การวิเคราะห์และตรวจจับมัลแวร์ด้วยการเรียนรู้เชิงลึก ผ่านโครงข่ายประสาทเทียม

Malware Analysis and Detection Using Deep Learning via Artificial Neural Networks

ประวีณ ไม้เกตุ

Praveen Maigate

สาขาวิชาธุรกิจดิจิทัล คณะดิจิทัล วิทยาลัยเทคโนโลยีสยาม

Digital Business Program , Faculty of Digital , Siam Technology College

E-Mail : praveenm@siamtechno.ac.th

(Received : December 18, 2024; Revised : April 17, 2025; Accepted : April 22, 2025)

บทคัดย่อ

การวิจัยครั้งนี้มีวัตถุประสงค์เพื่อ 1) พัฒนาโมเดลการเรียนรู้เชิงลึกโดยใช้โครงข่ายประสาทเทียมแบบเชื่อมต่ออย่างสมบูรณ์ (Fully Connected Neural Network: FCNN) สำหรับการตรวจจับและวิเคราะห์มัลแวร์ 2) ศึกษาพฤติกรรมและรูปแบบการโจมตีของมัลแวร์ เช่น บรูทฟอร์ซ ฟิชชิง แรนซัมแวร์ และดีดอส บนข้อมูลจากโปรโตคอลการเข้าถึงระยะไกล (Remote Desktop Protocol: RDP) และซอฟต์แวร์ AnyDesk และ 3) ประเมินประสิทธิภาพและความแม่นยำของโมเดลที่พัฒนาขึ้นในการระบุพฤติกรรมที่ผิดปกติในสภาพแวดล้อมที่หลากหลาย เครื่องมือที่ใช้ในการวิจัย ได้แก่ โปรแกรม AnyDesk, โปรโตคอล RDP, TensorFlow, Keras และ Wireshark โดยใช้ตัวชี้วัดทางสถิติ ได้แก่ ค่า Accuracy, Precision, Recall และ F1-score

ผลการวิจัยพบว่า 1) โมเดล FCNN ที่พัฒนาขึ้นสามารถตรวจจับพฤติกรรมผิดปกติจากข้อมูลที่ได้จากโปรโตคอล RDP และ AnyDesk ได้อย่างมีประสิทธิภาพ โดยมีค่า Accuracy สูงสุดถึง 91.49% 2) ผลการประเมินประสิทธิภาพจากชุดข้อมูลทดสอบ พบว่าโมเดลมีค่า Precision 95.45%, Recall 87.50% และ F1-score 91.30% ซึ่งแสดงถึงความสมดุลระหว่างความแม่นยำในการตรวจจับและอัตราการแจ้งเตือนผิดพลาดที่ต่ำ และ 3) โมเดลสามารถวิเคราะห์และตรวจจับพฤติกรรมการโจมตีของมัลแวร์แต่ละประเภทได้อย่างแม่นยำ เช่น การโจมตีแบบบรูทฟอร์ซ ฟิชชิง แรนซัมแวร์ และดีดอส ซึ่งช่วยลดความเสี่ยงจากภัยคุกคามไซเบอร์ในสภาพแวดล้อมการใช้งานจริงได้อย่างมีประสิทธิภาพ

คำสำคัญ : การเรียนรู้เชิงลึก, โครงข่ายประสาทเทียม, การตรวจจับมัลแวร์, ความปลอดภัยทางไซเบอร์

ABSTRACT

This study aimed to (1) develop a deep learning model using a Fully Connected Neural Network (FCNN) for malware detection and analysis, (2) investigate the behaviors and attack patterns of various types of malware such as brute-force, phishing, ransomware, and DDoS based on data from Remote Desktop Protocol (RDP) and AnyDesk software, and (3) evaluate the performance and accuracy of the proposed model in detecting anomalous behaviors across diverse environments. The tools used in this research included AnyDesk, RDP protocol, TensorFlow, Keras, and Wireshark. Statistical metrics used to assess model performance were Accuracy, Precision, Recall, and F1-score.

The results showed that (1) the developed FCNN model effectively detected anomalous behaviors from data captured via RDP and AnyDesk, achieving a maximum accuracy of 91.49%; (2) performance evaluation on the test dataset indicated that the model achieved a Precision of

95.45%, Recall of 87.50%, and F1-score of 91.30%, demonstrating a balanced trade-off between detection accuracy and false-positive rate; and (3) the model was capable of accurately analyzing and identifying distinct malware attack behaviors such as brute-force attacks, phishing, ransomware, and DDoS thus significantly reducing cybersecurity risks in real-world environments.

Keywords : Deep Learning, Neural Networks, Malware Detection, Cybersecurity

บทนำ

ในยุคดิจิทัลที่เทคโนโลยีมีบทบาทสำคัญในทุกมิติของชีวิตประจำวันและการดำเนินธุรกิจ ความปลอดภัยทางไซเบอร์ได้กลายเป็นปัจจัยสำคัญที่องค์กรและบุคคลต้องให้ความสำคัญ เนื่องจากภัยคุกคามทางไซเบอร์ เช่น มัลแวร์ (Malware) มีการพัฒนาอย่างซับซ้อนและสร้างความเสียหายร้ายแรงต่อระบบและข้อมูล โดยเฉพาะในกรณีของการโจมตีที่เพิ่มความซับซ้อนอย่าง ransomware, phishing และ Distributed Denial of Service (DDoS) ซึ่งส่งผลต่อความเชื่อมั่นและความปลอดภัยของระบบคอมพิวเตอร์

การตรวจจับและวิเคราะห์มัลแวร์จึงเป็นกระบวนการที่มีความสำคัญอย่างยิ่งในการป้องกันและลดความเสียหายจากการโจมตีเหล่านี้ งานวิจัยนี้นำเสนอการประยุกต์ใช้การเรียนรู้เชิงลึก (Deep Learning) [1] ผ่านโครงข่ายประสาทเทียมแบบ Fully Connected Neural Networks (FCNN) เพื่อเพิ่มประสิทธิภาพในการตรวจจับและวิเคราะห์มัลแวร์ โดยเน้นการพัฒนาโมเดลที่สามารถระบุพฤติกรรมผิดปกติในระบบได้อย่างแม่นยำ โดยใช้ข้อมูลจากโปรโตคอลการเข้าถึงระยะไกล (Remote Desktop Protocol: RDP) และซอฟต์แวร์ AnyDesk ซึ่งเป็นช่องทางที่มักถูกใช้เป็นเป้าหมายของการโจมตี

โดยการวิจัยครั้งนี้ สามารถเป็นประโยชน์ต่อผู้ดูแลระบบเครือข่ายและผู้รับผิดชอบด้านความปลอดภัยทางไซเบอร์ ที่ต้องการเครื่องมือช่วยในการตรวจจับและป้องกันการโจมตีของมัลแวร์ ผ่านระบบที่มีประสิทธิภาพสูง เช่น ฝ่ายไอทีขององค์กรหรือหน่วยงานราชการ นักวิจัยและนักพัฒนาเทคโนโลยีด้าน Cybersecurity และ AI ที่ต้องการศึกษาแนวทางใหม่ ๆ ในการประยุกต์ใช้การเรียนรู้เชิงลึก (Deep Learning) และโครงข่ายประสาทเทียมในการตรวจจับภัยคุกคามทางไซเบอร์

นอกจากนี้ งานวิจัยยังให้ความสำคัญกับการสำรวจรูปแบบการโจมตีที่หลากหลาย เช่น Brute Force Attack, Phishing, Ransomware และ DDoS เพื่อสร้างความเข้าใจเชิงลึกเกี่ยวกับลักษณะของภัยคุกคามเหล่านี้ และพัฒนาระบบที่มีความสามารถสูงในการป้องกันและรับมือกับการโจมตีทางไซเบอร์ในอนาคต

1. วัตถุประสงค์การวิจัย

1.1 เพื่อพัฒนาโมเดลการเรียนรู้เชิงลึก Deep Learning โดยใช้ FCNN สำหรับการตรวจจับและวิเคราะห์มัลแวร์ที่มีประสิทธิภาพสูง

1.2 เพื่อศึกษาพฤติกรรมและรูปแบบการโจมตีของมัลแวร์ เช่น Brute Force Attack Phishing Ransomware และ DDoS บนข้อมูลจากโปรโตคอลการเข้าถึงระยะไกล RDP และซอฟต์แวร์ AnyDesk

1.3 เพื่อประเมินประสิทธิภาพและความแม่นยำของโมเดลที่พัฒนาขึ้นในการระบุพฤติกรรมที่ผิดปกติและตรวจจับมัลแวร์ในสภาพแวดล้อมที่หลากหลาย

2. เอกสารและงานวิจัยที่เกี่ยวข้อง

การตรวจจับมัลแวร์โดยใช้โมเดล [2] Siamese Network แบบคู่งานวิจัยนี้นำเสนอการใช้ Dual Siamese Network สำหรับตรวจจับมัลแวร์ที่หลากหลาย โดยอาศัยการแปลงข้อมูลไบนารีเป็นภาพเกรย์สเกล และการวิเคราะห์ภาพจากความถี่ของ Opcode ด้วยเทคนิค Few-Shot Learning (FSL) เพื่อเรียนรู้ข้อมูลจำนวนน้อยและ

เปรียบเทียบความเหมือนของข้อมูล ผลการทดลองแสดงให้เห็นว่าระบบสามารถตรวจจับมัลแวร์ใหม่ได้อย่างแม่นยำถึง 95.9% และ 99.83% จากชุดข้อมูลที่ทดสอบ

การศึกษาเปรียบเทียบวิธี RNN สำหรับตรวจจับโค้ดอันตรายบนเว็บ งานวิจัยนี้เปรียบเทียบประสิทธิภาพของเทคนิค RNN-based Methods [3] เช่น LSTM, GRU, และ SRU ในการตรวจจับโค้ดอันตรายที่ฝังอยู่ในเว็บแอปพลิเคชัน พบว่า GRU และ SRU มีค่าการเรียกคืน สูงถึง 81.07% และ 80.96% ตามลำดับ โดยการเตรียมข้อมูลก่อนการประมวลผลมีผลกระทบต่อประสิทธิภาพของโมเดลอย่างมาก

วิธีการตรวจจับมัลแวร์โดยใช้ CNN สำหรับ IoT [4] งานวิจัยนี้เสนอวิธีการตรวจจับมัลแวร์รูปแบบใหม่ที่มุ่งเน้นอุปกรณ์ Internet of Things (IoT) โดยใช้เทคนิค Convolutional Neural Network (CNN) และการแปลงไฟล์ไบนารีเป็นภาพ RGB เพื่อดึงคุณสมบัติและความสัมพันธ์ของมัลแวร์ในรูปแบบต่าง ๆ ผลการทดลองแสดงให้เห็นว่าโมเดลสามารถปรับใช้ได้ข้ามแพลตฟอร์ม และมีความแม่นยำในการตรวจจับมัลแวร์ได้อย่างมีประสิทธิภาพ

การวิเคราะห์ทางนิติดิจิทัลของแอปพลิเคชัน AnyDesk [5] งานวิจัยนี้มุ่งเน้นการวิเคราะห์ข้อมูลทางนิติดิจิทัลจากซอฟต์แวร์ AnyDesk โดยใช้เครื่องมือทางนิติวิทยาศาสตร์เพื่อดึงข้อมูลที่เกี่ยวข้อง เช่น ไฟล์บันทึกการใช้งาน (Log Files), การเคลื่อนย้ายไฟล์, และกิจกรรมการเชื่อมต่อของผู้ใช้ โดยผลการศึกษาเน้นถึงความสำคัญของการวิเคราะห์ *artifacts* เพื่อระบุตัวผู้กระทำผิดและข้อมูลที่เกี่ยวข้องกับความปลอดภัย

การใช้ DeepLearning สำหรับตรวจจับการฉ้อโกงผ่านบัตรเครดิต [6] งานวิจัยนี้เป็นการทบทวนเทคนิค Deep Learning เช่น CNN, RNN, LSTM และ GRU ในการตรวจจับการฉ้อโกงบัตรเครดิต โดยเน้นการเปรียบเทียบประสิทธิภาพของโมเดล ความท้าทายในการฝึกฝนโมเดล และการแก้ไขปัญหา เช่น ความไม่สมดุลของข้อมูล (Class Imbalance) ผลการศึกษาระบุว่าการใช้โมเดล DL สามารถวิเคราะห์ข้อมูลเชิงลึกและตรวจจับการทุจริตได้อย่างมีประสิทธิภาพ

3. ขั้นตอนการดำเนินการวิจัย

3.1 การตั้งค่าและการสร้างสถานการณ์ (Setup & Scenario Creation) เพื่อทดสอบความสามารถของโมเดลการตรวจจับมัลแวร์ ระบบได้จำลองสถานการณ์ที่สอดคล้องกับรูปแบบภัยคุกคามทางไซเบอร์ที่พบได้บ่อยในสภาพแวดล้อมการทำงานจริง โดยแบ่งออกเป็น 4 รูปแบบการโจมตีหลัก ได้แก่ Brute Force, Phishing Ransomware และ DDoS ดังรายละเอียดในตารางที่ 1

ตารางที่ 1 แสดงรูปแบบการโจมตี รายละเอียดการจำลอง และข้อมูลเชิงประมาณ

รูปแบบการโจมตี	รายละเอียดการจำลอง	ข้อมูลเชิงปริมาณ
การโจมตีแบบ Brute Force	1. จำลองการพยายามเข้าสู่ระบบโดยใช้การคาดเดารหัสผ่านอัตโนมัติผ่าน Remote Desktop Protocol (RDP) และ AnyDesk 2. ใช้ชุดเครื่องมือเช่น Hydra และ Metasploit ในการทดสอบการเจาะระบบ 3. จำลองพฤติกรรมของแฮกเกอร์ที่พยายามเข้าถึงบัญชีที่ได้รับการป้องกัน	1. จำนวนครั้งในการพยายามเข้าสู่ระบบ500 ครั้งต่อบัญชี 2. ระยะเวลาของการโจมตีแต่ละครั้ง 1-3 นาที 3. ใช้ IP Address ที่แตกต่างกัน 10 หมายเลข 4. ทดสอบการเปลี่ยนรหัสผ่านแบบสุ่มและพจนานุกรม (Dictionary Attack)

ตารางที่ 1 แสดงรูปแบบการโจมตี รายละเอียดการจำลอง และข้อมูลเชิงประมาณ (ต่อ)

รูปแบบการโจมตี	รายละเอียดการจำลอง	ข้อมูลเชิงปริมาณ
การโจมตีแบบ Phishing	1. จำลองการส่งอีเมลฟิชชิ่งไปยังผู้ใช้ปลายทาง (เป้าหมายคือพนักงานในองค์กรจำลอง) 2. ฝังลิงก์อันตรายหรือไฟล์แนบ เช่น ไฟล์ .docx, .pdf, .zip ที่มีสคริปต์แฝง 3. วิเคราะห์พฤติกรรมของผู้ใช้ว่ามีการเปิดไฟล์หรือคลิกลิงก์อันตรายหรือไม่	1. จำนวนอีเมลฟิชชิ่งที่ส่งออก 200 ฉบับ 2. ประเมินอัตราการคลิกเปิดไฟล์หรือกดลิงก์ เป้าหมาย 10-30% 3. เวลาตรวจจับของระบบ น้อยกว่า 5 วินาทีหลังจากผู้ใช้เปิดไฟล์
การโจมตีแบบ Ransomware	1. จำลองการแพร่กระจายของ Ransomware ผ่านไฟล์แนบในอีเมล และ เว็บไซต์ดาวน์โหลดไฟล์ 2. ใช้มัลแวร์ตัวอย่างที่เข้ารหัสไฟล์ในเครื่อง เป้าหมายเป็นระบบ Windows และ Linux 3. วิเคราะห์การเชื่อมต่อกับเซิร์ฟเวอร์ Command & Control (C2)	1.จำนวนไฟล์ที่ถูกเข้ารหัสจำลอง 500 ไฟล์ 2.วิเคราะห์แพ็กเก็ตข้อมูลเพื่อดูการส่งข้อมูลออกไปยัง C2 3.อัตราความสำเร็จของการตรวจจับ 95% ขึ้นไป
การโจมตีแบบ DDoS	1. จำลองการโจมตีเพื่อทำให้ระบบ ไม่สามารถให้บริการได้ (Denialof Service) 2. ใช้ SYN Flood, UDP Flood และ HTTP Flood เพื่อเพิ่มโหลดบนเซิร์ฟเวอร์ 3. ทดสอบการป้องกันของ Firewall และ Intrusion Detection System (IDS)	1. จำนวนคำร้องขอที่ส่งไปยังเซิร์ฟเวอร์ 100,000-500,000 คำร้องขอ 2. ระยะเวลาของการโจมตี 5-15 นาที 3. การวัดประสิทธิภาพของระบบก่อนและหลังโจมตี (ลดลง 40-80%)

3.2 ความหลากหลายของสภาพแวดล้อมในการทดสอบ เพื่อให้การทดสอบครอบคลุมสภาพแวดล้อมการทำงานจริง ระบบได้จำลองการทดสอบใน 4 รูปแบบของสภาพแวดล้อม ดังรายละเอียดในตารางที่ 2

ตารางที่ 2 แสดงสภาพแวดล้อมการทำงาน และความหลากหลายของสภาพแวดล้อม

สภาพแวดล้อมการทำงาน	ความหลากหลายของสภาพแวดล้อม
เครือข่ายองค์กรภายใน (On-Premise Network)	1. ใช้ Firewall และ Intrusion Detection System (IDS) ในการวิเคราะห์และบล็อกการโจมตี 2. จำลองสถานการณ์ที่มัลแวร์แพร่กระจายภายในระบบ
ระบบบนคลาวด์ (Cloud-Based Environment)	1. จำลองเซิร์ฟเวอร์ใน AWS, Google Cloud และ Microsoft Azure 2.ทดสอบการเข้าถึงจาก IP Address ภายนอกและการโจมตีผ่าน API
อุปกรณ์ปลายทาง (Endpoint Devices)	1. จำลองพฤติกรรมของพนักงานที่ใช้ แล็ปท็อปและโทรศัพท์มือถือ ในการทำงาน 2. วิเคราะห์พฤติกรรมของมัลแวร์ที่พยายามเจาะอุปกรณ์ส่วนตัว
เครือข่ายสาธารณะ (Public Network)	1. จำลองการเชื่อมต่อผ่าน Wi-Fi สาธารณะ เช่น ที่ร้านกาแฟ หรือสถานที่สาธารณะ 2. ตรวจสอบพฤติกรรมของมัลแวร์ที่แพร่กระจายผ่านเครือข่ายที่ไม่ปลอดภัย

3.3. รวบรวมและเตรียมข้อมูล (Data Collection and Preparation) ในการวิจัยครั้งนี้ ได้ทำการรวบรวมและเตรียมข้อมูลสำหรับการพัฒนาโมเดลตรวจจับมัลแวร์ โดยข้อมูลที่ใช้ในการฝึกและทดสอบมากจากแหล่งข้อมูลสาธารณะ รายงานการโจมตี และการจำลองสถานการณจริง ข้อมูลที่ได้จาก Wireshark และเครื่องมือเฝ้าระวังเครือข่ายอื่นๆ ถูกแปลงให้อยู่ในรูปแบบที่สามารถใช้ในการเรียนรู้ของโมเดล โดยแสดงรายละเอียดในตารางที่ 3

ตารางที่ 3 แสดงรายละเอียดการรวบรวมและเตรียมข้อมูลในการวิจัย

หัวข้อ	รายละเอียด
ขนาดข้อมูล	รวม 100,000 ตัวอย่าง แบ่งเป็น Brute Force (25%) Phishing (20%) Ransomware (15%) DDoS (30%) และ Traffic ปกติ (10%)
การจัดการความไม่สมดุลของข้อมูล	1. Oversampling (SMOTE) เพิ่มตัวอย่างในกลุ่มที่มีน้อย 2. Undersampling ลดจำนวนข้อมูลกลุ่มที่มากเกินไป 3. Data Augmentation ปรับค่าฟีเจอร์ เช่น IP, Packet Size
การแบ่งข้อมูล	1. ใช้ Stratified Sampling และRandom Sampling แบ่งเป็น Training Set (70%) และTesting Set(30%)
เหตุผลของสัดส่วน 70/30	1. 70% ฝึกโมเดล ให้เรียนรู้ได้ดี 2. 30% ทดสอบโมเดล ประเมินผลได้แม่นยำ 3. เป็นสมดุลที่เหมาะสม ลด Overfitting และให้ผลลัพธ์เชื่อถือได้

3.4 การพัฒนาโมเดลการเรียนรู้เชิงลึก (Deep Learning Model Development)

ในการวิจัยครั้งนี้ ใช้ โมเดล Fully Connected Neural Network (FCNN) เป็นโมเดลหลักในการตรวจจับมัลแวร์ โดยมีการปรับแต่งโครงสร้างโมเดลและค่าพารามิเตอร์เพื่อเพิ่มประสิทธิภาพสูงสุด โมเดลประกอบด้วย 3 Hidden Layers ที่มีจำนวนหน่วยประมวลผล 64, 32 และ 16 โหนด ตามลำดับ โดยใช้ฟังก์ชันกระตุ้น ReLU ใน Hidden Layers และ Sigmoid ใน Output Layer เพื่อจำแนกประเภทข้อมูล การปรับแต่งพารามิเตอร์ (Hyperparameter Tuning) ทำโดยทดสอบค่าต่าง ๆ และพบว่าการตั้งค่า Learning Rate 0.001 Batch Size 32 และ Optimizer Adam ให้ผลลัพธ์ที่ดีที่สุด เพื่อป้องกัน Overfitting ใช้ Dropout 0.2, Early Stopping และ Data Augmentation ช่วยเพิ่มความสามารถในการจำแนกมัลแวร์และลดข้อผิดพลาดในการตรวจจับ ดังแสดงในรายละเอียดในตารางที่ 4

ตารางที่ 4 แสดงรายละเอียดของโมเดลที่ใช้และโมเดลที่ไม่ใช้

โมเดลที่ใช้และโมเดลที่ไม่ใช้	รายละเอียด
เหตุผลในการเลือก FCNN	- เหมาะสำหรับข้อมูลที่มีโครงสร้าง (Structured Data) - วิเคราะห์ความสัมพันธ์ของฟีเจอร์ได้ดี - ข้อมูลไม่ใช่ภาพ (CNN) หรืออนุกรมเวลาโดยตรง (RNN)
เหตุผลที่ไม่เลือก CNN หรือ RNN	- CNN เหมาะสำหรับข้อมูลเชิงพื้นที่ เช่น รูปภาพมัลแวร์ - RNN/LSTM เหมาะกับข้อมูลลำดับเวลา แต่งานนี้เน้นฟีเจอร์ที่เป็นอิสระกัน
โครงสร้างโมเดล FCNN	- Input Layer: รับฟีเจอร์ เช่น IP, Port, Packet Size - Hidden Layers: 3 ชั้น (64, 32, 16 โหนด) - Activation: ReLU (Hidden), Sigmoid (Output)
Optimizer และ Hyperparameters	- Optimizer: Adam - Learning Rate: 0.001, Batch Size: 32, Epochs: 20 - Dropout (0.2), Early Stopping, Data Augmentation

ตารางที่ 4 แสดงรายละเอียดของโมเดลที่ใช้และโมเดลที่ไม่ใช้ (ต่อ)

โมเดลที่ใช้และโมเดลที่ไม่ใช้	รายละเอียด
แนวทางทดลองเพิ่มเติม (อนาคต)	- CNN: แปลงข้อมูลแพ็กเก็ตเป็นภาพ (Byteplot) - LSTM: จัดข้อมูลตามลำดับเวลาเพื่อวิเคราะห์รูปแบบต่อเนื่องของข้อมูล เช่น DDoS

ตารางที่ 5 แสดงโครงสร้างและรายละเอียดของโมเดล FCNN

รายละเอียด	ค่าพารามิเตอร์ / โครงสร้างโมเดล
ประเภทของโมเดล	Fully Connected Neural Network (FCNN)
จำนวนชั้น (Layers)	รวม 5 ชั้น (Input 1 ชั้น, Hidden 3 ชั้น, Output 1 ชั้น)
จำนวนโหนดใน Hidden Layers	ชั้นที่ 1: 64 โหนดชั้นที่ 2: 32 โหนดชั้นที่ 3: 16 โหนด
ฟังก์ชันกระตุ้น (Activation Function)	Hidden Layers: ReLUOutput Layer: Sigmoid
Optimizer ที่ใช้	Adam Optimizer
Learning Rate	0.001
Batch Size	32
จำนวน Epoch	20
เทคนิคเพิ่มเติม (Regularization)	Dropout (อัตรา 0.2), Early Stopping, Data Augmentation

3.5 การฝึกอบรมและการทดสอบโมเดล (Training and Testing the Model)

โมเดลถูกฝึกอบรมโดยใช้ข้อมูลที่ได้จัดเตรียมไว้ และใช้การประเมินผลด้วยข้อมูลทดสอบเพื่อวัดความสามารถของโมเดลในการตรวจจับมัลแวร์ โดยตัวชี้วัดประสิทธิภาพหลักที่ใช้ในการประเมิน ได้แก่ ค่า Accuracy, Precision, Recall และ F1-score เพื่อประเมินความสามารถของโมเดลในการแจ้งเตือนการโจมตีและลดอัตราการแจ้งเตือนที่ผิดพลาด (False Positives) ขั้นตอนการฝึกโมเดลประกอบด้วย

3.5.1 Preprocessing ข้อมูลที่ได้จาก Wireshark ซึ่งอยู่ในรูปแบบ Packet Capture (PCAP) จะถูกแปลงให้อยู่ในรูปแบบที่สามารถนาเข้าสู่โมเดลได้ เช่น การทำ One-hot encoding สำหรับคำสั่งที่ตรวจจับได้ และการแปลง IP address หรือข้อมูลเชิงตัวเลขให้อยู่ในรูปแบบที่โมเดลเข้าใจได้

3.5.2 Training โมเดลจะถูกฝึกโดยใช้ Training Set ซึ่งประกอบไปด้วยข้อมูลการเชื่อมต่อระยะไกลที่มีทั้งกิจกรรมปกติ และกิจกรรมที่ผิดปกติ เช่น การเข้าถึงไฟล์ที่ไม่ได้รับอนุญาตหรือการส่งคำสั่งที่ผิดปกติ

3.5.3 Evaluation หลังจากฝึกโมเดลเรียบร้อยแล้ว ข้อมูลจาก Testing Set จะถูกนำมาทดสอบเพื่อประเมินประสิทธิภาพของโมเดล

3.6 การดึงฟีเจอร์ (Feature Extraction)

3.6.1 IP Address Analysis ใช้ IP Address เพื่อวิเคราะห์การเชื่อมต่อ ถ้ามีการเชื่อมต่อจาก IP ที่ไม่น่าเชื่อถือ ฟีเจอร์นี้จะช่วยระบุพฤติกรรมที่ผิดปกติ

3.6.2 Protocol Detection วิเคราะห์โปรโตคอล เช่น TCP, UDP, หรือ ICMP

3.6.3 Port Number ตรวจสอบพอร์ตที่ใช้ เช่น Port 3389 ของ RDP เพื่อตรวจหาพฤติกรรมที่น่าสงสัย

3.6.4 Packet Size ขนาดของแพ็กเก็ตอาจเป็นตัวชี้วัดถึงการโจมตี เช่น การโจมตี DDoS

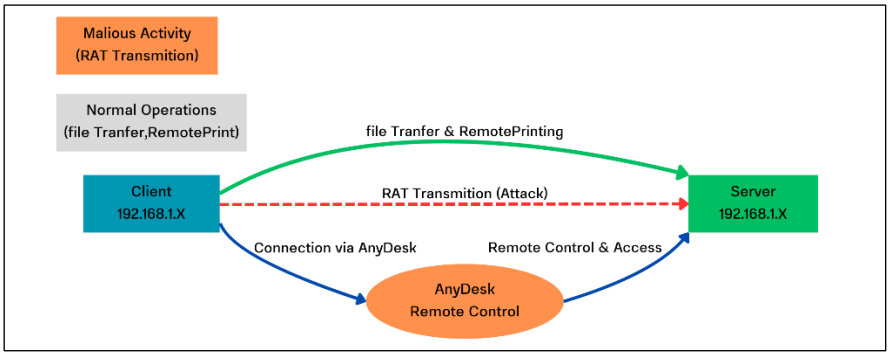
ตารางที่ 6 แสดงข้อมูลแพ็กเก็ตที่มีการกรองข้อมูล

Source IP	Destination IP	Protocol	Source Port	Destination Port	Packet Size (Bytes)	Timestamp
192.168.1.1	192.168.1.3	TCP	52345	443	345	2024-10-02 00:00:00
10.0.0.1	192.168.1.4	UDP	23422	3389	982	2024-10-02 00:01:00
172.16.0.1	192.168.1.2	ICMP	46712	7070	768	2024-10-02 00:02:00
192.168.1.2	192.168.1.3	TCP	51123	80	1280	2024-10-02 00:03:00

จากตารางที่ 6 แสดงข้อมูลแพ็กเก็ตที่มีการกรองข้อมูล และการแบ่งชุดข้อมูล (Data Splitting)Training Set ใช้สำหรับการฝึกโมเดล Test Set ใช้สำหรับการทดสอบโมเดล โดยแบ่งข้อมูลในสัดส่วน 70/30 เพื่อทดสอบประสิทธิภาพของโมเดล

3.7 การตั้งค่าและการสร้างสถานการณ์ (Setup & Scenario Creation)

ในงานวิจัยนี้ ได้ทำสร้างสถานการณ์ต่าง ๆ เพื่อทดสอบและวิเคราะห์การตรวจจับพฤติกรรมการใช้งานที่ผิดปกติผ่านโปรโตคอล RDP และซอฟต์แวร์ AnyDesk ข้อมูลจากสถานการณ์จำลองนี้จะถูกเก็บเป็นแพ็กเก็ตและวิเคราะห์เพื่อค้นหาสัญญาณการโจมตี เช่น การโจมตีจาก Remote Access Trojan (RAT)



ภาพที่ 1 ขั้นตอนในการตั้งค่าและสร้างสถานการณ์จำลอง

จากภาพที่ 1 เป็นขั้นตอนในการตั้งค่าและสร้างสถานการณ์จำลองสามารถจำแนกได้ดังนี้ การตั้งค่าเครือข่ายและ AnyDesk

1) เชื่อมต่อ Client และ Server ในเครือข่าย LAN โดยใช้ IP ภายใน (192.168.1.x)

2) ใช้ AnyDesk เพื่อควบคุมเครื่องปลายทางและจำลองการใช้งานปกติ เช่น ส่งไฟล์ และ พิมพ์งาน ระยะไกลสถานการณ์โจมตี (RAT Transmission)

1) สร้าง RAT ด้วยเครื่องมือ เช่น Metasploit Framework

2) ส่ง RAT ไปยังเครื่องปลายทางผ่าน AnyDesk และจับแพ็กเก็ตด้วย Wireshark

3) วิเคราะห์แพ็กเก็ตเพื่อหาพฤติกรรมผิดปกติ เช่น การสื่อสารกับ Command & Control (C2)

การสร้างและทดสอบโมเดล FCNN

1) ใช้ TensorFlow/PyTorch สร้างโมเดล FCNN วิเคราะห์ข้อมูลแพ็กเก็ต

2) ฝึกโมเดลด้วยข้อมูลที่แปลงเป็นตัวเลข เช่น IP, Protocol และ Timestamp แปลงเป็นค่าวินาทีหรือมิลลิวินาที

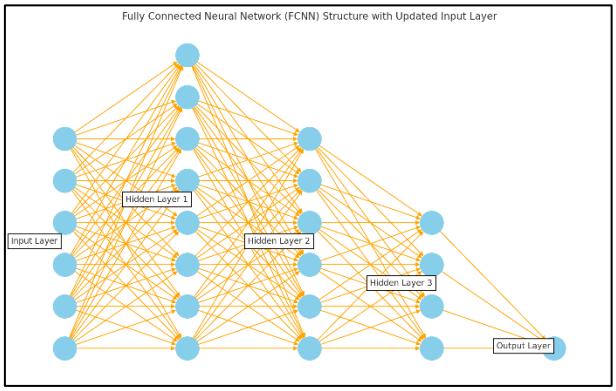
3) ประเมินโมเดลเพื่อจำแนกพฤติกรรมปกติและการโจมตี

ตารางที่ 7 แสดงตัวอย่างการแปลงข้อมูล

Source IP	Destination IP	Protocol	Source Port	Destination Port	Packet Size (Bytes)	Timestamp
3232235777	3232235779	1	52345	443	345	1693507200
167772161	3232235780	0	23422	3389	982	1693507260

จากตารางที่ 7 แสดงตัวอย่างการแปลงข้อมูลการแบ่งชุดข้อมูล (Train-Test Split) ข้อมูลที่เตรียมแล้วจะถูกแบ่งเป็นชุดฝึก Training Set และชุดทดสอบ Test Set ในอัตราส่วน 70:30 เพื่อประเมินความสามารถของโมเดลและป้องกัน Overfitting การปรับสเกลข้อมูล เนื่องจากฟีเจอร์บางส่วนมีค่าต่างกันมาก การปรับสเกลข้อมูลให้อยู่ในช่วงเดียวกันช่วยให้โมเดลเรียนรู้ได้ดีขึ้นและประมวลผลข้อมูลได้มีประสิทธิภาพ

3.8 การฝึกโมเดล (Model Training) หลังจากเตรียมข้อมูลแพ็กเก็ตเครือข่าย ขั้นตอนต่อไปคือการฝึกโมเดลสำหรับตรวจจับพฤติกรรมการโจมตีหรือการเชื่อมต่อที่ผิดปกติผ่านโปรโตคอล RDP และ AnyDesk โดยใช้ Neural Network ที่เรียนรู้จากฟีเจอร์ที่คัดเลือกมา เพื่อสร้างความสัมพันธ์และจำแนกข้อมูลว่าเป็นปกติหรือไม่ การเลือกโครงสร้างโมเดล (Model Architecture Selection) ใช้โครงข่ายประสาทเทียมแบบ FCNN ซึ่งเหมาะกับการวิเคราะห์ข้อมูลแบบมีโครงสร้าง เช่น ข้อมูลแพ็กเก็ตเครือข่าย



ภาพที่ 2 แสดงโครงสร้างทดสอบโมเดลด้วย FCNN

จากภาพที่ 2 โครงสร้างทดสอบโมเดลด้วย FCNN ประกอบด้วย

1. Input Layer รับข้อมูลฟีเจอร์ เช่น Source IP, Destination IP, Protocol, Source Port, Destination Port, Packet Size

2. Hidden Layers [Hidden Layer 1: 8 โหนด, ฟังก์ชันการกระตุ้น ReLU] [Hidden Layer 2: 6 โหนด, ฟังก์ชันการกระตุ้น ReLU] [Hidden Layer 3: 4 โหนด, ฟังก์ชันการกระตุ้น ReLU]

3. Output Layer 1 โหนด, ฟังก์ชันการกระตุ้น Sigmoid สาหรับจำแนกข้อมูลเป็นปกติ (0) หรือการโจมตี (1)

4. การตั้งค่าสูตรและพารามิเตอร์ (Hyperparameter Tuning) พารามิเตอร์ที่ใช้ในการทดลอง

4.1 Learning Rate : 0.001 เพื่อให้โมเดลปรับค่าพารามิเตอร์ได้ราบรื่น

4.2 Optimizer : Adam Optimizer ที่ปรับค่า Learning Rate ให้เหมาะสม

4.3 Loss Function : Binary Crossentropy สาหรับปัญหาการจำแนกแบบ Binary

Batch Size : 32 ,Epochs : 20

4. การวัดประสิทธิภาพ

การทดสอบเพื่อประเมินประสิทธิภาพของโมเดล [4] โดยการวัดค่า Accuracy ของการตรวจจับพฤติกรรมที่ผิดปกติ ค่า Precision ของการระบุพฤติกรรมที่เป็นมัลแวร์ ค่า Recall ซึ่งวัดความสามารถในการตรวจจับพฤติกรรมผิดปกติ และ F1-score ซึ่งเป็นการวัดความสมดุลระหว่าง Precision และ Recall

$$\text{Accuracy} = \frac{\text{True Positives+True Negatives}}{\text{True Positives+True Negatives+False Positives+False Negatives}} \quad (1)$$

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives+False Positives}} \quad (2)$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives+False Negatives}} \quad (3)$$

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

ผลการวิจัย

1. ผลการพัฒนาโมเดลการเรียนรู้เชิงลึก Deep Learning โดยใช้ FCNN สำหรับการตรวจจับและวิเคราะห์มัลแวร์ที่มีประสิทธิภาพสูงในงานวิจัยนี้ ได้สร้างสถานการณ์ต่าง ๆ โมเดลด้วย FCNN เพื่อทดสอบและวิเคราะห์การตรวจจับพฤติกรรมการใช้งานที่ผิดปกติผ่านโปรโตคอล RDP และซอฟต์แวร์ AnyDesk ข้อมูลจากสถานการณ์จำลองนี้จะถูกเก็บเป็นแพ็กเก็ตและวิเคราะห์เพื่อค้นหาสัญญาณการโจมตีผลการทดสอบแสดงในตารางที่ 6

ตารางที่ 8 แสดงข้อมูลแพ็กเก็ตที่มีการกรองข้อมูล

Epoch	Training Loss	Training Accuracy (%)	Validation Loss	Validation Accuracy (%)
1	0.465	76.8	0.478	75.2
2	0.392	83.5	0.422	82.4
3	0.352	86.9	0.399	84.1
4	0.352	88.5	0.385	85.6
5	0.301	90.2	0.372	86.3
6	0.287	91.1	0.360	87.1
7	0.275	92.0	0.350	87.8
8	0.260	92.8	0.340	88.2
9	0.250	93.2	0.330	88.7
10	0.240	93.4	0.320	89.2

ตารางที่ 8 แสดงข้อมูลแพ็กเก็ตที่มีการกรองข้อมูล (ต่อ)

Epoch	Training Loss	Training Accuracy (%)	Validation Loss	Validation Accuracy (%)
11	0.231	93.8	0.310	89.7
12	0.225	94.1	0.302	90.0
13	0.219	94.3	0.295	90.3
14	0.215	94.5	0.288	90.7
15	0.210	94.8	0.280	91.0
16	0.205	95.0	0.275	91.2
17	0.202	95.3	0.270	91.5
18	0.199	95.5	0.265	91.8
19	0.195	95.7	0.260	92.1
20	0.192	95.9	0.255	92.5

จากตารางที่ 8 ผลการทดสอบโมเดล Fully Connected Neural Network (FCNN) พบว่า ค่า Training Accuracy และ Validation Accuracy เพิ่มขึ้นอย่างต่อเนื่องตลอดระยะเวลาการฝึกอบรมทั้งหมด 20 Epoch โดย Epoch ที่ 20 มีค่า Training Accuracy สูงสุดที่ 95.9% และ Validation Accuracy สูงสุดที่ 92.5% ในขณะที่ค่า Training Loss และ Validation Loss ลดลงอย่างต่อเนื่องเช่นกัน แสดงให้เห็นว่าโมเดลที่พัฒนาขึ้นมีประสิทธิภาพดีในการเรียนรู้ สามารถวิเคราะห์ข้อมูลแพ็กเก็ตที่ถูกกรองเพื่อระบุพฤติกรรมผิดปกติและตรวจจับมัลแวร์ได้อย่างมีประสิทธิภาพสูง และสามารถนำไปใช้งานจริงได้อย่างน่าเชื่อถือ

2. ผลการทดลองหลังจากการฝึกโมเดลเสร็จสิ้น ผู้วิจัยได้ทำการทดสอบโมเดลกับชุดข้อมูลทดสอบ (Test Set) เพื่อประเมินความสามารถของโมเดลในการตรวจจับพฤติกรรมที่ผิดปกติ
การคำนวณตัวชี้วัดจากข้อมูลการทดลอง

2.1 จำนวนข้อมูลโจมตีจริงทั้งหมด (Actual Attacks, AT) = 1200 รายการ
2.2 จำนวนข้อมูลที่โมเดลระบุว่าเป็นการโจมตี (Predicted Attacks,PT) = 1100 รายการ
2.3 จำนวนการตรวจจับที่ถูกต้อง (True Positives, TP) = 1,050 รายการ
2.4 จำนวนข้อมูลโจมตีจริงที่โมเดลระบุว่าไม่ใช่การโจมตี (False Negatives, FN) = 150 รายการ
2.5 จำนวนข้อมูลปกติที่โมเดลระบุว่าถูกโจมตี (False Positives, FP) = 50 รายการ
2.6 จำนวนข้อมูลปกติที่โมเดลระบุได้ถูกต้อง (True Negatives, TN) = 1100 รายการ

ผลลัพธ์การคำนวณโมเดล

Accuracy: 91.49%

$$\text{Accuracy} = \frac{1{,}050+1{,}100}{1{,}050+1{,}100+50+150} = \frac{2{,}150}{2{,}350} \approx 0.9149$$ คิดเป็นร้อยละ 91.49

Precision: 95.45%

$$\text{Precision} = \frac{1{,}050}{1{,}050+50} = \frac{1{,}050}{1{,}100} \approx 0.9545$$ คิดเป็นร้อยละ95.45

Recall: 87.50%

$$\text{Recall} = \frac{1{,}050}{1{,}050+150} = \frac{1{,}050}{1{,}200} \approx 0.875$$ คิดเป็นร้อยละ87.50

F1-Score: 91.30%

$$\text{F1-Score} = 2x\frac{0.9545+0.875}{0.9545+0.875} = 2 \times \frac{0.8357}{1.8295} \approx 0.9130$$ คิดเป็นร้อยละ91.30

2. พฤติกรรมและรูปแบบการโจมตีของมัลแวร์ เช่น บูสฟอส แอซเท็ก ฟิชชิง แรนซัมแวร์ และ ดีดอส บนข้อมูลจากโปรโตคอลการเข้าถึงระยะไกล และซอฟต์แวร์ แอนี่เดสก์

งานวิจัยนี้ได้ศึกษาพฤติกรรมและรูปแบบการโจมตีที่สำคัญของมัลแวร์จำนวน 4 ประเภท ได้แก่ บรูทฟอร์ซ (Brute Force), ฟิชชิง (Phishing), แรนซัมแวร์ (Ransomware) และ ดีดอส (DDoS) โดยใช้ข้อมูลที่เก็บรวบรวมจากโปรโตคอลการเข้าถึงระยะไกล (Remote Desktop Protocol: RDP) และซอฟต์แวร์แอนี่เดสก์ (AnyDesk) ซึ่งเป็นช่องทางที่พบการโจมตีบ่อยครั้ง สรุปรายละเอียดของแต่ละรูปแบบในตารางที่ 9

ตารางที่ 9 แสดงรูปแบบการโจมตี และพฤติกรรมสำคัญที่พบจากการศึกษา

รูปแบบการโจมตี	พฤติกรรมสำคัญที่พบจากการศึกษา
บรูทฟอร์ซ (Brute Force)	มีลักษณะของการโจมตีแบบเดารหัสผ่านซ้ำ ๆ จากหลาย IP address ในช่วงเวลาสั้นๆ- พบการโจมตีในโปรโตคอล RDP และ AnyDesk โดยแฮกเกอร์ใช้ชุดเครื่องมืออย่าง Hydra หรือ Metasploit ในการพยายามเข้าถึงระบบ
ฟิชชิง (Phishing)	พบการส่งอีเมลหรือข้อความที่มีลิงก์หรือไฟล์แนบอันตราย โดยผู้ใช้ที่ตกเป็นเหยื่อจะมีพฤติกรรมคลิกลิงก์หรือเปิดไฟล์แนบที่คล้ายการใช้งานปกติทั่วไป- ข้อมูลแพ็กเก็ตแสดงการเชื่อมต่อกับเซิร์ฟเวอร์ที่มีความผิดปกติ
แรนซัมแวร์ (Ransomware)	มีการเข้ารหัสข้อมูลสำคัญในเครื่องของเหยื่อ และเชื่อมต่อกับเซิร์ฟเวอร์ Command and Control (C2) เพื่อส่งข้อมูลออกไปภายนอก- มัลแวร์มักส่งผ่านทางไฟล์แนบหรือดาวน์โหลดผ่านเว็บไซต์ที่ไม่น่าเชื่อถือ
ดีดอส (DDoS)	การโจมตีแบบปฏิเสธการให้บริการ พบการส่งคำร้องขอปริมาณมากจนเซิร์ฟเวอร์ปลายทางให้บริการไม่ได้ชั่วคราว- การโจมตีใช้วิธีการเช่น SYN Flood, UDP Flood และ HTTP Flood

จากการศึกษาข้อมูลแพ็กเก็ตผ่านโปรโตคอล RDP และ AnyDesk พบว่าโมเดลที่พัฒนาขึ้นสามารถวิเคราะห์และแยกแยะพฤติกรรมการโจมตีมัลแวร์แต่ละประเภทได้อย่างแม่นยำ โดยเฉพาะการตรวจสอบแพ็กเก็ตข้อมูลที่มีรูปแบบผิดปกติ เช่น ขนาดแพ็กเก็ต, หมายเลขพอร์ต, IP address ที่ไม่ปกติ และการเชื่อมต่อกับเซิร์ฟเวอร์ภายนอกที่น่าสงสัย ทำให้สามารถตรวจจับภัยคุกคามได้อย่างมีประสิทธิภาพสูงและป้องกันความเสียหายจากมัลแวร์ดังกล่าวได้ดีขึ้น

3. ผลการประเมินประสิทธิภาพและความแม่นยำของโมเดลที่พัฒนาขึ้นในการระบุพฤติกรรมที่ผิดปกติและตรวจจับมัลแวร์ในสภาพแวดล้อมที่หลากหลาย

ตารางที่ 10 ผลการประเมินประสิทธิภาพของโมเดล FCNN ในการตรวจจับพฤติกรรมที่ผิดปกติ

ตัวชี้วัด	ค่าที่ได้ (%)	ความหมาย
Accuracy	91.49	ความถูกต้องโดยรวมในการจำแนกข้อมูล
Precision	95.45	ความแม่นยำในการตรวจจับพฤติกรรมที่เป็นภัย
Recall	87.50	ความสามารถในการตรวจพบภัยทั้งหมด
F1-score	91.30	ความสมดุลระหว่าง Precision และ Recall

จากตารางที่ 10 แสดงค่าประสิทธิภาพของโมเดล FCNN ในการจำแนกพฤติกรรมปกติและพฤติกรรมที่เป็นภัยพบว่า โมเดลมีค่า Accuracy สูงถึง 91.49 % และมีค่า Precision, Recall และ F1-score ในระดับที่สมดุล แสดงให้เห็นว่าโมเดลมีความสามารถในการตรวจจับมัลแวร์ได้อย่างแม่นยำและลดการแจ้งเตือนผิดพลาด

อภิปรายผลการวิจัย

1. การพัฒนาโมเดล Fully Connected Neural Network (FCNN) สามารถเรียนรู้ข้อมูลได้อย่างต่อเนื่องและมีประสิทธิภาพ โดยค่าความถูกต้อง (Accuracy) เพิ่มขึ้นจาก 76.8% ใน Epoch แรก ไปถึง 95.9% ใน Epoch ที่ 20 และ Validation Accuracy เพิ่มขึ้นจนถึง 92.5% แสดงว่าโมเดลไม่มีภาวะ Overfitting และสามารถจำแนกข้อมูลได้ดีทั้งในชุดฝึกและชุดทดสอบ ซึ่งแสดงถึงความสามารถของโมเดลในการตรวจจับพฤติกรรมที่ผิดปกติจากข้อมูลแพ็กเก็ตจริงได้อย่างมีประสิทธิภาพ และมีศักยภาพในการนำไปประยุกต์ใช้กับสถานการณ์จริง

2. การศึกษาพฤติกรรมและรูปแบบการโจมตีของมัลแวร์ งานวิจัยได้จำแนกรูปแบบพฤติกรรมของมัลแวร์ 4 ประเภท ได้แก่ Brute Force, Phishing, Ransomware และ DDoS ซึ่งพบว่าแต่ละประเภทมีลักษณะเฉพาะที่สามารถตรวจจับได้จากข้อมูลพฤติกรรม เช่น ขนาดแพ็กเก็ต, พอร์ต, IP address และลักษณะการเชื่อมต่อ ทั้งนี้ โมเดล FCNN สามารถตรวจจับพฤติกรรมการโจมตีที่ซ่อนอยู่ในกระบวนการใช้งานทั่วไปได้อย่างแม่นยำ โดยเฉพาะการโจมตีผ่าน AnyDesk และ RDP ซึ่งเป็นช่องทางยอดนิยมของแฮกเกอร์ แสดงให้เห็นว่าโมเดลมีศักยภาพในการตรวจสอบภัยคุกคามที่เกิดขึ้นจริงในสภาพแวดล้อมการทำงาน

3. การประเมินประสิทธิภาพและความแม่นยำของโมเดลในการระบุพฤติกรรมผิดปกติในสภาพแวดล้อมที่หลากหลาย ผลการประเมินแสดงว่าโมเดลมีความแม่นยำสูงในการตรวจจับพฤติกรรมที่เป็นภัย โดยมีค่า Accuracy 91.49%, Precision 95.45%, Recall 87.50% และ F1-score 91.30% อย่างไรก็ตาม ยังพบข้อผิดพลาดบางประการ เช่น False Positives จำนวน 50 รายการ และ False Negatives จำนวน 150 รายการ ซึ่งมักเกิดในกรณีที่พฤติกรรมมีลักษณะคล้ายกับการใช้งานปกติ หรือมีการเข้ารหัสข้อมูลสูง เช่น Ransomware ผ่าน SSL/TLS เพื่อลดข้อผิดพลาดเหล่านี้ งานวิจัยได้เสนอแนวทางการปรับปรุงโมเดล เช่น การเพิ่มข้อมูลที่หลากหลาย การวิเคราะห์ฟีเจอร์เชิงลึก การใช้ Hybrid Learning และการนำเทคนิค Adaptive Learning มาเสริมความสามารถของโมเดลให้รองรับภัยคุกคามรูปแบบใหม่ได้อย่างมีประสิทธิภาพมากยิ่งขึ้น ซึ่งสอดคล้องกับงานวิจัยก่อนหน้า เช่น Li et al. (2021) และ Soni et al. (2024) ที่ชี้ให้เห็นว่าโมเดล Deep Learning ที่มีการเรียนรู้อย่างต่อเนื่องสามารถปรับตัวรับภัยใหม่ ๆ ได้ดี

ข้อเสนอแนะ

จากผลการวิจัยพบว่า โมเดลโครงข่ายประสาทเทียมแบบเชื่อมต่อสมบูรณ์ (Fully Connected Neural Network: FCNN) สามารถตรวจจับและวิเคราะห์พฤติกรรมที่ผิดปกติในโปรโตคอล RDP และซอฟต์แวร์ AnyDesk ได้อย่างแม่นยำ โดยมีค่า Accuracy อยู่ที่ 91.49% ซึ่งถือว่าอยู่ในระดับที่ดี อย่างไรก็ตาม โมเดลยังมีข้อจำกัดบางประการที่ควรได้รับการพัฒนาเพิ่มเติม เพื่อให้สามารถนำไปประยุกต์ใช้งานจริงได้อย่างมีประสิทธิภาพมากยิ่งขึ้นในอนาคตข้อเสนอแนะที่สำคัญประการแรก คือ ควรขยายชุดข้อมูลที่ใช้ในการฝึกโมเดลให้มีความหลากหลายมากขึ้น โดยเฉพาะการเพิ่มข้อมูลจากโปรโตคอลอื่น ๆ ที่มีความเสี่ยงต่อการถูกโจมตี เช่น VNC, TeamViewer และ SSH เพื่อเพิ่มความครอบคลุมของระบบตรวจจับ และลดความเอนเอียงที่อาจเกิดจากการใช้ข้อมูลจำลองเพียงบางประเภท ซึ่งอาจนำไปสู่ปัญหา Overfitting ได้ ควรพัฒนาเทคนิคการเรียนรู้แบบผสม (Hybrid Learning) โดยผสานการทำงานของโมเดล Deep Learning เข้ากับวิธีการแบบดั้งเดิม เช่น Decision Tree หรือ Support Vector Machine (SVM) เพื่อเพิ่มความสามารถของระบบในการตรวจจับพฤติกรรมการโจมตีที่มีลักษณะซับซ้อน หรือคล้ายคลึงกับพฤติกรรมของผู้ใช้งานทั่วไป ซึ่งอาจตรวจจับได้ยากด้วยโมเดลเพียงประเภทเดียว ควรมีการทดสอบโมเดลในสภาพแวดล้อมจริง เช่น เครือข่ายขององค์กร หรือระบบ Cloud Computing เพื่อประเมินความสามารถของโมเดลในสถานการณ์ที่มีปัจจัยแวดล้อมจริง และปรับแต่งโมเดลให้สามารถตอบสนองต่อเงื่อนไขการใช้งานที่ซับซ้อนและเปลี่ยนแปลงได้อย่างเหมาะสม ควรพัฒนาโมเดลให้สามารถเรียนรู้แบบปรับตัว (Adaptive Learning) เพื่อให้สามารถอัปเดตความรู้จากภัยคุกคามรูปแบบใหม่ได้โดยไม่ต้องเริ่มต้นฝึกโมเดลใหม่ทั้งหมด ซึ่งจะช่วยลดเวลาและต้นทุนในการฝึกโมเดลซ้ำ และยังเพิ่มศักยภาพในการรับมือกับภัยคุกคามทางไซเบอร์ที่มีการพัฒนาอย่างต่อเนื่อง

เอกสารอ้างอิง

[1] A. Sharma, M. U. Reddy, A. Lathigara, and A. Verma, “Automated malware classification using deep learning neural networks,” in *Proc. 2023 IEEE Int. Conf. ICT Bus. Ind. Gov. (ICTBIG)*, Indore, India, Dec. 2023, pp. 1–6, doi: 10.1109/ICTBIG59752.2023.10456242.

[2] B. An, J. Yang, S. Kim, and T. Kim, “Malware detection using dual Siamese network model,” *Comput. Model. Eng. Sci.* (CMES), vol. 141, no. 1, pp. 563–584, 2024, doi: 10.32604/cmes.2024.052403.

[3] Z. Guan, J. Wang, X. Wang, W. Xin, J. Cui, and X. Jing, “A comparative study of RNN-based methods for web malicious code detection,” in *Proc. 2021 IEEE 6th Int. Conf. Comput. Commun. Syst. (ICCCS)*, Chengdu, China, Apr. 2021, pp. 769–773, doi: 10.1109/ICCCS52626.2021.9449245.

[4] Q. Li, J. Mi, W. Li, J. Wang, and M. Cheng, “CNN-based malware variants detection method for Internet of Things,” *IEEE Internet Things J.*, vol. 8, no. 23, pp. 16946–16962, Dec. 2021, doi: 10.1109/JIOT.2021.3075694.

[5] N. Soni, M. Kaur, and V. Bhardwaj, “A forensic analysis of AnyDesk remote access application by using various forensic tools and techniques,” *Forensic Sci. Int.: Digit. Invest.*, vol. 48, p. 301695, Mar. 2024, doi: 10.1016/j.fsidi.2024.301695.

[6] I. D. Mienye and N. Jere, “Deep learning for credit card fraud detection: A review of algorithms, challenges, and solutions,” *IEEE Access*, vol. 12, pp. 96893–96910, 2024, doi: 10.1109/ACCESS.2024.3426955.