Rattanakosin Journal of Science and Technology: RJST
2025, Volume 7 Issue 3: 236-251
Research Article
Online ISSN: 2673-0553

การตรวจจับพฤติกรรมการโจมตีในเครือข่ายอินเทอร์เน็ตทุกสรรพสิ่ง ด้วยโครงข่ายประสาทเทียมคอนโวลูชันเชิงอนุกรมเวลา

ประวีณ ไม้เกตุ[1] และ กัญญณัฏฐ์ สุริยันต์[2*]

[1,2]สาขาวิชาธุรกิจดิจิทัล คณะดิจิทัล วิทยาลัยเทคโนโลยีสยาม

E-mail: [1]praveenm@siamtechno.ac.th, [2*]kunyanuts@siamtechno.ac.th

บทคัดย่อ

งานวิจัยนี้มีวัตถุประสงค์เพื่อ 1) พัฒนาเฟรมเวิร์คสำหรับตรวจจับพฤติกรรมการโจมตีในเครือข่าย IoT โดยใช้โมเดล Temporal Convolutional Networks (TCN) และ 2) ประเมินประสิทธิภาพของโมเดล TCN โดยเปรียบเทียบกับโมเดล RNN LSTM GRU และ DNN การทดลองใช้ระยะเวลา 12 สัปดาห์ โดยใช้ ภาษา Python การทดลองใช้ชุดข้อมูลสังเคราะห์ที่จำลองการโจมตีแบบการโจมตีแบบปฏิเสธการให้บริการ (Denial-of-Service :DoS) และการโจมตีทางไซเบอร์ที่ผู้ไม่หวังดีแอบดักฟัง และแก้ไขข้อมูล (Man-in-the-Middle: MITM) โดยเริ่มจากกระบวนการแปลงและปรับแต่งข้อมูล (Data Preprocessing) และฝึกโมเดลด้วยTensorFlow/Keras บน Google Colab ผลการทดลองพบว่า TCN Model มีค่า Recall สูง (0.9986) ซึ่งแสดงให้เห็นว่าสามารถตรวจจับเหตุการณ์โจมตีได้เกือบทั้งหมด อย่างไรก็ตาม Accuracy (≈0.50) และ Precision ต่ำ (0.4978) ทำให้เกิดอัตราการแจ้งเตือนผิดพลาด (False Positive) สูง และค่า Loss ผันผวนสูง สะท้อนว่าโมเดลยังไม่สามารถเรียนรู้ข้อมูลได้อย่างมีเสถียร สำหรับผลของการเปรียบเทียบกับโมเดลอื่น RNN ให้ผลลัพธ์ที่สมดุล LSTM และ GRU มีประสิทธิภาพสูงในข้อมูลที่ซับซ้อน และ DNN มีความแม่นยำสูงสุด แสดงให้เห็นว่าโมเดล TCN แม้จะสามารถตรวจจับพฤติกรรมการโจมตีได้ดีในแง่ของ Recall แต่ยังต้องปรับปรุงเพื่อเพิ่ม Precision และ Accuracy ดังนั้น การปรับปรุงคุณภาพของชุดข้อมูล การใช้กระบวนการปรับแต่งค่าพารามิเตอร์ของโมเดลที่กำหนดไว้ล่วงหน้า (Hyperparameter Tuning) เพื่อเพิ่มประสิทธิภาพของโมเดล การลด Overfitting ผ่าน Regularization Techniques และการทดสอบเฟรมเวิร์คกับชุดข้อมูลจริงเพื่อเพิ่มความแม่นยำ และลดอัตราการแจ้งเตือนผิดพลาด สิ่งนี้จะช่วยให้สามารถนำไปใช้งานในระบบรักษาความปลอดภัยของเครือข่าย IoT ได้อย่างมีประสิทธิภาพ

คำสำคัญ: อินเทอร์เน็ตในทุกสิ่ง (IoT) ความปลอดภัยทางไซเบอร์ โครงข่ายประสาทเทียมคอนโวลูชันเชิงอนุกรมเวลา (TCN) ระบบตรวจจับการบุกรุก (IDS) การวิเคราะห์ข้อมูลอนุกรมเวลา

* Corresponding author, e-mail: kunyanuts@siamtechno.ac.th

Received: 18 01 2025 / Revised: 04 03 2025 / Accepted: 01 05 2025

Rattanakosin Journal of Science and Technology: RJST
2025, Volume 7 Issue 3: 236-251
Research Article
Online ISSN: 2673-0553

Detection of Attack Behaviors in IoT Networks Using Temporal Convolutional Networks (TCN)

Praveen Maiget[1] and Kanyanut Suriyan[2*]
[1,2*]Department of Digital Business, Faculty of Digital, Siam Technology College
E-mail: [1]praveenm@siamtechno.ac.th, [2*]kunyanuts siamtechno.ac.th

Abstract

This research aims to (1) develop a framework for detecting attack behaviors in IoT networks using the Temporal Convolutional Networks (TCN) model and (2) evaluate the performance of the TCN model by comparing it with RNN, LSTM, GRU, and DNN models. The experiment was conducted over a 12-week period using Python as the primary programming language. A synthetic dataset was utilized, simulating Denial-of-Service (DoS) and Man-in-the-Middle (MITM) cyberattacks. The research process involved data transformation and preprocessing, followed by training the models using TensorFlow/Keras on Google Colab. The experimental results revealed that the TCN model achieved a high recall score (0.9986), indicating its ability to detect nearly all attack events. However, its accuracy (≈0.50) and precision (0.4978) were low, leading to a high false positive rate. Additionally, the loss values exhibited significant fluctuations, reflecting the model's instability in learning the data effectively. In comparison, the RNN model provided balanced results, LSTM and GRU demonstrated high efficiency in handling complex data, and DNN achieved the highest accuracy. These findings suggest that while the TCN model performs well in detecting attack behaviors in terms of recall, it requires further improvements to enhance precision and accuracy. To address these issues, improving dataset quality, applying hyperparameter tuning, implementing regularization techniques to reduce overfitting, and testing the framework with real-world datasets are recommended. These improvements will contribute to a more reliable and effective security system for IoT networks.

Keywords: Internet of things (IoT), Cybersecurity, Temporal convolutional networks (TCN), Intrusion detection system (IDS), Time-series analysis

[*] Corresponding author, e-mail: kunyanuts@siamtechno.ac.th
Received: 18 01 2025 / Revised: 04 03 2025 / Accepted: 01 05 2025

1. ที่มาและความสำคัญ

ในยุคดิจิทัลปัจจุบันเครือข่ายอินเทอร์เน็ตในทุกสิ่ง (Internet of Things: IoT) ได้กลายเป็นองค์ประกอบสำคัญของการดำเนินงานในหลากหลายภาคส่วน และเป็นแนวคิดที่กำลังได้รับความนิยมอย่างรวดเร็วในบริบทของการสื่อสารโทรคมนาคมไร้สายในยุคปัจจุบัน [1] แนวคิดนี้ช่วยให้เทคโนโลยีและอุปกรณ์อัจฉริยะสามารถเชื่อมต่อ และทำงานร่วมกันได้อย่างมีประสิทธิภาพ ส่งผลให้เกิดระบบอัตโนมัติที่ชาญฉลาดมากขึ้น ตัวอย่างที่โดดเด่น ได้แก่ เทอร์โมสแตต และระบบควบคุม HVAC (Heating Ventilation and Air Conditioning) ซึ่งช่วยยกระดับบ้านอัจฉริยะให้สามารถปรับอุณหภูมิและสภาพแวดล้อมได้อย่างแม่นยำและสะดวกสบายยิ่งขึ้น [2]

นอกจากนี้ IoT ยังถูกนำมาใช้อย่างแพร่หลายในเครือข่ายทั่วโลกโดยเชื่อมโยงวัตถุอัจฉริยะเข้ากับเทคโนโลยีอินเทอร์เน็ตที่ขยายออกไป ซึ่งครอบคลุมทั้งเทคโนโลยีสนับสนุนที่จำเป็น เช่น RFID เซ็นเซอร์ ตัวกระตุ้น (Actuators) และอุปกรณ์สื่อสารแบบเครื่องต่อเครื่อง (Machine-to-Machine Communication: M2M) รวมถึงชุดแอปพลิเคชันและบริการที่ใช้เทคโนโลยีเหล่านี้เพื่อสร้างโอกาสใหม่ทางธุรกิจและการตลาด [3]

อย่างไรก็ตามการเติบโตอย่างรวดเร็วของ IoT ได้นำมาทั้งโอกาสและความท้าทายโดยเฉพาะอย่างยิ่งในด้านความปลอดภัยทางไซเบอร์ [4] เนื่องจากอุปกรณ์ IoT จำนวนมากมีข้อจำกัดด้านทรัพยากรการประมวลผลพลังงานและหน่วยความจำ [5] อีกทั้งมีช่องโหว่ที่อาจถูกแสวงหาประโยชน์ได้ง่าย เช่นการขาดการเข้ารหัส (Encryption) และกลไกการพิสูจน์ตัวตนที่ไม่เพียงพอ [6] ทำให้ IoT ตกเป็นเป้าหมายของการโจมตีทางไซเบอร์ในหลากหลายรูปแบบไม่ว่าจะเป็นการโจมตีแบบ DoS (Denial-of-Service) การละเมิดข้อมูล (Data Breaches) และการแพร่กระจายของมัลแวร์ (Malware Infections) [7] ซึ่งล้วนเป็นปัจจัยที่ต้องให้ความสำคัญในการพัฒนาและเสริมสร้างมาตรการป้องกันด้านความปลอดภัยสำหรับระบบ IoT ในอนาคต

การรักษาความปลอดภัยในเครือข่าย IoT จึงเป็นสิ่งสำคัญระบบตรวจจับการบุกรุกที่ใช้ในการเฝ้าระวังและวิเคราะห์ทราฟฟิกในเครือข่าย หรือระบบคอมพิวเตอร์เพื่อระบุพฤติกรรมที่เป็นอันตราย หรือการโจมตีทางไซเบอร์ (Intrusion Detection System: IDS) เป็นกลไกพื้นฐานในการเฝ้าสังเกต (Monitor) และการตรวจจับกิจกรรมที่เป็นอันตราย (Detect malicious activities) [8] IDS แบบดั้งเดิมมักใช้วิธีการที่อาศัยกฎ (rules) หรือเงื่อนไขที่กำหนดไว้ล่วงหน้าเพื่อใช้ในการตรวจจับหรือจำแนกข้อมูลต่าง ๆ เช่น การระบุพฤติกรรมที่ผิดปกติ หรือการตรวจจับภัยคุกคามทางไซเบอร์ (Rule-based) หรือเทคนิคในการตรวจจับภัยคุกคามทางไซเบอร์ (Signature-based methods) ซึ่งอาศัยรูปแบบหรือโครงสร้างที่ถูกกำหนดล่วงหน้า (Predefined patterns) เพื่อใช้ในการตรวจจับ จัดประเภทหรือวิเคราะห์ข้อมูลในระบบต่าง ๆ เช่น ความปลอดภัยทางไซเบอร์ หรือการประมวลผลข้อมูล หรือลายเซ็นทางดิจิทัล (Signatures) ของการโจมตีที่เป็นที่รู้จักและถูกบันทึกไว้แล้วในฐานข้อมูลของระบบรักษาความปลอดภัยทางไซเบอร์ (Known attacks) ในการระบุและการจำแนก (classify) และจัดประเภททราฟฟิกที่เป็นอันตราย (Malicious traffic) ที่ไหลเวียนอยู่ในระบบเครือข่ายหรืออุปกรณ์โดยอาศัยกฎ (rules) ลักษณะเฉพาะหรือรูปแบบที่สามารถใช้ระบุและตรวจจับข้อมูลบางอย่างได้ (Signature) หรืออัลกอริธึมในการวิเคราะห์พฤติกรรมของทราฟฟิก เพื่อตรวจจับ

ภัยคุกคามทางไซเบอร์ [9] อย่างไรก็ตามวิธีการเหล่านี้อาจไม่เพียงพอในการตรวจจับการโจมตีรูปแบบใหม่ ๆ หรือการโจมตีทางไซเบอร์ที่อาศัยช่องโหว่ (vulnerabilities) ของซอฟต์แวร์หรือระบบที่ยังไม่มีการแก้ไข หรือยังไม่เป็นที่รู้จักโดยเจ้าของระบบหรือผู้พัฒนา (zero-day vulnerability) [10] เนื่องจากผู้โจมตีทางไซเบอร์ (Attackers) มักพัฒนาวิธีการโจมตีที่ซับซ้อนเพื่อการหลบเลี่ยง หรือข้ามผ่านกลไกความปลอดภัยแบบดั้งเดิม (Bypass traditional security mechanisms) เช่น ระบบป้องกันเครือข่ายที่ทำหน้าที่กรอง และควบคุมทราฟฟิกที่เข้าและออกจากเครือข่ายตามกฎที่กำหนดไว้ (Firewall) และซอฟต์แวร์ป้องกันไวรัส (Antivirus) โดยที่ระบบเหล่านี้ไม่สามารถตรวจจับหรือบล็อกการโจมตีได้ [11] เช่น มัลแวร์ที่สามารถเปลี่ยนแปลงโค้ดของตัวเอง (Mutate) โดยอัตโนมัติ เพื่อหลบเลี่ยงการตรวจจับจากซอฟต์แวร์ป้องกันไวรัส (Antivirus Software) และระบบรักษาความปลอดภัย (Plymorphic malware) เทคนิคในการแปลงหรือดัดแปลงโค้ดโปรแกรมให้มีลักษณะที่ซับซ้อน และอ่านยากเพื่อป้องกันการวิเคราะห์ (0bfuscate code) และการใช้ประโยชน์จากช่องโหว่ (Vulnerabilities) ในระบบคอมพิวเตอร์ซอฟต์แวร์ หรือเครือข่ายเพื่อเจาะระบบหรือโจมตีเป้าหมาย (Exploit vulnerabilities) ที่ยังไม่ถูกค้นพบ การเรียนรู้ของเครื่อง (Machine Learning: ML) เป็นเทคโนโลยีที่ถูกนำมาประยุกต์ใช้ในระบบตรวจจับการบุกรุก (Intrusion Detection System: IDS) เพื่อยกระดับความสามารถในการตรวจจับความผิดปกติหรือสิ่งที่แตกต่างจากรูปแบบปกติในข้อมูลหรือระบบ (Anomalies) และกิจกรรมที่เป็นอันตรายหรือมีเจตนาร้ายในระบบคอมพิวเตอร์เครือข่าย (Malicious activities) โดยอาศัยขั้นตอนวิธี (Algorithms) ที่สามารถเรียนรู้รูปแบบจากข้อมูลในอดีตที่ถูกเก็บรวบรวมและบันทึก (Historical data) และการจำแนกประเภทของทราฟฟิกในเครือข่าย (Classify network traffic) เป็นทราฟฟิกที่ปลอดภัย (Benign) หรือทราฟฟิกที่เป็นอันตราย (Malicious) [12] ML algorithms เช่น เครื่องเวกเตอร์สนับสนุน (Support Vector Machine: SVM) ต้นไม้ตัดสินใจ (Decision Tree) และป่าการตัดสินใจแบบสุ่ม (Random Forest) ได้ถูกนำมาใช้ใน IDS [13,14] อย่างแพร่หลาย อย่างไรก็ตาม โมเดลการเรียนรู้ของเครื่องแบบดั้งเดิม (Traditional ML models) อาจมีข้อจำกัดในการดึงความสัมพันธ์ของข้อมูลที่ขึ้นอยู่กับช่วงเวลา (Capture temporal dependencies) ในข้อมูลทราฟฟิกเครือข่าย (Network traffic data) ซึ่งเป็นสิ่งสำคัญในการตรวจจับการโจมตี (Detect attacks) ที่มีรูปแบบความซับซ้อนตามช่วงเวลา (Complex temporal patterns) [15] ตัวอย่างเช่น การโจมตีแบบปฏิเสธการให้บริการ (Denial-of-Service: DoS attack) อาจมีรูปแบบการส่งทราฟฟิกจำนวนมาก (Flood traffic pattern) ที่เปลี่ยนแปลงไปตามเวลา หรือการโจมตีแบบคนกลาง (Man-in-the-Middle attack) อาจมีรูปแบบการดักจับและแก้ไขข้อมูลที่ซับซ้อน (Intercept and modify data) การเรียนรู้เชิงลึก (Deep Learning: DL) เป็นสาขาย่อยของการเรียนรู้ของเครื่อง (Subfield of ML) ที่ได้รับความนิยมเพิ่มขึ้นในช่วงไม่กี่ปีที่ผ่านมา โมเดลการเรียนรู้เชิงลึก เช่น โครงข่ายประสาทเทียมเชิงลึก (Deep Neural Networks: DNNs) โครงข่ายประสาทเทียมเชิงคอนโวลูชัน (Convolutional Neural Networks: CNNs) และโครงข่ายประสาทเทียมเชิงซ้ำซ้อน (Recurrent Neural Networks: RNNs) ได้แสดงให้เห็นถึงประสิทธิภาพที่เหนือกว่าโมเดลการเรียนรู้ของเครื่องแบบดั้งเดิม (Traditional ML models) ในหลายงาน เช่น การรู้จำภาพ (Image recognition) การประมวลผลภาษาธรรมชาติ (Natural language processing) และการวิเคราะห์ข้อมูลแบบอนุกรมเวลา (Time-series

analysis) [16] โดยเฉพาะอย่างยิ่งโครงข่ายประสาทเทียมเชิงซ้ำซ้อน (RNNs) เช่นหน่วยความจำระยะยาวระยะสั้น (Long Short-Term Memory: LSTM) และหน่วยความจำแบบซ้ำซ้อน (Gated Recurrent Unit: GRU) ได้ถูกนำมาใช้เพื่อแก้ปัญหาการดึงความสัมพันธ์ของข้อมูลที่ขึ้นอยู่กับช่วงเวลาในระบบตรวจจับการบุกรุก (Capture temporal dependencies in IDS) เนื่องจาก RNNs สามารถเรียนรู้รูปแบบตามลำดับของข้อมูล (Learn temporal patterns from sequential data) [17] อย่างไรก็ตาม RNNs มีข้อจำกัดเรื่องการลดทอนของเกรเดียนท์ (Vanishing gradient) ซึ่งทำให้การฝึกฝนโมเดลที่มีลำดับข้อมูลยาวมากเป็นไปได้ยาก (Train models with long sequences) และความซับซ้อนในการคำนวณที่สูง (High computational complexity) ทำให้การนำไปใช้งานในอุปกรณ์ IoT ที่มีทรัพยากรจำกัด (Deploy in resource-constrained IoT devices) เป็นไปได้ยาก โครงข่ายประสาทเทียมเชิงอนุกรมเวลาคอนโวลูชัน (Temporal Convolutional Networks: TCN) เป็นโมเดลการเรียนรู้เชิงลึกที่ได้รับความนิยมในการวิเคราะห์ข้อมูลแบบอนุกรมเวลา (Time-series data analysis) [18] โดยมีสถาปัตยกรรมที่คล้ายกับโครงข่ายประสาทเทียมเชิงคอนโวลูชัน (Architecture similar to CNNs) แต่มีการปรับเปลี่ยนเพื่อให้สามารถดึงความสัมพันธ์ของข้อมูลที่ขึ้นอยู่กับช่วงเวลาได้อย่างมีประสิทธิภาพ (Modified to capture temporal dependencies effectively) [19] ข้อดีของ TCN ที่ทำให้โดดเด่นกว่า RNNs ได้แก่ความเร็วในการฝึกฝนที่รวดเร็วกว่า (Faster training speed) เนื่องจากสามารถประมวลผลข้อมูลแบบขนาน (Process data in parallel) [20] การเรียนรู้ลำดับข้อมูลระยะยาวได้ดีกว่า (Better long-term sequence learning) โดยใช้การขยายขนาดของคอนโวลูชัน (Dilated convolutions) [21] การใช้หน่วยความจำน้อยกว่า (Lower memory usage) เนื่องจากไม่ต้องเก็บสถานะภายใน (Hidden state) ของเครือข่าย [18] ด้วยคุณสมบัติเหล่านี้ TCN จึงเหมาะสมกับการตรวจจับการโจมตีทางไซเบอร์ (Detect cyber-attacks) ในเครือข่าย IoT ซึ่งมีข้อมูลแบบอนุกรมเวลาจำนวนมาก (Large time-series data) และต้องการโมเดลที่สามารถฝึกฝนได้รวดเร็ว และนำไปใช้ในอุปกรณ์ที่มีทรัพยากรจำกัดได้ (Fast-training models that can be deployed on resource-constrained devices)

งานวิจัยนี้มุ่งเน้นเพื่อพัฒนาเฟรมเวิร์คสำหรับตรวจจับพฤติกรรมการโจมตีบนเครือข่ายทุกสรรพสิ่งด้วย Temporal Convolutional Networks (TCN) และเพื่อประเมินประสิทธิภาพ และความแม่นยำของโมเดล Temporal Convolutional เพื่อหาวิธีที่มีประสิทธิภาพที่สุดสำหรับการป้องกันภัยคุกคามไซเบอร์ในยุคดิจิทัล โดยมุ่งหวังให้โมเดลสามารถตรวจจับพฤติกรรมที่เป็นอันตรายได้อย่างแม่นยำมีความสามารถในการประมวลผลแบบเรียลไทม์ และสามารถนำไปปรับใช้ในระบบที่มีทรัพยากรจำกัด เช่นเครือข่าย IoT และโครงสร้างพื้นฐานด้านความปลอดภัยทางไซเบอร์ในอนาคต

2. วัตถุประสงค์

2.1 เพื่อพัฒนาเฟรมเวิร์คสำหรับตรวจจับพฤติกรรมการโจมตีบนเครือข่ายทุกสรรพสิ่งด้วย Temporal Convolutional Networks (TCN)

2.2 เพื่อประเมินประสิทธิภาพของโมเดล TCN ในการตรวจจับพฤติกรรมการโจมตีโดยการเปรียบเทียบกับโมเดล RNN LSTM GRU และ DNN

3. ทฤษฎี และงานวิจัยที่เกี่ยวข้อง

3.1 Internet of Things (IoT) หรือ "อินเทอร์เน็ตในทุกสิ่ง" หมายถึง เครือข่ายของอุปกรณ์เครื่องจักรและเซ็นเซอร์ที่สามารถเชื่อมต่อกันผ่านอินเทอร์เน็ต [1] อุปกรณ์เหล่านี้สามารถรวบรวม วิเคราะห์ และแลกเปลี่ยนข้อมูลระหว่างกันได้โดยอัตโนมัติ [2] ปัจจุบัน IoT ถูกนำมาใช้ในหลากหลายแอปพลิเคชัน เช่น บ้านอัจฉริยะ (Smart Home) เมืองอัจฉริยะ (Smart City) ระบบขนส่งอัจฉริยะ (Smart Transportation) และระบบอัตโนมัติในภาคอุตสาหกรรม (Industrial Automation) [3] อย่างไรก็ตามการเติบโตอย่างรวดเร็วของ IoT นำมาซึ่งความท้าทายด้านความปลอดภัย [4] เนื่องจากอุปกรณ์ IoTจำนวนมากมีทรัพยากรจำกัด (Resource Constraints) และมีช่องโหว่ (Vulnerabilities) ที่สามารถถูกโจมตีหรือแสวงหาประโยชน์ได้ง่าย (Exploit) ทำให้อุปกรณ์ IoT กลายเป็นเป้าหมายสำคัญของภัยคุกคามทางไซเบอร์

3.2 การโจมตีแบบปฏิเสธการให้บริการ (Denial-of-Service: DoS) ภัยคุกคามในเครือข่าย IoT มีหลากหลายรูปแบบ เช่น การโจมตีแบบปฏิเสธการให้บริการ (Denial-of-Service: DoS) การละเมิดข้อมูล (Data Breaches) และการโจมตีโดยมัลแวร์ (Malware Attacks) [5] ช่องโหว่ด้านความปลอดภัยในอุปกรณ์ IoT อาจเกิดจากการใช้ข้อมูลประจำตัวเริ่มต้น (Default Credentials) การขาดการเข้ารหัสข้อมูล (Lack of Encryption) และกลไกการพิสูจน์ตัวตนที่ไม่แข็งแกร่งเพียงพอ (Weak Authentication Mechanisms) [6] การโจมตีเหล่านี้อาจส่งผลกระทบร้ายแรงต่อความเป็นส่วนตัวของข้อมูล (Privacy) ความสมบูรณ์ของข้อมูล (Data Integrity) และความพร้อมใช้งานของระบบ (System Availability) ในเครือข่าย IoT [7] โดยเฉพาะอย่างยิ่งการโจมตีแบบ DoS อาจทำให้อุปกรณ์ IoT ไม่สามารถให้บริการได้ตามปกติ ส่งผลต่อประสิทธิภาพและความน่าเชื่อถือของระบบอัจฉริยะในภาคส่วนต่าง ๆ เช่น โครงสร้างพื้นฐานด้านพลังงาน การแพทย์ และอุตสาหกรรม

3.3 ระบบตรวจจับการบุกรุก (Intrusion Detection System: IDS) เป็นซอฟต์แวร์หรือฮาร์ดแวร์ที่ใช้เฝ้าระวังและวิเคราะห์ทราฟฟิกเครือข่ายหรือกิจกรรมในระบบ เพื่อระบุพฤติกรรมที่เป็นอันตรายหรือการโจมตีทางไซเบอร์ [8] IDS สามารถแบ่งออกเป็นสองประเภทหลักตามวิธีการตรวจจับพฤติกรรมที่เบี่ยงเบนไปจากรูปแบบปกติ (Deviations from Normal Behavior) [9] ได้แก่ 1) Signature-based IDS ใช้การจับคู่รูปแบบ (Pattern Matching) กับลายเซ็นของการโจมตีที่รู้จัก (Known Signatures) เพื่อตรวจจับภัยคุกคาม เช่นมัลแวร์ หรือการโจมตีที่เคยถูกบันทึกไว้ในฐานข้อมูลความปลอดภัย 2) Anomaly-based IDS ใช้การวิเคราะห์พฤติกรรม (Behavioral Analysis) และการเรียนรู้ของเครื่อง (Machine Learning) เพื่อตรวจจับพฤติกรรมที่ผิดปกติในระบบ โดยไม่จำเป็นต้องอ้างอิงกับฐานข้อมูลของการโจมตีที่รู้จักมาก่อน ซึ่งช่วยให้สามารถตรวจจับ Zero-day Attacks และภัยคุกคามที่ไม่เคยพบมาก่อนได้ IDS ถือเป็นกลไกสำคัญในการเสริมสร้างความปลอดภัยของเครือข่าย IoT เนื่องจากสามารถเฝ้าระวังพฤติกรรมที่เป็นอันตราย

และแจ้งเตือนผู้ดูแลระบบให้ดำเนินการป้องกัน หรือบรรเทาผลกระทบได้อย่างทันท่วงที อย่างไรก็ตามระบบ IDS แบบเดิมยังมีข้อจำกัดในเรื่องของความแม่นยำ และการประมวลผลข้อมูลแบบเรียลไทม์ทำให้มีการพัฒนาระบบป้องกันการบุกรุกตรวจจับป้องกันภัยคุกคามทางไซเบอร์แบบเรียลไทม์ (Intrusion Prevention System :IPS) ซึ่งสามารถตรวจจับและบล็อกการโจมตีได้โดยอัตโนมัติเพื่อเพิ่มขีดความสามารถในการป้องกันภัยคุกคามทางไซเบอร์ในเครือข่าย IoT

3.4 การเรียนรู้เชิงลึก (Deep Learning: DL) เป็นสาขาย่อยของ Machine Learning ที่ใช้โครงข่ายประสาทเทียม (Artificial Neural Networks: ANN) เพื่อเรียนรู้รูปแบบข้อมูลที่ซับซ้อน [15] DL ถูกนำมาใช้ใน IDS เพื่อเพิ่มความสามารถในการตรวจจับพฤติกรรมผิดปกติ (Anomalies) และกิจกรรมที่เป็นอันตราย (Malicious Activities) ได้อย่างแม่นยำ [12] DL มีข้อได้เปรียบ IDS แบบดั้งเดิมเพราะสามารถเรียนรู้จากข้อมูล และตรวจจับภัยคุกคามที่ไม่เคยพบมาก่อน อย่างไรก็ตามความต้องการทรัพยากรสูง และความซับซ้อนของโมเดลยังคงเป็นความท้าทายที่ต้องได้รับการพัฒนาเพื่อให้สามารถใช้งานได้อย่างมีประสิทธิภาพในเครือข่าย IoT และระบบความปลอดภัยทางไซเบอร์

3.5 โครงข่ายประสาทเทียมเชิงเวลาคอนโวลูชัน (Temporal Convolutional Networks: TCN) โมเดลการเรียนรู้เชิงลึกที่เหมาะสำหรับการวิเคราะห์ข้อมูลแบบอนุกรมเวลา (Time-Series Data) [18] โครงสร้างของ TCN คล้ายกับโครงข่ายประสาทเทียมเชิงคอนโวลูชัน (Convolutional Neural Networks: CNNs) แต่ได้รับการปรับปรุงเพื่อให้สามารถเรียนรู้ความสัมพันธ์ของข้อมูลในช่วงเวลาต่าง ๆ (Capture Temporal Dependencies) ได้อย่างมีประสิทธิภาพ [19] ข้อดี TCN ที่มากกว่าโครงข่ายประสาทเทียมเชิงซ้ำซ้อน (Recurrent Neural Networks: RNNs) ได้แก่ ความเร็วในการฝึกฝนที่สูงกว่า การเรียนรู้ข้อมูลระยะยาวที่ดีกว่า และการใช้หน่วยความจำน้อยลง ทำให้ TCN เป็นทางเลือกที่เหมาะสมสำหรับงานที่ต้องการวิเคราะห์ข้อมูลลำดับเวลา เช่นการตรวจจับภัยคุกคามทางไซเบอร์ในเครือข่าย IoT

3.6 งานวิจัยการประยุกต์ใช้ TCN ในการพยากรณ์ข้อมูลพลังงาน โดย Lara-Benitez et al. ได้นำ TCN มาใช้พยากรณ์ความต้องการพลังงานไฟฟ้า โดยทำการเปรียบเทียบกับ โมเดล Long Short-Term Memory (LSTM) โดยแบบจำลองมาตรฐานในการพยากรณ์ข้อมูลชุดเวลาโดยใช้ข้อมูลพลังงานจากประเทศสเปน 2 ชุด ได้แก่ ความต้องการใช้ไฟฟ้าระดับประเทศ และการใช้พลังงานของสถานีชาร์จรถยนต์ไฟฟ้าสำหรับการศึกษาความสามารถของโมเดลที่แตกต่างกัน จากการทดลองพบว่า TCN สามารถให้ผลการพยากรณ์ที่แม่นยำกว่า LSTM ในทุกกรณี นอกจากนี้ TCN ยังสามารถจับรูปแบบของข้อมูลในระยะยาวได้ดีกว่า LSTM และใช้ทรัพยากรคอมพิวเตอร์น้อยลง [22]

4. วิธีดำเนินการวิจัย

ในงานวิจัยนี้ มีวิธีการดำเนินการวิจัยสำหรับพัฒนาเฟรมเวิร์คสำหรับตรวจจับพฤติกรรมการโจมตีบนเครือข่ายทุกสรรพสิ่งด้วย Temporal Convolutional Networks (TCN) และประเมินประสิทธิภาพของโมเดล TCN ในการตรวจจับพฤติกรรมการโจมตี โดยการเปรียบเทียบกับโมเดล RNN LSTM GRU และ DNN โดยมีวิธีการดำเนินการวิจัยประกอบด้วย 6 หัวข้อหลัก ดังนี้

4.1 การเก็บรวบรวมข้อมูล (Data Collection) เนื่องจากข้อจำกัดในการเข้าถึงข้อมูลการโจมตีในเครือข่าย IoT จริง งานวิจัยนี้จึงเลือกใช้ชุดข้อมูลสังเคราะห์ที่สร้างขึ้นโดยใช้ make classification` function จาก Scikit-learn library ใน Python ชุดข้อมูลนี้ประกอบด้วยข้อมูลอนุกรมเวลา (time-series data) ที่จำลองการรับส่งข้อมูลในเครือข่าย IoT โดยมี ตัวแปรและคุณลักษณะของข้อมูล (features) ที่หลากหลาย เช่น ปริมาณทราฟฟิก ความถี่ใน การส่งข้อมูล โปรโตคอล (protocol) ช่องทางสื่อสารระหว่างอุปกรณ์หรือโปรแกรมที่ทำงานบนเครือข่าย (port) และ IP address ชุดข้อมูลแบ่งออกเป็น 3 ส่วน คือ ข้อมูลปกติ ข้อมูลที่จำลองพฤติกรรมการโจมตีแบบการโจมตีทางไซเบอร์ที่มีเป้าหมายเพื่อทำให้ระบบเครือข่ายบริการ หรือเซิร์ฟเวอร์ไม่สามารถให้บริการได้ตามปกติ (Denial-of-Service :DoS) ซึ่งมีปริมาณข้อมูลมหาศาลที่ถูกส่งไปยังเซิร์ฟเวอร์ หรือเครือข่ายอย่างต่อเนื่อง (flood traffic) ไปยังอุปกรณ์เป้าหมาย และข้อมูลที่จำลองพฤติกรรมการโจมตีทางไซเบอร์ที่แฮกเกอร์แอบดักจับ และแก้ไขข้อมูลที่ถูกส่งระหว่างสองฝ่ายที่คิดว่ากำลังสื่อสารกันอย่างปลอดภัย (Man-in-the-Middle) ซึ่งมีลักษณะดักจับหรือขัดขวางข้อมูลที่ถูกส่งผ่านเครือข่ายโดยบุคคลที่สามที่ไม่ได้รับอนุญาตสามารถนำไปใช้เพื่อดักฟัง แก้ไข หรือขโมยข้อมูลที่กำลังถูกส่งระหว่างสองฝ่าย (Intercept) และปรับปรุงข้อมูลระหว่างผู้ใช้ (Client) และผู้ให้บริการ (Server) การใช้ชุดข้อมูลสังเคราะห์ช่วยให้สามารถควบคุมลักษณะ และปริมาณของข้อมูลรวมถึงรูปแบบการโจมตีได้ตามต้องการ

4.2 ชุดข้อมูลสังเคราะห์ (Dataset)

ตารางที่ 1 ชุดข้อมูลสังเคราะห์สร้างขึ้นโดยมี ตัวแปรที่ใช้ควบคุมหรือกำหนดพฤติกรรมของระบบ (Parameter)

Parameter	คำอธิบาย	ค่าที่ใช้
จำนวนอุปกรณ์ IoT (n_devices)	จำนวนอุปกรณ์ IoT ที่ใช้ในการจำลอง	10
ระยะเวลา (n_days)	ระยะเวลาที่ใช้ในการจำลอง	7 วัน
ความถี่ในการเก็บ ข้อมูล (frequency)	ความถี่ในการเก็บข้อมูล	1440 ครั้ง/วัน (1 นาที/ครั้ง)
คุณลักษณะ (Features)	คุณลักษณะของข้อมูล เช่น ปริมาณ traffic, ความถี่, protocol, port, IP address	5 features
รูปแบบการโจมตี	รูปแบบการโจมตีที่ใช้ในการจำลอง คือ DoS, Man-in-the-Middle	DoS, Man-in-the-Middle

4.3 เครื่องมือวิจัย (Research Instruments) งานวิจัยนี้เลือกใช้ภาษาไพทอล (Python Language) เป็นภาษาโปรแกรมหลักเนื่องจากมีความยืดหยุ่นสูงและมีไลบรารีสำหรับ Data Science และ Machine learning โดยใช้ Scikit-learn ในการสร้าง Dataset และ preprocess ข้อมูล TensorFlow/Keras ซึ่งใช้ในการสร้างและฝึกฝน TCN Model การทดลองดำเนินการบน Google Colab Platform ซึ่งเป็น Cloud platform เพื่อเร่งการฝึกฝนโมเดลใช้

Wireshark สำหรับ capture และ วิเคราะห์ network traffic ใช้ Nmap สำหรับ network scanning และ vulnerability analysis ตามความเหมาะสม

4.4 ขั้นตอนการพัฒนาโมเดล

4.4.1 กำหนดวัตถุประสงค์ของโมเดลโดยการระบุเป้าหมายของการพัฒนาโมเดล และกำหนดขอบเขตการทำงานของโมเดล

4.4.2 เตรียมชุดข้อมูล โดยการรวบรวมข้อมูลที่เกี่ยวข้อง จากนั้นทำความสะอาดข้อมูลและจัดการค่าที่ขาดหายไป และแปลงและเลือกฟีเจอร์ที่สำคัญ

4.4.3 แบ่งชุดข้อมูล ทำการแบ่งข้อมูลเป็น Training Set, Validation Set และ Test Set พร้อมทั้งตรวจสอบความสมดุลของข้อมูล

4.4.4 ออกแบบโครงสร้างของโมเดล เลือกประเภทของโมเดลที่เหมาะสม และกำหนดจำนวนชั้นของโมเดลและพารามิเตอร์ต่าง ๆ

4.4.5 เลือก Loss Function และ Optimizer ทำการกำหนด Loss Function ที่เหมาะสมกับปัญหาและเลือก Optimizer ที่ช่วยให้โมเดลเรียนรู้ได้อย่างมีประสิทธิภาพ

4.4.6 ฝึกโมเดล (Model Training) ทำการป้อนข้อมูลเข้าโมเดลเพื่อให้เรียนรู้ รวมถึงปรับพารามิเตอร์และค่า Hyperparameters เพื่อเพิ่มประสิทธิภาพ

4.4.7 ประเมินผลการฝึกโมเดล ทำการตรวจสอบค่า Accuracy, Precision, Recall และ F1-Scor จากนั้นวิเคราะห์ค่า Loss และ Validation Loss และเลือกใช้ Confusion Matrix และ ROC Curve เพื่อประเมินผล

4.4.8 ปรับปรุงและปรับแต่งโมเดล ปรับโครงสร้างของโมเดลเพื่อลด Overfitting โดยใช้เทคนิค Regularization หรือเพิ่มข้อมูลฝึก

4.4.9 ทดสอบโมเดลกับชุดข้อมูลที่ไม่เคยเห็นมาก่อน ทำการตรวจสอบความสามารถของโมเดลในการทำงานกับข้อมูลใหม่ พร้อมวิเคราะห์ข้อผิดพลาดและปรับปรุงโมเดลเพิ่มเติม

4.4.10 นำโมเดลไปใช้งาน (Deployment) บันทึกโมเดลที่ผ่านการฝึกแล้ว จากนั้นนำไปใช้งานในสภาพแวดล้อมจริงหรือสร้าง API สำหรับการใช้งาน

4.5 การประเมินประสิทธิภาพ และความแม่นยำ ประกอบด้วย ขั้นตอนดังต่อไปนี้

4.5.1 เตรียมชุดข้อมูลทดสอบ (Test Set Preparation) ประกอบด้วย การแบ่งข้อมูลออกเป็น Training Set Validation Set และ Test Set จากนั้นทำการตรวจสอบความสมดุลของข้อมูล (Data Imbalance)

4.5.2 คำนวณค่า Accuracy Precision Recall และ F1-Score ใช้สูตรในการคำนวณค่า Accuracy เพื่อวัดความแม่นยำของโมเดล คำนวณค่า Precision และ Recall เพื่อตรวจสอบอัตราการทำนายผิดพลาดและการตรวจจับที่ถูกต้อง และคำนวณค่า F1-Score เพื่อประเมินประสิทธิภาพโดยรวมของโมเดล

4.5.3 วิเคราะห์ค่า Loss และ Validation Loss ทำการตรวจสอบว่า Loss ลดลงอย่างต่อเนื่องหรือไม่ จากนั้นเปรียบเทียบระหว่าง Training Loss และ Validation Loss เพื่อดูว่ามี Overfitting หรือไม่

4.5.4 วิเคราะห์ผลลัพธ์ผ่าน Confusion Matrix ตรวจสอบค่าที่โมเดลทำนายถูกต้องและผิดพลาด และวิเคราะห์อัตราการเกิด False Positives และ False Negatives

4.5.6 สร้าง ROC Curve และคำนวณ AUC Score วัดความสามารถของโมเดลในการแยกแยะระหว่างคลาส พร้อมทั้งคำนวณค่า AUC (Area Under Curve) เพื่อดูประสิทธิภาพของโมเดล

4.5.7 ตรวจสอบแนวโน้มของค่า Accuracy และ Loss ตลอดการฝึกโมเดล ใช้กราฟแสดงผล Accuracy กับ Epochs และ Loss กับ Epochs และวิเคราะห์แนวโน้มว่าค่า Accuracy เพิ่มขึ้น และค่า Loss ลดลงหรือไม่

4.5.8 เปรียบเทียบผลลัพธ์กับโมเดลอื่น ๆ นำค่า Accuracy Precision Recall และ F1-Score มาเปรียบเทียบกับโมเดลอื่น เช่น RNN LSTM GRU และ DNN จากนั้นวิเคราะห์ข้อดีและข้อเสียของแต่ละโมเดล

4.5.9 ปรับปรุงโมเดลเพื่อเพิ่มประสิทธิภาพ ปรับพารามิเตอร์ของโมเดล เช่น Learning Rate Batch Size และจำนวน Epochs ใช้เทคนิค Regularization เช่น Dropout เพื่อลด Overfitting และปรับปรุงโครงสร้างของโมเดลเพื่อเพิ่มประสิทธิภาพในการเรียนรู้

4.6 การวิเคราะห์ข้อมูล ประกอบด้วย

4.6.1 เตรียมข้อมูล โดยการตรวจสอบค่าที่ขาดหายไป ปรับขนาดข้อมูล และแบ่งชุดข้อมูล

4.6.2 สร้างชุดข้อมูลสังเคราะห์ จำลองพฤติกรรมปกติ และการโจมตี เช่น DoS และ MITM

4.6.3 วิเคราะห์เชิงพรรณนา คำนวณสถิติเบื้องต้น เช่น ค่าเฉลี่ย และการกระจายตัวของข้อมูล

4.6.4 วิเคราะห์ความสัมพันธ์ของตัวแปร ใช้ Correlation Matrix และ Feature Selection

4.6.5 ประเมินโมเดล คำนวณ Accuracy, Precision, Recall, F1-Score และ Loss

4.6.6 วิเคราะห์ Confusion Matrix ตรวจสอบอัตราการเกิดข้อผิดพลาด (False Positives False Negatives)

4.6.7 ปรับปรุงโมเดล วิเคราะห์แนวโน้มการเรียนรู้ และปรับพารามิเตอร์ให้เหมาะสม

4.6.8 เปรียบเทียบกับโมเดลอื่น วิเคราะห์ประสิทธิภาพของ TCN เทียบกับ RNN LSTM GRU และDNN

5. ผลและวิจารณ์

5.1 ผลการทดลองการพัฒนาเฟรมเวิร์คสำหรับตรวจจับพฤติกรรมการโจมตีบนเครือข่ายทุกสรรพสิ่งด้วย Temporal Convolutional Networks (TCN) เป็นไปดังตารางที่ 2

ตารางที่ 2 ผลการทดลองที่ได้รับจากการ Train TCN Model

รอบการฝึกโมเดล (Epoch)	ความแม่นยำของโมเดลจากชุดข้อมูลฝึก (Accuracy)	ค่าความผิดพลาดของโมเดลจากชุดข้อมูลฝึก (Loss)	ความแม่นยำของโมเดลจากชุดข้อมูลตรวจสอบ (Val_Accuracy)	ค่าความผิดพลาดของโมเดลจากชุดข้อมูลตรวจสอบ (ValLoss)	อัตราการเรียนรู้ของโมเดล (Learning Rate)	เวลาที่ใช้ในการฝึกแต่ละรอบ (Time/Epoch)
1	0.5814	1.4105	0.5096	0.5096	0.01	29s
2	0.5299	2.8932	0.5008	2.0804	0.01	29s
3	0.4998	1.8371	0.5013	0.7905	0.01	42s
4	0.5007	2.9269	0.4984	2.9081	0.01	27s
5	0.4995	1.3288	0.4992	2.0398	0.01	27s
6	0.4992	1.4011	0.5023	0.7518	0.01	28s
7	0.4998	0.9236	0.4977	0.7134	0.01	30s
8	0.5000	1.5425	0.4987	2.5159	0.01	28s
9	0.5021	1.3080	0.5021	0.9836	0.01	41s
10	0.5009	1.0523	0.5020	1.7645	0.01	41s

จากตารางที่ 2 ผลการทดลองแสดงให้เห็นว่า TCN Model ที่พัฒนาขึ้นยังไม่สามารถเรียนรู้รูปแบบของข้อมูลได้อย่างมีประสิทธิภาพโดยมีรายละเอียดดังนี้ ค่า Accuracy ต่ำ โดยค่า Accuracy ของโมเดลอยู่ที่ประมาณ 0.49 - 0.50 ซึ่งถือว่าต่ำกว่าเกณฑ์มาตรฐานที่ยอมรับได้สำหรับงานตรวจจับการโจมตีทางไซเบอร์ (Cyber Attack) โดยทั่วไปแล้วค่า Accuracy ควรอยู่ที่ 80% ขึ้นไปเพื่อให้มั่นใจได้ว่า Model มีความน่าเชื่อถือในการใช้งานจริง

ค่า Loss ผันผวน ค่า Loss ในตารางที่ 2 มีความผันผวนสูงในแต่ละรอบการฝึก Epoch ชี้ให้เห็นว่าโมเดลยังไม่สามารถเรียนรู้จนถึงจุดที่ค่าความผิดพลาด (Loss) ลดลงอย่างมีเสถียรภาพ (Converge) และไม่เปลี่ยนแปลงมากนักเมื่อฝึกต่อไป (Converge) และอาจเกิดปัญหาภาวะที่โมเดลเรียนรู้ข้อมูลฝึก ได้ดีเกินไป จนไม่สามารถทำงานได้ดีกับข้อมูลใหม่ (Overfitting) ซึ่งส่งผลให้โมเดลมีประสิทธิภาพต่ำในการทำงานกับข้อมูลใหม่โดยยังคงให้ผลลัพธ์ที่แม่นยำ (Generalize)

ค่า Learning Rate ที่ใช้ในการฝึกโมเดลสูงเกินไป ทำให้โมเดลไม่สามารถปรับพารามิเตอร์ได้อย่างเหมาะสมและพลาดจุดที่ให้ผลลัพธ์ดีที่สุด (Optimal Point) ในการเรียนรู้

ตารางที่ 3 Confusion Matrix แสดงประสิทธิภาพของโมเดลการเรียนรู้เชิงลึก (Deep Learning) และการเรียนรู้ของเครื่อง (Machine Learning) โดยใช้แสดงผลลัพธ์ของการจำแนกประเภท (Classification) เปรียบเทียบระหว่างค่าจริง (Actual) และค่าที่โมเดลทำนาย (Predicted)

ทำนายว่าไม่ใช่การโจมตี (Predicted Negative)	ทำนายว่าเป็นการโจมตี (Predicted Positive)
108	60,648
85	60,119

จากตารางที่ 3 พบว่า โมเดลมีข้อผิดพลาดในการจำแนกพฤติกรรมการโจมตีในเครือข่าย IOT แสดงค่าทำนายว่าไม่ใช่การโจมตี (Predicted Negative) และทำนายว่าเป็นการโจมตี (Predicted Positive) โดยโมดลสามารถทำนายค่าที่ไม่ใช่การโมตี ได้ถูกต้อง 108 แต่กลับทำนายผิดพลาดว่าเป็นการโจมตี มากถึง 60,648 ครั้ง ซึ่งหมายถึงว่าโมเดลมีอัตราการแจ้งเตือนผิดพลาดสูงมาก (False Positive) ในทางกลับกัน โมเดลสามารถทำนายว่าเป็นการโจมตีได้ถูกต้อง 60,119 ครั้ง แต่ยังมีการทำนายว่าไม่ใช่การโจมตี อยู่ 85 ครั้ง แสดงให้เห็นว่าโมดลมีความสามารถในการตรวจจับการโจมตีได้ดี (พิจารณาค่า Recall สูง) แต่มีปัญหาในด้านความแม้นยำของการแจ้งเตือน(Precision ต่ำ) ซึ่งแสดงให้เห็นว่า โมเดลควรได้รับการปรับปรุงเพื่อเพิ่มความแม่นยำในการตรวจจับพฤติกรรมการโจมตี และลดอัตราการแจ้งเตือนผิดพลาด และทำให้โมเดลสามารถใช้งานจริงได้อย่างมีประสิทธิภาพมากขึ้น

ตารางที่ 4 ผลคำนวณการฝึกโมเดล TCN

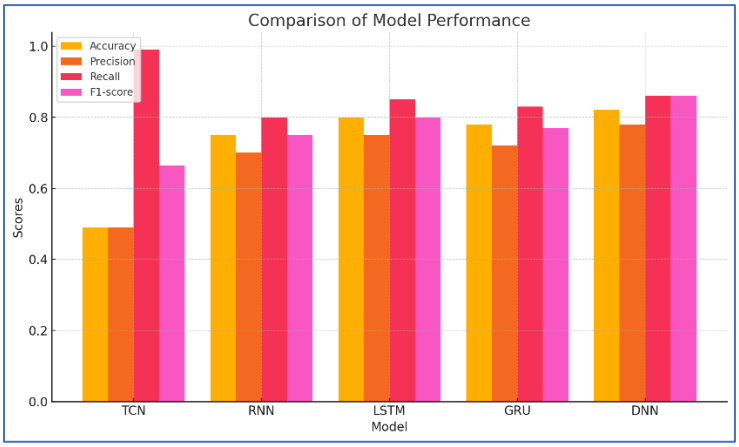
Metric	Value
Accuracy	0.4979
Precision	0.4978
Recall	0.9986
F1-score	0.6644

จากตารางที่ 4 ผลการคำนวณการฝึกโมเดล TCN พบว่า มีความแม่นยำไม่เพียงพอสำหรับการนำไปใช้งานจริง พิจารณา ค่า Accuracy เท่ากับ 0.4979 สามารถทำนายได้ถูกต้องเพียงประมาณ 49.79 % แสดงว่าค่าต่ำกว่ามาตรฐานที่เหมาะสม ค่า Precision เท่ากับ 0.4978 แสดงว่าโมเดลมีอัตราการทำนายว่ามีการโจมตี (Positive) ได้ถูกต้องเพียง 49.78 % ซึ่งหมายความว่ามี False Positive สูง ค่า Recall เท่ากับ 0.9986 แสดงว่า โมเดลสามารถตรวจจับเหตุการณ์ที่โจมตีที่เกิดขึ้นจริงได้ถึง 99.86 % ซึ่งเป็นค่าสูง แสดงว่าโมเดลสามารถตรวจจับการโจมตีทางไซเบอร์ได้แทบทุกกรณี ค่า F1-score เท่ากับ 0.6644 ซึ่งเป็นค่าเฉลี่ยระหว่างค่า Precision และค่า Recall แสดงให้เห็นว่าโมเดลต้องปรับปรุงให้สามารถจำแนกได้แม่นยำขึ้น โมเดลTCN มีความสามารถในการตรวจกับภัยคุกคามทางไซเบอร์ได้ดี เนื่องจากมีค่า Recall ที่สูง แต่ยังมีปัญหาในด้านของ Precision ต่ำ และAccuracyต่ำ หมายความว่า โมเดลมีอัตราการแจ้งเตือน

ข้อผิดพลาด (False Positive) สูง จำเป็นต้องปรับปรุงให้สามารถแยกแยะพฤติกรรมที่เป็นอันตรายและไม่เป็นอันตรายให้ดีขึ้น เช่น การปรับพารามิเตอร์ของโมเดลหรือเพิ่มคุณภาพของข้อมูลฝึก

5.2 ประเมินประสิทธิภาพของโมเดล TCN ในการตรวจจับพฤติกรรมการโจมตี โดยการเปรียบเทียบกับโมเดล RNN LSTM GRU และDNN

ผลการเปรียบเทียบประสิทธิภาพของ TCN Model กับ Model ที่นิยมใช้ในการตรวจจับ Cyber Attack จำนวน 4 โมเดล ได้แก่ RNN, LSTM, GRU หรือ DNN โดยใช้ชุดข้อมูลเดียวกัน และเปรียบเทียบผลลัพธ์โดยใช้ค่า Accuracy Precision Recall F1-score และ Confusion Matrix เป็นเกณฑ์ในการวัดผล โดยแสดงการเปรียบเทียบในกราฟ รูปที่ 1



รูปที่ 1 แสดงการเปรียบเทียบประสิทธิภาพ ของ TCN Model กับ Model อื่น ๆ

จากรูปที่ 1 จากกราฟแสดงให้เห็นถึงว่า โมเดล TCN มีค่า Recall สูงที่สุด แต่ค่า Precision และ Accuracy ต่ำ ทำให้เหมาะกับงานที่เน้นการตรวจจับทุกกรณีสำคัญ สำหรับโมเดล RNN มีผลลัพธ์สมดุลเหมาะกับงานทั่วไป โมเดล LSTM ประสิทธิภาพสูงในทุกด้านเหมาะสำหรับข้อมูลที่มีลำดับซับซ้อน โมเดล GRU ใกล้เคียงกับ LSTM แต่เหมาะกับงานที่ต้องการประสิทธิภาพสูง และโมเดล DNN มีความสมดุลและประสิทธิภาพสูงสุดในทุกด้านเหมาะกับงานที่ต้องการความแม่นยำและความครอบคลุมสูง

ประสิทธิภาพการนำเครื่องมือเหล่านี้มาประยุกต์ใช้จะช่วยพัฒนาโมเดลที่มีประสิทธิภาพสูง สำหรับการตรวจจับภัยคุกคามทางไซเบอร์ได้อย่างมีประสิทธิภาพยิ่งขึ้น ซึ่งจะส่งผลดีต่อการวิจัยและพัฒนา ระบบรักษาความปลอดภัยในอนาคตการนำเสนอผลการเรียนรู้ของโมเดล และการใช้เครื่องมือช่วยในการพัฒนาโมเดลเป็นขั้นตอนสำคัญในการสร้างโมเดลสำหรับงานตรวจจับไซเบอร์โดยการวิเคราะห์กราฟการฝึกฝนและการเลือกใช้ Framework ที่เหมาะสมจะช่วยให้สามารถเข้าใจและปรับปรุงประสิทธิภาพของโมเดลได้อย่างมีประสิทธิภาพซึ่งจะนำไปสู่การพัฒนาระบบรักษาความปลอดภัยที่มีประสิทธิภาพสูงขึ้นในอนาคต

6. สรุปผล

การพัฒนาเฟรมเวิร์คโดยใช้ Temporal Convolutional Networks (TCN) ซึ่งเป็นโมเดล Deep Learning ที่เหมาะกับการวิเคราะห์ข้อมูลแบบอนุกรมเวลา (Time-Series Data) สำหรับการตรวจจับพฤติกรรมการโจมตีในเครือข่าย IoT พบว่า เฟรมเวิร์คนี้ใช้ชุดข้อมูลสังเคราะห์ที่จำลองการโจมตีแบบ Denial-of-Service (DoS) และ Man-in-the-Middle (MITM) โดยใช้ Python และ TensorFlow/Keras บน Google Colab โดยโมเดลถูกฝึกฝนด้วยชุดข้อมูลที่ผ่านการ Standardization One-Hot Encoding และ Feature Engineering เพื่อให้สามารถเรียนรู้พฤติกรรมปกติและพฤติกรรมการโจมตี ผลการทดลองพบว่า โมเดล TCN ยังไม่สามารถเรียนรู้รูปแบบของข้อมูลได้อย่างมีประสิทธิภาพ ค่า Accuracy อยู่ที่ประมาณ 0.49 - 0.50 ซึ่งต่ำกว่าเกณฑ์มาตรฐาน (≥ 80%) สำหรับการใช้งานจริง สำหรับ ค่า Loss มีความผันผวนสูงในแต่ละ Epoch แสดงให้เห็นว่าโมเดลยังไม่สามารถเรียนรู้ได้อย่างมีเสถียรภาพ และอาจเกิดปัญหา Overfitting ทำให้ประสิทธิภาพลดลงเมื่อใช้กับข้อมูลใหม่ รวมถึงค่า Learning Rate สูงเกินไป ทำให้โมเดลไม่สามารถปรับค่าพารามิเตอร์ได้อย่างเหมาะสม ส่งผลให้พลาดจุดที่ให้ผลลัพธ์ดีที่สุด (Optimal Point)

การประเมินประสิทธิภาพของโมเดล TCN ในการตรวจจับพฤติกรรมการโจมตีโดยการเปรียบเทียบกับโมเดล RNN, LSTM, GRU และ DNN โดยใช้ Accuracy Precision Recall F1-score และ Confusion Matrix เป็นเกณฑ์วัดผล สามารถสรุปได้ว่า TCN มีค่า Recall สูงสุด (0.9986) แสดงว่าโมเดลสามารถตรวจจับเหตุการณ์โจมตีได้แทบทุกกรณี Precision ต่ำมาก (0.4978) ซึ่งหมายถึงอัตราการแจ้งเตือนผิดพลาด (False Positive) สูง Accuracy เพียง 0.4979 ทำให้ไม่สามารถใช้งานจริงได้อย่างมีประสิทธิภาพ F1-score อยู่ที่ 0.6644 แสดงถึงความไม่สมดุลของ Precision และ Recall เมื่อ Confusion Matrix พบว่า โมเดลมีอัตราการแจ้งเตือนผิดพลาดสูงมาก (False Positive = 60,648 ครั้ง) แม้จะสามารถตรวจจับการโจมตีได้แม่นยำ (True Positive = 60,119 ครั้ง) แต่ยังคงมีปัญหาในการแยกแยะระหว่างพฤติกรรมปกติและพฤติกรรมโจมตีแต่เมื่อเทียบกับโมเดลอื่น ๆ LSTM และ GRU มีประสิทธิภาพดีในทุกด้าน โดยเหมาะกับข้อมูลที่มีลำดับซับซ้อน DNN มีความสมดุลและมีประสิทธิภาพสูงสุดสำหรับงานที่ต้องการความแม่นยำ RNN มีผลลัพธ์สมดุล เหมาะสำหรับงานทั่วไป และTCN เหมาะกับงานที่ต้องการตรวจจับทุกกรณีสำคัญ แต่ยังต้องปรับปรุงเรื่อง Precision และ Accuracy

7. ข้อเสนอแนะ

เพื่อเพิ่มประสิทธิภาพของ framework ในการตรวจจับพฤติกรรมการโจมตีในเครือข่าย IoT ควรปรับปรุงคุณภาพของชุดข้อมูลโดยใช้ Dataset จริงหรือ Dataset สังเคราะห์ที่มีความซับซ้อนมากขึ้น พร้อมทั้งเพิ่มขนาดข้อมูลกำจัด noise และ outliers และใช้ Data Augmentation เพื่อเพิ่มความหลากหลายของข้อมูล นอกจากนี้ควรปรับปรุงโครงสร้างของ TCN Model โดยทดลองเปลี่ยน จำนวนชั้น (layers) kernel size และactivation functions รวมถึงใช้ Attention Mechanism เพื่อให้โมเดลสามารถโฟกัสกับลักษณะสำคัญของข้อมูล

ในส่วนของการปรับแต่งค่าพารามิเตอร์ (Hyperparameter Tuning) ควรลด Learning Rate ปรับ Batch Size และ Epochs ให้เหมาะสม และใช้ Regularization Techniques เช่น Dropout หรือ L1/L2 Regularization เพื่อลด Overfitting สุดท้าย ควรทดสอบโมเดลกับ ชุดข้อมูลจริง, วิเคราะห์ข้อผิดพลาดผ่าน Confusion Matrix และใช้ Ensemble Learning โดยรวมโมเดลอื่น เช่น LSTM หรือ GRU เพื่อเพิ่มความแม่นยำและลดอัตราการแจ้งเตือนผิดพลาด ซึ่งจะช่วยให้เฟรมเวิร์คมีประสิทธิภาพและสามารถนำไปใช้งานจริงได้อย่างมีประสิทธิภาพยิ่งขึ้น

8. เอกสารอ้างอิง

[1] Atzori, L., Iera, A., & Morabito, G. (2010). The internet of things: A survey. *Computer networks*, 54(15), 2787-2805.

[2] Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys & Tutorials*, 17(4), 2347-2376.

[3] Miorandi, D., Sicari, S., De Pellegrini, F., & Chlamtac, I. (2012). Internet of things: Vision, applications and research challenges. *Ad Hoc Networks*, 10(7), 1497-1516.

[4] Sicari, S., Rizzardi, A., Grieco, L. A., & Coen-Porisini, A. (2015). Security, privacy and trust in Internet of Things: The road ahead. *Computer Networks*, 76, 146-164.

[5] Stergiou, C., Psannis, K. E., Kim, B. G., & Gupta, B. B. (2020). Secure integration of IoT and cloud computing. *Future Generation Computer Systems*, 107, 964-975.

[6] Su, S., Tian, H., Huang, H., & Tong, X. (2019). IoT security and privacy: Challenges and solutions. *Journal of Network and Computer Applications*, 144, 101-109.

[7] Bhuyan, M. H., Bhattacharyya, D. K., & Kalita, J. K. (2014). Network anomaly detection: Methods, systems and tools. *IEEE Communications Surveys & Tutorials*, 16(1), 303-336.

[8] Axelsson, S. (2000). Intrusion detection systems: A survey and taxonomy. *Technical report*, 99(15), 1-39.

[9] Debar, H., Dacier, M., & Wespi, A. (1999). Towards a taxonomy of intrusion-detection systems. *Computer Networks*, 31(8), 805-822.

[10] Sommer, R., & Paxson, V. (2010). Outside the closed world: On using machine learning for network intrusion detection. *2010 IEEE symposium on security and privacy*, 305-316.

[11] Buczak, A. L., & Guven, E. (2016). A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys & Tutorials*, 18(2), 1153-1176.

[12] Xin, Y., Kong, L., Liu, Z., Chen, Y., Li, Y., Zhu, H., ... & Wang, C. (2018). Machine learning and deep learning methods for cybersecurity. *IEEE Access*, 6, 35365-35381.

[13] Ingre, B., & Yadav, A. (2015). *Performance analysis of NSL-KDD dataset using ANN*. 2015 International Conference on Signal Processing and Communication Engineering Systems (SPACES), 92-96.

[14] Moustafa, N., Adi, E., Turnbull, B., & Hu, J. (2018). *Deep learning for anomaly detection and classification in IoT sensors*. 2018 IEEE International Conference on Big Data (Big Data), 3333-3342.

[15] Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018). *Toward generating a new intrusion detection dataset and intrusion traffic characterization*. 4th international conference on information systems security and privacy (ICISSP), 108-116.

[16] Li, Z., Qin, Z., Huang, K., Ye, X., & Ye, D. (2019). Intrusion detection system based on convolutional neural networks and gated recurrent unit. *IEEE Access*, 7, 138175-138188.

[17] Yin, C., Zhu, Y., Fei, J., & He, X. (2017). A deep learning approach for intrusion detection using recurrent neural networks. *IEEE Access*, 5, 21954-21961.

[18] Bai, S., Kolter, J. Z., & Koltun, V. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint*, arXiv:1803.01271.

[19] Lea, C., Vidal, R., Reiter, A., & Hager, G. D. (2016). Temporal convolutional networks: A unified approach to action segmentation. European conference on computer vision, 473-487.

[20] van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., ... & Kavukcuoglu, K. (2016). Wavenet: A generative model for raw audio. *arXiv preprint*, arXiv:1609.03499.

[21] Yu, F., & Koltun, V. (2015). Multi-scale context aggregation by dilated convolutions. *arXiv preprint*, arXiv:1511.07122.

[22] Lara-Benítez, P., Carranza-García, M., Luna-Romera, J. M., & Riquelme, J. C. (2020). Temporal Convolutional Networks Applied to Energy-Related Time Series Forecasting. *Applied Sciences,* 10(7), 2322.