
<android.os.Bundle: void putShortArray(java.lang.String,short[])> -> _SINK_ (NO_CATEGORY)
<android.os.Bundle: void putSparseParcelableArray(java.lang.String,android.util.SparseArray)> -> _SINK_ (NO_CATEGORY)
<android.os.Bundle: void putString(java.lang.String,java.lang.String)> -> _SINK_ (NO_CATEGORY)
<android.os.Bundle: void putStringArray(java.lang.String,java.lang.String[])> -> _SINK_ (NO_CATEGORY)
<android.os.Bundle: void putStringArrayList(java.lang.String,java.util.ArrayList)> -> _SINK_ (NO_CATEGORY)
<android.os.Bundle: void putAll(android.os.Bundle)> -> _SINK_ (NO_CATEGORY)

<android.media.AudioRecord: int read(short[],int,int)> -> _SOURCE_ (OTHERS)
<android.media.AudioRecord: int read(byte[],int,int)> -> _SOURCE_ (OTHERS)
<android.media.AudioRecord: int read(java.nio.ByteBuffer,int)> -> _SOURCE_ (OTHERS)
<android.content.pm.PackageManager: java.util.List getInstalledApplications(int)> -> _SOURCE_ (OTHERS)
<android.content.pm.PackageManager: java.util.List getInstalledPackages(int)> -> _SOURCE_ (OTHERS)
<android.content.pm.PackageManager: java.util.List queryIntentActivities(android.content.Intent,int)> -> _SOURCE_ (NO_CATEGORY)
<android.content.pm.PackageManager: java.util.List queryIntentServices(android.content.Intent,int)> -> _SOURCE_ (NO_CATEGORY)
<android.content.pm.PackageManager: java.util.List queryBroadcastReceivers(android.content.Intent,int)> -> _SOURCE_ (NO_CATEGORY)
<android.content.pm.PackageManager: java.util.List queryContentProviders(java.lang.String,int,int)> -> _SOURCE_ (NO_CATEGORY)

<android.util.Log: int d(java.lang.String,java.lang.String)> -> _SINK_ (NO_CATEGORY)
<android.util.Log: int d(java.lang.String,java.lang.String,java.lang.Throwable)> -> _SINK_ (NO_CATEGORY)
<android.util.Log: int e(java.lang.String,java.lang.String)> -> _SINK_ (NO_CATEGORY)
<android.util.Log: int e(java.lang.String,java.lang.String,java.lang.Throwable)> -> _SINK_ (NO_CATEGORY)
<android.util.Log: int i(java.lang.String,java.lang.String)> -> _SINK_ (NO_CATEGORY)

<android.util.Log: int i(java.lang.String,java.lang.String,java.lang.Throwable)> -> _SINK_ (NO_CATEGORY)
<android.util.Log: int v(java.lang.String,java.lang.String)> -> _SINK_ (NO_CATEGORY)
<android.util.Log: int v(java.lang.String,java.lang.String,java.lang.Throwable)> -> _SINK_ (NO_CATEGORY)
<android.util.Log: int w(java.lang.String,java.lang.Throwable)> -> _SINK_ (NO_CATEGORY)
<android.util.Log: int w(java.lang.String,java.lang.String)> -> _SINK_ (NO_CATEGORY)
<android.util.Log: int w(java.lang.String,java.lang.String,java.lang.Throwable)> -> _SINK_ (NO_CATEGORY)
<android.util.Log: int wtf(java.lang.String,java.lang.Throwable)> -> _SINK_ (NO_CATEGORY)
<android.util.Log: int wtf(java.lang.String,java.lang.String)> -> _SINK_ (NO_CATEGORY)
<android.util.Log: int wtf(java.lang.String,java.lang.String,java.lang.Throwable)> -> _SINK_ (NO_CATEGORY)

<java.io.OutputStream: void write(byte[])> -> _SINK_ (NO_CATEGORY)
<java.io.OutputStream: void write(byte[],int,int)> -> _SINK_ (NO_CATEGORY)
<java.io.OutputStream: void write(int)> -> _SINK_ (NO_CATEGORY)

<java.io.FileOutputStream: void write(byte[])> -> _SINK_ (OTHERS)
<java.io.FileOutputStream: void write(byte[],int,int)> -> _SINK_ (OTHERS)
<java.io.FileOutputStream: void write(int)> -> _SINK_ (OTHERS)

<java.io.Writer: void write(char[])> -> _SINK_ (NO_CATEGORY)
<java.io.Writer: void write(char[],int,int)> -> _SINK_ (NO_CATEGORY)
<java.io.Writer: void write(int)> -> _SINK_ (NO_CATEGORY)
<java.io.Writer: void write(java.lang.String)> -> _SINK_ (NO_CATEGORY)
<java.io.Writer: void write(java.lang.String,int,int)> -> _SINK_ (NO_CATEGORY)
<java.io.Writer: java.io.Writer append(java.lang.CharSequence)> -> _SINK_ (NO_CATEGORY)

<java.io.OutputStreamWriter: java.io.Writer append(java.lang.CharSequence)> -> _SINK_ (NO_CATEGORY)

<android.content.Intent: android.content.Intent setAction(java.lang.String)> -> _SINK_
<android.content.Intent: android.content.Intent setClassName(android.content.Context,java.lang.Class)> -> _SINK_

<android.content.Intent: android.content.Intent
setClassName(android.content.Context,java.lang.String)> -> _SINK_
<android.content.Intent: android.content.Intent
setComponent(android.content.ComponentName)> -> _SINK_

%<android.content.Intent: android.content.Intent putExtra(java.lang.String,double[])> -> _SINK_
(NO_CATEGORY)
%<android.content.Intent: android.content.Intent putExtra(java.lang.String,int)> -> _SINK_
(NO_CATEGORY)
%<android.content.Intent: android.content.Intent
putExtra(java.lang.String,java.lang.CharSequence)> -> _SINK_ (NO_CATEGORY)
%<android.content.Intent: android.content.Intent putExtra(java.lang.String,char)> -> _SINK_
(NO_CATEGORY)
%<android.content.Intent: android.content.Intent putExtra(java.lang.String,android.os.Bundle)> ->
SINK (NO_CATEGORY)
%<android.content.Intent: android.content.Intent
putExtra(java.lang.String,android.os.Parcelable[])> -> _SINK_ (NO_CATEGORY)
%<android.content.Intent: android.content.Intent putExtra(java.lang.String,java.io.Serializable)> ->
SINK (NO_CATEGORY)
%<android.content.Intent: android.content.Intent putExtra(java.lang.String,int[])> -> _SINK_
(NO_CATEGORY)
%<android.content.Intent: android.content.Intent putExtra(java.lang.String,float)> -> _SINK_
(NO_CATEGORY)
%<android.content.Intent: android.content.Intent putExtra(java.lang.String,byte[])> -> _SINK_
(NO_CATEGORY)
%<android.content.Intent: android.content.Intent putExtra(java.lang.String,long[])> -> _SINK_
(NO_CATEGORY)
%<android.content.Intent: android.content.Intent putExtra(java.lang.String,android.os.Parcelable)>
-> _SINK_ (NO_CATEGORY)
%<android.content.Intent: android.content.Intent putExtra(java.lang.String,float[])> -> _SINK_
(NO_CATEGORY)
%<android.content.Intent: android.content.Intent putExtra(java.lang.String,long)> -> _SINK_
(NO_CATEGORY)
%<android.content.Intent: android.content.Intent putExtra(java.lang.String,java.lang.String[])> ->
SINK (NO_CATEGORY)

%<android.content.Intent: android.content.Intent putExtra(java.lang.String,boolean)> -> _SINK_ (NO_CATEGORY)
%<android.content.Intent: android.content.Intent putExtra(java.lang.String,boolean[])> -> _SINK_ (NO_CATEGORY)
%<android.content.Intent: android.content.Intent putExtra(java.lang.String,short)> -> _SINK_ (NO_CATEGORY)
%<android.content.Intent: android.content.Intent putExtra(java.lang.String,double)> -> _SINK_ (NO_CATEGORY)
%<android.content.Intent: android.content.Intent putExtra(java.lang.String,short[])> -> _SINK_ (NO_CATEGORY)
%<android.content.Intent: android.content.Intent putExtra(java.lang.String,java.lang.String)> -> _SINK_ (NO_CATEGORY)
%<android.content.Intent: android.content.Intent putExtra(java.lang.String,byte)> -> _SINK_ (NO_CATEGORY)
%<android.content.Intent: android.content.Intent putExtra(java.lang.String,char[])> -> _SINK_ (NO_CATEGORY)
%<android.content.Intent: android.content.Intent putExtra(java.lang.String,java.lang.CharSequence[])> -> _SINK_ (NO_CATEGORY)

<android.content.Context: void sendBroadcast(android.content.Intent)> -> _SINK_ (NO_CATEGORY)
<android.content.Context: void sendBroadcast(android.content.Intent,java.lang.String)> -> _SINK_ (NO_CATEGORY)
<android.content.Context: void sendOrderedBroadcast(android.content.Intent,java.lang.String)> -> _SINK_ (NO_CATEGORY)

<android.content.ContextWrapper: void sendOrderedBroadcast(android.content.Intent,java.lang.String)> -> _SINK_ (NO_CATEGORY)

<android.media.MediaRecorder: void setVideoSource(int)> -> _SINK_ (NO_CATEGORY)
<android.media.MediaRecorder: void setPreviewDisplay(android.view.Surface)> -> _SINK_ (NO_CATEGORY)
<android.media.MediaRecorder: void start()> -> _SINK_ (NO_CATEGORY)

<android.content.Context: android.content.Intent registerReceiver(android.content.BroadcastReceiver,android.content.IntentFilter)> -> _SINK_ (NO_CATEGORY)
<android.content.Context: android.content.Intent registerReceiver(android.content.BroadcastReceiver,android.content.IntentFilter,java.lang.String,android.os.Handler)> -> _SINK_ (NO_CATEGORY)

<android.content.IntentFilter: void addAction(java.lang.String)> -> _SINK_ (OTHERS)
<android.telephony.SmsManager: void sendTextMessage(java.lang.String,java.lang.String,java.lang.String,android.app.PendingIntent,android.app.PendingIntent)> android.permission.SEND_SMS -> _SINK_ (OTHERS)
<android.telephony.SmsManager: void sendDataMessage(java.lang.String,java.lang.String,short,byte[],android.app.PendingIntent,android.app.PendingIntent)> android.permission.SEND_SMS -> _SINK_ (OTHERS)
<android.telephony.SmsManager: void sendMultipartTextMessage(java.lang.String,java.lang.String,java.util.ArrayList,java.util.ArrayList,java.util.ArrayList)> android.permission.SEND_SMS -> _SINK_ (OTHERS)
<java.net.Socket: void connect(java.net.SocketAddress)> -> _SINK_ (OTHERS)
<android.os.Handler: boolean sendMessage(android.os.Message)> -> _SINK_ (OTHERS)

<android.content.SharedPreferences$Editor: android.content.SharedPreferences$Editor putBoolean(java.lang.String,boolean)> -> _SINK_ (OTHERS)
<android.content.SharedPreferences$Editor: android.content.SharedPreferences$Editor putFloat(java.lang.String,float)> -> _SINK_ (OTHERS)
<android.content.SharedPreferences$Editor: android.content.SharedPreferences$Editor putInt(java.lang.String,int)> -> _SINK_ (OTHERS)
<android.content.SharedPreferences$Editor: android.content.SharedPreferences$Editor putLong(java.lang.String,long)> -> _SINK_ (OTHERS)
<android.content.SharedPreferences$Editor: android.content.SharedPreferences$Editor putString(java.lang.String,java.lang.String)> -> _SINK_ (OTHERS)

<android.content.SharedPreferences: android.content.SharedPreferences getDefaultSharedPreferences(android.content.Context)> -> _SOURCE_ (OTHERS)
<android.provider.Telephony$Mms: android.database.Cursor query(android.content.ContentResolver,java.lang.String[])> -> _SOURCE_ (OTHERS)

<android.provider.ContactsContract$Contacts: android.net.Uri getLookupUri(long,java.lang.String)>
-> _SOURCE_ (USER_IDENTIFIERS)
<android.provider.ContactsContract$SyncState: android.util.Pair
getWithUri(android.content.ContentProviderClient,android.accounts.Account)>
android.permission.READ_SOCIAL_STREAM android.permission.READ_CONTACTS
android.permission.WRITE_CONTACTS -> _SOURCE_ (USER_IDENTIFIERS)
<android.provider.ContactsContract$ProfileSyncState: android.util.Pair
getWithUri(android.content.ContentProviderClient,android.accounts.Account)> -> _SOURCE_
(USER_IDENTIFIERS)

<android.bluetooth.BluetoothAdapter: java.lang.String getAddress()> -> _SOURCE_
(USER_IDENTIFIERS)
<android.net.wifi.WifiInfo: java.lang.String getMacAddress()> -> _SOURCE_ (UNIQUE_IDENTIFIER)
<java.util.Locale: java.lang.String getCountry()> -> _SOURCE_ (USER_IDENTIFIERS)
<android.net.wifi.WifiInfo: java.lang.String getSSID()> -> _SOURCE_ (UNIQUE_IDENTIFIER)
<android.telephony.gsm.GsmCellLocation: int getCid()> -> _SOURCE_ (LOCATION_INFORMATION)
<android.telephony.gsm.GsmCellLocation: int getLac()> -> _SOURCE_ (LOCATION_INFORMATION)

<android.accounts.AccountManager: android.accounts.Account[] getAccounts()> -> _SOURCE_
(USER_IDENTIFIERS)
<java.util.Calendar: java.util.TimeZone getTimeZone()> -> _SOURCE_ (OTHERS)
<android.provider.Browser: android.database.Cursor getAllBookmarks()> -> _SOURCE_
(USER_IDENTIFIERS)
<android.provider.Browser: android.database.Cursor getAllVisitedUrls()> -> _SOURCE_
(USER_IDENTIFIERS)

<org.apache.http.impl.client.DefaultHttpClient: org.apache.http.HttpResponse
execute(org.apache.http.client.methods.HttpUriRequest)> -> _SINK_ (NO_CATEGORY)
<org.apache.http.client.HttpClient: org.apache.http.HttpResponse
execute(org.apache.http.client.methods.HttpUriRequest)> -> _SINK_ (NO_CATEGORY)

<android.content.ContentResolver: android.database.Cursor
query(android.net.Uri,java.lang.String[],java.lang.String,java.lang.String[],java.lang.String)> ->
SOURCE (NO_CATEGORY)

<android.content.ContentResolver: android.database.Cursor
query(android.net.Uri,java.lang.String[],java.lang.String,java.lang.String[],java.lang.String,android.os.CancellationSignal)> -> _SOURCE_ (NO_CATEGORY)

%This is handled by the Easy Taint Wrapper given that the URL is used afterwards
%<java.net.URL: void <init>(java.lang.String,java.lang.String,int,java.lang.String)> -> _SINK_ (OTHERS)
%<java.net.URL: void <init>(java.lang.String,java.lang.String,java.lang.String)> -> _SINK_ (OTHERS)
%<java.net.URL: void
<init>(java.lang.String,java.lang.String,int,java.lang.String,java.net.URLStreamHandler)> -> _SINK_
(OTHERS)
%<java.net.URL: void <init>(java.lang.String)> -> _SINK_ (OTHERS)
%<java.net.URL: void <init>(java.net.URL,java.lang.String)> -> _SINK_ (OTHERS)
%<java.net.URL: void <init>(java.net.URL,java.lang.String,java.net.URLStreamHandler)> -> _SINK_
(OTHERS)

%<android.content.Context: void startActivity(android.content.Intent)> -> _SINK_ (NO_CATEGORY)
%<android.content.ContextWrapper: void startActivity(android.content.Intent)> -> _SINK_
(NO_CATEGORY)
%<android.content.Context: void startActivity(android.content.Intent,android.os.Bundle)> ->
SINK (NO_CATEGORY)
<android.content.Context: void startActivities(android.content.Intent[])> -> _SINK_ (NO_CATEGORY)
<android.content.Context: void startActivities(android.content.Intent[],android.os.Bundle)> ->
SINK (NO_CATEGORY)
<android.content.Context: android.content.ComponentName
startService(android.content.Intent)> -> _SINK_ (NO_CATEGORY)
<android.content.Context: boolean
bindService(android.content.Intent,android.content.ServiceConnection,int)> -> _SINK_
(NO_CATEGORY)
<android.content.Context: void sendBroadcast(android.content.Intent)> -> _SINK_
(NO_CATEGORY)
<android.content.Context: void sendBroadcast(android.content.Intent,java.lang.String)> -> _SINK_
(NO_CATEGORY)

%<android.app.Activity: android.content.Intent getIntent()> -> _SOURCE_ (NO_CATEGORY)

<android.app.Activity: void setResult(int,android.content.Intent)> -> _SINK_ (NO_CATEGORY)

%Do not enter this method as a source. Our callback parameter handling will take care
%of the parameters of this method anyway. Adding this method taints the whole activity!
%<android.app.Activity: void onActivityResult(int,int,android.content.Intent)> -> _SOURCE_

%<android.app.Activity: void startActivity(android.content.Intent)> -> _SINK_ (NO_CATEGORY)
%<android.app.Activity: void startActivity(android.content.Intent,android.os.Bundle)> -> _SINK_ (NO_CATEGORY)
<android.app.Activity: void startActivities(android.content.Intent[])> -> _SINK_ (NO_CATEGORY)
<android.app.Activity: void startActivities(android.content.Intent[],android.os.Bundle)> -> _SINK_ (NO_CATEGORY)
<android.app.Activity: void startActivityForResult(android.content.Intent,int)> -> _SINK_ (NO_CATEGORY)
<android.app.Activity: void startActivityForResult(android.content.Intent,int,android.os.Bundle)> -> _SINK_ (NO_CATEGORY)
<android.app.Activity: void startActivityFromChild(android.app.Activity,android.content.Intent,int,android.os.Bundle)> -> _SINK_ (NO_CATEGORY)
<android.app.Activity: void startActivityFromChild(android.app.Activity,android.content.Intent,int)> -> _SINK_ (NO_CATEGORY)
<android.app.Activity: void startActivityFromFragment(android.app.Fragment,android.content.Intent,int,android.os.Bundle)> -> _SINK_ (NO_CATEGORY)
<android.app.Activity: void startActivityFromFragment(android.app.Fragment,android.content.Intent,int)> -> _SINK_ (NO_CATEGORY)
<android.app.Activity: void startActivityIfNeeded(android.content.Intent,int,android.os.Bundle)> -> _SINK_ (NO_CATEGORY)
<android.app.Activity: void startActivityIfNeeded(android.content.Intent,int)> -> _SINK_ (NO_CATEGORY)
<android.app.Activity: android.content.ComponentName startService(android.content.Intent)> -> _SINK_ (NO_CATEGORY)

<android.app.Activity: boolean
bindService(android.content.Intent,android.content.ServiceConnection,int)> -> _SINK_
(NO_CATEGORY)
<android.app.Activity: void sendBroadcast(android.content.Intent)> -> _SINK_ (NO_CATEGORY)
<android.app.Activity: void sendBroadcast(android.content.Intent,java.lang.String)> -> _SINK_
(NO_CATEGORY)
<android.app.Activity: void sendBroadcastAsUser(android.content.Intent,android.os.UserHandle)> -
> _SINK_ (NO_CATEGORY)
<android.app.Activity: void
sendBroadcastAsUser(android.content.Intent,android.os.UserHandle,java.lang.String)> -> _SINK_
<android.app.Activity: void
sendOrderedBroadcast(android.content.Intent,java.lang.String,android.content.BroadcastReceiver,a
ndroid.os.Handler,int,java.lang.String,android.os.Bundle)> -> _SINK_ (NO_CATEGORY)
<android.app.Activity: void sendOrderedBroadcast(android.content.Intent,java.lang.String)> ->
SINK (NO_CATEGORY)
<android.app.Activity: void
sendOrderedBroadcastAsUser(android.content.Intent,android.os.UserHandle,java.lang.String,androi
d.content.BroadcastReceiver,android.os.Handler,int,java.lang.String,android.os.Bundle)> -> _SINK_
(NO_CATEGORY)
<android.app.Activity: void sendStickyBroadcast(android.content.Intent)> -> _SINK_
(NO_CATEGORY)
<android.app.Activity: void
sendStickyBroadcastAsUser(android.content.Intent,android.os.UserHandle)> -> _SINK_
(NO_CATEGORY)
<android.app.Activity: void
sendStickyOrderedBroadcast(android.content.Intent,android.content.BroadcastReceiver,android.os.
Handler,int,java.lang.String,android.os.Bundle)> -> _SINK_ (NO_CATEGORY)
<android.app.Activity: void
sendStickyOrderedBroadcastAsUser(android.content.Intent,android.os.UserHandle,android.content.
BroadcastReceiver,android.os.Handler,int,java.lang.String,android.os.Bundle)> -> _SINK_
(NO_CATEGORY)

<android.content.ContentResolver: android.net.Uri
insert(android.net.Uri,android.content.ContentValues)> -> _SINK_ (NO_CATEGORY)

<android.content.ContentResolver: int delete(android.net.Uri,java.lang.String,java.lang.String[])> -> _SINK_ (NO_CATEGORY)
<android.content.ContentResolver: int update(android.net.Uri,android.content.ContentValues,java.lang.String,java.lang.String[])> -> _SINK_ (NO_CATEGORY)
<android.content.ContentResolver: android.database.Cursor query(android.net.Uri,java.lang.String[],java.lang.String,java.lang.String[],java.lang.String)> -> _SINK_ (NO_CATEGORY)
<android.content.ContentResolver: android.database.Cursor query(android.net.Uri,java.lang.String[],java.lang.String,java.lang.String[],java.lang.String,android.os.CancellationSignal)> -> _SINK_ (NO_CATEGORY)

%<android.app.Activity: android.view.View findViewById(int)> -> _SOURCE_ (NO_CATEGORY)
<android.database.Cursor: java.lang.String getString(int)> -> _SOURCE_ (OTHERS)
<android.database.sqlite.SQLiteDatabase: android.database.Cursor query(android.net.Uri,java.lang.String[],java.lang.String,java.lang.String[],java.lang.String)> -> _SOURCE_ (OTHERS)
<android.database.sqlite.SQLiteDatabase: android.database.Cursor query(android.net.Uri,java.lang.String[],java.lang.String,java.lang.String[],java.lang.String,android.os.CancellationSignal)> -> _SOURCE_ (OTHERS)

<java.lang.ProcessBuilder: java.lang.Process start()> -> _SINK_ (NO_CATEGORY)

ภาคผนวก ค

ตารางที่ 19 แสดงถึงการเปรียบเทียบความแม่นยำของแบบจำลอง (มีการคัดกรองข้อมูล - เพิ่มข้อมูลแบบสุ่ม – แปลงค่าความถี่ของคำที่เกิดขึ้นบนเอกสารทั้งหมด) โดยมีป้ายชื่อข้อมูลประเภท “USER_IDENTIFIER”

แบ่งชุดข้อมูลสำหรับสอนและทดสอบ	Logistic Regression			Support Vector Machine			K-Nearest Neighbors		
	Precision	Recall	Accuracy	Precision	Recall	Accuracy	Precision	Recall	Accuracy
ข้อมูลชุด ที่ 1 (Clean-Random Oversampling-TF-IDF-USER-IDENTIFIER)									
Avg 5-folds CV	**36.8**	**60.8**	**81.8**	**38.8**	**61.6**	**82.8**	**32.6**	**68**	**78.2**
Fold 1	39	60	83	43	63	85	35	67	80
Fold 2	36	61	81	36	63	81	32	70	77
Fold 3	37	61	81	39	60	82	34	63	79
Fold 4	36	57	82	38	57	83	34	64	80
Fold 5	36	65	82	38	65	83	28	76	75

ตารางที่ 20 แสดงถึงการเปรียบเทียบความแม่นยำของแบบจำลอง (ไม่มีการคัดกรองข้อมูล - เพิ่มข้อมูลแบบสุ่ม - แปลงค่าความถี่ของคำที่เกิดขึ้นบนเอกสารทั้งหมด) โดยมีป้ายชื่อข้อมูลประเภท “USER_IDENTIFIER”

แบ่งชุดข้อมูลสำหรับสอนและทดสอบ	Logistic Regression			Support Vector Machine			K-Nearest Neighbors		
	Precision	Recall	Accuracy	Precision	Recall	Accuracy	Precision	Recall	Accuracy
ข้อมูลชุด ที่ 1 (UnClean-Random Oversampling-TF-IDF-USER-IDENTIFIER)									
Avg 5-folds CV	**34.2**	**59.6**	**82.2**	**36.6**	**60.8**	**83.4**	**31.8**	**65**	**80**
Fold 1	34	59	83	35	59	84	29	64	80
Fold 2	30	57	81	32	58	82	30	59	80
Fold 3	36	60	82	38	61	83	33	67	80
Fold 4	33	58	81	37	61	83	35	64	81
Fold 5	38	64	84	41	65	85	32	71	79

ตารางที่ 21 แสดงถึงการเปรียบเทียบความแม่นยำของแบบจำลอง (มีการคัดกรองข้อมูล - เพิ่มข้อมูลแบบสังเคราะห์ - แปลงค่าความถี่ของคำที่เกิดขึ้นบนเอกสารทั้งหมด) โดยมีป้ายชื่อข้อมูลประเภท “USER_IDENTIFIER”

แบ่งชุดข้อมูลสำหรับสอนและทดสอบ	Logistic Regression			Support Vector Machine			K-Nearest Neighbors		
	Precision	Recall	Accuracy	Precision	Recall	Accuracy	Precision	Recall	Accuracy
ข้อมูลชุด ที่ 1 (Clean-SMOTE Oversampling-TF-IDF-USER-IDENTIFIER)									
Avg 5-folds CV	**33.2**	**59.4**	**79.6**	**36**	**62.4**	**81**	**29**	**63.6**	**75.8**
Fold 1	35	58	81	39	61	83	31	66	77
Fold 2	33	60	79	36	64	81	29	63	76
Fold 3	36	62	80	36	62	80	28	59	75
Fold 4	32	58	79	35	59	81	28	61	75
Fold 5	30	59	79	34	66	80	29	69	76

ตารางที่ 22 แสดงถึงการเปรียบเทียบความแม่นยำของแบบจำลอง (ไม่มีการคัดกรองข้อมูล - เพิ่มข้อมูลแบบสังเคราะห์ - แปลงค่าความถี่ของคำที่เกิดขึ้นบนเอกสารทั้งหมด) โดยมีป้ายชื่อข้อมูลประเภท “USER_IDENTIFIER”

แบ่งชุดข้อมูลสำหรับสอนและทดสอบ	Logistic Regression			Support Vector Machine			K-Nearest Neighbors		
	Precision	Recall	Accuracy	Precision	Recall	Accuracy	Precision	Recall	Accuracy
ข้อมูลชุด ที่ 1 (UnClean-SMOTE Oversampling-TF-IDF-USER-IDENTIFIER)									
Avg 5-folds CV	**31.2**	**57**	**80.4**	**33.4**	**60.8**	**81.8**	**26.2**	**60.6**	**76**
Fold 1	30	55	81	31	62	82	25	65	76
Fold 2	27	54	79	29	57	80	24	59	75
Fold 3	34	60	81	35	61	82	27	58	76
Fold 4	32	55	80	35	60	82	27	56	76
Fold 5	33	61	81	37	64	83	28	65	77

ตารางที่ 23 แสดงถึงการเปรียบเทียบความแม่นยำของแบบจำลอง (มีการคัดกรองข้อมูล - เพิ่มข้อมูลแบบสุ่ม - แปลงพารากราฟเอกสารเป็นเวกเตอร์) โดยมีป้ายชื่อข้อมูลประเภท “USER_IDENTIFIER”

แบ่งชุดข้อมูลสำหรับสอนและทดสอบ	Logistic Regression			Support Vector Machine			K-Nearest Neighbors		
	Precision	Recall	Accuracy	Precision	Recall	Accuracy	Precision	Recall	Accuracy
ข้อมูลชุด ที่ 1 (Clean-Random Oversampling-doc2vec-USER-IDENTIFIER)									
Avg 5-folds CV	30.4	60.4	77.4	32.6	60.2	79	44.8	59.4	85.4
Fold 1	33	59	79	35	59	80	50	62	87
Fold 2	29	61	76	31	60	78	42	59	84
Fold 3	32	64	77	33	62	78	45	57	85
Fold 4	31	60	78	33	59	80	45	56	86
Fold 5	27	58	77	31	61	79	42	63	85

ตารางที่ 24 แสดงถึงการเปรียบเทียบความแม่นยำของแบบจำลอง (มีการคัดกรองข้อมูล - เพิ่มข้อมูลแบบสุ่ม - แปลงค่าความถี่ของคำที่เกิดขึ้นบนเอกสารทั้งหมด) โดยมีป้ายชื่อข้อมูลประเภท UNIQUE_IDENTIFIER

แบ่งชุดข้อมูลสำหรับสอนและทดสอบ	Logistic Regression			Support Vector Machine			K-Nearest Neighbors		
	Precision	Recall	Accuracy	Precision	Recall	Accuracy	Precision	Recall	Accuracy
ข้อมูลชุด ที่ 1 (Clean-Random Oversampling-TF-IDF-UNIQUE-IDENTIFIER)									
Avg 5-folds CV	**34.4**	**64**	**86.4**	**38.2**	**64.8**	**80.4**	**25.2**	**76.2**	**77.8**
Fold 1	38	64	87	42	65	89	29	73	81
Fold 2	35	66	87	37	63	87	23	77	75
Fold 3	35	66	86	39	68	87	25	76	77
Fold 4	36	73	87	40	73	52	25	79	78
Fold 5	28	51	85	33	55	87	24	76	78

ตารางที่ 25 แสดงถึงการเปรียบเทียบความแม่นยำของแบบจำลอง (ไม่มีการคัดกรองข้อมูล - เพิ่มข้อมูลแบบสุ่ม - แปลงค่าความถี่ของคำที่เกิดขึ้นบนเอกสารทั้งหมด) โดยมีป้ายชื่อข้อมูลประเภท UNIQUE_IDENTIFIER

แบ่งชุดข้อมูลสำหรับสอนและทดสอบ	Logistic Regression			Support Vector Machine			K-Nearest Neighbors		
	Precision	Recall	Accuracy	Precision	Recall	Accuracy	Precision	Recall	Accuracy
ข้อมูลชุด ที่ 1 (UnClean-Random Oversampling-TF-IDF-UNIQUE-IDENTIFIER)									
Avg 5-folds CV	**29.8**	**56.2**	**82.6**	**30.8**	**58**	**83**	**23.4**	**68.6**	**74.8**
Fold 1	30	51	82	32	55	83	24	69	74
Fold 2	27	56	82	29	57	83	22	70	74
Fold 3	31	56	83	31	56	83	25	63	76
Fold 4	31	59	83	33	60	84	24	70	75
Fold 5	30	59	83	29	62	82	22	71	75

ตารางที่ 26 แสดงถึงการเปรียบเทียบความแม่นยำของแบบจำลอง (มีการคัดกรองข้อมูล - เพิ่มข้อมูลแบบสังเคราะห์- แปลงค่าความถี่ของคำที่เกิดขึ้นบนเอกสารทั้งหมด) โดยมีป้ายชื่อข้อมูลประเภท UNIQUE_IDENTIFIER

แบ่งชุดข้อมูลสำหรับสอนและทดสอบ	Logistic Regression			Support Vector Machine			K-Nearest Neighbors		
	Precision	Recall	Accuracy	Precision	Recall	Accuracy	Precision	Recall	Accuracy
ข้อมูลชุด ที่ 1 (Clean-SMOTE Oversampling-TF-IDF-UNIQUE-IDENTIFIER)									
Avg 5-folds CV	**31.2**	**61.4**	**85**	**34.2**	**66.2**	**86.2**	**24.2**	**68.2**	**79**
Fold 1	33	61	85	36	68	86	27	69	80
Fold 2	33	60	86	34	63	87	25	71	80
Fold 3	32	64	85	36	67	87	22	62	78
Fold 4	31	69	84	35	75	86	24	72	78
Fold 5	27	53	85	30	58	85	23	67	79

ตารางที่ 27 แสดงถึงการเปรียบเทียบความแม่นยำของแบบจำลอง (ไม่มีการคัดกรองข้อมูล - เพิ่มข้อมูลแบบสังเคราะห์- แปลงค่าความถี่ของคำที่เกิดขึ้นบนเอกสารทั้งหมด) โดยมีป้ายชื่อข้อมูลประเภท UNIQUE_IDENTIFIER

แบ่งชุดข้อมูลสำหรับสอนและทดสอบ	Logistic Regression			Support Vector Machine			K-Nearest Neighbors		
	Precision	Recall	Accuracy	Precision	Recall	Accuracy	Precision	Recall	Accuracy
ข้อมูลชุด ที่ 1 (UnClean-SMOTE Oversampling-TF-IDF-UNIQUE-IDENTIFIER)									
Avg 5-folds CV	**27.8**	**55.2**	**81.6**	**29.4**	**59.6**	**82.4**	**22.6**	**63.2**	**75.4**
Fold 1	28	52	81	30	57	82	24	67	75
Fold 2	26	59	81	27	63	82	20	60	74
Fold 3	30	57	82	31	58	82	23	64	75
Fold 4	29	53	83	31	61	83	25	61	78
Fold 5	26	55	81	28	59	83	21	64	75

ตารางที่ 28 แสดงถึงการเปรียบเทียบความแม่นยำของแบบจำลอง (มีการคัดกรองข้อมูล - เพิ่มข้อมูลแบบสุ่ม - แปลงพารากราฟเอกสารเป็นเวกเตอร์) โดยมีป้ายชื่อข้อมูลประเภท UNIQUE_IDENTIFIER

แบ่งชุดข้อมูลสำหรับสอนและทดสอบ	Logistic Regression			Support Vector Machine			K-Nearest Neighbors		
	Precision	Recall	Accuracy	Precision	Recall	Accuracy	Precision	Recall	Accuracy
ข้อมูลชุด ที่ 1 (Clean-Random Oversampling-doc2vec-UNIQUE-IDENTIFIER)									
Avg 5-folds CV	33.4	67	85.6	34.6	66	86.4	46.6	61.2	90.4
Fold 1	35	76	87	33	71	86	49	71	91
Fold 2	36	66	86	38	66	86	53	62	91
Fold 3	30	64	84	33	68	86	45	59	90
Fold 4	31	61	85	34	61	87	40	51	89
Fold 5	35	68	86	35	64	87	46	63	91

ตารางที่ 29 แสดงถึงการเปรียบเทียบความแม่นยำของแบบจำลอง (มีการคัดกรองข้อมูล - เพิ่มข้อมูลแบบสุ่ม - แปลงค่าความถี่ของคำที่เกิดขึ้นบนเอกสารทั้งหมด) โดยมีป้ายชื่อข้อมูลประเภท “LOCATION_INFORMATION”

แบ่งชุดข้อมูลสำหรับสอนและทดสอบ	Logistic Regression			Support Vector Machine			K-Nearest Neighbors		
	Precision	Recall	Accuracy	Precision	Recall	Accuracy	Precision	Recall	Accuracy
ข้อมูลชุด ที่ 1 (Clean-Random Oversampling-TF-IDF-LOCATION_INFORMATION)									
Avg 5-folds CV	**19.2**	**45.2**	**81.2**	**20.8**	**45.2**	**82.6**	**15.6**	**63**	**71**
Fold 1	15	33	82	16	34	83	15	54	75
Fold 2	20	47	82	23	47	84	17	69	72
Fold 3	18	46	79	19	43	81	15	63	69
Fold 4	23	54	82	23	49	83	15	66	68
Fold 5	20	46	81	23	53	82	16	63	71

ตารางที่ 30 แสดงถึงการเปรียบเทียบความแม่นยำของแบบจำลอง (ไม่มีการคัดกรองข้อมูล - เพิ่มข้อมูลแบบสุ่ม - แปลงค่าความถี่ของคำที่เกิดขึ้นบนเอกสารทั้งหมด) โดยมีป้ายชื่อข้อมูลประเภท “LOCATION_INFORMATION”

แบ่งชุดข้อมูลสำหรับสอนและทดสอบ	Logistic Regression			Support Vector Machine			K-Nearest Neighbors		
	Precision	Recall	Accuracy	Precision	Recall	Accuracy	Precision	Recall	Accuracy
ข้อมูลชุด ที่ 1 (UnClean-Random Oversampling-TF-IDF-LOCATION_INFORMATION)									
Avg 5-folds CV	**20**	**47.8**	**80.6**	**21.8**	**47.6**	**82.2**	**18.6**	**55.6**	**77.2**
Fold 1	19	44	80	22	44	82	18	51	77
Fold 2	19	51	80	21	49	82	17	56	76
Fold 3	21	52	80	22	51	81	21	60	78
Fold 4	21	46	81	24	50	83	20	60	77
Fold 5	20	46	82	20	44	83	17	51	78

ตารางที่ 31 แสดงถึงการเปรียบเทียบความแม่นยำของแบบจำลอง (มีการคัดกรองข้อมูล - เพิ่มข้อมูลแบบสังเคราะห์ - แปลงค่าความถี่ของคำที่เกิดขึ้นบนเอกสารทั้งหมด) โดยมีป้ายชื่อข้อมูลประเภท “LOCATION_INFORMATION”

แบ่งชุดข้อมูลสำหรับสอนและทดสอบ	Logistic Regression			Support Vector Machine			K-Nearest Neighbors		
	Precision	Recall	Accuracy	Precision	Recall	Accuracy	Precision	Recall	Accuracy
ข้อมูลชุด ที่ 1 (Clean-SMOTE Oversampling-TF-IDF-LOCATION_INFORMATION)									
Avg 5-folds CV	**19.2**	**45.6**	**80.8**	**20.8**	**47.2**	**82**	**17.8**	**53.4**	**77.2**
Fold 1	15	37	80	17	39	82	17	54	78
Fold 2	21	48	82	23	50	83	19	53	79
Fold 3	20	46	81	20	48	81	17	52	76
Fold 4	20	52	80	21	51	81	18	54	77
Fold 5	20	45	81	23	48	83	18	54	76

ตารางที่ 32 แสดงถึงการเปรียบเทียบความแม่นยำของแบบจำลอง (ไม่มีการคัดกรองข้อมูล - เพิ่มข้อมูลแบบสังเคราะห์ - แปลงค่าความถี่ของคำที่เกิดขึ้นบนเอกสารทั้งหมด) โดยมีป้ายชื่อข้อมูลประเภท “LOCATION_INFORMATION”

แบ่งชุดข้อมูลสำหรับสอนและทดสอบ	Logistic Regression			Support Vector Machine			K-Nearest Neighbors		
	Precision	Recall	Accuracy	Precision	Recall	Accuracy	Precision	Recall	Accuracy
ข้อมูลชุด ที่ 1 (UnClean-SMOTE Oversampling-TF-IDF-LOCATION_INFORMATION)									
Avg 5-folds CV	**18.8**	**46.6**	**79.8**	**20.2**	**48.4**	**81**	**17.6**	**53**	**76.6**
Fold 1	18	44	80	20	46	81	16	49	75
Fold 2	18	55	78	19	55	79	18	57	77
Fold 3	19	45	79	20	50	80	20	59	77
Fold 4	20	44	81	24	51	83	19	53	78
Fold 5	19	45	81	18	40	82	15	47	76

ตารางที่ 33 แสดงถึงการเปรียบเทียบความแม่นยำของแบบจำลอง (มีการคัดกรองข้อมูล - เพิ่มข้อมูลแบบสุ่ม - แปลงพารากราฟเอกสารเป็นเวกเตอร์) โดยมีป้ายชื่อข้อมูลประเภท “LOCATION_INFORMATION”

แบ่งชุดข้อมูลสำหรับสอนและทดสอบ	Logistic Regression			Support Vector Machine			K-Nearest Neighbors		
	Precision	Recall	Accuracy	Precision	Recall	Accuracy	Precision	Recall	Accuracy
ข้อมูลชุด ที่ 1 (Clean-Random Oversampling-doc2vec-LOCATION_INFORMATION)									
Avg 5-folds CV	15.4	44	78.4	17.6	46	80.4	25.2	41.4	87.2
Fold 1	11	42	76	13	45	78	18	34	87
Fold 2	19	49	79	21	50	81	27	36	87
Fold 3	13	34	78	17	39	81	29	44	88
Fold 4	18	49	79	20	51	80	27	48	86
Fold 5	16	46	80	17	45	82	25	45	88

ตารางที่ 34 แสดงถึงการเปรียบเทียบความแม่นยำของแบบจำลอง (มีการคัดกรองข้อมูล - เพิ่มข้อมูลแบบสุ่ม - แปลงค่าความถี่ของคำที่เกิดขึ้นบนเอกสารทั้งหมด) โดยมีป้ายชื่อข้อมูลประเภท “OTHERS”

แบ่งชุดข้อมูลสำหรับสอนและทดสอบ	Logistic Regression			Support Vector Machine			K-Nearest Neighbors		
	Precision	Recall	Accuracy	Precision	Recall	Accuracy	Precision	Recall	Accuracy
ข้อมูลชุด ที่ 1 (Clean-Random Oversampling-TF-IDF-OTHER)									
Avg 5-folds CV	72.4	82.8	70	74.8	75.4	69.6	73	78.4	68.4
Fold 1	74	83	72	75	76	70	75	78	70
Fold 2	71	85	70	75	77	71	72	79	68
Fold 3	72	80	68	74	74	68	72	77	66
Fold 4	74	81	70	75	73	68	74	79	70
Fold 5	71	85	70	75	77	71	72	79	68

ตารางที่ 35 แสดงถึงการเปรียบเทียบความแม่นยำของแบบจำลอง (ไม่มีการคัดกรองข้อมูล - เพิ่มข้อมูลแบบสุ่ม - แปลงค่าความถี่ของคำที่เกิดขึ้นบนเอกสารทั้งหมด) โดยมีป้ายชื่อข้อมูลประเภท “OTHERS”

แบ่งชุดข้อมูลสำหรับสอนและทดสอบ	Logistic Regression			Support Vector Machine			K-Nearest Neighbors		
	Precision	Recall	Accuracy	Precision	Recall	Accuracy	Precision	Recall	Accuracy
ข้อมูลชุด ที่ 1 (UnClean-Random Oversampling-TF-IDF-OTHER)									
Avg 5-folds CV	71	79.2	68.2	73.4	73.2	68.4	72	74	67
Fold 1	71	80	68	73	71	68	72	76	68
Fold 2	71	78	68	73	73	68	71	72	65
Fold 3	71	79	68	74	74	69	72	71	66
Fold 4	71	79	68	73	74	68	72	76	68
Fold 5	71	80	69	74	74	69	73	75	68

ตารางที่ 36 แสดงถึงการเปรียบเทียบความแม่นยำของแบบจำลอง (ไม่มีการคัดกรองข้อมูล - เพิ่มข้อมูลแบบสุ่ม - แปลงค่าความถี่ของคำที่เกิดขึ้นบนเอกสารทั้งหมด) โดยมีป้ายชื่อข้อมูลประเภท “OTHERS”

แบ่งชุดข้อมูลสำหรับสอนและทดสอบ	Logistic Regression			Support Vector Machine			K-Nearest Neighbors		
	Precision	Recall	Accuracy	Precision	Recall	Accuracy	Precision	Recall	Accuracy
ข้อมูลชุด ที่ 1 (UnClean-SMOTE Oversampling-TF-IDF-OTHER)									
Avg 5-folds CV	75.2	67	67.2	76.4	67.6	68.2	78	65.2	68.8
Fold 1	75	68	68	75	66	67	76	64	67
Fold 2	75	67	67	76	68	68	77	67	68
Fold 3	75	67	67	77	68	68	80	65	69
Fold 4	75	65	66	76	67	68	78	62	67
Fold 5	76	68	68	78	69	70	79	68	73

ตารางที่ 37 แสดงถึงการเปรียบเทียบความแม่นยำของแบบจำลอง (มีการคัดกรองข้อมูล - เพิ่มข้อมูลแบบสังเคราะห์ - แปลงค่าความถี่ของคำที่เกิดขึ้นบนเอกสารทั้งหมด) โดยมีป้ายชื่อข้อมูลประเภท “OTHERS”

แบ่งชุดข้อมูลสำหรับสอนและทดสอบ	Logistic Regression			Support Vector Machine			K-Nearest Neighbors		
	Precision	Recall	Accuracy	Precision	Recall	Accuracy	Precision	Recall	Accuracy
ข้อมูลชุด ที่ 1 (Clean-SMOTE Oversampling-TF-IDF-OTHER)									
Avg 5-folds CV	76.8	69	68	78.2	69	68.8	78.6	64.2	66.6
Fold 1	77	68	67	78	70	68	78	65	66
Fold 2	79	71	70	80	70	71	80	64	67
Fold 3	75	71	69	77	68	68	78	65	68
Fold 4	75	68	66	77	69	68	79	63	66
Fold 5	78	67	68	79	68	69	78	64	66

ตารางที่ 38 แสดงถึงการเปรียบเทียบความแม่นยำของแบบจำลอง (มีการคัดกรองข้อมูล - เพิ่มข้อมูลแบบสุ่ม - แปลงพารากราฟเอกสารเป็นเวกเตอร์) โดยมีป้ายชื่อข้อมูลประเภท “OTHERS”

แบ่งชุดข้อมูลสำหรับสอนและทดสอบ	Logistic Regression			Support Vector Machine			K-Nearest Neighbors		
	Precision	Recall	Accuracy	Precision	Recall	Accuracy	Precision	Recall	Accuracy
ข้อมูลชุด ที่ 1 (Clean-Random Oversampling-doc2vec-OTHER)									
Avg 5-folds CV	70.8	80.6	67.4	71.8	80	68.4	77.4	72.2	69.6
Fold 1	71	80	67	72	81	68	79	73	70
Fold 2	72	82	69	72	80	69	77	73	70
Fold 3	69	82	67	70	81	68	76	74	70
Fold 4	71	79	67	72	79	68	77	68	67
Fold 5	71	80	67	73	79	69	78	73	71

บรรณานุกรม

1. *Global market share held by the leading smartphone operating systems in sales to end users from 1st quarter 2009 to 2nd quarter 2018.* . 2018 22 January 2019]; Available from: https://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-systems.
2. *General Data Protection Regulation*. 25/07/2019]; Available from: https://gdpr-info.eu/.
3. F Liu, S.W., P Story, S Zimmeck, N Sadeh, *Towards Automatic Classification of Privacy Policy Text*. 2018, Carnegie Mellon University.
4. M.I. Gordon, D.K., J.H. Perkins, L. Gilham, N. Nguyen, M.C. Rinard, *Information flow analysis of android applications in droidsafe*, in *22nd Annual Network and Distributed System Security Symposium NDSS 2015*. 2015.
5. Arzt, S., et al., *FlowDroid: precise context, flow, field, object-sensitive and lifecycle-aware taint analysis for Android apps.* SIGPLAN Not., 2014. **49**(6): p. 259-269.
6. Enck, W., et al., *TaintDroid: an information-flow tracking system for realtime privacy monitoring on smartphones*, in *Proceedings of the 9th USENIX conference on Operating systems design and implementation*. 2010, USENIX Association: Vancouver, BC, Canada. p. 393-407.
7. *Google Play Store*. 22 January 2019]; Available from: https://play.google.com/store/apps.
8. Bird, S., E. Klein, and E. Loper, *Natural Language Processing with Python*. 2009: O'Reilly Media, Inc. 512.
9. James, D., *Introduction to Machine Learning with Python: A Guide for Beginners in Data Science*. 2018: CreateSpace Independent Publishing Platform. 233.
10. *TF–IDF*, in *Encyclopedia of Machine Learning*, C. Sammut and G.I. Webb, Editors. 2010, Springer US: Boston, MA. p. 986-987.
11. Le, Q. and T. Mikolov, *Distributed representations of sentences and documents*,

in *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*. 2014, JMLR.org: Beijing, China. p. II-1188-II-1196.

12. Mikolov, T., et al., *Distributed representations of words and phrases and their compositionality*, in *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*. 2013, Curran Associates Inc.: Lake Tahoe, Nevada. p. 3111-3119.
13. Yap, B.W., et al. *An Application of Oversampling, Undersampling, Bagging and Boosting in Handling Imbalanced Datasets*. 2014. Singapore: Springer Singapore.
14. Refaeilzadeh, P., L. Tang, and H. Liu, *Cross-Validation*, in *Encyclopedia of Database Systems*, L. Liu and M.T. ÖZsu, Editors. 2009, Springer US: Boston, MA. p. 532-538.
15. Ting, K.M., *Confusion Matrix*, in *Encyclopedia of Machine Learning and Data Mining*, C. Sammut and G.I. Webb, Editors. 2017, Springer US: Boston, MA. p. 260-260.
16. S. Arzt, S.R., C. Fritz, E. Bodden, A. Bartel, J. Klein, Y. Le Traon, D. Octeau, P. McDaniel. *Flowdroid: Precise context flow field object-sensitive and lifecycle-aware taint analysis for android apps*. 2014.
17. Feng, P., Ma, J., Sun, C., Xu, X., & Ma, Y, *A Novel Dynamic Android Malware Detection System With Ensemble Learning*, in *IEEE Access*. 2018.
18. *Usableprivacy*. 12 December 2019]; Available from: https://explore.usableprivacy.org/.
19. Shomir Wilson, F.S., Aswarth Abhilash Dara, Frederick Liu, Sushain Cherivirala, Pedro Giovanni Leon, Mads Schaarup Andersen, Sebastian Zimmeck, Kanthashree Mysore Sathyendra, N. Cameron Russell, Thomas B. Norton, Eduard Hovy, Joel Reidenberg, and Norman Sadeh. *The Creation and Analysis of a Website Privacy Policy Corpus*. in *In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. August 2016. Berlin, Germany.
20. *Flowdroid Source Sink File*. 12 December 2019]; Available from: https://github.com/secure-software-engineering/FlowDroid/blob/master/soot-infoflow-android/SourcesAndSinks.txt.

21. *Android API Package*. Available from: https://developer.android.com/reference/packages.

22. *DroidBench*. 12 December 2019]; Available from: https://github.com/secure-software-engineering/DroidBench.

23. Xia, M., et al., *Effective Real-Time Android Application Auditing.* 2015 IEEE Symposium on Security and Privacy, 2015: p. 899-914.

ประวัติผู้เขียน

ชื่อ-สกุล	เมธัส นาคเสนีย์
วัน เดือน ปี เกิด	16 August 1995
สถานที่เกิด	ที่จังหวัดกรุงเทพมหานคร
วุฒิการศึกษา	สำเร็จการศึกษาระดับปริญญาบัณฑิต หลักสูตรวิศวกรรม สาขาวิศวกรรมคอมพิวเตอร์ จากมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี เมื่อ พ.ศ. 2560 และเข้าศึกษาต่อใน หลักสูตรวิทยาศาสตร์มหาบัณฑิต สาขาวิศวกรรมซอฟต์แวร์ คณะ วิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2560