การระบุตัวตนของอุปกรณ์ไอโอทีโดยใช้การวิเคราะห์การรับส่งข้อมูลเครือข่าย

IOT DEVICE IDENTIFICATION USING NETWORK TRAFFIC ANALYSIS

ณัฐพนธ์ อิทธินิรันดร

บัณฑิตวิทยาลัย มหาวิทยาลัยศรีนครินทรวิโรฒ

2565

การระบุตัวตนของอุปกรณ์ไอโอทีโดยใช้การวิเคราะห์การรับส่งข้อมูลเครือข่าย

ณัฐพนธ์ อิทธินิรันดร

สารนิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
วิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการข้อมูล
คณะวิทยาศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ
ปีการศึกษา 2565

IOT DEVICE IDENTIFICATION USING NETWORK TRAFFIC ANALYSIS

NUTTAPOL ITTINIRUNDORN

A Master's Project Submitted in Partial Fulfillment of the Requirements

for the Degree of MASTER OF SCIENCE

(Data Science)

Faculty of Science, Srinakharinwirot University

2022

สารนิพนธ์

เรื่อง

การระบุตัวตนของอุปกรณ์ไอโอทีโดยใช้การวิเคราะห์การรับส่งข้อมูลเครือข่าย

ของ

ณัฐพนธ์ อิทธินิรันดร

ได้รับอนุมัติจากบัณฑิตวิทยาลัยให้นับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการข้อมูล

ของมหาวิทยาลัยศรีนครินทรวิโรฒ

.....

(รองศาสตราจารย์ นายแพทย์ฉัตรชัย เอกปัญญาสกุล)

คณบดีบัณฑิตวิทยาลัย

.....

คณะกรรมการสอบปากเปล่าสารนิพนธ์

..... ที่ปรึกษาหลัก

(อาจารย์ ดร.โสภณ มงคลลักษมี)

..... ประธาน

(รองศาสตราจารย์ ดร.อรรณพ หมั่นสกุล)

..... กรรมการ

(อาจารย์ ดร.ศิริสรรพ เหล่าหะเกียรติ)

ชื่อเรื่อง	การระบุตัวตนของอุปกรณ์ไอโอทีโดยใช้การวิเคราะห์การรับส่งข้อมูลเครือข่าย
ผู้วิจัย	ณัฐพนธ์ อิทธินิรันดร
ปริญญา	วิทยาศาสตรมหาบัณฑิต
ปีการศึกษา	2565
อาจารย์ที่ปรึกษา	อาจารย์ ดร. โสภณ มงคลลักษมี

การใช้งานอุปกรณ์ไอโอที (IoT) มีความนิยมสูงขึ้น ด้วยปริมาณการใช้งานที่เพิ่มขึ้นนี้ส่งผลโดยตรงต่อการบริหารและความปลอดภัย เพื่อรองรับกับปัญหาดังกล่าว งานวิจัยนี้นำเสนอการแยกแยะอุปกรณ์ไอโอทีด้วยข้อมูลทางเครือข่ายในระดับ Network Layer และ Transport Layer โดยใช้การเรียนรู้ของเครื่องซึ่งช่วยรักษาความเป็นส่วนตัวของข้อมูลผู้ใช้งานและการแยกแยะอุปกรณ์อย่างมีประสิทธิภาพ ข้อมูลเครือข่ายของอุปกรณ์จะถูกจัดเก็บและสกัดข้อมูลทางสถิติที่จำเป็นในช่วงเวลา 10 นาที ซึ่งประกอบด้วยข้อมูลเช่น ขนาดแพคเก็ต จำนวนแพคเก็ต และพฤติกรรมของ IPID โดยข้อมูลที่ได้จะถูกนำมาใช้ในการเรียนรู้ของเครื่อง ซึ่งมีวิธีการเรียนรู้ที่ใช้คือ K-Nearest Neighbors, Naïve Bayesian, Random Forest, และ Support Vector Machine ในการวัดผลใช้ตัวชี้วัดที่ประกอบด้วย Accuracy, Precision, Recall, และ F-1 พบว่าค่าผลเฉลี่ยสูงกว่า 0.9 ในทุกด้าน นอกจากนี้การใช้ข้อมูลพฤติกรรมของ IPID สามารถเพิ่มประสิทธิภาพได้โดยเฉพาะเมื่อใช้ร่วมกับ Naïve Bayesian ซึ่งให้ผลดีขึ้นเฉลี่ยถึง 0.08 ในทุกตัวชี้วัด

คำสำคัญ : อุปกรณ์ไอโอที, การเรียนรู้ของเครื่อง, การจำแนก, การจัดการเครือข่าย

Title	IOT DEVICE IDENTIFICATION USING NETWORK TRAFFIC ANALYSIS
Author	NUTTAPOL ITTINIRUNDORN
Degree	MASTER OF SCIENCE
Academic Year	2022
Thesis Advisor	Professor Doctor Sophon Mongkolluksamee

The popularity of Internet of Things (IoT) devices has increased significantly, directly impacting management and security. Therefore, this research proposes a method to classify IoT devices using network-level and transport-level data, combined with machine learning techniques. This approach helps maintain user data privacy and enables rapid device classification. The network data from devices is collected and statistically analyzed over a ten-minute period, including packet size, packet count, and IPID behavior. The collected data is used to train four types of machine learning models: K-Nearest Neighbours, Naïve Bayesian, Random Forest, and Support Vector Machine. The performance evaluation used accuracy, precision, recall, and F-1 measures, with average scores exceeding 0.9 in all aspects. Furthermore, incorporating IPID behavior data improved effectiveness, particularly when combined with Naïve Bayesian, resulting in an average improvement of 0.08 across all evaluation metrics.

Keyword : IoT Devices, Machine learning, Classification, Network management

กิตติกรรมประกาศ

งานวิจัยเรื่องนี้สำเร็จลุล่วงได้เพราะความอนุเคราะห์และความช่วยเหลือในทุกๆด้าน ขอขอบพระคุณ อ.ดร.โสภณ มงคลลักษมี อาจารย์ที่ปรึกษาและช่วยการดูแลแนวทางการลงมือปฏิบัติและความเข้าใจในระบบเครือข่ายที่มากขึ้น ขอขอบพระคุณคณาจารย์ ภาควิชาวิทยาการคอมพิวเตอร์ สาขา วิทยาการข้อมูลทุกท่าน ที่ได้ปูพื้นฐานการทำงานของการเรียนรู้ด้วยคอมพิวเตอร์และวิชาข้างเคียงซึ่งบูรณาการเข้าด้วยกัน ขอขอบพระคุณ อ.สิทธิชัย วรโชติกำจร ผู้สอนวิชาระบบเครือข่ายในระดับปริญญาตรี

ณัฐพนธ์ อิทธินิรันดร

สารบัญ

หน้า

สารบัญตาราง

หน้า

สารบัญรูปภาพ

หน้า

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญ

อุปกรณ์ไอโอที(IoT: Internet of Things)หรืออุปกรณ์อินเตอร์เน็ตของสรรพสิ่งนั้น ในปัจจุบันได้เข้ามามีบทบาทอยากมากต่อผู้คน อุปกรณ์เหล่านี้ถูกติดตั้งและใช้งานเพื่อสนับสนุนการใช้ชีวิตประจำวันและการทำงาน ด้วยอุปกรณ์มีขนาดเล็กและการติดตั้งได้ง่าย อุปกรณ์เหล่านี้อาจถูกติดตั้งและใช้งานโดยไม่ได้รับการตรวจสอบและลงทะเบียน ซึ่งอาจมีผลกระทบต่อการบริหารจัดการระบบเครือข่ายรวมถึงด้านความปลอดภัย โดยเฉพาะอย่างยิ่งในองค์กรต่างๆที่คำนึงถึงเรื่องของความมั่นคงปลอดภัยทางไซเบอร์ ด้วยอุปกรณ์ IoT อาจเป็นช่องโหว่ในการถูกโจมจากผู้ไม่หวังดี ด้วยความสำคัญของปัญหาดังกล่าวงานวิจัยนี้จำได้นำเสนอวิธีการในการจำแนกอุปกรณ์ IoT ต่างๆด้วยการวิเคราะห์ข้อมูลที่รับส่งในเครือข่าย ซึ่งสามารถนำไปประยุกต์ใช้ในการบริหารจัดการเครือข่ายและเพิ่มความมั่นคงปลอดภัยทางไซเบอร์ ให้กับองค์กร

เมื่อมีอุปกรณ์ IoT จำนวนมากเชื่อมต่อเครือข่าย การบริหารจัดการระบบเครือข่ายจะมีความจำเป็นมากขึ้นตามด้วย เพื่อรักษาเสถียรภาพของระบบเครือข่ายให้มีความเร็วเหมาะสม การที่สามารถแยกแยะอุปกรณ์ IoT ที่ใช้งานเครือข่ายได้นั้น ช่วยให้สามารถสร้างการรับประกันคุณภาพ (Quality of Service) หรือ QoS ช่วยให้ระบบสามารถจัดการข้อมูลในระบบเครือข่ายได้อย่างถูกต้องและความเหมาะสม

อุปกรณ์ IoT นั้นถูกผลิตออกมาเพื่อตอบสนองต่อความต้องการที่สูงขึ้นอย่างรวดเร็วของผู้ใช้ อีกทั้งด้วยเรื่องของ IoT ยังเป็นเรื่องใหม่ทำให้ยังขาดแคลนองค์กรในการตรวจสอบและรับประกันคุณภาพของอุปกรณ์ IoT ส่งผลให้อุปกรณ์ IoT ในปัจจุบันยังคงมีความเสี่ยงสูงในเรื่องของความปลอดภัย ทำให้มีความเสียงสูงในการเกิดภัยคุกคามกับองค์กรที่นำอุปกรณ์มาใช้งาน อุปกรณ์ดังกล่าวนั้นสามารถสร้างความเสียหายให้กับระบบเครือข่ายหรือข้อมูลขององค์กรได้ เช่น การโจรกรรมข้อมูลที่มีความสำคัญต่อองค์กร การก่อกวนระบบเครือข่ายขององค์กรเพื่อให้สามารถดำเนินการช้าลง หรือไม่สามารถดำเนินการใดๆ ภายใต้เครือข่ายขององค์กรที่ถูกโจมได้ โดยอุปกรณ์ IoT โดยส่วนใหญ่จะมีขนาดเล็กและสามารถซ่อนตามสถานที่ต่างๆได้ และยากต่อการสังเกตุ ดังนั้นหากสามารถแยกแยะอุปกรณ์ IoT ได้ด้วยข้อมูลเครือข่ายที่พวกมันใช้ในการสื่อสารย่อมเป็นผลดีอย่างมากในด้านความมั่นคงปลอดภัย

ทั้งนี้ ผู้วิจัยได้พัฒนาวิธีการในการแยกแยะอุปกรณ์ IoT ต่างๆ ด้วยการวิเคราะห์ข้อมูลเครือข่ายและการไหลเวียนของข้อมูล เพื่อช่วยในการค้นหาอุปกรณ์ภายในเครือข่ายที่มีแนวโน้วจะ

มีคุณสมบัติเป็นอุปกรณ์ IoT เพื่อซึ่งสามารถนำไปประยุกต์ใช้งานได้ทั้งด้านการบริหารจัดการ และการตรวจสอบด้านความปลอดภัยอาทิเช่นค้นหาอุปกรณ์ที่ไม่พึงประสงค์ภายในเครือข่ายเพื่อความปลอดภัยของระบบเครือข่ายและป้องกันการโจรกรรมข้อมูลภายในระบบเครือข่าย

1.2 วัตถุประสงค์ของงานวิจัย

1.2.1 ศึกษาและทำความเข้าใจเกี่ยวกับข้อมูลเครือข่ายของอุปกรณ์ IoT

1.2.2 ศึกษาการใช้งานการเรียนรู้ของเครื่องเพื่อจำแนกอุปกรณ์ที่เป็น IoT ด้วยข้อมูลเครือข่าย

1.3 ขอบเขตงานวิจัย

1.3.1 ใช้งานข้อมูลเครือข่าย(Network Traffic)ในระดับชั้นที่ 3 (network layer - IP) และชั้นที่ 4 (transport layer -TCP/UDP) เป็นข้อมูลหลัก

1.3.2 ใช้การเรียนรู้ของเครื่องสำหรับการจำแนกอุปกรณ์ IoT

1.3.3 ประเมินผลลัพธ์ของงานวิจัยด้วยค่าต่าง ๆ เช่น Accuracy, Precision, Recall และ F1

1.4 วิธีดำเนินการวิจัย

1.4.1 จัดหาและสืบค้นวรรณกรรมและงานวิจัยที่เกี่ยวข้อง

1.4.2 เตรียมความพร้อมอุปกรณ์ที่จำเป็นต่อการใช้งานเพื่อการวิจัย

1.4.3 จัดหาชุดข้อมูลเพื่อทดลองการสร้างแบบจำลองการทำนายเกี่ยวกับบันทึกการใช้งานของระบบเครือข่าย

1.4.4 จัดเตรียมชุดข้อมูลให้เหมาะสมกับการใช้งานเพื่อประสิทธิภาพของแบบจำลอง

1.4.5 นำชุดข้อมูลที่ผ่านการเตรียมความพร้อมเพื่อสร้างแบบการจำลองการทำนายโดยการเรียนรู้ของเครื่อง

1.4.6 ทำการทดลองและสรุปผลลัพธ์

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1.5.1 เข้าใจหลักการระหว่างการทำงานของอุปกรณ์ IoT และระบบเครือข่าย

1.5.2 ได้รับความเข้าใจเรื่องหลักการการเรียนรู้ของเครื่อง

1.5.3 สามารถสร้างแบบจำลองการทำนายที่มีประสิทธิภาพที่น่าพึงพอใจ

1.5.4 ได้วิธีการในการแยกแยะระหว่างอุปกรณ์ IoT

บทที่ 2
แนวคิด ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

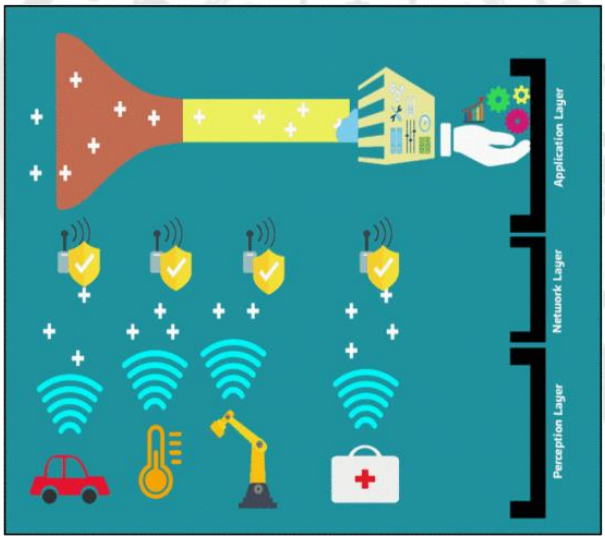
2.1.1 Internet of Thing (IoT)

อินเทอร์เน็ตในทุกสรรพสิ่ง หรือ Internet of Thing (IoT) หมายถึงเครือข่ายของอุปกรณ์ทางกายภาพ ยานพาหนะ เครื่องใช้ภายในบ้าน และอุปกรณ์ใดๆ ที่ฝังเซ็นเซอร์ ซอฟต์แวร์ และการเชื่อมต่อ ที่ทำให้อุปกรณ์เหล่านั้นสามารถเชื่อมต่อและแลกเปลี่ยนข้อมูลผ่านอินเทอร์เน็ตได้ การนำเทคโนโลยี IoT มาใช้นั้นมีประโยชน์หลายด้านเช่นปรับปรุงประสิทธิภาพ: เทคโนโลยี IoT ช่วยให้อุปกรณ์ต่างสามารถสื่อสารระหว่างกัน แบ่งปันข้อมูล และทำการตัดสินใจตามข้อมูลนั้น ซึ่งส่งผลโดยตรงนำไปสู่การปรับปรุงประสิทธิภาพในอุตสาหกรรมต่างๆ เช่น การผลิต การขนส่ง และการดูแลสุขภาพ ซึ่งสามารถใช้ IoT เพื่อทำให้กระบวนการต่างๆเป็นอัตโนมัติและเพิ่มประสิทธิภาพการดำเนินงานได้

- ลดค่าใช้จ่าย: เทคโนโลยี IoT สามารถช่วยบริษัทต่างๆ ประหยัดค่าใช้จ่ายได้ โดยการลดของเสีย ปรับปรุงประสิทธิภาพการใช้พลังงาน และลดความจำเป็นในการใช้แรงงานคน ตัวอย่างเช่น สามารถใช้เซ็นเซอร์ IoT เพื่อตรวจสอบอุปกรณ์และคาดการณ์ความต้องการในการบำรุงรักษา ซึ่งสามารถลดเวลาหยุดทำงานและค่าใช้จ่ายในการซ่อมแซม
- ปรับปรุงประสบการณ์ของลูกค้า: สามารถใช้ IoT เพื่อสร้างผลิตภัณฑ์และบริการใหม่ๆ ที่ช่วยยกระดับประสบการณ์ของลูกค้า ตัวอย่างเช่น สามารถใช้อุปกรณ์สมาร์ทโฮมเพื่อควบคุมแสง อุณหภูมิ และความปลอดภัย ในขณะที่อุปกรณ์สวมใส่สามารถใช้ตรวจสอบสมรรถภาพและสุขภาพได้
- ความปลอดภัยที่เพิ่มขึ้น: สามารถใช้ IoT เพื่อปรับปรุงความปลอดภัยในอุตสาหกรรมต่างๆ เช่น การขนส่งและการดูแลสุขภาพ ตัวอย่างเช่น สามารถใช้เซ็นเซอร์ IoT เพื่อตรวจสอบประสิทธิภาพของยานพาหนะและตรวจจับปัญหาด้านความปลอดภัยที่อาจเกิดขึ้นได้ รวมถึงการใช้อุปกรณ์สวมใส่เพื่อตรวจสอบสุขภาพของผู้ป่วยและแจ้งเตือนผู้ให้บริการด้านสุขภาพถึงปัญหาที่อาจเกิดขึ้น

สถาปัตยกรรมของ IoT นั้นโดยทั่วไปแล้วสามารถแบ่งออกสองแบบคือ แบบสามชั้น (Three-layer architecture)และห้าชั้น (Five-layer architecture)

สถาปัตยกรรมแบบสามชั้น (Al-Qaseemi et al., 2016) ถือเป็นรูปแบบพื้นฐานของระบบ IoT ประกอบด้วย ชั้นการรับรู้(perception layer), ชั้นเครือข่าย(network layer) และชั้นแอพพลิเคชั่น(application layer) ดังแสดงในรูปที่ 1 ชั้นการรับรู้นี้เป็นที่รู้จักกันในอีกชื่อว่าชั้นกายภาพเพราะจะประกอบด้วยเซ็นเซอร์ที่ฝังอยู่ในสิ่งของทางกายภาพเพื่อตรวจรับและแปลงข้อมูลให้อยู่ในรูปของสัญญาณไฟฟ้า และจะรวบรวมข้อมูลจากอุปกรณ์และส่งข้อมูลนี้ไปยังชั้นเครือข่าย ชั้นเครือข่ายรับผิดชอบในการเชื่อมต่อจากชั้นการรับรู้ ในชั้นนี้สามารถเป็นเทคโนโลยีการเชื่อมต่อใดๆก็ได้ที่เหมาะสมกับงาน เช่น การเชื่อมต่อแบบมีสายหรือไร้สายที่ปลอดภัย และข้อมูลที่ชั้นแอพพลิเคชั่นได้รับจาก ชั้นเครือข่ายจะถูกวิเคราะห์และประมวลผลเพื่อให้บริการและตัดสินใจส่งผลลัพธ์กลับจากชั้นแอพพลิเคชั่น ไปยังชั้นการรับรู้ผ่านทางชั้นเครือข่าย

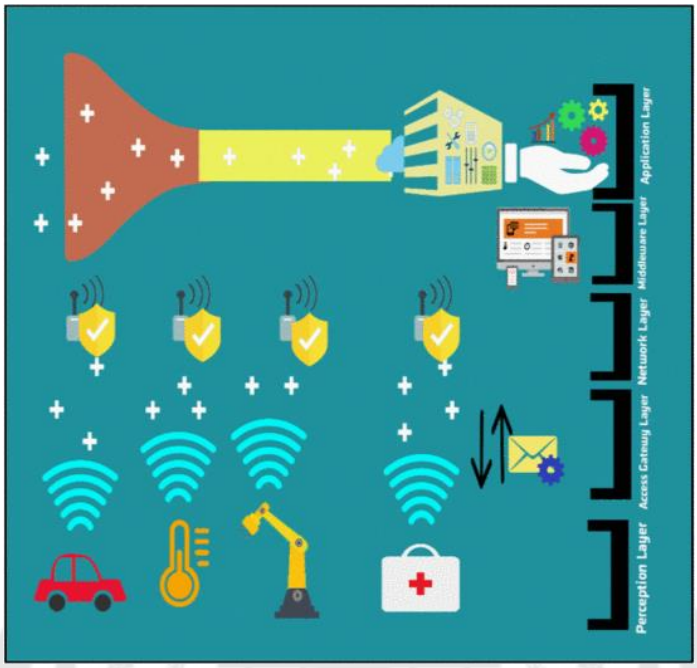


ภาพประกอบ 1 สถาปัตยกรรม IoT แบบสามชั้น

ที่มา: (Al-Qaseemi et al., 2016)

สถาปัตยกรรมแบบห้าชั้น (Al-Qaseemi et al., 2016) สถาปัตยกรรมแบบห้าชั้นนั้นมีสามชั้นหลักเหมือนกับแบบสามชั้นแต่มีเพิ่มขึ้นมาอีกสองชั้นคือชั้นการเข้าถึง(access getaway layer) และชั้นมิดเดิลแวร์ (middleware layer) ชั้นการเข้าถึงมีหน้าที่จัดการการสื่อสาร

IoT ในสภาพแวดล้อมเฉพาะ และแลกเปลี่ยนข้อความระหว่างวัตถุและระบบ ชั้นมิดเดิลแวร์เพิ่มการเชื่อมโยงที่ยืดหยุ่นมากขึ้นระหว่างอุปกรณ์ฮาร์ดแวร์และแอปพลิเคชันตามที่แสดงในรูปที่ 2



ภาพประกอบ 2 สถาปัตยกรรม IoT แบบห้าชั้น

ที่มา: (Al-Qaseemi et al., 2016)

2.1.2 Open Systems Interconnection (OSI) Reference Model

ในงานวิจัยนี้ให้ความสนใจในการใช้ข้อมูลที่อยู่ในชั้นเครือข่ายของสถาปัตยกรรมเพื่อการแยกแยะอุปกรณ์ IoT โดยใช้ข้อมูลที่เกิดในชั้น Network และชั้นTransport ของ OSI 7 Layers โมเดล ซึ่งก็เทียบได้กับ IP protocol และ TCP protocol ตามลำดับ โดนแต่ละชั้นของ OSI 7 Layers ดังนี้

- **Physical Layer**: ชั้นนี้เป็นชั้นต่ำสุดของแบบจำลอง OSI เป็นชั้นที่ดูแลการส่งบิตข้อมูลดิบที่ทางไฟฟ้าหรือทางแสงผ่านเครือข่ายจากชั้นทางกายภาพของอุปกรณ์ส่งไปยังชั้นทางกายภาพของอุปกรณ์รับ ซึ่งอาจรวมถึงข้อกำหนดต่างๆ เช่น แรงดันไฟฟ้า การเดินสาย และความถี่วิทยุที่ชั้นกายภาพ
- **Data-Link Layer**: ในชั้น Data-Link นี้ จะดูแลในเรื่องของการการถ่ายโอนข้อมูลแบบโหนดต่อโหนด โดยที่ข้อมูลจะถูกบรรจุเป็นเฟรม ในชั้นนี้

สามารถแก้ไขข้อผิดพลาดที่อาจเกิดขึ้นที่ชั้นทางกายภาพ ชั้น Data-Link ครอบคลุมสองชั้นย่อยประกอบด้วยชั้น Media Access Control (MAC) ให้การควบคุมการไหลของข้อมูลและมัลติเพล็กซ์สำหรับการส่งผ่านอุปกรณ์ผ่านเครือข่าย ชั้นที่สองคือ Logical Link Control (LLC) ซึ่งให้การควบคุมการไหลและข้อผิดพลาดบนสื่อกายภาพ

- **Network Layer**: ชั้นนี้มีหน้าที่รับเฟรมจากชั้น Data-Link และส่งไปยังปลายทางที่ต้องการตามที่อยู่ที่ระบุอยู่ภายในเฟรม ชั้น Network จะค้นหาปลายทางโดยใช้ที่อยู่แบบลอจิคัล เช่น IP (อินเทอร์เน็ตโปรโตคอล) ที่ชั้นนี้ เร้าเตอร์เป็นส่วนประกอบสำคัญที่ใช้ในการกำหนดเส้นทางข้อมูลที่จำเป็นในการไประหว่างเครือข่าย
- **Transport Layer**: ชั้นนี้จัดการการจัดส่งและการตรวจสอบข้อผิดพลาดของแพ็กเก็ตข้อมูล ควบคุมขนาด จัดลำดับ และถ่ายโอนข้อมูลระหว่างระบบและเครื่องปลายทาง ตัวอย่างโปรโตคอลที่พบบ่อยที่สุดในชั้นนี้คือ TCP หรือ Transmission Control Protocol
- **Session Layer**: ชั้นนี้จะควบคุมการสนทนาระหว่างคอมพิวเตอร์เครื่องต่างๆ เซสชันหรือการเชื่อมต่อระหว่างเครื่องได้รับการตั้งค่า จัดการ และสิ้นสุดที่ชั้นนี้ นอกจากนั้นแล้วชั้นนี้จะให้บริการรวมถึงการรับรองความถูกต้องและการเชื่อมต่อใหม่
- **Presentation Layer**: ชั้นนี้รับผิดชองการนำเสนอ การจัดรูปแบบหรือแปลข้อมูลสำหรับชั้นแอปพลิเคชันตามไวยากรณ์หรือความหมายที่แอปพลิเคชันยอมรับ ในชั้นนี้ยังสามารถจัดการการเข้ารหัสและถอดรหัสที่ชั้นแอปพลิเคชันต้องการ
- **Application Layer**: ผู้ใช้ทั้งสองฝั่งจะปฏิสัมพันธ์กับชั้นแอปพลิเคชันผ่านทางแอปพลิเคชันที่ใช้งาน เช่นเว็บเบราว์เซอร์

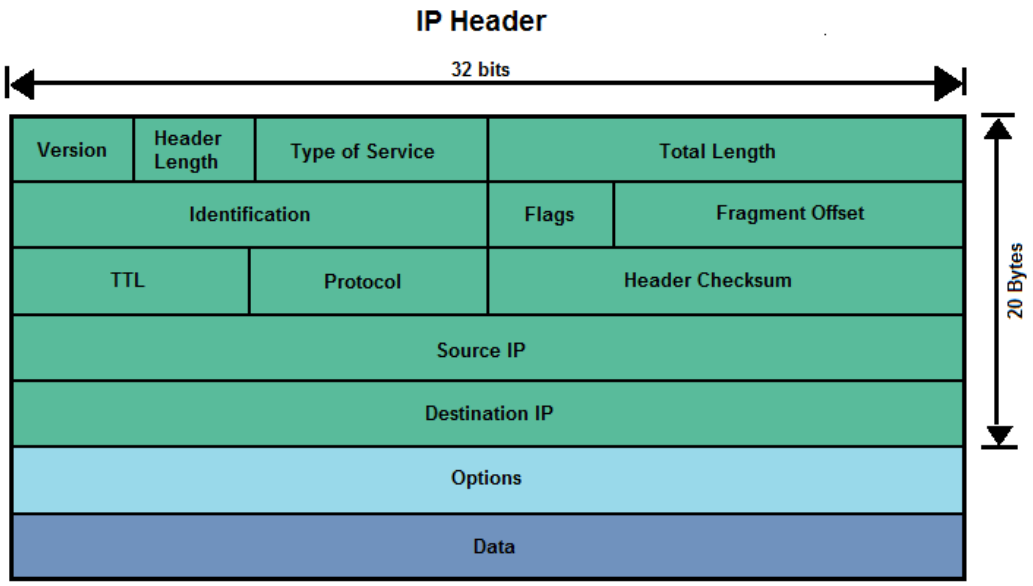
2.1.3 **โปรโตคอล** TCP/IP

TCP (Transmission Control Protocol) เป็นโปรโตคอลแบบ connected-oriented ที่ให้การส่งข้อมูลที่เชื่อถือได้และเป็นระเบียบระหว่างแอปพลิเคชันที่ทำงานบนเครื่องต่างๆ ในเครือข่าย เป็นหนึ่งในโปรโตคอลหลักของชุดอินเทอร์เน็ตโปรโตคอลที่มักถูกเรียกว่า TCP/IP ตัว

TCP มีหน้าที่แบ่งข้อมูลออกเป็นส่วนๆ (segment) และส่งผ่านเครือข่าย นอกจากนี้ยังมีกลไกสำหรับการตรวจจับข้อผิดพลาด การควบคุมการไหล และการควบคุมความแออัดของข้อมูลที่เกิดขึ้นในระบบเครือข่าย

TCP ใช้วิธีการจับมือแบบสามทางเพื่อสร้างการเชื่อมต่อระหว่างสองโฮสต์ในระหว่างการจับมือกัน โฮสต์ทั้งสองจะแลกเปลี่ยนแพ็กเก็ตการซิงโครไนซ์ (SYN) และการตอบรับ (ACK) เพื่อตกลงหมายเลขลำดับ (sequence number) สำหรับการส่งข้อมูล เมื่อสร้างการเชื่อมต่อแล้ว TCP จะส่งข้อมูลเป็นเซ็กเมนต์ แต่ละเซกเมนต์ประกอบด้วยหมายเลขลำดับและหมายเลขการตอบรับ ซึ่งใช้เพื่อให้แน่ใจว่าการส่งข้อมูลมีความน่าเชื่อถือและเป็นลำดับ หากเซ็กเมนต์สูญหายหรือเสียหายระหว่างการส่ง ผู้รับสามารถร้องขอการส่งสัญญาณเซ็กเมนต์นั้นซ้ำได้ โดยสรุปแล้ว TCP เป็นระเบียบวิธีการถ่ายโอนข้อมูลผ่านเครือข่ายที่มีเชื่อถือได้ รักษาลำดับของข้อมูล และมีประสิทธิภาพ

IP (Internet Protocol) เป็นโปรโตคอลในชั้นเครือข่าย เป็นวิธีการพื้นฐานสำหรับการสื่อสารข้ามเครือข่าย มีหน้าที่กำหนดเส้นทางแพ็กเก็ตข้อมูลจากโฮสต์หนึ่งไปยังอีกโฮสต์หนึ่งในเครือข่ายของคอมพิวเตอร์ที่เชื่อมต่อถึงกัน แพ็กเก็ต IP ประกอบด้วยส่วนหัวและเพย์โหลด ส่วนหัวประกอบด้วยข้อมูล เช่น ที่อยู่ IP ต้นทางและปลายทาง ความยาวแพ็คเก็ต และประเภทโปรโตคอล ในส่วนที่เรียกว่า payload คือส่วนข้อมูลข้อมูลจริงที่กำลังส่ง IP เป็นโปรโตคอลที่ไม่จำเป็นต้องไม่มีการสร้างการเชื่อมต่อไว้ล่วงหน้าก่อนที่จะส่งข้อมูล แต่ละแพ็กเก็ตจะถูกส่งโดยอิสระและอาจใช้เส้นทางที่แตกต่างกันผ่านเครือข่ายเพื่อไปยังปลายทาง ซึ่งโครงสร้างของ IP แพ็กเก็ตแสดงในภาพที่ 3



ภาพประกอบ 3 IPv4 Header

ที่มา: https://www.networkurge.com/2017/10/ip-header-details.html

เพื่อให้แน่ใจว่าแพ็กเก็ตถูกส่งไปยังปลายทางที่ถูกต้อง แต่ละโฮสต์บนเครือข่ายจะได้รับที่อยู่ IP (IP address) ที่ไม่ซ้ำกัน ซึ่งเป็นตัวเลข 32 บิตสำหรับ IP โปรโตคอลรุ่นที่4 ที่แสดงในรูปแบบจุดทศนิยม ที่อยู่ IP แบ่งออกเป็นสองส่วน: รหัสเครือข่ายและรหัสโฮสต์ รหัสเครือข่ายระบุเครือข่ายที่เป็นเจ้าของโฮสต์ ในขณะที่รหัสโฮสต์ระบุโฮสต์แต่ละรายการภายในเครือข่ายนั้น

โดยสรุปแล้ว IP เป็นโปรโตคอลพื้นฐานที่ช่วยให้สามารถสื่อสารผ่านเครือข่ายคอมพิวเตอร์ได้ เป็นกลไกพื้นฐานสำหรับกำหนดเส้นทางแพ็กเก็ตข้อมูลระหว่างโฮสต์และจำเป็นต่อการทำงานของอินเทอร์เน็ต

คำอธิบายของฟิลด์สำคัญบางส่วนในส่วนหัวของ IP แสดงได้ดังนี้:

- **Version**: ฟิลด์นี้ระบุเวอร์ชันของโปรโตคอล IP ที่ใช้ เช่น 4 สำหรับ IP เวอร์ชัน 4
- **Header Length**: ฟิลด์นี้ระบุความยาวของส่วนหัว IP หากเป็น IPv4 ค่าต่ำสุดคือ 5 ซึ่งแสดงถึงความยาวส่วนหัว 20 ไบต์ ในขณะที่ค่าสูงสุดคือ 15
- **Total Length**: ฟิลด์นี้ระบุความยาวทั้งหมดของแพ็กเก็ต IP รวมทั้งส่วนหัวและเพย์โหลด

- **Identification**: ฟิลด์นี้ใช้เพื่อระบุแต่ละแพ็กเก็ต IP โดยไม่ซ้ำกัน โดยทั่วไปจะเพิ่มขึ้นสำหรับแพ็กเก็ตใหม่แต่ละแพ็กเก็ต
- **Time to Live (TTL)**: ฟิลน์นี้ระบุจำนวนฮ็อพสูงสุดที่แพ็คเก็ตสามารถรับได้ก่อนที่จะถูกยกเลิก
- **Source IP Address**: หมายเลข IP address ของผู้ส่ง
- **Destination IP Address**: หมายเลข IP address ของผู้รับ

2.1.4 **การเรียนรู้ด้วยเครื่อง** (Machine Learning)

การเรียนรู้ของเครื่องเป็นแขนงหนึ่งของปัญญาประดิษฐ์ (AI) ที่เกี่ยวข้องกับการสอนคอมพิวเตอร์ให้เรียนรู้จากข้อมูลโดยไม่ต้องตั้งโปรแกรมไว้อย่างชัดเจน กล่าวอีกนัยหนึ่ง อัลกอริทึมการเรียนรู้ของเครื่องใช้เทคนิคทางสถิติเพื่อวิเคราะห์ข้อมูลและระบุรูปแบบ แล้วใช้รูปแบบเหล่านั้นเพื่อทำการทำนายหรือตัดสินใจ

อัลกอริทึมการเรียนรู้ของเครื่องประเภทหนึ่งที่พบได้บ่อยคือโครงข่ายประสาทเทียม ซึ่งได้รับแรงบันดาลใจจากโครงสร้างและการทำงานของสมองมนุษย์ อัลกอริทึมแมชชีนเลิร์นนิงประเภทอื่นๆ ได้แก่ Decision Tree, support vector machine และ K-Nearest Neighbors

- **K-Nearest Neighbors (KNN)**

K-Nearest Neighbors (KNN) เป็นประเภทของอัลกอริทึมการเรียนรู้ของเครื่องภายใต้การดูแลที่ใช้สำหรับงานจัดประเภทและการถดถอย KNN เป็นอัลกอริทึมที่เรียบง่ายและใช้งานง่ายซึ่งทำงานโดยการค้นหาจุดข้อมูล K ในชุดการฝึกที่ใกล้กับจุดข้อมูลใหม่มากที่สุด จากนั้นจัดประเภทหรือคาดการณ์จุดข้อมูลใหม่ตามป้ายกำกับของเพื่อนบ้าน K ที่ใกล้ที่สุด กล่าวอีกนัยหนึ่ง เมื่อกำหนดจุดข้อมูลใหม่ KNN จะพิจารณาจุดข้อมูล K ที่ใกล้เคียงที่สุดในชุดการฝึก และกำหนดคลาสหรือป้ายกำกับที่พบได้บ่อยที่สุดในจุด K เหล่านั้น ค่าของ K คือไฮเปอร์พารามิเตอร์ที่ผู้ใช้สามารถตั้งค่าได้ และกำหนดจำนวนเพื่อนบ้านที่ใช้ในการจัดประเภทหรือการคาดคะเน

KNN มักใช้สำหรับปัญหาการจำแนกประเภท โดยเป้าหมายคือการทำนายระดับหรือหมวดหมู่ของจุดข้อมูลใหม่ตามคุณลักษณะ ตัวอย่างเช่น เมื่อพิจารณาชุดข้อมูลของสัตว์ที่มีคุณสมบัติต่างๆ เช่น น้ำหนัก ส่วนสูง และความยาวขน KNN สามารถใช้เพื่อจำแนกสัตว์ชนิดใหม่เป็นสุนัข แมว หรือนกตามคุณลักษณะของมัน

KNN เป็นอัลกอริทึมแบบไม่มีพารามิเตอร์ ซึ่งหมายความว่าจะไม่ตั้งสมมติฐานใดๆ เกี่ยวกับการกระจายข้อมูลพื้นฐาน สิ่งนี้ทำให้เป็นอัลกอริทึมที่ยืดหยุ่นและทรงพลังที่สามารถ

ทำงานได้ดีกับชุดข้อมูลที่หลากหลาย อย่างไรก็ตาม KNN อาจมีราคาแพงในการคำนวณสำหรับชุดข้อมูลขนาดใหญ่ เนื่องจากต้องมีการคำนวณระยะห่างระหว่างจุดข้อมูลใหม่และจุดทั้งหมดในชุดการฝึก นอกจากนี้ การเลือก K อาจมีผลกระทบอย่างมากต่อความแม่นยำของอัลกอริทึม และการหาค่าที่เหมาะสมที่สุดของ K อาจเป็นงานที่ท้าทาย

- **Decision Tree**

ต้นไม้การตัดสินใจเป็นประเภทของอัลกอริทึมการเรียนรู้ของเครื่องภายใต้การดูแลที่ใช้สำหรับการจำแนกประเภทและการถดถอย เป็นแบบจำลองกราฟิกที่แสดงถึงชุดของการตัดสินใจและผลที่ตามมาที่เป็นไปได้ ในต้นไม้การตัดสินใจ โหนดแสดงถึงจุดตัดสินใจหรือคุณลักษณะ กิ่งก้านแสดงถึงค่าที่เป็นไปได้ของคุณลักษณะ และใบไม้แสดงถึงผลลัพธ์สุดท้ายหรือการจำแนกประเภท เป้าหมายของอัลกอริทึมคือการสร้างแผนผังที่สามารถจำแนกหรือทำนายผลลัพธ์ของจุดข้อมูลใหม่ได้อย่างแม่นยำตามคุณลักษณะ

อัลกอริทึมแผนผังการตัดสินใจทำงานโดยแบ่งข้อมูลซ้ำๆ ออกเป็นส่วนย่อยตามคุณลักษณะที่มีข้อมูลมากที่สุด จนกว่าแต่ละชุดย่อยจะมีจุดข้อมูลที่มีป้ายกำกับหรือคลาสเดียวกัน อัลกอริทึมใช้ตัวชี้วัดเช่น information gain หรือดัชนี Gini เพื่อกำหนดการแยกที่ดีที่สุดในแต่ละโหนด และเพื่อกำหนดว่าเมื่อใดควรหยุดการแยกข้อมูล

ต้นไม้การตัดสินใจมีข้อดีหลายประการเหนืออัลกอริทึมการเรียนรู้ของเครื่องอื่นๆ รวมถึงความเรียบง่าย ความสามารถในการตีความ และความสามารถในการจัดการข้อมูลทั้งแบบหมวดหมู่และแบบต่อเนื่อง นอกจากนี้ยังสามารถใช้สำหรับการเลือกคุณลักษณะและการแสดงข้อมูล

อย่างไรก็ตาม โครงสร้างการตัดสินใจอาจมีแนวโน้มที่จะเกินพอดี (Over fitting) โดยที่แบบจำลองนั้นซับซ้อนเกินไปและทำงานได้ดีกับข้อมูลการฝึกอบรม แต่ไม่ดีสำหรับข้อมูลการทดสอบ เพื่อแก้ไขปัญหานี้ เทคนิคต่างๆ เช่น pruning, ensemble methods, และrandom forests สามารถใช้เพื่อปรับปรุงความแม่นยำและการวางนัยทั่วไป(generalization)ของแบบจำลองต้นไม้ตัดสินใจ

- **Random Forest**

Random Forest เป็นอัลกอริทึมการเรียนรู้ของเครื่องภายใต้การดูแลซึ่งอิงจากแผนผังการตัดสินใจ เป็นอัลกอริทึมที่ทรงพลังและเป็นที่นิยมซึ่งใช้สำหรับทั้งการจัดหมวดหมู่และการถดถอย Random Forest ทำงานโดยการสร้างแผนผังการตัดสินใจหลายชุดบนชุดข้อมูลย่อยที่เลือกแบบสุ่ม จากนั้นจึงรวมผลลัพธ์ของแผนผังเหล่านั้นเพื่อทำการทำนายขั้นสุดท้ายหรือการ

จำแนกประเภท การเลือกชุดย่อยและฟีเจอร์แบบสุ่มช่วยลดการโอเวอร์ฟิตติ้งและเพิ่มความแม่นยำของโมเดล

ใน Random forest ต้นไม้การตัดสินใจแต่ละต้นจะถูกสร้างขึ้นจากชุดข้อมูลย่อยที่เลือกแบบสุ่มและชุดย่อยที่เลือกแบบสุ่มของคุณสมบัติ สิ่งนี้ช่วยลดความสัมพันธ์ระหว่างต้นไม้และทำให้แน่ใจว่าต้นไม้แต่ละต้นมีการตัดสินใจที่แตกต่างกัน เมื่อสร้างต้นไม้ทั้งหมดแล้ว อัลกอริทึมจะรวมผลลัพธ์ของต้นไม้เหล่านี้เพื่อทำการทำนายหรือจัดประเภทขั้นสุดท้าย

Random forest มีข้อดีหลายประการเหนืออัลกอริทึมการเรียนรู้ของเครื่องอื่นๆ รวมถึงความสามารถในการจัดการข้อมูลมิติสูง ความสัมพันธ์ที่ไม่ใช่เชิงเส้น และค่าที่ขาดหายไป นอกจากนี้ยังมีแนวโน้มที่จะ overfitting น้อยกว่าแผนผังการตัดสินใจเดี่ยว และสามารถให้การวัดความสำคัญของคุณลักษณะและการโต้ตอบที่แปรผันได้

อย่างไรก็ตาม โมเดล Random forest อาจมีราคาแพงในการคำนวณและอาจต้องใช้หน่วยความจำจำนวนมาก โดยเฉพาะอย่างยิ่งสำหรับชุดข้อมูลขนาดใหญ่ที่มีคุณสมบัติมากมาย นอกจากนี้ อาจเป็นเรื่องยากที่จะตีความและทำให้เห็นภาพ เนื่องจากการคาดคะเนหรือการจำแนกขั้นสุดท้ายจะขึ้นอยู่กับผลรวมของแผนผังการตัดสินใจหลายต้น

- **Support Vector Machine**

Support Vector Machine (SVM) เป็นอัลกอริธึมการเรียนรู้ของเครื่องภายใต้การดูแลที่ใช้สำหรับงานจำแนกประเภทและการถดถอย SVM เป็นอัลกอริทึมที่ทรงพลังและเป็นที่นิยม ซึ่งทำงานโดยการค้นหาไฮเปอร์เพลนที่เหมาะสมที่สุดซึ่งแยกจุดข้อมูลออกเป็นคลาสตามลำดับ

ในปัญหาการจำแนกประเภทแบบไบนารี SVM จะค้นหาไฮเปอร์เพลนที่เพิ่มระยะขอบระหว่างคลาสทั้งสองให้มากที่สุด นั่นคือ ระยะห่างระหว่างไฮเปอร์เพลนและจุดข้อมูลที่ใกล้เคียงที่สุดจากแต่ละคลาส ไฮเปอร์เพลนนี้สามารถเป็นแบบเชิงเส้นหรือไม่เชิงเส้น ขึ้นอยู่กับชนิดของฟังก์ชันเคอร์เนลที่ใช้ ฟังก์ชันเคอร์เนลแมปข้อมูลอินพุตในพื้นที่มิติที่สูงกว่า ซึ่งไฮเปอร์เพลนเชิงเส้นสามารถแยกคลาสได้อย่างมีประสิทธิภาพมากขึ้น

SVM ยังสามารถใช้กับปัญหาการจำแนกประเภทหลายชั้นโดยใช้เทคนิคต่างๆ เช่น หนึ่งต่อหนึ่งหรือหนึ่งต่อหนึ่ง ในรูปแบบหนึ่งต่อหนึ่ง SVM จะสร้างตัวแยกประเภทไบนารีหลายตัวสำหรับแต่ละคู่ของคลาส จากนั้นจึงรวมผลลัพธ์เข้าด้วยกันเพื่อทำการทำนายขั้นสุดท้าย ในแบบหนึ่งต่อทั้งหมด SVM จะสร้างตัวแยกประเภทไบนารีสำหรับแต่ละคลาสเทียบกับที่เหลือ จากนั้นเลือกคลาสที่มีคะแนนความเชื่อมั่นสูงสุดเป็นตัวทำนายสุดท้าย

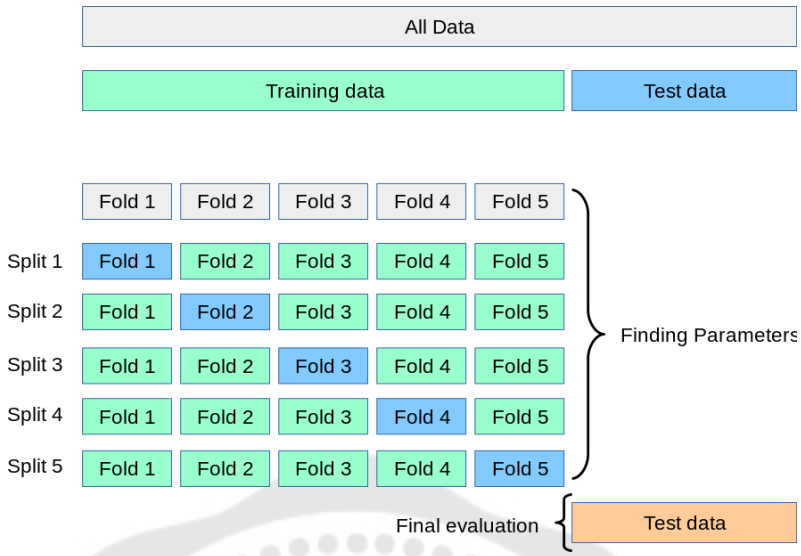
SVM มีข้อดีหลายประการเหนืออัลกอริทึมการเรียนรู้ของเครื่องอื่นๆ รวมถึงความสามารถในการจัดการข้อมูลมิติสูง ความสัมพันธ์ที่ไม่ใช่เชิงเส้น และข้อมูลที่มีสัญญาณรบกวน นอกจากนี้ ยังมีแนวโน้มน้อยที่จะเกินพอดีกว่ารุ่นอื่น ๆ และสามารถให้การวัดความสำคัญของคุณลักษณะและการโต้ตอบที่ผันแปรได้

อย่างไรก็ตาม SVM อาจมีราคาแพงในการคำนวณสำหรับชุดข้อมูลขนาดใหญ่ โดยเฉพาะอย่างยิ่งสำหรับเคอร์เนลที่ไม่ใช่เชิงเส้น นอกจากนี้ การเลือกฟังก์ชันเคอร์เนลและพารามิเตอร์อาจมีผลกระทบอย่างมากต่อความแม่นยำของโมเดล และการค้นหาค่าที่เหมาะสมที่สุดอาจเป็นงานที่ท้าทาย

2.1.4 K-Fold Cross Validation

K-Fold Cross Validation เป็นเทคนิคที่ใช้เพื่อประเมินประสิทธิภาพของแบบจำลองการเรียนรู้ของเครื่อง แนวคิดพื้นฐานคือการแบ่งชุดข้อมูลที่ใช้ในสอนออกเป็น k ส่วนเท่าๆ กันหรือ "Fold" การพับแบบ k-1 ใช้สำหรับการฝึกโมเดล และการพับที่เหลือใช้สำหรับทดสอบประสิทธิภาพของโมเดล กระบวนการนี้ทำซ้ำ k ครั้ง โดยแต่ละ k ครั้งจะถูกใช้เป็นชุดทดสอบหนึ่งครั้ง จากนั้นประสิทธิภาพของโมเดลจะถูกเฉลี่ยจากการวนซ้ำ k ครั้ง เพื่อให้ค่าประมาณโดยรวมของประสิทธิภาพ ข้อได้เปรียบการใช้ k-fold cross validation คือให้การประมาณประสิทธิภาพของแบบจำลองที่เชื่อถือได้มากกว่าการแบ่งข้อมูลใช้สอนและใช้ทดสอบเพียงชุดเดียว ด้วยการทำซ้ำขั้นตอนหลาย ๆ ครั้ง การประเมินด้วยวิธีการนี้มีโอกาสน้อยที่จะมีอคติ(bias)ได้น้อยกว่าการที่แบ่งข้อมูลสำหรับสอนและทดสอบเพียงชุดเดียว

ค่า k ที่ใช้กันโดยทั่วไปสำหรับคือ 10 ซึ่งหมายถึงว่าชุดข้อมูลถูกแบ่งออกเป็น 10 ส่วนเท่าๆ กัน และดำเนินการซ้ำ 10 ครั้ง ดังแสดงในรูปที่ 4 อย่างไรก็ตาม ค่า k สามารถปรับได้ตามขนาดของชุดข้อมูลและระดับความแม่นยำที่ต้องการ



ภาพประกอบ 4 K-Flod Cross validation

ที่มา https://scikit-learn.org/stable/modules/cross_validation.html

2.1.5 พฤติกรรมและลักษณะของ IP Identification

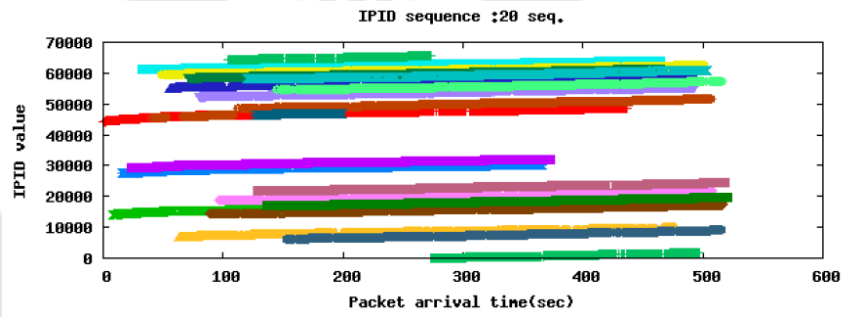
ในเลเยอร์ IP (Internet Protocol) Identification (หรือ ID) หมายถึงฟิลด์ภายในส่วนหัวของ IP ที่ช่วยระบุและประกอบแพกเก็ต IP ที่กระจัดกระจายกลับมาอีกครั้ง เมื่อข้อมูลถูกส่งผ่านเครือข่าย IP ข้อมูลจะถูกแบ่งออกเป็นหน่วยเล็กๆ เรียกว่า IP แพกเก็ต แพกเก็ตเหล่านี้อาจต้องถูกแยกส่วนออกเป็นชิ้นเล็กๆ หากขนาดของแพกเก็ตเกินหน่วยส่งข้อมูลสูงสุด (MTU) ของเครือข่าย

ID ฟิลด์มีขนาด 16 บิตที่ไม่ซ้ำกันในแต่ละ IP แพ็กเก็ต หาก IP แพ็กเก็ต IP ขนาดใหญ่ถูกแยกส่วนออกเป็นแฟรกเมนต์ที่มีขนาดเล็กลง แต่ละแฟรกเมนต์จะมีค่า ID เหมือนกันกับแพ็กเก็ตดั้งเดิมของมัน สิ่งนี้ทำให้อุปกรณ์ที่รับข้อมูลสามารถระบุและจัดกลุ่มชิ้นส่วนเข้าด้วยกันเพื่อสร้างแพกเก็ตเดิมขึ้นใหม่

ฟิลด์ ID ในส่วนหัว IP สร้างขึ้นโดยระบบปฏิบัติการที่ส่งและใช้เพื่อระบุแพกเก็ต IP แต่ละรายการ อย่างไรก็ตาม ลักษณะการทำงานเฉพาะและการใช้งานฟิลด์ ID อาจแตกต่างกันไปตามระบบปฏิบัติการต่างๆ ลักษณะของ IPID อาจเปลี่ยนแปลงได้ด้วยการอัปเดตหรือการกำหนดค่าต่างๆ ภายในระบบปฏิบัติการเช่นกัน นอกจากนี้ เวอร์ชันหรือรีลีสของระบบปฏิบัติการอาจมีลักษณะการทำงานที่แตกต่างกัน ดังนั้นค่าความแตกต่างนี้อาจนำมาใช้ในการแยกแยะระหว่างอุปกรณ์ได้เช่นกัน

จากงาน [Counting NATted Hosts by Observing TCP/IP Field Behaviors] (Mongkolluksamee et al., 2012) แสดงข้อแตกต่างทั่วไปบางประการที่สามารถสังเกตได้จากลักษณะการทำงานของ IPID ในระบบปฏิบัติการต่างๆ

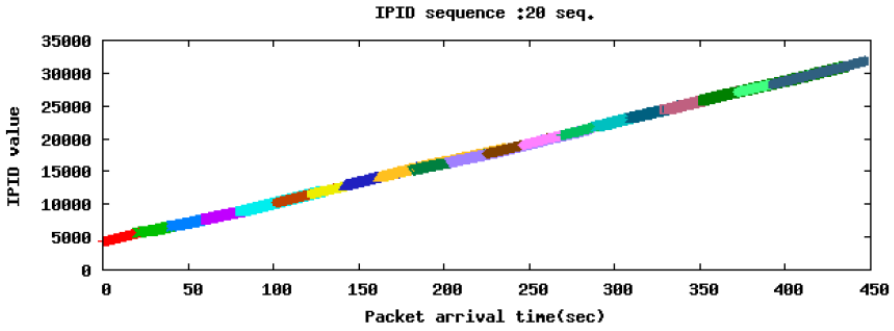
Sequential IPID: ระบบปฏิบัติการบางระบบกำหนดค่า IPID ตามลำดับสำหรับแต่ละแพกเก็ตที่ส่ง ซึ่งหมายความว่าแต่ละแพกเก็ตที่ตามมาซึ่งส่งโดยอุปกรณ์เดียวกันจะมีค่า IPID เพิ่มขึ้น IPID ตามลำดับสามารถให้ความรู้สึกถึงลำดับแพกเก็ตและสามารถช่วยในการประกอบใหม่ได้ จากรูปที่ 5 ด้านล่างจะเห็นได้ว่าแต่ละเส้นเป็นแถวของข้อมูลในแต่ละ session ในแต่ละ session จะมีค่า IPID ที่เรียงกันไป



ภาพประกอบ 5 Sequential IPID

ที่มา: (Mongkolluksamee et al., 2012)

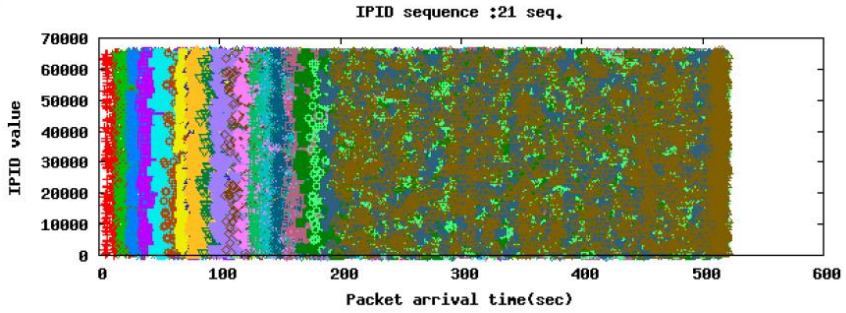
Incremental IPID: ระบบปฏิบัติการบางระบบสร้างค่า IPID โดยการเพิ่มค่าคงที่สำหรับแต่ละแพ็กเก็ตที่ส่ง โดยไม่คำนึงถึงแหล่งที่มาเช่นมาจากคนละโปรแกรม วิธีการนี้สามารถคาดการณ์ได้น้อยกว่าเมื่อเทียบกับ IPID ตามลำดับ โดยเฉพาะอย่างยิ่งหากอุปกรณ์หลายเครื่องสร้างแพกเก็ตพร้อมกัน จากรูปที่ 6 ด้านล่างจะเห็นได้ว่าจะมีเพียงส้นเดียวยาวแต่ละช่วงของสีคือแต่ละ session แสดงให้เห็นว่าทุก session ใช้ค่า IPID ที่ต่อเนื่องกัน



ภาพประกอบ 6 Incremental IPID

ที่มา: (Mongkolluksamee et al., 2012)

Random IPID: ระบบปฏิบัติการบางระบบใช้ค่า IPID แบบสุ่มสำหรับแต่ละแพกเก็ตที่ส่ง การสุ่ม IPID สามารถช่วยปรับปรุงความปลอดภัยโดยทำให้ผู้โจมตีคาดเดาหรือติดตามการไหลของแพกเก็ตได้ยากขึ้น อย่างไรก็ตาม การทำเช่นนี้อาจทำให้การประกอบชิ้นส่วนใหม่และการติดตามแพกเก็ตมีความท้าทายมากขึ้น จากรูป 7 จะเห็นได้ว่าในแต่ละสีคือแต่ละ session มีค่า IPID ที่สุ่มกระจายเป็นแถบตั้งแต่ค่าน้อยสุดไปค่ามากสุดตลอดช่วงเวลาที่ส่งข้อมูล



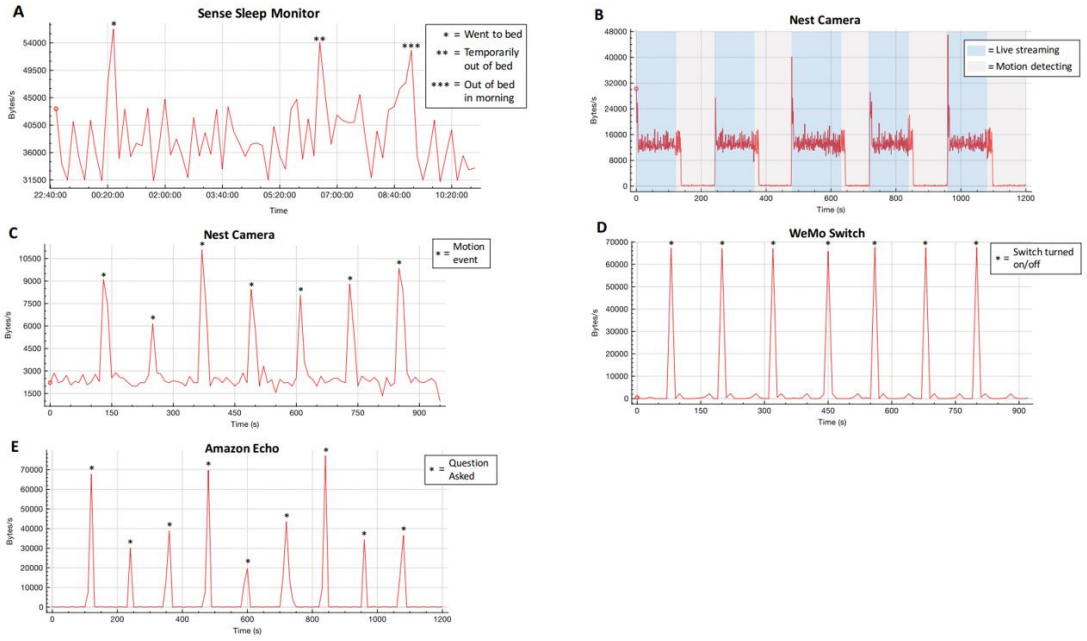
ภาพประกอบ 7 Random IPID

ที่มา: (Mongkolluksamee et al., 2012)

2.1.6 พฤติกรรมการใช้งานเครือข่ายของ IoT และไม่ใช่ IoT

ด้วยความแตกต่างในลักษณะการทำงานหรือใช้งาน ทำให้อุปกรณ์ IoT และอุปกรณ์ที่ไม่ใช่ IoT นั้นมีพฤติกรรมในการใช้งานเครือข่ายที่แตกต่างเช่น

อัตราการส่ง/รับข้อมูลในเครือข่ายทสอดคล้องกับการโต้ตอบของผู้ใช้ จากงาน[A Smart Home is No Castle: Privacy Vulnerabilities of Encrypted IoT Traffic] (Apthorpe et al., 2017) จากรูปที่ 8 แสดงตัวอย่างข้อมูลในเครือข่ายที่เกิดขึ้นจากอุปกรณ์ได้รับการกระตุ้นจากภายนอกหรือจากผู้ใช้โดยตรง จะเห็นได้ว่าเมื่ออุปกรณ์ IoT ตรวจพบเหตุการณ์อาทิเช่นมีการขยับตัวของผู้ใช้ในกลุ่มอุปกรณ์ตรวจจับการเคลื่อนไหว อุปกรณ์ก็จะมีการส่งข้อมูลที่สูงขึ้นในทันทีที่มีการตรวจพบเหตุการณ์ หรือเกิดการส่งข้อมูลที่มากขึ้นในอุปกรณ์กลุ่มของ switch เมื่อผู้ใช้มีการสั่งงานเปิดหรือปิด

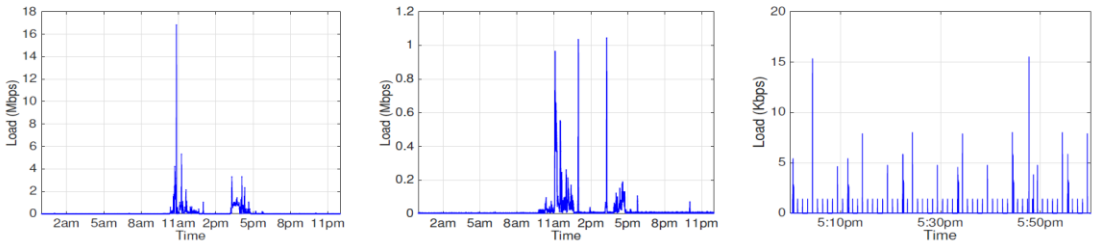


ภาพประกอบ 8 อัตราการส่ง/รับข้อมูลในเครือข่ายของอุปกรณ์ IoT 5 ชนิด ที่แสดงการอัตราการส่ง/รับที่เพิ่มขึ้นอย่างเห็นได้ชัดซึ่งสอดคล้องกับการโต้ตอบของผู้ใช้

ที่มา: (Apthorpe et al., 2017)

รูปแบบของโหลดที่เกิดขึ้นจากอุปกรณ์ IoT และที่ไม่ใช่ IoT มีความแตกต่างกัน ซึ่งเกี่ยวข้องโดยตรงจากการออกแบบเพื่อการใช้งานของอุปกรณ์ จากงาน [Characterizing and classifying IoT traffic in smart cities and campuses] (Sivanathan et al., 2017) แสดงให้เห็นว่าอุปกรณ์ IoT นั้นโดยปกติแล้วจะส่งข้อมูลในเครือข่ายน้อยมากดังแสดงในรูปที่ 9 (b) และ (c) ซึ่งต่างจากอุปกรณ์ที่ไม่ใช้ IoT เช่น computer หรือ mobile phone ซึ่งมักจะมีการใช้งานเครือข่ายมี่

สูงดังแสดงในรูปที่ 9 (a) เมื่อนำข้อมูลทั้ง IoT และไม่ใช่ IoT มารวมกันจะเห็นได้ว่าปริมาณข้อมูลส่วนใหญ่เกิดจากอุปกรณ์ที่ไม่ใช่ IoT



(a) Load: IoT and non-IoT. (b) Load: IoT only. (c) Load: LiFX light bulb.

ภาพประกอบ 9รูปแบบของโหลดที่เกิดขึ้นจากอุปกรณ์ IoT และที่ไม่ใช่ IoT

ที่มา: (Sivanathan et al., 2017)

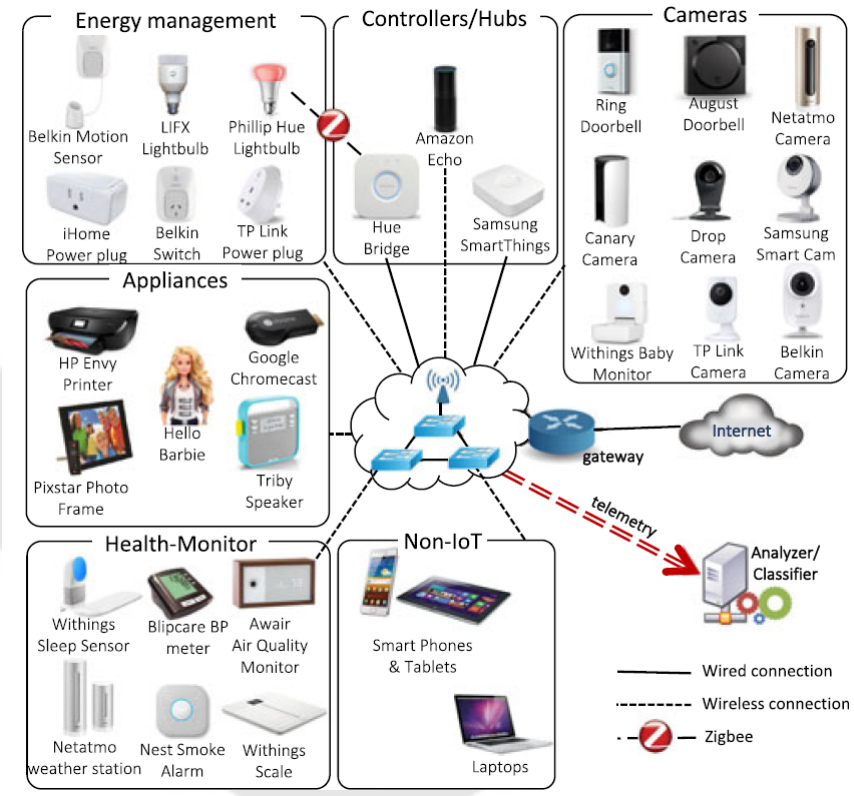
ปลายทางการสื่อสารของอุปกรณ์ที่เจาะจง อุปกรณ์ IoT นั้นเป็นอุปกรณ์เป็นอุปกรณ์ที่มักระบุปลายทางการสื่อสารไว้ล่วงหน้า เช่นอุปกรณ์พวกนี้มักส่งไปยังเซิร์ฟเวอร์หรือกลุ่มของเซิร์ฟเวอร์ที่เจ้าของผลิตภัณฑ์ได้มีการกำหนดไว้เพื่อให้บริการเฉพาะอุปกรณ์

2.2 Dataset

คุณ Sivanathan และเพื่อน [Classifying IoT Devices in Smart Environments Using Network Traffic Characteristics] (Sivanathan et al., 2019) นำเสนอชุดข้อมูลสำหรับการจำแนกประเภทอุปกรณ์ IoT ตามลักษณะการรับส่งข้อมูลในเครือข่าย โดยมีโครงสร้างการเชื่อมต่อเพื่อจัดเก็บข้อมูลดังรูปที่ 10 โดยแสดงอุปกรณ์ IoT ที่แตกต่างกัน 28 อุปกรณ์ที่เชื่อมต่อพร้อมกับอุปกรณ์ที่ไม่ใช่ IoT จากนั้นข้อมูลการสื่อสารในเครือข่ายจะถูกจัดเก็บเพื่อใช้ในการวิเคราะห์

อุปกรณ์ต่างๆในเครือข่ายมีดังนี้ อุปกรณ์ IoT ประกอบด้วยกลุ่มกล้อง (Nest Dropcam, Samsung SmartCam, Netatmo Welcome, Belkin camera, TP-Link Day Night Cloud camera, Withings Smart Baby Monitor, Canary camera, August door bell, Ring door bell), กลุ่มสวิตช์และทริกเกอร์ (iHome, TP-Link Smart Plug, Belkin Wemo Motion Sensor, Belkin Wemo Switch), กลุ่มฮับ (Smart Things, Amazon Echo), กลุ่มเซ็นเซอร์คุณภาพอากาศ (NEST Protect smoke alarm, Netatmo Weather station, Awair air quality monitor), กลุ่มอุปกรณ์อิเล็กทรอนิกส์ (Triby Speaker, PIXSTAR Photoframe, HP Printer, Hello barbie,

Google Chromecast) , กลุ่มอุปกรณ์ดูแลสุขภาพ (Withings Smart scale, Withings Aura smart sleep sensor, Blipcare lood pressure meter) และกลุ่มหลอดไฟ (Phips Hue และ LiFX Smart Bulb) อุปกรณ์ที่ไม่ใช่ IoT จำนวนมากยังเชื่อมต่อกับเครือข่าย เช่น แล็ปท็อป โทรศัพท์มือถือ และแท็บเล็ต Android แท็บเล็ตใช้เพื่อกำหนดค่าอุปกรณ์ IoT ตามคำแนะนำของผู้ผลิตอุปกรณ์ที่เกี่ยวข้อง



ภาพประกอบ 10 การเชื่อมต่ออุปกรณ์เพื่อการจัดเก็บข้อมูลของชุดข้อมูล

ที่มา: https://iotanalytics.unsw.edu.au/traffic_analysis.html

ชุดข้อมูลที่จะนำมาใช้สำหรับงานวิจัยชิ้นนี้จะเป็นข้อมูลที่นำมาจากเว็บไซต์ https://iotanalytics.unsw.edu.au/ โดยข้อมูลที่ได้รับมานั้นจะเป็นข้อมูลที่เป็นไฟล์นามสกุล .pcap โดยมีไฟล์ทั้งหมด 20 ไฟล์ โดยแต่ละไฟล์นั้นจะเป็นข้อมูลการไหลเวียนของ packet ของเครือข่ายภายใน 1 วัน โดยเริ่มเก็บข้อมูลวันที่ 23 กันยายน ปี พ.ศ. 2559 จนถึงวันที่ 12 ตุลาคม

พ.ศ. 2559 แสดงดังรูปที่ 11 โดยข้อมูลภายใน PCAP ประกอบด้วยข้อมูลทั้งหมดที่ถูกส่งออกมาก ซึ่งประกอบด้วยส่วนหัวของแพกเก็ตและข้อมูลทั้งหมด

ภาพประกอบ 11วันที่จัดเก็บข้อมูลใน Dataset

2.3 งานวิจัยที่เกี่ยวข้อง

2.3.1 A machine learning approach for IoT device identification based on network traffic analysis (Meidan, Bohadana, Shabtai, Guarnizo, et al., 2017)

เป็นงานวิจัยที่สร้างแบบจำองการทำนายอุปกรณ์ IoT และไม่ใช่ IoT ออกจากกัน โดยการใช้งานการเรียนรู้ของเครื่องเพื่อระบุถึงอุปกรณ์ที่เชื่อมต่อภายในระบบเครือข่ายโดยการใช้งานการวิเคราะห์เครือข่ายจากการรับไฟล์นามสกุล PCAP มาวิเคราห์ package ที่ไหลเวียนภายในเครือข่าย โดยอาศัยการสังเกต หมายเลข IP ต้นทางและปลายทาง และหมายเลข port โดยโดยสังเกตุการตื่อสารตั้งแต่จุดเริ่มต้น (SYN) จนถึงจุดสิ้นสุด (FIN) เป็นราย session ของการเชื่อมต่อโดยแบ่งข้อมูลออกเป็นสามส่วน ได้แก่ Single-session Multi-session classifier และ Test set และประเมินผลด้วยวิธีการ GBM, Ramdom Forest และ XGBoost

2.3.2 Characterizing and classifying IoT traffic in smart cities and campuses (Sivanathan et al., 2017)

เป็นงานวิจัยที่มุ่งเน้นถึงความปลอดภัยของ ระบบเครือข่าย ทางด้านการโจมตี ทางระบบไซเบอร์ โดยการวิเคราะห์ระบบเครือข่าย และแสดงให้เห็นถึง ลักษณะบุคลิกภาพของอุปกรณ์ IoT และพฤติกรรมของอุปกรณ์ โดยการรวบรวมข้อมูล การไหลเวียนของระบบเครือข่ายจากสภาพแวดล้อม ของมหาวิทยาลัย ที่มีการติดตั้งอุปกรณ์ IoT มากกว่า 20 อุปกรณ์

2.3.3 Classifying IoT Devices in Smart Environments Using Network Traffic Characteristics (Sivanathan et al., 2019)

เป็นงานวิจัยที่มุ่งเน้นถึงความปลอดภัย ภายในระบบเครือข่าย โดยการติดตั้งอุปกรณ์ ทั้งหมด 28 ชิ้น และสังเกตถึง ลักษณะการใช้งานภายในจราจรของระบบเครือข่าย เป็นเวลากว่า 6 เดือน แต่ดูถึงคุณสมบัติทางสถิติเช่น หมายเลข port, การเข้ารหัส และบุคลิกภาพการทำงาน ของอุปกรณ์ เพื่อให้สามารถสร้างระบบตรวจจับอุปกรณ์ใด โดยไม่จำเป็นต้อง มีอุปกรณ์ที่เฉพาะทาง

2.3.4 Counting NATted Hosts by Observing TCP/IP FieldBehaviors (Mongkolluksamee et al., 2012)

เป็นงานวิจัยที่ แสดงถึง พฤติกรรมการใช้งานของระบบเครือข่าย ภายใต้โปรโตคอล TCP/IP โดยมุ่งเน้นวิเคราะห์ การใช้งานระบบเครือข่ายเป็นรายอุปกรณ์ เพื่อสังเกตบุคลิกภาพของการเปลี่ยนแปลงของค่า IPID ที่เกิดขึ้น

2.3.5 Detection of Unauthorized IoT Devices Using Machine Learning Techniques (Meidan, Bohadana, Shabtai, Ochoa, et al., 2017)

เป็นงานวิจัยที่มุ่งเน้นถึงการสร้างระบบความปลอดภัยภายในระบบเครือข่าย โดยมีจุดมุ่งหมายให้สามารถระบุอุปกรณ์ที่เป็น IoT อีทางองค์กรให้การอนุมัติการใช้งาน โดยงานวิจัยดังกล่าวจะใช้งานการเรียนรู้ของเครื่อง โดยแบบจำลองการทำนาย Random Forest โดยอยู่ในประเภท supervised machine learning

2.3.6 IoT device fingerprinting with sequence-based features Department of Computer Science (Aluthge, 2017)

เป็นงานวิจัยที่ มีจุดมุ่งหมายในการ สร้างระบบการระบุ อุปกรณ์ ที่ใช้พลังงานต่ำจากการเรียงลำดับของ packet ภายในระบบเครือข่าย โดยใช้งานการเรียนรู้ของเครื่องโดยใช้วิธีการจำแนกอุปกรณ์ โดยดูข้อมูล และเปรียบเทียบทั้งในรูปแบบของ Packet-based และ sequence-based

2.3.7 A Smart Home is No Castle: Privacy Vulnerabilities of Encrypted IoT Traffic (Apthorpe et al., 2017)

งานวิจัยที่ สำรวจลักษณะการใช้งาน ระบบเครือข่าย ของอุปกรณ์ในแต่ละชิ้น ทั้งที่เป็น IoT และไม่ใช่ IoT ด้วยความกังวลต่อความเป็นส่วนตัวของผู้ใช้งานอุปกรณ์ ถึงแม้ว่าข้อมูลดังกล่าวนั้นจะได้รับการเข้ารหัสแล้ว

บทที่ 3
วิธีการดำเนินการวิจัย

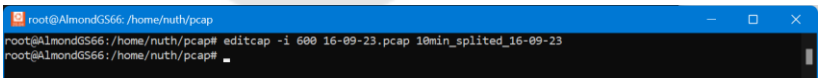
ในการทำวิจัยในครั้งนี้ผู้วิจัยได้ดำเนินการตามขั้นตอนดังนี้ จัดเตรียมชุดข้อมูลออกแบบการทดลองเพื่อศึกษาพฤติกรรมของอุปกรณ์ IoT และสร้างแบบจำลอง ML ต่างๆเพื่อแยกแยะระหว่างอุปกรณ์ IoT พร้อมทั้งเปรียบเทียบประสิทธิภาพการทำงานของแบบจำลองแต่ละชนิด

3.1 การจัดเตรียมเตรียมข้อมูล

เนื่องด้วยภายใน PCAP ไฟล์เป็นข้อมูลของอุปกรณ์ต่างๆที่ส่งข้อมูลพร้อมกันในเครือข่าย และแต่ละไฟล์มีเก็บข้อมูลเป็นเวลา 24 ชั่วโมงจึงจำเป็นต้องทำการเตรียมข้อมูลให้พร้อมในการสอน ML โมเดล โดยมีขั้นตอนหลักดังนี้ 1. แบ่งไฟล์ PCAP ยาว 24 ชม ออกเป็นไฟล์ละ 10 นาที, 2. สกัดข้อมูล PCAP ไฟล์ 10 นาทีให้อยู่ในรูป CSV ไฟล์ 10 นาที, 3. การแยกไฟล์ CSV 10 นาที เป็นไฟล์ย่อยของแต่ละอุปกรณ์, 4. รวมข้อมูลในทุก 10 นาทีของแต่ละอุปกรณ์เพื่อสร้างเป็น 1 sample ของ dataset

3.1.1 แบ่งไฟล์ PCAP ยาว 24 ชั่วโมงออกเป็นไฟล์ละ 10 นาที

PCAP ไฟล์ความยาว 24 ชั่วโมง นั้นมีข้อมูลยาวเกินไปและจุดประสงค์สำคัญของงานนี้คือการแยกแยะอุปกรณ์ได้ในระยะเวลาอันสั้น จึงได้แบ่ง PCAP ไฟล์ออกเป็นไฟล์ย่อยความยาว 10 นาที ด้วยโปรแกรม editcap -i 600 ดังแสดงในภาพ โดยมีความหมายให้ตัดไฟล์ PCAP ต้นฉบับออกเป็นไฟล์ย่อยความยาว 600 วินาทีเท่ากัน



ภาพประกอบ 12 การตัดแบ่งไฟล์ PCAP ออกเป็นไฟล์ละ 10 นาที

3.1.2 สกัดข้อมูลจาก PCAP ไฟล์ 10 นาทีให้อยู่ในรูป CSV ไฟล์ 10 นาที

เนื่องจากข้อมูลแบบ PCAP ไฟล์นั้นไม่สะดวกในการใช้งานสอน ML model โดยตรง จะทำให้ต้องสกัดเอาข้อมูลที่ต้องการจาก PCAP ไฟล์ ออกมาให้อยู่ในรูปแบบ CSV เพื่อความสะดวกในการทำงาน ซึ่งในงานวิจัยนี้เลือกที่จะตรวจอุปกรณ์ให้ได้ด้วยข้อมูลในชั้น Network layer (IP) และ Transport layer (TCP/UDP) เท่านั้นจะไม่ใช้ข้อมูลในระดับอื่นๆเช่น Application layer

ในการสกัดข้องมูล Network layer และ Transport layer จาก Packet ทำได้ด้วย Python Library ชื่อว่า Scrapy โดยจะทำการอ่านข้อมูลดังตารางด้านล่างจากแต่ละ Packet โดยข้อมูลแต่ละ Packet จะเป็น 1 แถวใน CSV ไฟล์ ทำให้ได้ผลลัพธ์เป็นไฟล์ CSV ของ PCAP ไฟล์ที่มีความยาว 10 นาที ซึ่งภายในประกอบด้วยอุกรณ์ต่างๆอยู่ด้วยกัน

ตาราง 1 คุณสมบัติของข้อมูลที่เลือกใช้

ลำดับ	Field name	Description
1	Source IP address	หมายเลข IP Address ของผู้ส่งข้อมูล
2	Destination IP Address	หมายเลข IP Address ของผู้รับข้อมูล
3	Source Port	หมายเลข Port ของต้นทางผู้ส่ง
4	Destination Port	หมายเลข Port ของต้นทางผู้รับ
5	IPDI	หมายเลข IPID ของ packet
6	IP packet Length	ความยาวของ IP packet
7	Proto	หมายเลข Protocol
8	TCP seq	หมายเลข TCP sequence number
9	Source Mac Address	หมายเลข MAC ต้นทาง

3.1.3 การแยกไฟล์ CSV 10 นาที เป็นไฟล์ย่อยของแต่ละอุปกรณ์

เนื่องจากต้องการข้อมูลของแต่ละอุปกรณ์เพื่อใช้ในการสอน ML model ในขั้นตอนนี้จึงทำการแยกแต่ข้อมูลของแต่ละอุปกรณ์จาก CSV 10 นาที ให้เป็น ไฟล์ CSV ของแต่ละอุปกรณ์ ด้วยการใช้ข้อมูล MAC Address ที่ Dataset เตรียมไว้ให้ ดังแสดงในตารางด้านล่าง ส่งผลให้ได้ไฟล์ข้อมูลของแต่ละอุปกรณ์ในช่วงเวลาแต่ละ 10 นาที

ตาราง 2 รายชื่ออุปกรณ์และ MAC Address

ลำดับ	รายชื่ออุปกรณ์	MAC ADDRESS
1	Smart Things	d0:52:a8:00:67:5e
2	Amazon Echo	44:65:0d:56:cc:d3
3	Netatmo Welcome	70:ee:50:18:34:43
4	TP-Link Day Night Cloud camera	f4:f2:6d:93:51:f1
5	Samsung SmartCam	00:16:6c:ab:6b:88
6	Dropcam	30:8c:fb:2f:e4:b2
7	Insteon Camera	00:62:6e:51:27:2e
8	Insteon Camera	e8:ab:fa:19:de:4f
9	Withings Smart Baby Monitor	00:24:e4:11:18:a8
10	Belkin Wemo switch	ec:1a:59:79:f4:89
11	TP-Link Smart plug	50:c7:bf:00:56:39
12	iHome	74:c6:3b:29:d7:1d
13	Belkin wemo motion sensor	ec:1a:59:83:28:11
14	NEST Protect smoke alarm	18:b4:30:25:be:e4
15	Netatmo weather station	70:ee:50:03:b8:ac
16	Withings Smart scale	00:24:e4:1b:6f:96
17	Blipcare Blood Pressure meter	74:6a:89:00:2e:25
18	Withings Aura smart sleep sensor	00:24:e4:20:28:c6
19	Light Bulbs LiFX Smart Bulb	d0:73:d5:01:83:08
20	Triby Speaker	18:b7:9e:02:20:44
21	PIX-STAR Photo-frame	e0:76:d0:33:bb:85
22	HP Printer	70:5a:0f:e4:9b:c0
23	Samsung Galaxy Tab	08:21:ef:3b:fc:e3
24	Nest Dropcam	30:8c:fb:b6:ea:45
25	Android Phone	40:f3:08:ff:1e:da
26	Laptop	74:2f:68:81:69:42
27	MacBook	ac:bc:32:d4:6f:2f
28	Android Phone	b4:ce:f6:a7:a3:c2
29	IPhone	d0:a6:37:df:a1:e1
30	MacBook/Iphone	f4:5c:89:93:cc:85
31	TPLink Router Bridge LAN (Gateway)	14:cc:20:51:33:ea

3.1.4 **รวมข้อมูลในทุก 10 นาทีสร้างเป็น 1 sample ของ dataset**

เมื่อเสร็จกระบวนการแยกไฟล์แต่ละอุปกรณ์แล้วจะนำไฟล์ที่ได้ทั้งไฟล์นั้นมาคำนวณค่าทางสถิติของข้อมูลในแต่ละไฟล์ออกมาเป็น 1 sample ในชุดข้อมูลสำหรับการสอน ML Model โดยข้อมูลที่ได้แสดงดังตารางด้านล่าง

ตาราง 3 รายชื่อคุณสมบัติของข้อมูลที่จัดเก็บเป็น 1 sample

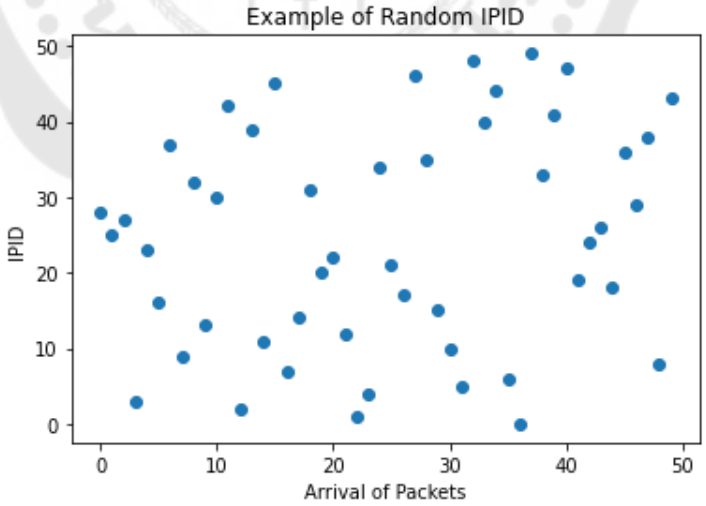
	Field name	**รายละเอียดของคุณสมบัติ**
1	Number of Flow	เป็นจำนวนครั้งที่ Source IP, Destination IP, Source Port, Destination Port และหมายเลข Protocol ของ packet (Proto) มีค่าแตกต่างกัน
2	Number of Packets	เป็นจำนวน packet ที่เกิดขึ้นทั้งหมดภายในช่วงเวลา 10 นาทีที่ผ่านมา
3	Avg Packet Size	ค่าเฉลี่ยของขนาดของ Packets ในช่วงเวลาทุก 10 นาที
4	Max Packet Size	ค่าที่สูงที่สุดของขนาดของ Packets ในช่วงเวลาทุก 10 นาที
5	Min Packet Size	ค่าที่ต่ำที่สุดของขนาดของ Packets ในช่วงเวลาทุก 10 นาที
6	SD Packet size	ค่าที่มัธยฐานของขนาดของ Packets ในช่วงเวลาทุก 10 นาที
7	IPID_DIFF_negative_ratio	ค่าความแตกต่างระหว่างบรรทัดที่เป็นหมายลบที่ลดลงของ IPID
8	Label	ชนิดของอุปกรณ์

ในการคำนวณค่าทางสถิติต่างๆนั้นทำด้วย Library Numpy ในภาษา Python สำหรับค่าที่ต้องให้ความสนใจเป็นพิเศษคือค่า IPID_DIFF_negative_ratio เป็นค่าที่นักวิจัยคิดขึ้นมาเพื่อแสดงลักษณะของการเปลี่ยนแปลงของ IPID ดังที่แสดงในบทที่ 2 ซึ่ง IPID มีลักษณะการทำหนดค่าอยู่ 3 แบบคือ Sequence, Incremental , และ Random โดยค่า IPID_DIFF_negative_ratio จะใช้ในการบอกถึงความแตกต่างของรูปแบบการกำหนด IPID ของอุปกรณ์แต่ละชนิด โดยการคำนวณผลต่าง(Diff)แบบไม่ต่อเนื่องที่ n ของข้อมูล IPID ด้วยคำสั่ง numpy.diff() จากนั้นหากสัดส่วนของค่า Diff ที่เป็นลบต่อค่า Diff ทั้งหมด โดยมีตัวอย่างของฟังก์ชันแสดงดังด้านล่าง

```
def find_negative_ipid_ratio(x):
    print("IPID: ")
    print(list(x))
    IPID_DIFF_array = np.diff(x)
    print("IPID Diff: ")
    print(list(IPID_DIFF_array))
    IPID_DIFF_array = IPID_DIFF_array[(IPID_DIFF_array < 0)]
    print("Negative Diff value: ")
    print(list(IPID_DIFF_array))
    IPID_DIFF_negative_ratio = len(IPID_DIFF_array) / len(x)
    print("Ratio of Negative Diff Value")
    print(IPID_DIFF_negative_ratio)
```

ตัวอย่างการหาค่า IPID_DIFF_negative_ratio

กรณีเป็น Random IPID แสดงดังรูปที่ 13 และมีค่าต่างๆดังนี้



ภาพประกอบ 13 ตัวอย่างการจำลองค่า IPID แบบสุ่ม

IPID:

[28, 25, 27, 3, 23, 16, 37, 9, 32, 13, 30, 42, 2, 39, 11, 45, 7, 14, 31, 20, 22, 12, 1, 4, 34, 21, 17, 46, 35, 15, 10, 5, 48, 40, 44, 6, 0, 49, 33, 41, 47, 19, 24, 26, 18, 36, 29, 38, 8, 43]

ค่า IPID Diff:

[-3, 2, -24, 20, -7, 21, -28, 23, -19, 17, 12, -40, 37, -28, 34, -38, 7, 17, -11, 2, -10, -11, 3, 30, -13, -4, 29, -11, -20, -5, -5, 43, -8, 4, -38, -6, 49, -16, 8, 6, -28, 5, 2, -8, 18, -7, 9, -30, 35]

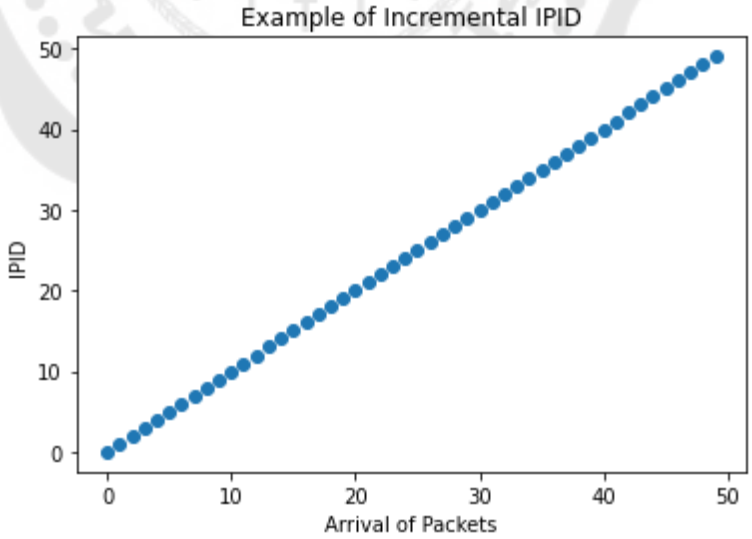
Negative Diff value:

[-3, -24, -7, -28, -19, -40, -28, -38, -11, -10, -11, -13, -4, -11, -20, -5, -5, -8, -38, -6, -16, -28, -8, -7, -30]

Ratio of Negative Diff Value:

0.5

กรณีเป็น Incremental IPID แสดงดังรูปที่ 14 และมีค่าต่างๆดังนี้



ภาพประกอบ 14 ตัวอย่างการเรียงกันของค่า IPID ในลักษณะ Incremental

IPID:

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49]

IPID Diff:

[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]

Negative Diff value:

[]

Ratio of Negative Diff Value

0.0

จากทั้งสองจะเห็นได้ว่าในการณี Random IPID จะมีค่า Ratio of Negative Diff ที่เป็นค่าลบ ซึ่งแสดงถึงมีการเปลี่ยนแปลงของค่า IPID ที่เปลี่ยนไปแบบสุ่ม แต่ในกรณีของ Incremental IPID จะพบว่ามีค่า Ratio of Negative Diff ที่เป็น 0 เนื่องจากข้อมูล IPID มีการเพิ่มขึ้นทีละหนึ่งทำให้มีความแตกต่างระหว่างคู่ของ IPID ที่ติดกันเป็นค่าบวกเสมอทำให้จำนวนของค่า Diff ที่เป็นลบมีค่าเป็น 0 เมื่อนำมาหาค่า Ratio of Negative Diff แล้วจึงได้ค่าเป็น 0 ซึ่งค่า Ratio of Negative Diff นี้จะถูกใช้ในการบอกลักษณะของ IPID ของแต่ละอุปกรณ์

3.2 **การสำรวจและการคัดเลือกข้อมูล** dataset

หลังจากการเตรียมข้อมูลซึ่งข้อมูล 1 record ได้จากข้อมูลเครือข่าย 10 นาทีของแต่ละอุปกรณ์มีข้อมูลจำนวนทั้งหมด 41,708 records โดยแบ่งตามอุปกรณ์ได้ดังตารางด้านล่าง จะเห็นได้ว่าบางอุปกรณ์นั้นมีจำนวนข้อมูลน้อยมากและไม่เหมาะสมสำหรับการใช้เพื่อสอน ML model

ตาราง 4 รายการ sample ของแต่ละอุปกรณ์

ชื่ออุปกรณ์	จำนวน record
Smart_Things	2817
HP_Printer	2817

ชื่ออุปกรณ์	จำนวน record
Amazon_Echo	2817
Samsung_SmartCam	2817
Dropcam	2817
Belkin_Wemo_switch	2817
Belkin_wemo_motion_sensor	2817
Triby_Speaker	2799
Netatmo_weather_station	2227
Withings_Smart_Baby_Monitor	2101
TP_Link_Smart_plug	2100
Samsung_Galaxy_Tab	2056
Withings_Aura_smart_sleep_sensor	2035
Light_Bulbs_LiFX_Smart_Bulb	2032
Insteon_Camera	2030
TP_Link_Day_Night_Cloud_camera	1815
PIX_STAR_Photo_frame	1162
iHome	1044
TPLink_Router_Bridge_LAN_Gateway	248
Laptop	155
Android_Phone_2	49
Withings_Smart_scale	37
MacBook	27
Android_Phone_1	27
NEST_Protect_smoke_alarm	22
iPhone	10
Nest_Dropcam	5
Blipcare_Blood_Pressure_meter	4
MacBook_Iphone	3
Insteon_Camera_2	1

ด้วยจำนวนข้อมูลของอุปกรณ์บางตัวที่น้อย ผู้วิจัยจึงเลือกเอาเฉพาะอุปกรณ์ที่มีจำนวน record มากกว่า 2,000 recordsและเป็นอุปกรณ์ IoT เท่านั้นเท่านั้น เพื่อให้ข้อมูลที่ได้รับการเรียนรู้นั้นมีความแม่นยำ ซึ่งเหลือเพียง 13 อุปกรณ์สรุปได้ดังตารางด้านล่าง

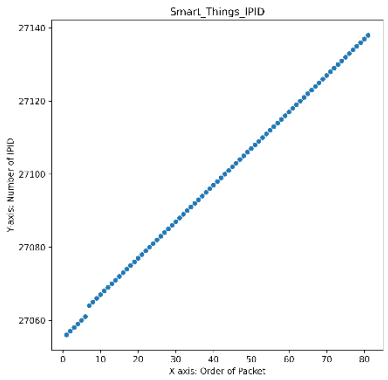
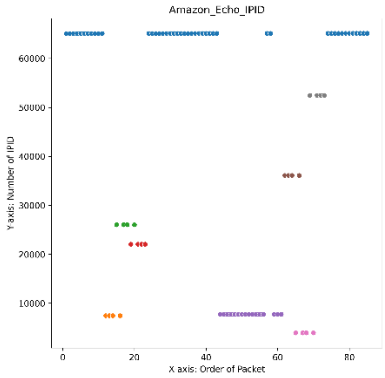
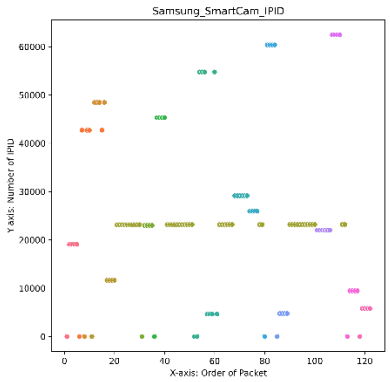
ตาราง 5 รายการ sample ของแต่ละอุปกรณ์ที่คงเหลือหลังทำความสะอาดข้อมูล

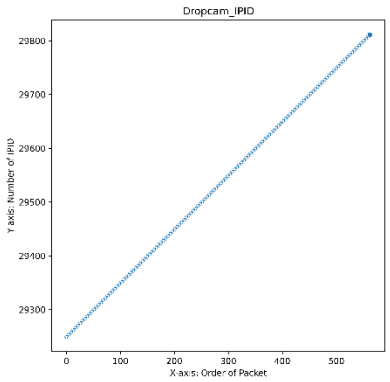
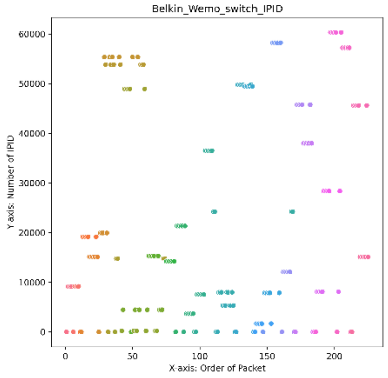
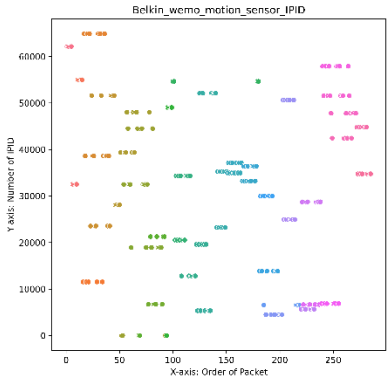
ชื่ออุปกรณ์	จำนวน record
Smart_Things	2817
Amazon_Echo	2817
Samsung_SmartCam	2817
Dropcam	2817
Belkin_Wemo_switch	2817
Belkin_wemo_motion_sensor	2817
Triby_Speaker	2799
Netatmo_weather_station	2227
Withings_Smart_Baby_Monitor	2101
TP_Link_Smart_plug	2100
Withings_Aura_smart_sleep_sensor	2035
Light_Bulbs_LiFX_Smart_Bulb	2032
Insteon_Camera	2030

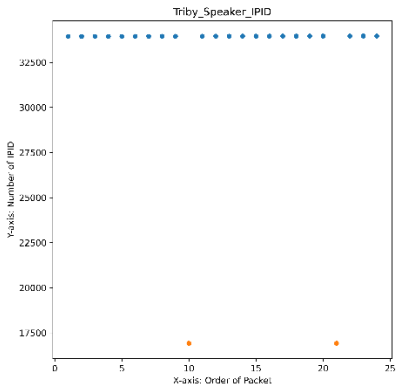
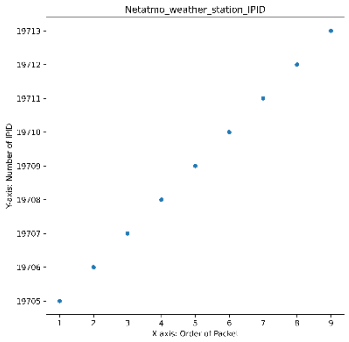
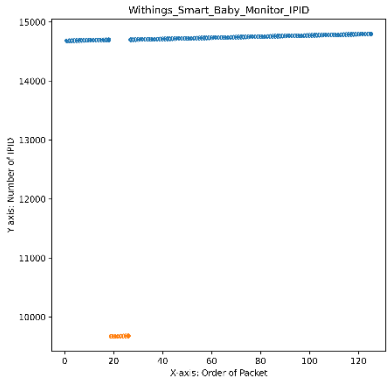
3.3 การศึกษาลักษณะของ IPID ของอุปกรณ์ที่มีจำนวน sample เกิน 2000 Records

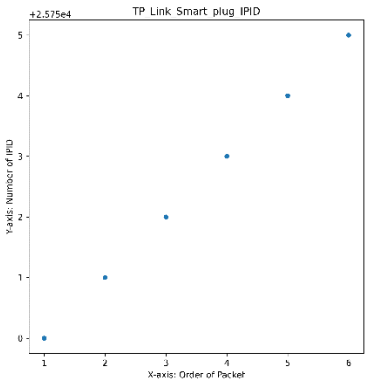
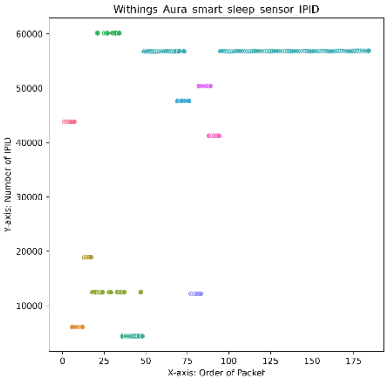
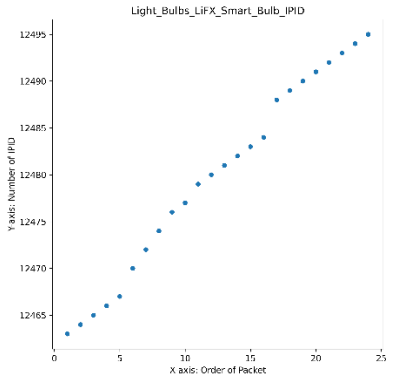
เพื่อให้เห็นลักษณะของ IPID ของอุปกรณ์ต่างๆ ผู้วิจัยได้ทำกราฟ IPID ของอุปกรณ์ดังแสดงในตารางด้านล่าง

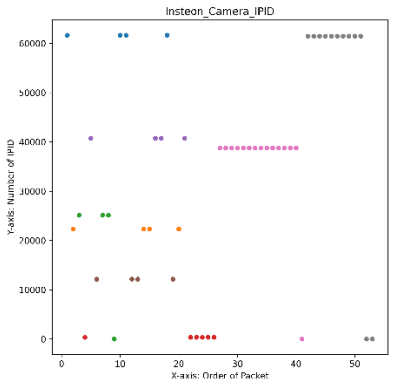
ตาราง 6 ตัวอย่างลักษณะ IPID ของอุปกรณ์

อุปกรณ์	IPID Behaviour	Description
Smart_Things		Incremental IPID
Amazon_Echo		Sequential IPID
Samsung_SmartCam		Sequential IPID

อุปกรณ์	IPID Behaviour	Description
Dropcam		Incrementa l IPID
Belkin_Wemo_switch		Sequential IPID
Belkin_wemo_motion_sensor		Sequential IPID

อุปกรณ์	IPID Behaviour	Description
Triby_Speaker		Incrementa l IPID
Netatmo_weather_station		Incrementa l IPID
Withings_Smart_Baby_Monitor		Sequential IPID

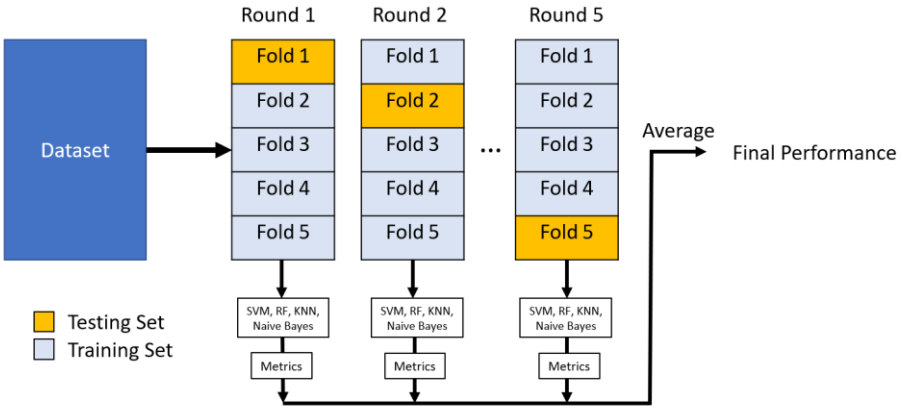
อุปกรณ์	IPID Behaviour	Description
TP_Link_Smart_plug		Incrementa l IPID
Withings_Aura_smart_sleep_senso r		Sequential IPID
Light_Bulbs_LiFX_Smart_Bulb		Incrementa l IPID

อุปกรณ์	IPID Behaviour	Description
Insteon_Camera		Sequential IPID

3.4 การทดลอง

ในการวิจัยนี้ทำการศึกษาความสามารถในการแยกแยะอุปกรณ์ IoT ชนิดต่างๆด้วย ML model ต่างๆประกอบด้วย Random Forest, Support Vector Machines (SVM), Naïve Bayesian,และ K-Nearest Neighbors โดยใช้ข้อมูลทางเครือข่าย 7 ชนิดที่สนใจประกอบด้วย Number of Flow, Number of Packets, Avg Packet Size, Max Packet Size, Min Packet Size, SD Packet size, และ IPID_DIFF_negative_ratio ซึ่งนำมากจากข้อมูลระดับ Network Layer และ Transport Layer เท่านั้น โดยข้อมูล IPID_DIFF_negative_ratio นั้นเป็นข้อมูลที่ผู้วิจัยสร้างขึ้นมาเพื่อใช้ในการบอกลักษณะของ IPID

ดังนั้นในการทดลองนี้จะแบ่งการทดลองออกเป็น 2 แบบคือการสร้าง ML Model ทั้ง 4 แบบด้วยชุดข้อมูล ที่ประกอบด้วย IPID_DIFF_negative_ratio และไม่รวม IPID_DIFF_negative_ratio เพื่อเปรียบเทียบประสิทธิภาพของ Model ระหว่างการใช้ Feature ที่เราสร้างขึ้นมากับไม่ใช้ ในการทดลองนี้ทำการทดลองโดยทำ 5 Fold Cross Validation จากนั้นจึงหาค่าเฉลี่ยของตัวชี้วัด โดยตัวชี้วัดในงานประกอบด้วย Accuracy, Precision, Recall, F-1 score ดังแสดงในรูปที่ 15 ด้านล่าง



ภาพประกอบ 15 การทำ Cross Validation แบบ 5 fold

ในการทดลองการสร้างแบบจำลองการทำนาย ผู้วิจัยได้แบ่งข้องมูลเพื่อการทำ Stratified K-Fold cross validation ออกเป็น 5 ส่วนเพื่อทดสอบความแม่นยำที่แน่นอนในการสร้างแบบจำลองการทำนายกับชุดข้อมูลเพื่อการทดสอบ แต่เนื่องจากข้อมูลที่ได้รับมานั้นเป็นข้อมูลที่มีจำนวนการบันทึกที่ไม่เท่ากันจึงจำเป็นต้องใช้งาน Stratified K Fold Cross Validation เพื่อแก้ไขปัญหาที่เกิดขึ้นโดยไม่ตัดทอนข้อมูลเดิมให้มีปริมาณที่เท่ากัน และโมเดลต่างๆที่ใช้นั้นนำมาจาก sklearn library และเลือกใช้ Parameter ต่างๆเป็นแบบค่า default ทั้งหมด

```
from sklearn.model_selection import StratifiedKFold
skf = StratifiedKFold(n_splits=5, shuffle=True, random_state=20)
```

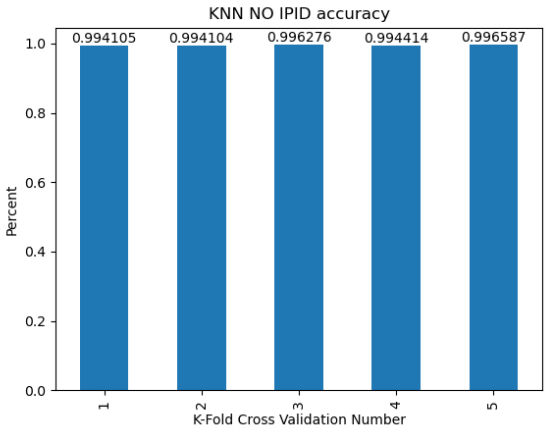
ภาพประกอบ 16 ตัวอย่างการเรียกใช้งาน Sklearn Library

บทที่ 4
ผลการทดลอง

4.1 ผลการทดลองประสิทธิภาพการสร้างแบบจำลองการทำนายโดยปารศจากการใช้งานคุณสมบัติ IPID Negative Ratio

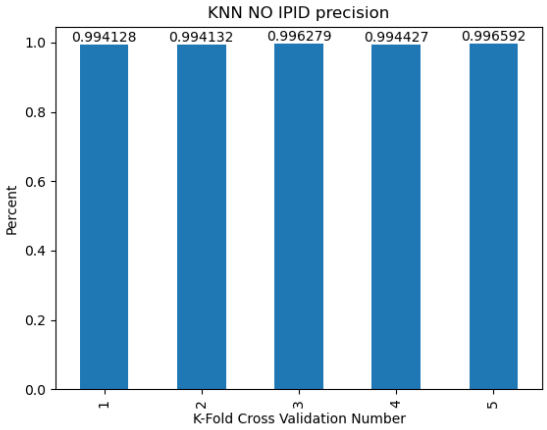
4.1.1 K-nearest Neighbors

ในการทดลองการสร้างแบบจำลองการทำนายนั้นได้ผลการทดลองค่า accuracy แบบ Stratified K Fold Cross Validation ทั้ง 5 ครั้งโดยมีผลลัพธ์คือ 0.994 0.994 0.996 0.994 และ 0.997 ตามลำดับ fold ดังแสดงในรูปที่ 17



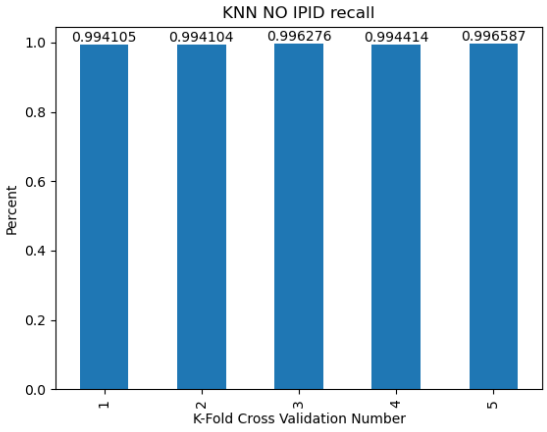
ภาพประกอบ 17กราฟแสดงค่า accuracy ของการใช้งานค่า IPID ในการไม่ใช้งาน K-Nearest Neighbors

ในการทดลองการสร้างแบบจำลองการทำนายนั้นได้ผลการทดลองค่า precision แบบ Stratified K Fold Cross Validation ทั้ง 5 ครั้งโดยมีผลลัพธ์คือ 0.994 0.994 0.996 0.994 และ 0.997 ตามลำดับ fold ดังแสดงในรูปที่ 18



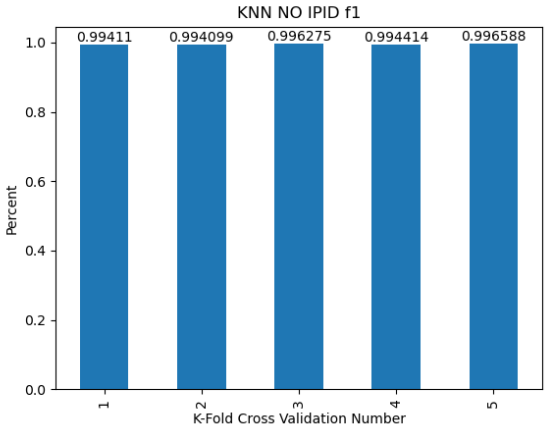
ภาพประกอบ 18 กราฟแสดงค่า precision ของการใช้งานค่า IPID ในการไม่ใช้งาน K-Nearest Neighbors

ในการทดลองการสร้างแบบจำลองการทำนายนั้นได้ผลการทดลองค่า recall แบบ Stratified K Fold Cross Validation ทั้ง 5 ครั้งโดยมีผลลัพธ์คือ 0.994 0.994 0.996 0.994 และ 0.997 ตามลำดับ fold ดังแสดงในรูปที่ 19



ภาพประกอบ 19 กราฟแสดงค่า recall ของการใช้งานค่า IPID ในการไม่ใช้งาน K-Nearest Neighbors

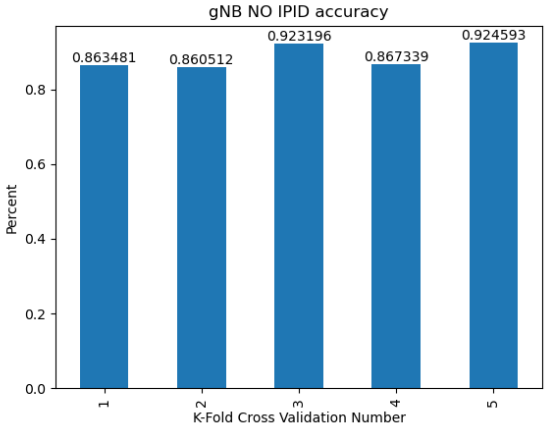
ในการทดลองการสร้างแบบจำลองการทำนายนั้นได้ผลการทดลองค่า f-1 แบบ Stratified K Fold Cross Validation ทั้ง 5 ครั้งโดยมีผลลัพธ์คือ 0.994 0.994 0.996 0.994 และ 0.997 ตามลำดับ fold ดังแสดงในรูปที่ 20



ภาพประกอบ 20 กราฟแสดงค่า f-1 ของการใช้งานค่า IPID ในการไม่ใช้งาน K-Nearest Neighbors

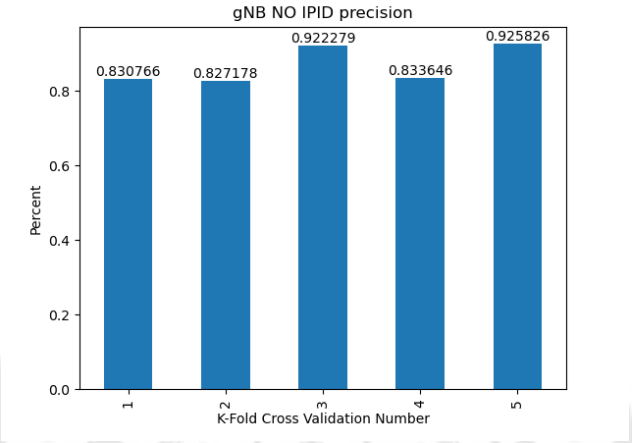
4.1.2 Naïve Bayesian

ในการทดลองการสร้างแบบจำลองการทำนายนั้นได้ผลการทดลองค่า accuracy แบบ Stratified K Fold Cross Validation ทั้ง 5 ครั้งโดยมีผลลัพธ์คือ 0.863 0.861 0.923 0.867 และ 0.925 ตามลำดับ fold ดังแสดงในรูปที่ 21



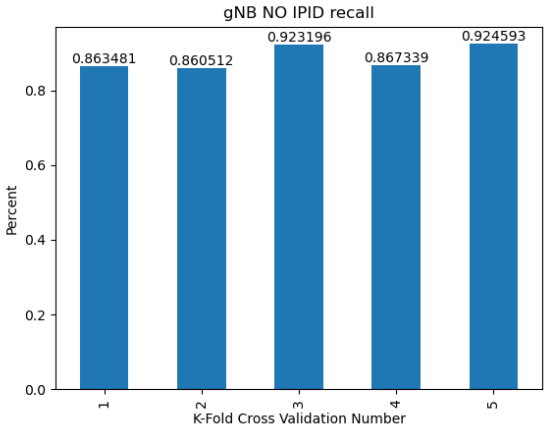
ภาพประกอบ 21 กราฟแสดงค่า accuracy ของการใช้งานค่า IPID ในการไม่ใช้งาน Gaussian Naive Bayes

ในการทดลองการสร้างแบบจำลองการทำนายนั้นได้ผลการทดลองค่า precision แบบ Stratified K Fold Cross Validation ทั้ง 5 ครั้งโดยมีผลลัพธ์คือ 0.83 0.827 0.922 0.834 และ 0.926 ตามลำดับ fold ดังแสดงในรูปที่ 22



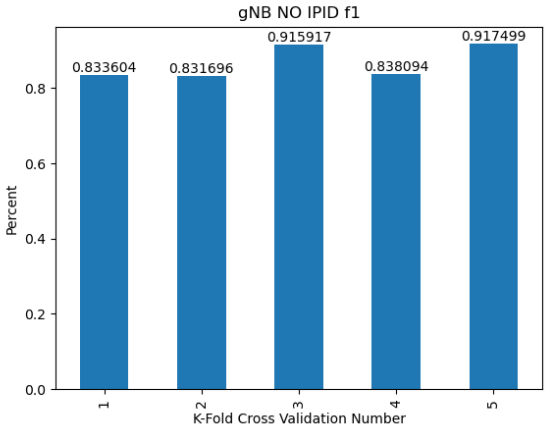
ภาพประกอบ 22 กราฟแสดงค่า precision ของการใช้งานค่า IPID ในการไม่ใช้งาน Gaussian Naive Bayes

ในการทดลองการสร้างแบบจำลองการทำนายนั้นได้ผลการทดลองค่า recall แบบ Stratified K Fold Cross Validation ทั้ง 5 ครั้งโดยมีผลลัพธ์คือ 0.863 0.861 0.923 0.867 และ 0.925 ตามลำดับ fold ดังแสดงในรูปที่ 23



ภาพประกอบ 23 กราฟแสดงค่า recall ของการใช้งานค่า IPID ในการไม่ใช้งาน Gaussian Naive Bayes

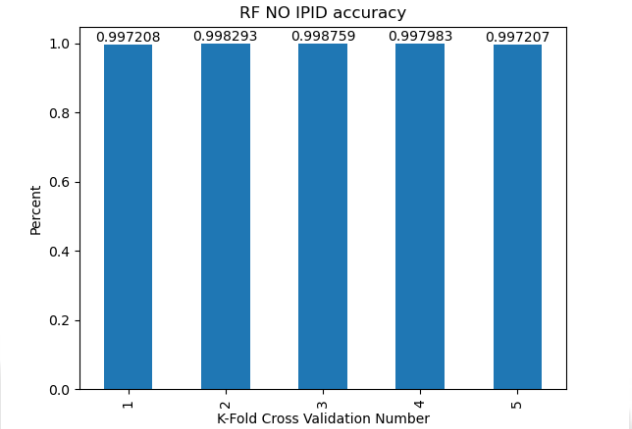
ในการทดลองการสร้างแบบจำลองการทำนายนั้นได้ผลการทดลองค่า f-1 แบบ Stratified K Fold Cross Validation ทั้ง 5 ครั้งโดยมีผลลัพธ์คือ 0.833 0.832 0.916 0.838 และ 0.918 ตามลำดับ fold ดังแสดงในรูปที่ 24



ภาพประกอบ 24 กราฟแสดงค่า f-1 ของการใช้งานค่า IPID ในการไม่ใช้งาน Gaussian Naive Bayes

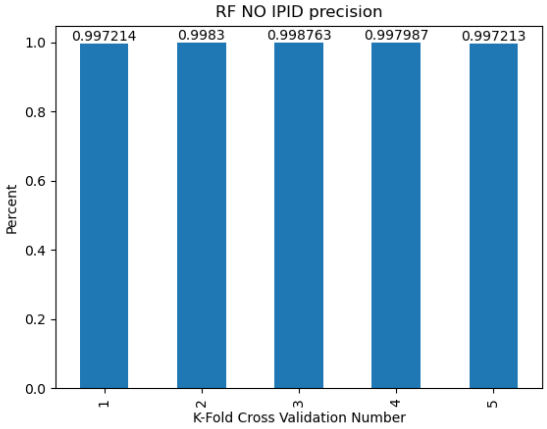
4.1.3 Random Forest

ในการทดลองการสร้างแบบจำลองการทำนายนั้นได้ผลการทดลองค่า accuracy แบบ Stratified K Fold Cross Validation ทั้ง 5 ครั้งโดยมีผลลัพธ์คือ 0.997 0.998 0.999 0.998 และ 0.997 ตามลำดับ fold ดังแสดงในรูปที่ 25



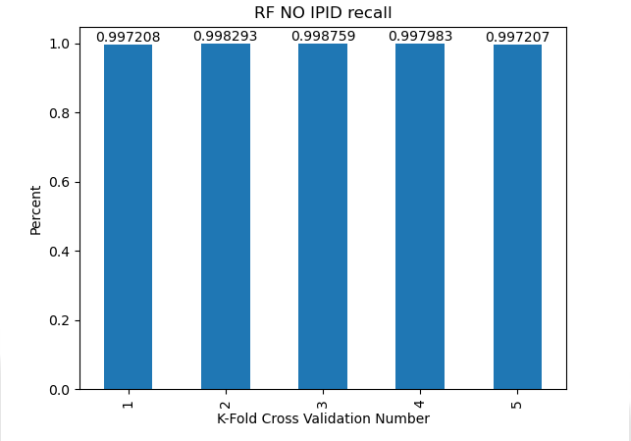
ภาพประกอบ 25 กราฟแสดงค่า accuracy ของการใช้งานค่า IPID ในการไม่ใช้งาน Random forest

ในการทดลองการสร้างแบบจำลองการทำนายนั้นได้ผลการทดลองค่า precision แบบ Stratified K Fold Cross Validation ทั้ง 5 ครั้งโดยมีผลลัพธ์คือ 0.997 0.998 0.999 0.998 และ 0.997 ตามลำดับ fold ดังแสดงในรูปที่ 26



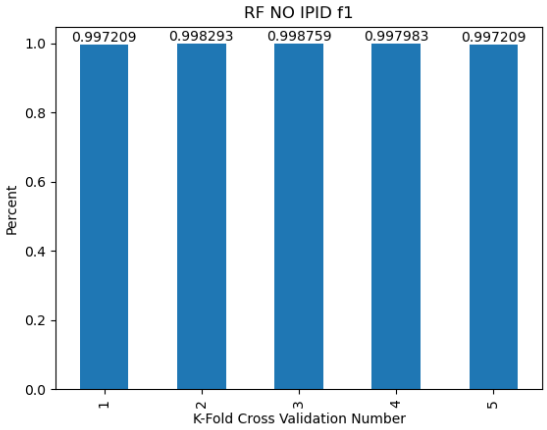
ภาพประกอบ 26 กราฟแสดงค่า precision ของการใช้งานค่า IPID ในการไม่ใช้งาน Random forest

ในการทดลองการสร้างแบบจำลองการทำนายนั้นได้ผลการทดลองค่า recall แบบ Stratified K Fold Cross Validation ทั้ง 5 ครั้งโดยมีผลลัพธ์คือ 0.997 0.998 0.999 0.998 และ 0.997 ตามลำดับ fold ดังแสดงในรูปที่ 27



ภาพประกอบ 27 กราฟแสดงค่า recall ของการใช้งานค่า IPID ในการไม่ใช้งาน Random forest

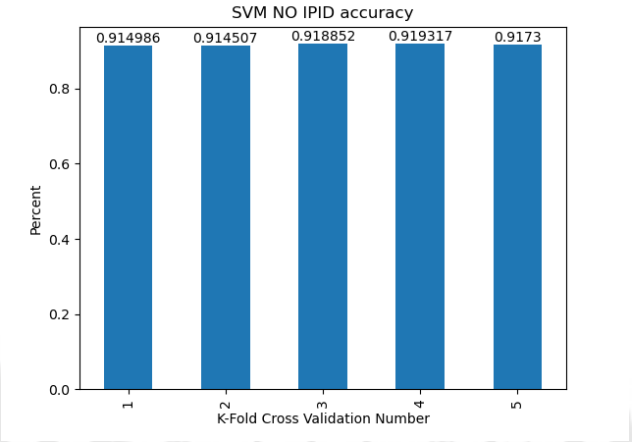
ในการทดลองการสร้างแบบจำลองการทำนายนั้นได้ผลการทดลองค่า f-1 แบบ Stratified K Fold Cross Validation ทั้ง 5 ครั้งโดยมีผลลัพธ์คือ 0.997 0.998 0.999 0.998 และ 0.997 ตามลำดับ fold ดังแสดงในรูปที่ 28



ภาพประกอบ 28 กราฟแสดงค่า f-1 ของการใช้งานค่า IPID ในการไม่ใช้งาน Random forest

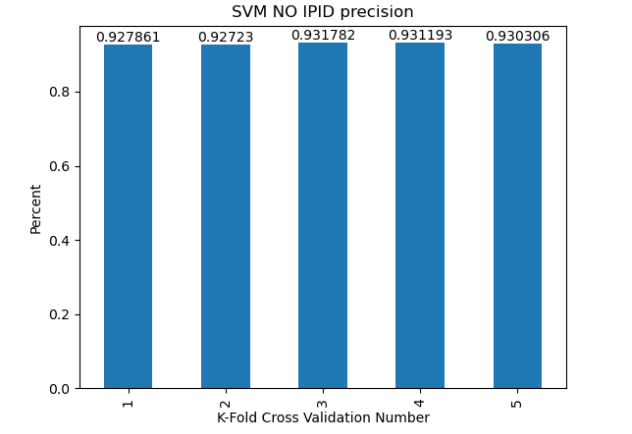
4.1.4 Support Vector Machine

ในการทดลองการสร้างแบบจำลองการทำนายนั้นได้ผลการทดลองค่า accuracy แบบ Stratified K Fold Cross Validation ทั้ง 5 ครั้งโดยมีผลลัพธ์คือ 0.915 0.915 0.919 0.919 และ 0.917 ตามลำดับ fold ดังแสดงในรูปที่ 29



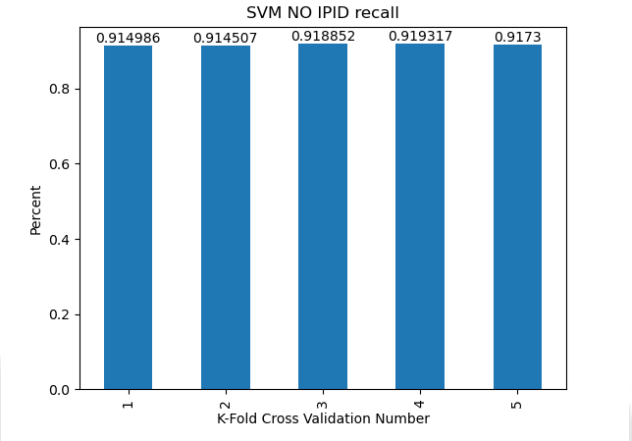
ภาพประกอบ 29 กราฟแสดงค่า accuracy ของการใช้งานค่า IPID ในการไม่ใช้งาน Support Vector Machines

ในการทดลองการสร้างแบบจำลองการทำนายนั้นได้ผลการทดลองค่า precision แบบ Stratified K Fold Cross Validation ทั้ง 5 ครั้งโดยมีผลลัพธ์คือ 0.928 0.928 0.932 0.932 และ 0.93 ตามลำดับ fold ดังแสดงในรูปที่ 30



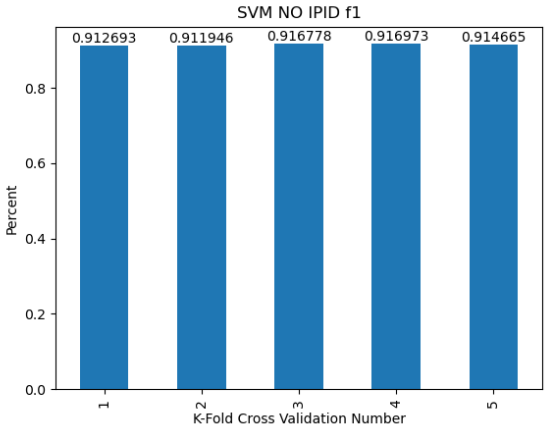
ภาพประกอบ 30กราฟแสดงค่า precision ของการใช้งานค่า IPID ในการไม่ใช้งาน Support Vector Machines

ในการทดลองการสร้างแบบจำลองการทำนายนั้นได้ผลการทดลองค่า recall แบบ Stratified K Fold Cross Validation ทั้ง 5 ครั้งโดยมีผลลัพธ์คือ 0.914 0.914 0.919 0.919 และ 0.917 ตามลำดับ fold ดังแสดงในรูปที่ 31



ภาพประกอบ 31 กราฟแสดงค่า recall ของการใช้งานค่า IPID ในการไม่ใช้งาน Support Vector Machines

ในการทดลองการสร้างแบบจำลองการทำนายนั้นได้ผลการทดลองค่า f-1 แบบ Stratified K Fold Cross Validation ทั้ง 5 ครั้งโดยมีผลลัพธ์คือ 0.913 0.912 0.917 0.917 และ 0.915 ตามลำดับ fold ดังแสดงในรูปที่ 32

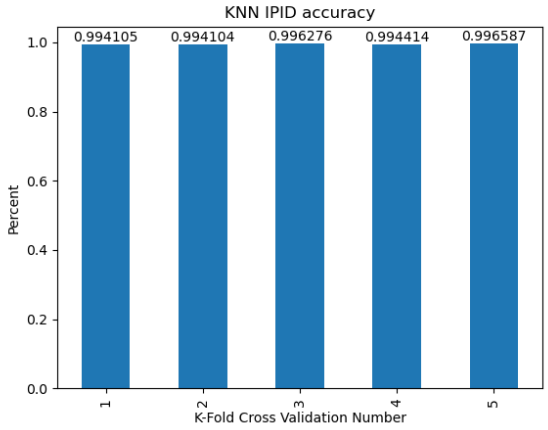


ภาพประกอบ 32 กราฟแสดงค่า f-1 ของการใช้งานค่า IPID ในการไม่ใช้งาน Support Vector Machines

4.2 การทดลองประสิทธิภาพการสร้างแบบจำลองการทำนายโดยใช้งานคุณสมบัติ IPID Negative Ratio

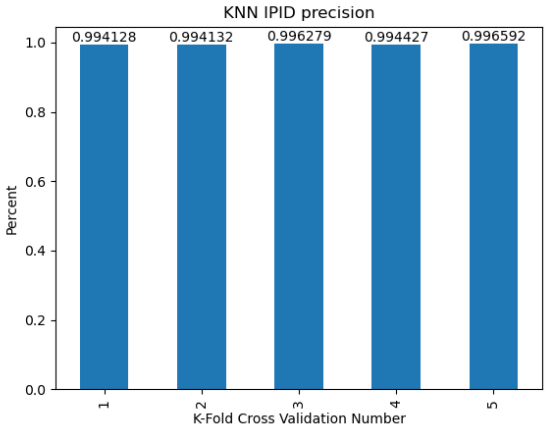
4.2.1 K-nearest Neighbors

ในการทดลองการสร้างแบบจำลองการทำนายนั้นได้ผลการทดลองค่า accuracy แบบ Stratified K Fold Cross Validation ทั้ง 5 ครั้งโดยมีผลลัพธ์คือ 0.994 0.994 0.996 0.994 และ 0.997 ตามลำดับ fold ดังแสดงในรูปที่ 33



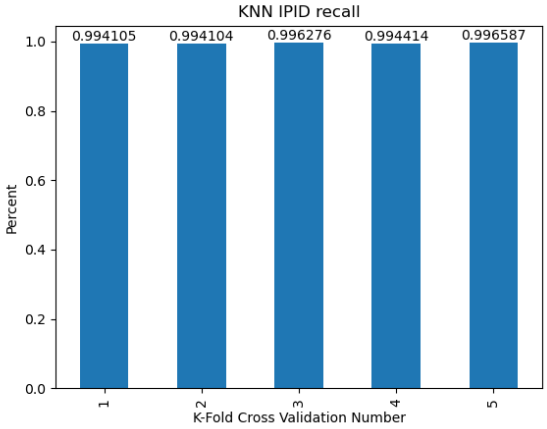
ภาพประกอบ 33 กราฟแสดงค่า accuracy ของการไม่ใช้งานค่า IPID ในการใช้งาน K-Nearest Neighbors

ในการทดลองการสร้างแบบจำลองการทำนายนั้นได้ผลการทดลองค่า precision แบบ Stratified K Fold Cross Validation ทั้ง 5 ครั้งโดยมีผลลัพธ์คือ 0.994 0.994 0.996 0.994 และ 0.997 ตามลำดับ fold ดังแสดงในรูปที่ 34



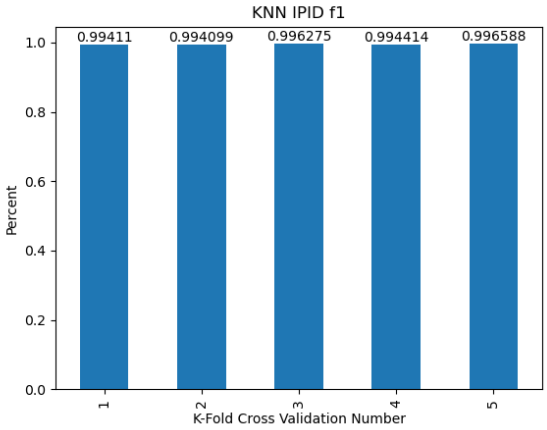
ภาพประกอบ 34 กราฟแสดงค่า precision ของการไม่ใช้งานค่า IPID ในการใช้งาน K-Nearest Neighbors

ในการทดลองการสร้างแบบจำลองการทำนายนั้นได้ผลการทดลองค่า recall แบบ Stratified K Fold Cross Validation ทั้ง 5 ครั้งโดยมีผลลัพธ์คือ 0.994 0.994 0.996 0.994 และ 0.997 ตามลำดับ fold ดังแสดงในรูปที่ 35



ภาพประกอบ 35 กราฟแสดงค่า recall ของการไม่ใช้งานค่า IPID ในการใช้งาน K-Nearest Neighbors

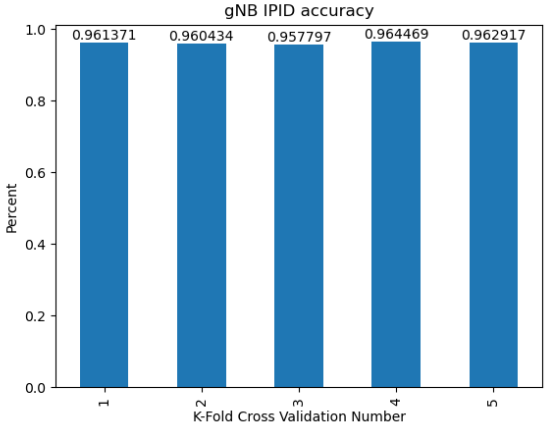
ในการทดลองการสร้างแบบจำลองการทำนายนั้นได้ผลการทดลองค่า f-1 แบบ Stratified K Fold Cross Validation ทั้ง 5 ครั้งโดยมีผลลัพธ์คือ 0.994 0.994 0.996 0.994 และ 0.997 ตามลำดับ fold ดังแสดงในรูปที่ 36



ภาพประกอบ 36 กราฟแสดงค่า f-1 ของการไม่ใช้งานค่า IPID ในการใช้งาน K-Nearest Neighbors

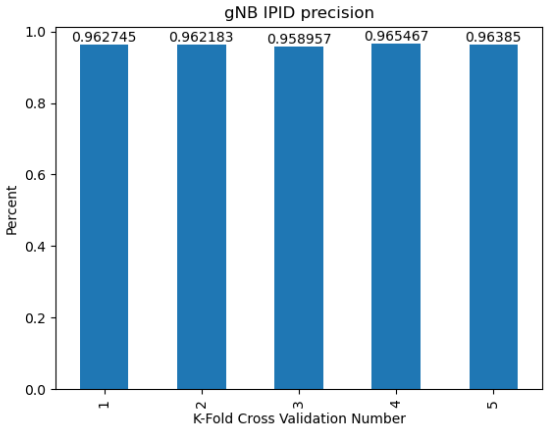
4.2.2 Naïve Bayesian

ในการทดลองการสร้างแบบจำลองการทำนายนั้นได้ผลการทดลองค่า accuracy แบบ Stratified K Fold Cross Validation ทั้ง 5 ครั้งโดยมีผลลัพธ์คือ 0.961 0.960 0.958 0.964 และ 0.963 ตามลำดับ fold ดังแสดงในรูปที่ 37



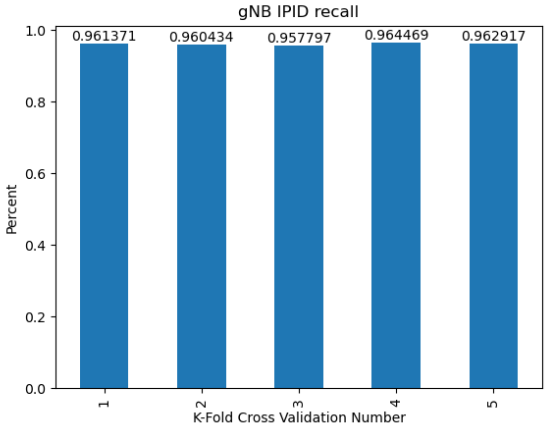
ภาพประกอบ 37 กราฟแสดงค่า accuracy ของการใช้งานค่า IPID ในการไม่ใช้งาน Gaussian Naive Bayes

ในการทดลองการสร้างแบบจำลองการทำนายนั้นได้ผลการทดลองค่า precision แบบ Stratified K Fold Cross Validation ทั้ง 5 ครั้งโดยมีผลลัพธ์คือ 0.963 0.962 0.959 0.964 และ 0.964 ตามลำดับ fold ดังแสดงในรูปที่ 38



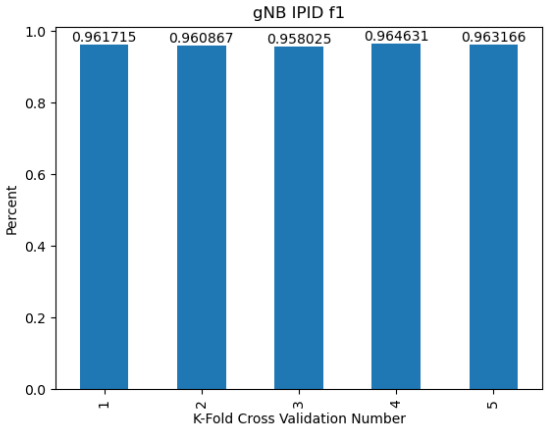
ภาพประกอบ 38 กราฟแสดงค่า precision ของการใช้งานค่า IPID ในการไม่ใช้งาน Gaussian Naive Bayes

ในการทดลองการสร้างแบบจำลองการทำนายนั้นได้ผลการทดลองค่า recall แบบ Stratified K Fold Cross Validation ทั้ง 5 ครั้งโดยมีผลลัพธ์คือ 0.961 0.960 0.958 0.961 และ 0.963 ตามลำดับ fold ดังแสดงในรูปที่ 39



ภาพประกอบ 39 กราฟแสดงค่า recall ของการใช้งานค่า IPID ในการไม่ใช้งาน Gaussian Naive Bayes

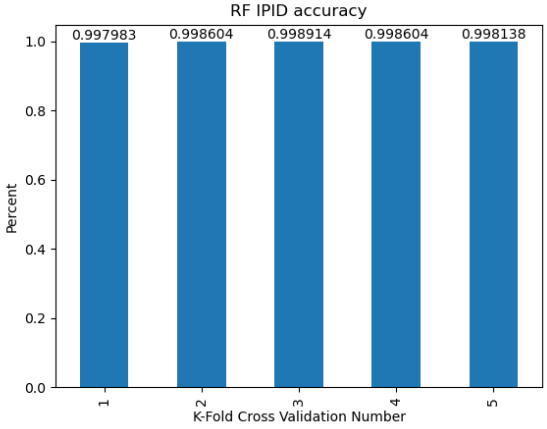
ในการทดลองการสร้างแบบจำลองการทำนายนั้นได้ผลการทดลองค่า f-1 แบบ Stratified K Fold Cross Validation ทั้ง 5 ครั้งโดยมีผลลัพธ์คือ 0.962 0.961 0.958 0.965 และ 0.963 ตามลำดับ fold ดังแสดงในรูปที่ 40



ภาพประกอบ 40 กราฟแสดงค่า f-1 ของการใช้งานค่า IPID ในการไม่ใช้งาน Gaussian Naive Bayes

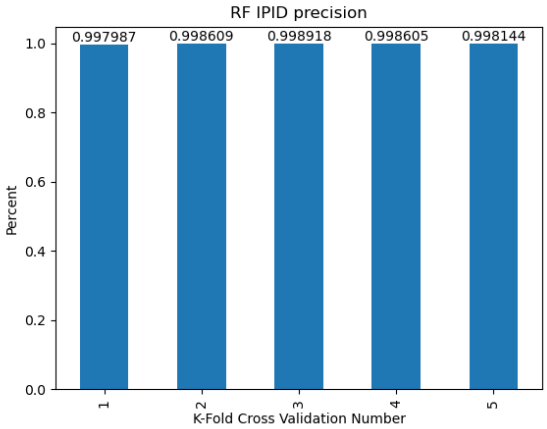
4.2.3 Random Forest

ในการทดลองการสร้างแบบจำลองการทำนายนั้นได้ผลการทดลองค่า accuracy แบบ Stratified K Fold Cross Validation ทั้ง 5 ครั้งโดยมีผลลัพธ์คือ 0.998 0.999 0.999 0.999 และ 0.998 ตามลำดับ fold ดังแสดงในรูปที่ 41



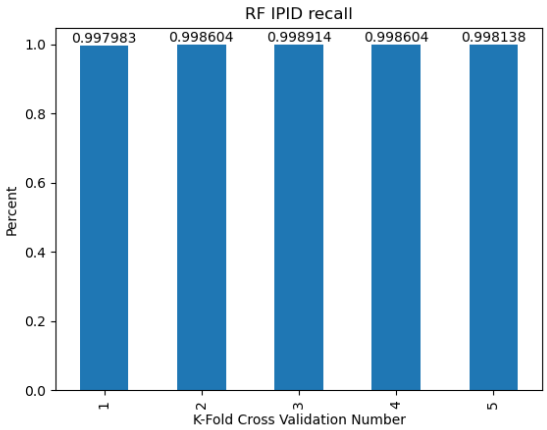
ภาพประกอบ 41 กราฟแสดงค่า accuracy ของการใช้งานค่า IPID ในการไม่ใช้งาน Random forest

ในการทดลองการสร้างแบบจำลองการทำนายนั้นได้ผลการทดลองค่า precision แบบ Stratified K Fold Cross Validation ทั้ง 5 ครั้งโดยมีผลลัพธ์คือ 0.998 0.999 0.999 0.999 และ 0.998 ตามลำดับ fold ดังแสดงในรูปที่ 42



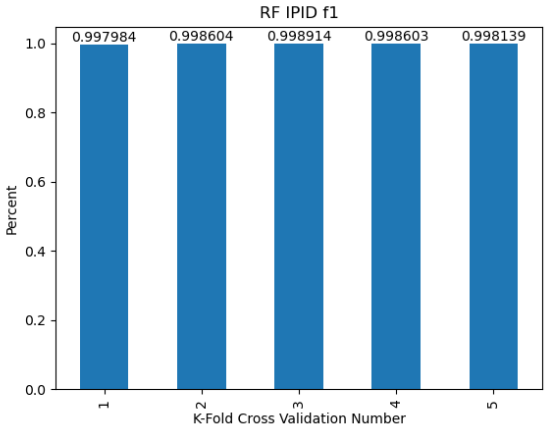
ภาพประกอบ 42 กราฟแสดงค่า precision ของการใช้งานค่า IPID ในการไม่ใช้งาน Random forest

ในการทดลองการสร้างแบบจำลองการทำนายนั้นได้ผลการทดลองค่า recall แบบ Stratified K Fold Cross Validation ทั้ง 5 ครั้งโดยมีผลลัพธ์คือ 0.998 0.999 0.999 0.999 และ 0.998 ตามลำดับ fold ดังแสดงในรูปที่ 43



ภาพประกอบ 43 กราฟแสดงค่า recall ของการใช้งานค่า IPID ในการไม่ใช้งาน Random forest

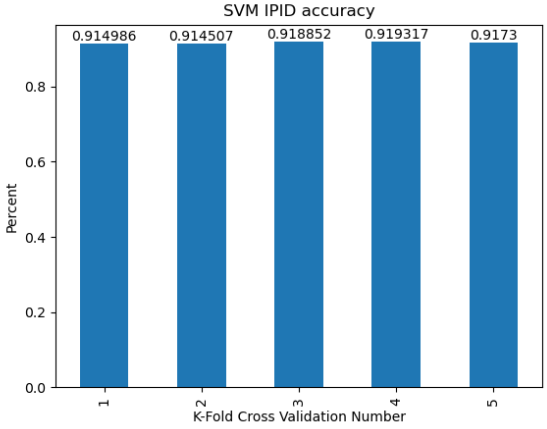
ในการทดลองการสร้างแบบจำลองการทำนายนั้นได้ผลการทดลองค่า f-1 แบบ Stratified K Fold Cross Validation ทั้ง 5 ครั้งโดยมีผลลัพธ์คือ 0.998 0.999 0.999 0.999 และ 0.998 ตามลำดับ fold ดังแสดงในรูปที่ 44



ภาพประกอบ 44 กราฟแสดงค่า f-1 ของการใช้งานค่า IPID ในการไม่ใช้งาน Random forest

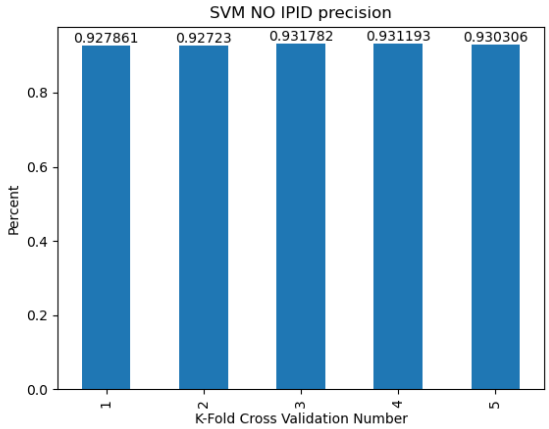
4.2.4.4 Support Vector Machine

ในการทดลองการสร้างแบบจำลองการทำนายนั้นได้ผลการทดลองค่า accuracy แบบ Stratified K Fold Cross Validation ทั้ง 5 ครั้งโดยมีผลลัพธ์คือ 0.915 0.919 0.919 0.919 และ 0.917 ตามลำดับ fold ดังแสดงในรูปที่ 45



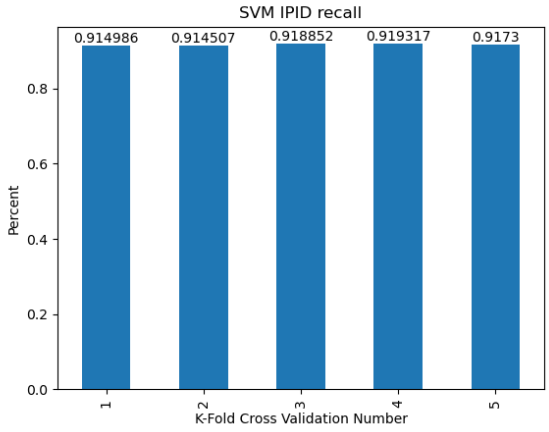
ภาพประกอบ 45 กราฟแสดงค่า accuracy ของการใช้งานค่า IPID ในการไม่ใช้งาน Support Vector Machines

ในการทดลองการสร้างแบบจำลองการทำนายนั้นได้ผลการทดลองค่า precision แบบ Stratified K Fold Cross Validation ทั้ง 5 ครั้งโดยมีผลลัพธ์คือ 0.928 0.927 0.932 0.931 และ 0.930 ตามลำดับ fold ดังแสดงในรูปที่ 46



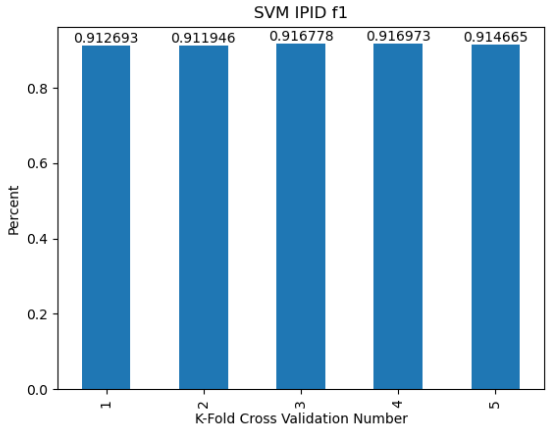
ภาพประกอบ 46 กราฟแสดงค่า precision ของการใช้งานค่า IPID ในการไม่ใช้งาน Support Vector Machines

ในการทดลองการสร้างแบบจำลองการทำนายนั้นได้ผลการทดลองค่า recall แบบ Stratified K Fold Cross Validation ทั้ง 5 ครั้งโดยมีผลลัพธ์คือ 0.915 0.915 0.919 0.918 และ 0.917 ตามลำดับ fold ดังแสดงในรูปที่ 47



ภาพประกอบ 47 กราฟแสดงค่า recall ของการใช้งานค่า IPID ในการไม่ใช้งาน Support Vector Machines

ในการทดลองการสร้างแบบจำลองการทำนายนั้นได้ผลการทดลองค่า f-1 แบบ Stratified K Fold Cross Validation ทั้ง 5 ครั้งโดยมีผลลัพธ์คือ 0.913 0.912 0.917 0.917 และ 0.915 ตามลำดับ fold ดังแสดงในรูปที่ 48



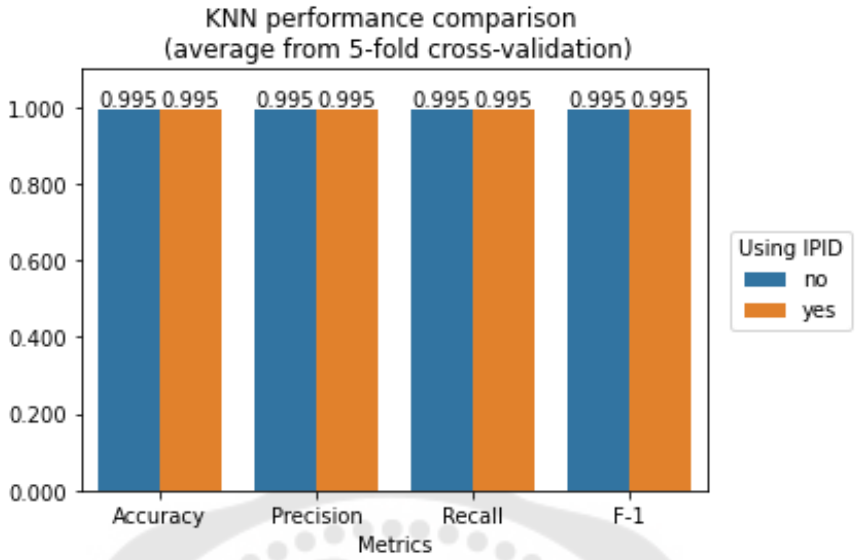
ภาพประกอบ 48 กราฟแสดงค่า f-1 ของการใช้งานค่า IPID ในการไม่ใช้งาน Support Vector

บทที่ 5
สรุปและอภิปรายผล

จากการทดลองการสร้างแบบจำลองการทำนายอุปกรณ์ IoT จากการเรียนรู้ของเครื่องโดยใช้งานชุดข้อมูลที่เป็นสาธารณะ IoT Traffic Trace จาก UNSW ด้วยการแปลงข้อมูลดิบในรูปของไฟล์นามสกุล PCAP ให้อยู่ในรูปของข้อมูลทางสถิติของอุปกรณ์ในช่วงเวลา 10 นาที โดยอุปกรณ์ที่สนใจเป็นอุปกรณ์ IoT ที่มีปริมาณข้อมูลเกิน 2,000 ตัวอย่างและเป็นอุปกรณ์ IoT ประกอบด้วย Smart Things, Amazon Echo, Samsung SmartCam, Dropcam, Belkin Wemo switch, Belkin wemo motion sensor, Triby Speaker, Netatmo weather station, Withings Smart Baby Monitor, TP Link Smart plug, Withings Aura smart sleep sensor, Light Bulbs LiFX Smart Bulb และ Insteon Camera

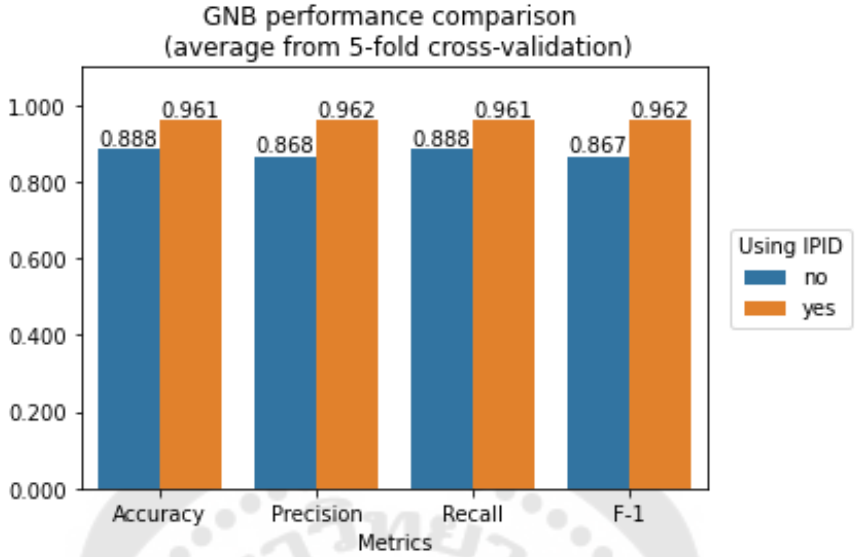
โดยนำข้อมูลนั้นเข้าไปแบบจำลองการทำนายทั้ง 4 รูปแบบได้แก่ K-nearest Neighbors, Naïve Bayesian, Random Forest และ Support Vector Machine โดยเสริมด้วยการแบ่งข้อมูลแบบ 5-Fold cross validation จากนั้นหาค่า average ของตัวชี้วัด และทำการเปรียบเทียบผลลัพธ์ระหว่างการใช้งานค่า IPID และไม่ใช้งานค่า IPID นั้นได้ผลลัพธ์ที่เป็นส่วนต่างออกมาดังนี้

เมื่อใช้งานด้วยวิธีการ K-nearest Neighbors นั้นพบว่าค่าตัวชี้วัดต่างๆนั้นมีค่าเท่ากันดังแสดงในภาพประกอบที่ 49 การสร้างแบบจำลองการทำนายนี้อาจไม่ได้นำค่า IPID มาเป็นตัวชี้วัดในการจำแนกอุปกรณ์เพิ่มเติม



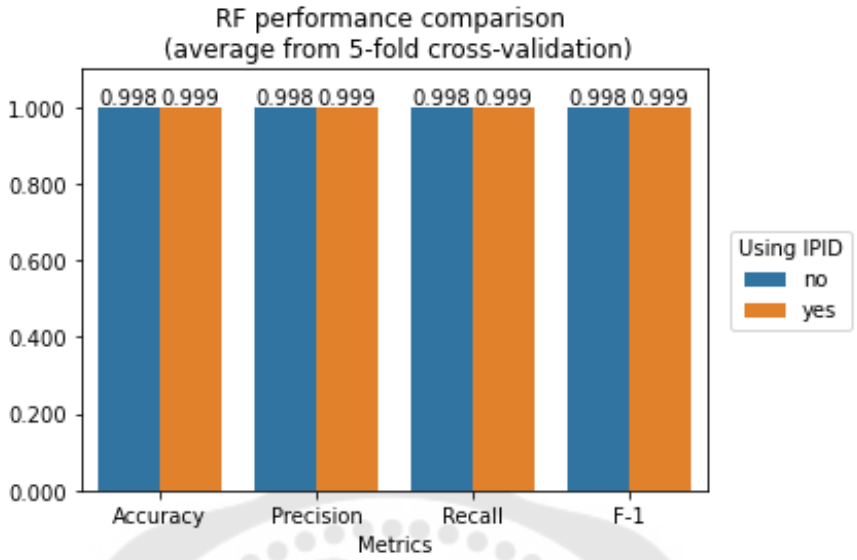
ภาพประกอบ 49 กราฟแสดงค่าสมรรถนะของแบบจำลองการทำนายของการใช้งานค่า IPID และไม่ใช้งาน IPID ของแบบจำลองการทำนาย K-nearest Neighbors

เมื่อใช้งานแบบจำลองการทำนาย Naïve Bayesian แล้วเมื่อนำผลลัพธ์มาเปรียบเทียบดังแสดงในภาพประกอบที่ 50 พบว่าได้รับค่า Accuracy สูงขึ้นเป็น 7.3 เปอร์เซ็นต์ ในส่วนของ Precision นั้นสูงขึ้น 9.4 เปอร์เซ็นต์ ค่า Recall นั้นสูงขึ้นที่ 7.3 เปอร์เซ็นต์ และ F1 นั้นสูงขึ้นที่ 9.5 เปอร์เซ็นต์ แสดงให้เห็นว่าพฤติกรรมของ IPID นั้นสามารถทำให้แบบจำลองการทำนาย Naïve Bayesian นั้นสามารถคาดการณ์อุปกรณ์ได้ง่ายและแม่นยำขึ้น



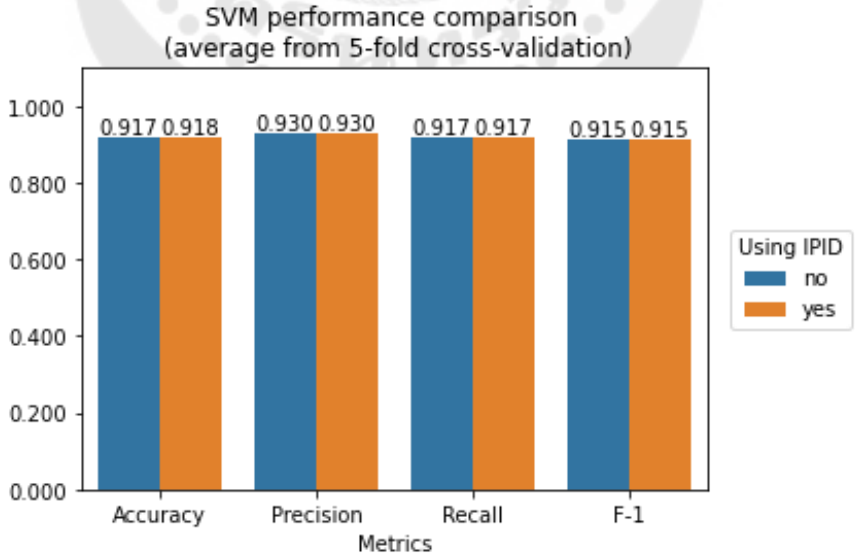
ภาพประกอบ 50 กราฟแสดงค่าสมรรถนะของแบบจำลองการทำนายของการใช้งานค่า IPID และไม่ใช้งาน IPID ของแบบจำลองการทำนาย Naïve Bayesian

เมื่อใช้นงานแบบจำลองการทำนาย Random Forest แล้วเมื่อนำผลลัพธ์มาเปรียบเทียบ ดังแสดงในภาพประกอบที่ 51 พบว่าได้รับค่า Accuracy, Precision, Recall และ F1 สูงขึ้นเป็น 0.1 เปอร์เซ็นต์เท่านั้นในทั้งหมด 4 ค่าตัวชี้วัดสมรรถนะของแบบจำลองการทำนาย



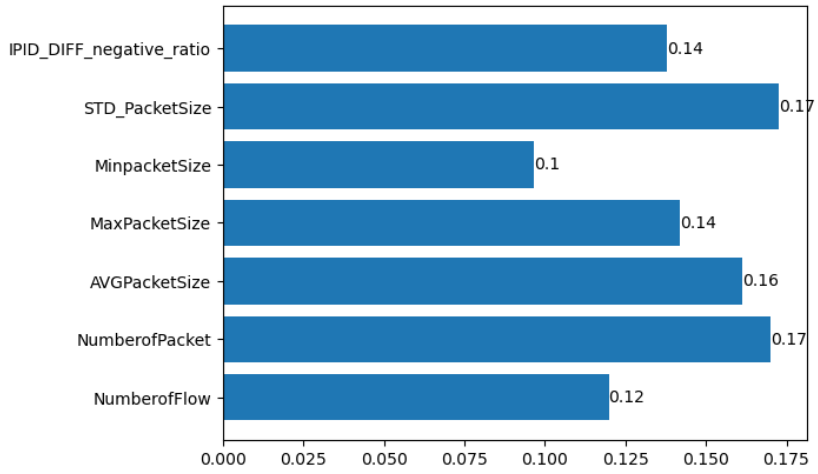
ภาพประกอบ 51 กราฟแสดงค่าสมรรถนะของแบบจำลองการทำนายของการใช้งานค่า IPID และไม่ใช้งาน IPID ของแบบจำลองการทำนาย Random Forest

เมื่อใช้นงานแบบจำลองการทำนาย Support Vector Machine แล้วเมื่อนำผลลัพธ์มาเปรียบเทียบดังแสดงในภาพประกอบที่ 52 พบว่าได้รับค่าความแตกต่างของค่า Accuracy เพียงแค่ 0.1 เปอร์เซ็นต์เพียงเท่านั้น



ภาพประกอบ 52 กราฟแสดงค่าสมรรถนะของแบบจำลองการทำนายของการใช้งานค่า IPID และไม่ใช้งาน IPID ของแบบจำลองการทำนาย Support Vector Machine

จากผลลัพธ์ต้น จะแสดงให้เห็นถึง ผลลัพธ์ต่างๆ ของแบบจำลองการทำนาย ว่าน้ำคุณสมบัติของข้อมูลที่ได้นั้น มาตรวจสอบค้นหาความสำคัญของคุณสมบัติ (Feature Importance) จะพบได้ว่าในคุณสมบัติของ IPID_DIFF_negative_ratio นั้นมีค่าคุณสมบัติอยู่ที่ 0.14 ซึ่งไม่ใช่ค่าที่สูงที่สุดโดยมีค่าที่สูงที่สุดคือ STD_Packet นั้นมีค่าคุณสมบัติอยู่ที่ 0.17 ซึ่งมากกว่า IPID_DIFF_negative_ratio อยู่ที่ 0.03



ภาพประกอบ 53 ความสำคัญของคุณสมบัติ (Feature Importance) ของชุดข้อมูล

จากผลการทดลองพบว่าการใช้งานข้อมูลเครือข่ายเฉพาะส่วนของ Network Layer และ Transport Layer ในช่วงเวลา 10 นาที นั้นสามารถทำงานได้ดีในการแยกแยะอุปกรณ์ IoT แต่ละชนิดกับ ML Model ที่เลือกมาทั้ง 4 ตัว นอกจากนั้นแล้วการใช้พฤติกรรมของ IPID นั้นยังสามารถเพิ่มความสามารถให้กับ Model ในการแยกแยะอุปกรณ์ดังจะเห็นได้อย่างชัดเจนเมื่อใช้งานร่วมกับแบบจำลองการทำนาย Naïve Bayesian พบว่าค่าตัวชี้วัดมีค่ามากขึ้นอย่างเห็นได้ชัด

หากผู้วิจัยนั้นมีความจำเป็นที่จะต้องใช้งานแบบจำลองการทำนาย แม้การประยุกต์การใช้งานจริงนั้น ทางผู้วิจัยสนใจเลือกใช้แบบจำลองการทำนาย ในรูปแบบของ Random Forest อันเนื่องมาจาก ผลลัพธ์ของการทำนายนั้น 3 สามารถสร้างผลการทำนายที่ดีที่สุดเมื่อเทียบกันทั้ง 4 แบบจำลองการทำนาย

บรรณานุกรม

Al-Qaseemi, S. A., Almulhim, H. A., Almulhim, M., & Chaudhry, S. R. (2016). IoT architecture challenges and issues: Lack of standardization. *2016 Future Technologies Conference (FTC)*, 731-738.

Aluthge, N. (2017). IoT device fingerprinting with sequence-based features Department of Computer Science. 71-71.

Apthorpe, N., Reisman, D., & Feamster, N. (2017). A Smart Home is No Castle: Privacy Vulnerabilities of Encrypted IoT Traffic. http://arxiv.org/abs/1705.06805

Meidan, Y., Bohadana, M., Shabtai, A., Guarnizo, J. D., Ochoa, M., Tippenhauer, N. O., & Elovici, Y. (2017). ProfilIoT: a machine learning approach for IoT device identification based on network traffic analysis. *Proceedings of the ACM Symposium on Applied Computing*, *Part F1280*, 506-509. https://doi.org/10.1145/3019612.3019878

Meidan, Y., Bohadana, M., Shabtai, A., Ochoa, M., Tippenhauer, N. O., Guarnizo, J. D., & Elovici, Y. (2017). Detection of Unauthorized IoT Devices Using Machine Learning Techniques. http://arxiv.org/abs/1709.04647

Mongkolluksamee, S., Fukuda, K., & Pongpaibool, P. (2012). Counting NATted hosts by observing TCP/IP field behaviors. 2012 IEEE International Conference on Communications (ICC),

Sivanathan, A., Gharakheili, H. H., Loi, F., Radford, A., Wijenayake, C., Vishwanath, A., & Sivaraman, V. (2019). Classifying IoT Devices in Smart Environments Using Network Traffic Characteristics. *IEEE Transactions on Mobile Computing*, *18*(8), 1745-1759. https://doi.org/10.1109/tmc.2018.2866249

Sivanathan, A., Sherratt, D., Gharakheili, H. H., Radford, A., Wijenayake, C., Vishwanath, A., & Sivaraman, V. (2017). Characterizing and classifying IoT traffic in smart cities and campuses. *2017 IEEE Conference on Computer Communications Workshops, INFOCOM WKSHPS 2017*, 559-564. https://doi.org/10.1109/INFCOMW.2017.8116438

ประวัติผู้เขียน