โครงงานเลขที่ วศ.คพ. S808-1/2568

เรื่อง

ระบบตรวจจับภัยคุกคามในเทคโนโลยีปฏิบัติการ (OT) สำหรับโรงงานอุตสาหกรรม

โดย

นายธนกฤต บุญยัง รหัส **650612084**
นายศุภกร สุวรรณภพ รหัส **650612100**
นายอดิศร สันเจริญ รหัส **650612104**

รายงานนี้เป็นส่วนหนึ่งของวิชาสำรวจเพื่อโครงงาน
ตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่
ปีการศึกษา 2568

PROJECT No. CPE S808-1/2568

Anomaly Detection in Operational Technology for Industrial Control Systems

Tanakrit boonyoung 650612084
Suppakorn suwannapop 650612100
Adisorn sancharoen 650612104

A Report Submitted in Partial Fulfillment of Project Survey Course
as Required by the Degree of Bachelor of Engineering

Department of Computer Engineering
Faculty of Engineering
Chiang Mai University

2025

หัวข้อโครงงาน	: ระบบตรวจจับภัยคุกคามในเทคโนโลยีปฏิบัติการ (OT) สำหรับโรงงานอุตสาหกรรม	
	: Anomaly Detection in Operational Technology for Industrial Control Systems	
โดย	: นายธนกฤต บุญยัง	รหัส 650612084
	นายศุภกร สุวรรณภพ	รหัส 650612100
	นายอดิศร สันเจริญ	รหัส 650612104
ภาควิชา	: วิศวกรรมคอมพิวเตอร์	
อาจารย์ที่ปรึกษา	: ผศ.ดร. กำพล วรดิษฐ์	
ปริญญา	: วิศวกรรมศาสตรบัณฑิต	
สาขา	: วิศวกรรมคอมพิวเตอร์	
ปีการศึกษา	: 2568	

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่ ได้อนุมัติให้โครงงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต (สาขาวิศวกรรมคอมพิวเตอร์)

..... หัวหน้าภาควิชาวิศวกรรมคอมพิวเตอร์
(รศ.ดร. สันติ พิทักษ์กิจนุกูร)

คณะกรรมการสอบโครงงาน

..... ประธานกรรมการ
(ผศ.ดร. กำพล วรดิษฐ์)

..... กรรมการ
(ผศ.ดร. เกษมสิทธิ์ ตียพันธ์)

..... กรรมการ
(อ.นพรุจ ซื่อตรง)

สารบัญ

บทที่ 1
บทนำ

1.1 ที่มาของโครงงาน

ในปัจจุบันโรงงานอุตสาหกรรมจำนวนมากได้ยกระดับกระบวนการผลิตให้มีความเป็นอัตโนมัติสูงขึ้น โดยอาศัยระบบควบคุมอุตสาหกรรม (Industrial Control Systems: ICS) ที่ทำงานร่วมกับระบบเทคโนโลยีปฏิบัติการ (Operational Technology: OT) ซึ่งเชื่อมโยงเซนเซอร์และแอกชูเอเตอร์เข้ากับเครื่องจักรจริง อย่างไรก็ตาม การที่ระบบ OT ถูกเชื่อมต่อเข้ากับเครือข่ายดิจิทัลมากขึ้น ย่อมทำให้ความเสี่ยงจากภัยคุกคามทางไซเบอร์เพิ่มสูงขึ้นตามไปด้วย การโจมตีดังกล่าวอาจทำให้ข้อมูลสูญหายหรือบิดเบือน รวมถึงก่อให้เกิดความเสียหายต่อเครื่องจักร กระบวนการผลิต และความปลอดภัยของบุคลากรได้โดยตรง ดังนั้นโครงงานนี้จึงถูกริเริ่มขึ้นเพื่อพัฒนากลไกการตรวจจับพฤติกรรมที่ผิดปกติในระบบ OT โดยใช้เทคนิคการเรียนรู้ของเครื่อง (Machine Learning) เพื่อเสริมสร้างความปลอดภัยและลดความเสียหายที่อาจเกิดขึ้นในโรงงานอุตสาหกรรม

1.2 วัตถุประสงค์ของโครงงาน

1. ออกแบบและพัฒนาโมเดลตรวจจับความผิดปกติจากข้อมูลเซนเซอร์และแอกชูเอเตอร์ในระบบ OT โดยใช้การวิเคราะห์เชิงเวลา (time-series analysis) ร่วมกับโมเดลเชิงลึก เช่น CNN และ LSTM
2. ระบุเหตุการณ์ที่มีลักษณะเป็นการโจมตีทางไซเบอร์หรือพฤติกรรมที่ผิดปกติในกระบวนการผลิต
3. พัฒนาต้นแบบ (prototype) ที่สามารถนำไปประยุกต์ใช้กับสภาพแวดล้อมในโรงงานจริงได้

1.3 ขอบเขตของโครงงาน

1.3.1 ขอบเขตด้านฮาร์ดแวร์

1. โครงงานนี้ ไม่ได้มุ่งเน้นการพัฒนาหรือใช้งานฮาร์ดแวร์จริง เช่น PLC, ปั๊ม, หรือวาล์ว
2. ส่วนประกอบเหล่านี้ถูก จำลองผ่านข้อมูลจาก SWaT dataset เท่านั้น
3. การดำเนินงานทั้งหมดจึงมุ่งเน้นที่การวิเคราะห์เชิงข้อมูลและการออกแบบโมเดล Machine Learning โดยไม่เกี่ยวข้องกับการสร้างหรือทดสอบระบบฮาร์ดแวร์จริง

1.3.2 ขอบเขตด้านซอฟต์แวร์

การเก็บและใช้ข้อมูล (Data Collection)

1. ใช้ชุดข้อมูล SWaT Dataset (Secure Water Treatment) ซึ่งเป็นข้อมูลจากระบบจำลองโรงงานน้ำประปาที่ใช้กันอย่างแพร่หลายในการศึกษาด้านความปลอดภัยของระบบ ICS/OT
2. ข้อมูลประกอบด้วยค่าการทำงานของเซนเซอร์และแอกชูเอเตอร์ที่มีทั้งสภาวะปกติและสภาวะถูกโจมตี
3. การนำเข้าข้อมูลจะใช้วิธีการอ่านไฟล์ CSV และเตรียมข้อมูลให้อยู่ในรูปแบบที่เหมาะสมต่อการประมวลผล

การตรวจจับและจำแนกความผิดปกติ (Threat Detection & Classification)

1. มุ่งเน้นการตรวจจับเหตุการณ์ที่มีลักษณะผิดปกติ เช่น การเปลี่ยนค่าของเซนเซอร์อย่างผิดธรรมชาติ หรือการสั่งการอุปกรณ์ที่ไม่สอดคล้องกับกระบวนการจริง
2. การจำแนกความผิดปกติออกเป็นสองกลุ่มหลัก ได้แก่
   (a) เหตุการณ์ปกติ (Normal events)
   (b) เหตุการณ์โจมตี/ผิดปกติ (Anomaly/Attack events)
3. การสร้าง Label จะอ้างอิงจากช่วงเวลาที่กำหนดไว้ในเอกสาร SWaT dataset

การประมวลผลและจัดการข้อมูล (Data Implementation)

1. การทำความสะอาดข้อมูล (data preprocessing) เช่น การจัดการค่าที่หายไป, การแทนค่าผิดปกติ, และการ normalize ข้อมูลให้อยู่ในสเกลเดียวกัน
2. การแบ่งข้อมูลออกเป็น Training set, Validation set และ Test set โดยอ้างอิงตามลำดับเวลาเพื่อเลี่ยงการรั่วไหลของข้อมูล (data leakage)
3. การสร้าง sliding windows สำหรับข้อมูลเชิงเวลา (time-series) เพื่อเตรียมให้เป็นอินพุตของโมเดล

การสร้างโมเดลและเปรียบเทียบประสิทธิภาพ (Model Comparison & Performance Evaluation)

1. พัฒนาโมเดลตรวจจับความผิดปกติที่อิงกับ Deep Learning ได้แก่
   (a) Convolutional Neural Network (CNN)
   (b) Long Short-Term Memory (LSTM)
2. ทดสอบโมเดลหลายรูปแบบ เช่น CNN 1D สำหรับจับ pattern ตามลำดับเวลา และ LSTM สำหรับตรวจจับ dependency ระหว่างข้อมูลในช่วงยาว
3. ประเมินผลลัพธ์ด้วยตัวชี้วัดมาตรฐาน
   (a) Accuracy, Precision
   (b) Recall, F1-score
   (c) Confusion Matrix
   (d) AUC-PR/ROC
4. เปรียบเทียบประสิทธิภาพของแต่ละโมเดลเพื่อหาวิธีที่เหมาะสมที่สุดต่อการใช้งาน

ผลลัพธ์ที่คาดว่าจะได้รับ (Expected Outcomes)

1. ได้ต้นแบบ (Prototype) ของระบบตรวจจับความผิดปกติในข้อมูลจากระบบ ICS/OT
2. โมเดลที่พัฒนาแล้วสามารถแยกความแตกต่างระหว่างเหตุการณ์ปกติและเหตุการณ์ผิดปกติได้ในระดับที่แม่นยำ
3. ได้ชุดข้อมูลและโค้ดที่สามารถนำไปปรับใช้หรือต่อยอดในงานวิจัยด้านความปลอดภัยไซเบอร์สำหรับระบบอุตสาหกรรม
4. สนับสนุนการเพิ่มมาตรการด้าน ความมั่นคงปลอดภัยเชิงปฏิบัติการ (Operational Security) ในโรงงานอุตสาหกรรม

1.4 ประโยชน์ที่ได้รับ

ด้านความรู้และความเข้าใจ

1. ได้ความเข้าใจในเชิงลึกเกี่ยวกับประเด็นด้านความมั่นคงปลอดภัยในระบบควบคุมอุตสาหกรรม (OT/ICS)
2. เพิ่มพูนทักษะในการนำ Machine Learning มาประยุกต์ใช้กับข้อมูลเชิงเวลา (time-series)

ด้านการพัฒนาเทคโนโลยี

1. ได้ต้นแบบ (Prototype) ของระบบตรวจจับความผิดปกติที่สามารถนำไปต่อยอดการพัฒนาระบบจริงได้
2. แสดงให้เห็นความเป็นไปได้ของการนำ AI/ML มาใช้แก้ปัญหาความปลอดภัยในภาคอุตสาหกรรม

ด้านการประยุกต์ใช้

1. ผลงานที่ได้สามารถนำไปใช้ตรวจจับการโจมตีหรือพฤติกรรมผิดปกติในโรงงานอุตสาหกรรมจริง
2. ช่วยเพิ่มความปลอดภัยทั้งในด้านข้อมูล เครื่องจักร และบุคลากร ลดความเสี่ยงจากการหยุดชะงักของกระบวนการผลิต

ด้านการวิจัยและการศึกษา

1. สามารถนำองค์ความรู้และผลลัพธ์ไปต่อยอดในเชิงวิจัยด้านความปลอดภัยไซเบอร์สำหรับระบบอุตสาหกรรม
2. เป็นกรณีศึกษาในการบูรณาการ Machine Learning กับความปลอดภัยไซเบอร์ (AI for Cybersecurity)

1.5 เทคโนโลยีและเครื่องมือที่ใช้

1.5.1 เทคโนโลยีด้านฮาร์ดแวร์

Graphics Processing Unit (GPU)

1. ใช้ GPU จากบริการ Google Colab เพื่อเร่งความเร็วในการฝึกโมเดลเชิงลึก (Deep Learning)

1.5.2 เทคโนโลยีด้านซอฟต์แวร์

ภาษาโปรแกรม (Programming Language)

1. ใช้ GPU จากบริการ Google Colab เพื่อเร่งความเร็วในการฝึกโมเดลเชิงลึก (Deep Learning)

เฟรมเวิร์กและไลบรารีสำหรับ Machine Learning (Frameworks & Libraries)

1. Scikit-learn: สำหรับ preprocessing การสร้าง baseline model และการประเมินผลเบื้องต้น
2. TensorFlow และ PyTorch: สำหรับการพัฒนาและฝึกโมเดลเชิงลึก (Deep Learning) เช่น CNN และ LSTM
3. Pandas และ NumPy: สำหรับการจัดการข้อมูลเชิงตารางและการคำนวณเชิงตัวเลข
4. Matplotlib และ Seaborn: สำหรับการสร้าง Visualization วิเคราะห์ข้อมูลและผลลัพธ์ของโมเดล

สภาพแวดล้อมการพัฒนา (Development Environment)

1. Jupyter Notebook และ Google Colab: สำหรับการทดลองเชิงโค้ด การประเมินผล และการรันโมเดลบน GPU
2. GitHub: สำหรับการจัดการซอร์สโค้ดและการทำงานร่วมกัน

1.6 แผนการดำเนินงาน

ขั้นตอนการดำเนินงาน	มิ.ย. 2568	ก.ค. 2568	ส.ค. 2568	ก.ย. 2568	ต.ค. 2568	พ.ย. 2568	ธ.ค. 2568	ม.ค. 2569	ก.พ. 2569
ขอใช้ข้อมูลและศึกษา Dataset จาก iTrust SUTD									
การเตรียมและทำความสะอาดข้อมูล (preprocessing)									
การออกแบบคุณลักษณะ (feature engineering)									
การกำหนดป้ายกำกับข้อมูล (Labeling)									
การสร้างและฝึกโมเดลหลัก CNN-LSTM									
การสร้างและฝึกโมเดลเปรียบเทียบ CNN และ LSTM)									
การประเมินผลและวิเคราะห์									
การจัดทำต้นแบบสาธิต เขียนรายงานและนำเสนอผลงาน									

ตารางที่ 1.1: Project Plan

1.7 บทบาทและความรับผิดชอบ

1. นายธนกฤต บุญยัง รหัส 650612084
    - พัฒนาและปรับแต่งโมเดล CNN-LSTM แบบบูรณาการ รวมถึงการปรับค่า hyperparameters ให้เหมาะสม
    - จัดทำการแสดงผลลัพธ์ เช่น กราฟ ตาราง Confusion Matrix และ Anomaly Detection Plot
    - เขียนรายงาน สรุปผลการทดลอง จัดทำสไลด์ และนำเสนอผลงาน
    - โมเดลที่รับผิดชอบ: CNN-LSTM (เวอร์ชันที่ปรับปรุงแล้ว) และการวิเคราะห์เปรียบเทียบ
2. นายศุภกร สุวรรณภพ รหัส 650612100
    - ออกแบบและพัฒนาโมเดล LSTM สำหรับการตรวจจับความผิดปกติของข้อมูลลำดับเวลา
    - ประเมินประสิทธิภาพของโมเดล วิเคราะห์ตัวชี้วัด (Precision, Recall, F1-score, AUC) และเปรียบเทียบกับ baseline models
    - สนับสนุนการทดสอบความทนทาน (robustness testing) เช่น การเพิ่ม noise และการจำลองข้อมูลที่หายไป
    - โมเดลที่รับผิดชอบ: LSTM และ CNN-LSTM
3. นายอดิศร สันเจริญ รหัส 650612104
    - ค้นหาและจัดการข้อมูล SWaT dataset รวมถึงการทำความสะอาดข้อมูล (cleaning), การทำ normalization, การจัดการ missing values และการทำ windowing สำหรับ time-series

- มีส่วนร่วมในกระบวนการเตรียมข้อมูล การฝึกสอนโมเดล การทดสอบ และการตรวจสอบความถูกต้อง (validation)
- ดำเนินการทดสอบประสิทธิภาพและการเลือก threshold ที่เหมาะสม
- โมเดลที่รับผิดชอบ: CNN และ CNN-LSTM

1.8 ผลกระทบด้านสังคม สุขภาพ ความปลอดภัย กฎหมาย และวัฒนธรรม

โครงงานนี้มุ่งเน้นการพัฒนาระบบตรวจจับความผิดปกติในระบบควบคุมอุตสาหกรรม (OT) ซึ่งมีผลกระทบในหลายมิติ ดังนี้

ด้านสังคม

การเพิ่มประสิทธิภาพในการตรวจจับและป้องกันการโจมตีไซเบอร์ในระบบอุตสาหกรรม สามารถลดความเสี่ยงของเหตุการณ์ที่อาจกระทบต่อประชาชน เช่น การปนเปื้อนของระบบน้ำในกรณีศึกษา SWaT dataset หรือการหยุดชะงักของการให้บริการสาธารณะ ซึ่งมีผลโดยตรงต่อคุณภาพชีวิตของสังคม

ด้านสุขภาพและความปลอดภัย

ระบบ OT ที่ถูกรุกรานสามารถส่งผลกระทบต่อความปลอดภัยของพนักงานและชุมชนโดยรอบ เช่น การทำงานผิดพลาดของปั๊ม วาล์ว หรือเซนเซอร์ในกระบวนการผลิตน้ำ หากตรวจจับและตอบสนองได้เร็ว จะช่วยลดความเสี่ยงต่ออุบัติเหตุและผลกระทบต่อสุขภาพของผู้บริโภคและผู้ปฏิบัติงาน

ด้านกฎหมายและมาตรฐาน

อุตสาหกรรมสมัยใหม่ต้องสอดคล้องกับข้อกำหนดด้านความมั่นคงปลอดภัยไซเบอร์ (Cybersecurity Regulations) และมาตรฐานสากล เช่น IEC 62443 และ ISO/IEC 27001 การพัฒนาโครงงานนี้ช่วยสร้างแนวทางที่สอดคล้องกับข้อกำหนดดังกล่าว และสามารถนำไปปรับใช้เพื่อตอบสนองต่อกฎหมายและมาตรการควบคุมของหน่วยงานที่เกี่ยวข้อง

ด้านวัฒนธรรมองค์กร

การนำเทคโนโลยี Machine Learning มาปรับใช้กับระบบ OT ช่วยเสริมสร้างวัฒนธรรมด้าน ความตระหนักรู้ทางไซเบอร์ (Cybersecurity Awareness) ในองค์กร ส่งเสริมให้ผู้ปฏิบัติงานและผู้บริหารเห็นความสำคัญของความปลอดภัยเชิงข้อมูลควบคู่ไปกับความปลอดภัยเชิงกายภาพ

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

การทำโครงงาน เริ่มต้นด้วยการศึกษาค้นคว้า ทฤษฎีที่เกี่ยวข้อง หรือ งานวิจัย/โครงงาน ที่เคยมีผู้นำเสนอไว้แล้ว ซึ่งเนื้อหาในบทนี้ก็จะเกี่ยวกับการอธิบายถึงสิ่งที่เกี่ยวข้องกับโครงงาน เพื่อให้ผู้อ่านเข้าใจเนื้อหาในบทถัดๆ ไปได้ง่ายขึ้น

2.1 Operational Technology (OT) และ Industrial Control Systems (ICS)

Operational Technology (OT) คือเทคโนโลยีที่ใช้สำหรับตรวจสอบและควบคุมกระบวนการทางกายภาพในโรงงานอุตสาหกรรมหรือโครงสร้างพื้นฐานที่สำคัญ เช่น ระบบไฟฟ้า ระบบน้ำมัน และระบบบำบัดน้ำเสีย จุดเด่นของ OT คือการทำงานที่ต้องเน้น ความต่อเนื่อง ความเสถียร และความปลอดภัย มากกว่า IT (Information Technology) ที่มุ่งเน้นการประมวลผลข้อมูลและธุรกรรมเป็นหลัก

Industrial Control Systems (ICS) เป็นกลุ่มของระบบที่ใช้ควบคุมและจัดการการทำงานของ OT โดยมีตัวอย่างที่สำคัญ ได้แก่

1. Supervisory Control and Data Acquisition (SCADA) ใช้สำหรับควบคุมและตรวจสอบระบบขนาดใหญ่ที่กระจายตัว
2. Distributed Control System (DCS) ใช้ควบคุมกระบวนการแบบต่อเนื่อง เช่น โรงกลั่นน้ำมัน
3. Programmable Logic Controller (PLC) ใช้ในกระบวนการที่ต้องการการควบคุมแบบเฉพาะกิจ

2.2 ความมั่นคงปลอดภัยไซเบอร์ในระบบ OT/ICS

ระบบ ICS เดิมถูกออกแบบมาเพื่อเน้นความทนทานและการทำงานอย่างต่อเนื่อง โดยไม่ได้คำนึงถึงการป้องกันภัยไซเบอร์ ทำให้มีช่องโหว่ต่อการโจมตี เช่น:

1. Stuxnet (2010): มัลแวร์ที่แทรกแซงการทำงานของ PLC ในโรงงานนิวเคลียร์อิหร่าน
2. Ukraine Power Grid Attack (2015): การโจมตีระบบไฟฟ้าทำให้เกิดการดับไฟในวงกว้าง

กรณีศึกษาดังกล่าวสะท้อนว่าภัยคุกคามไซเบอร์สามารถสร้างความเสียหายทั้งด้านเศรษฐกิจ ความมั่นคง และความปลอดภัยของประชาชน การพัฒนา ระบบตรวจจับความผิดปกติ (Anomaly Detection) จึงมีความสำคัญอย่างยิ่งใน OT

2.3 Anomaly Detection

Anomaly Detection คือกระบวนการระบุข้อมูลที่เบี่ยงเบนไปจากรูปแบบปกติ ซึ่งอาจเกิดจากการโจมตี ความผิดพลาดของอุปกรณ์ หรือความผิดปกติของกระบวนการ โดยทั่วไปแบ่งได้เป็น 3 ประเภท:

1. Point Anomaly: ข้อมูลจุดเดียวที่ผิดปกติ (เช่น ค่า Sensor พุ่งสูงกว่าปกติ)

2. Contextual Anomaly: ข้อมูลที่ปกติในบางบริบท แต่ผิดปกติในอีกบริบทหนึ่ง (เช่น ค่าน้ำสูงในฤดูฝนเป็นเรื่องปกติ แต่สูงในฤดูแล้งถือว่าผิดปกติ)
3. Collective Anomaly: ลำดับข้อมูลหลายจุดที่รวมกันแล้วผิดปกติ (เช่น การทำงานของ Pump และ Valve ที่ไม่สัมพันธ์กัน)

สำหรับระบบ ICS Collective Anomaly มีความสำคัญมาก เนื่องจากข้อมูลจาก Sensor และ Actuator อยู่ในรูปแบบ Time-Series ที่ต้องพิจารณาลำดับเวลา

2.4 Secure Water Treatment (SWaT) Dataset

Secure Water Treatment (SWaT) Dataset เป็นชุดข้อมูลที่สร้างขึ้นจากโรงงานจำลองระบบบำบัดน้ำ (Water Treatment Testbed) โดย iTrust Lab, Singapore University of Technology and Design (SUTD) มีวัตถุประสงค์เพื่อใช้เป็น มาตรฐานกลาง (benchmark) สำหรับการวิจัยด้านความปลอดภัยไซเบอร์ในระบบควบคุมอุตสาหกรรม (ICS/SCADA)

โครงสร้างของระบบ SWaT

โรงงานจำลอง SWaT ออกแบบให้มีลักษณะใกล้เคียงโรงงานจริง โดยแบ่งออกเป็น 6 กระบวนการย่อย (Processes/Stages):

1. P1: Raw Water Supply – ระบบสูบน้ำดิบเข้าสู่ถังเก็บ
2. P2: Pre-treatment – การกรองเบื้องต้น (Ultrafiltration)
3. P3: Chemical Dosing – การเติมสารเคมี เช่น กรด-ด่าง เพื่อปรับค่า pH
4. P4: Membrane-based Ultra-Filtration – การกรองละเอียดด้วยเมมเบรน
5. P5: Dechlorination – การกำจัดคลอรีนที่เหลืออยู่
6. P6: Backwash and Product Storage – การล้างย้อนและการเก็บน้ำสะอาดในถังเก็บสุดท้าย

ในแต่ละกระบวนการจะมีการติดตั้ง เซนเซอร์ (sensors) และ แอกชูเอเตอร์ (actuators) เช่น ปั๊ม (pumps), วาล์ว (valves), และเครื่องวัดระดับน้ำ (level sensors) ซึ่งควบคุมด้วย PLC (Programmable Logic Controllers) และเฝ้าติดตามผ่านระบบ SCADA

ประเภทตัวแปรใน Dataset

ข้อมูลที่บันทึกจาก SWaT เป็น Time-Series รายวินาที (1 sample/second) ครอบคลุมช่วงเวลา 11 วัน รวม 946,722 records และ 51 attributes

ข้อมูลทางกายภาพ (Physical Properties)

1. Sensors (ตัวแปรต่อเนื่อง)
   (a) Flow (อัตราการไหล)
   (b) Level (ระดับน้ำ)
   (c) Pressure (ความดัน)
   (d) Conductivity (ค่าการนำไฟฟ้า)
   (e) pH (กรด-ด่าง)
   (f) ORP (Oxidation-Reduction Potential)
   (g) Temperature (อุณหภูมิ)
2. Actuators (ตัวแปรไม่ต่อเนื่อง)
   (a) ปั๊ม (Pumps – เปิด/ปิด)
   (b) วาล์ว (Motorized Valves – เปิด/ปิด)
   (c) Dosing Pumps (สำหรับจ่ายสารเคมี)

สถานการณ์การโจมตี (Attack Scenarios)

ในการทดลอง มีการออกแบบการโจมตีทั้งหมด 36 รูปแบบ แบ่งเป็น:

1. Single Stage Single Point (SSSP) – โจมตีจุดเดียวในกระบวนการเดียว (26 ครั้ง)
2. Single Stage Multi Point (SSMP) – โจมตีหลายจุดในกระบวนการเดียว (4 ครั้ง)
3. Multi Stage Single Point (MSSP) – โจมตี 1 จุด แต่มีผลกระทบหลายกระบวนการ (2 ครั้ง)
4. Multi Stage Multi Point (MSMP) – โจมตีหลายจุดในหลายกระบวนการ (4 ครั้ง)

ตัวอย่าง:

1. ปลอมค่าจาก LIT101 (Level Sensor) ทำให้ระบบเข้าใจผิดว่าน้ำเต็มถัง และสั่งหยุดปั๊ม ส่งผลให้น้ำล้น (Overflow)
2. ปลอมค่าจาก LIT301 ให้แสดงค่าสูงผิดปกติ ส่งผลให้ปั๊มทำงานต่อแม้น้ำหมด และเกิด Underflow และอาจทำให้ปั๊มเสียหาย

2.5 Machine Learning

Machine Learning (ML) เป็นแขนงหนึ่งของปัญญาประดิษฐ์ (Artificial Intelligence: AI) ที่ช่วยให้ระบบสามารถเรียนรู้และปรับปรุงประสิทธิภาพจากข้อมูล โดยไม่ต้องเขียนกฎแบบตายตัวล่วงหน้า การป้อนข้อมูลจำนวนมากที่หลากหลายช่วยให้โมเดลสามารถสกัดรูปแบบ (patterns) และปรับปรุงความแม่นยำได้อย่างต่อเนื่อง

สำหรับโครงการนี้ ML ถูกนำมาใช้ในการตรวจจับความผิดปกติ (Anomaly Detection) ใน ข้อมูลเชิงเวลา (time-series) ที่ได้จาก ระบบควบคุมอุตสาหกรรม (ICS/OT) ซึ่งประกอบด้วยเซนเซอร์และแอกชูเอเตอร์ในระบบบำบัดน้ำ (SWaT dataset)

2.6 Convolutional Neural Network – Long Short-Term Memory (CNN-LSTM)

การตรวจจับความผิดปกติในข้อมูลเชิงเวลา (time-series anomaly detection) จำเป็นต้องอาศัยโมเดลที่สามารถสกัดคุณลักษณะเชิงลึกและในขณะเดียวกันต้องเข้าใจความสัมพันธ์เชิงลำดับเวลา โมเดล CNN-LSTM จึงถูกนำมาใช้เพราะผสานข้อดีของ Convolutional Neural Network (CNN) และ Long Short-Term Memory (LSTM) เข้าด้วยกัน ทำให้สามารถจัดการกับข้อมูลจากระบบควบคุมอุตสาหกรรม (ICS) ที่มีความซับซ้อนสูงได้อย่างมีประสิทธิภาพ โดยเฉพาะข้อมูลจาก SWaT dataset ซึ่งประกอบด้วยข้อมูลจาก sensor และ actuator ที่บันทึกเป็นลำดับเวลา

2.6.1 Convolutional Neural Network (CNN)

CNN เป็นโครงข่ายประสาทเทียมที่พัฒนาขึ้นเพื่อตรวจจับและสกัดคุณลักษณะสำคัญจากข้อมูล โดยอาศัยการทำงานของ convolution filters ที่เลื่อนผ่านข้อมูลเพื่อค้นหารูปแบบซ้ำหรือความเปลี่ยนแปลงที่มีนัยสำคัญ ในบริบทของข้อมูลเชิงเวลา CNN สามารถทำหน้าที่ตรวจสอบการเปลี่ยนแปลงของค่า sensor ในช่วงสั้น ๆ ได้อย่างมีประสิทธิภาพ เช่น การแกว่งของระดับน้ำหรือการเปลี่ยนแปลงค่าความดันอย่างเฉียบพลัน นอกจากนี้ CNN ยังช่วยลดสัญญาณรบกวน (noise) และเน้นคุณลักษณะที่บ่งบอกถึงความผิดปกติได้อย่างชัดเจน

2.6.2 Long Short-Term Memory (LSTM)

การผสมผสาน CNN และ LSTM เข้าไว้ด้วยกันในโมเดล CNN-LSTM ทำให้สามารถใช้ CNN ในการสกัดคุณลักษณะสำคัญจากข้อมูลเชิงเวลาในช่วงสั้น แล้วส่งผลลัพธ์ต่อให้ LSTM เพื่อเรียนรู้ความสัมพันธ์ที่ต่อเนื่องในระยะยาว แนวทางนี้ช่วยให้โมเดลสามารถจับพฤติกรรมผิดปกติที่ซับซ้อนในระบบ ICS ได้ดียิ่งขึ้น เมื่อเปรียบเทียบกับการใช้ CNN หรือ LSTM เพียงอย่างเดียว งานวิจัยจำนวนมากชี้ให้เห็นว่าโมเดล CNN-LSTM มีศักยภาพในการลดอัตราการตรวจจับผิดพลาด (false positives) และเพิ่มความแม่นยำในการตรวจจับ anomaly โดยเฉพาะอย่างยิ่งเมื่อใช้กับข้อมูลที่มีลักษณะ time-series แบบหลายตัวแปร (multivariate time-series) เช่นใน SWaT dataset

2.6.3 CNN-LSTM for Anomaly Detection

การผสมผสาน CNN และ LSTM เข้าไว้ด้วยกันในโมเดล CNN-LSTM ทำให้สามารถใช้ CNN ในการสกัดคุณลักษณะสำคัญจากข้อมูลเชิงเวลาในช่วงสั้น แล้วส่งผลลัพธ์ต่อให้ LSTM เพื่อเรียนรู้ความสัมพันธ์ที่ต่อเนื่องในระยะยาว แนวทางนี้ช่วยให้โมเดลสามารถจับพฤติกรรมผิดปกติที่ซับซ้อนในระบบ ICS ได้ดียิ่งขึ้น เมื่อเปรียบเทียบกับการใช้ CNN หรือ LSTM เพียงอย่างเดียว งานวิจัยจำนวนมากยืนยันว่าโมเดล CNN-LSTM มีประสิทธิภาพในการลดอัตราการตรวจจับผิดพลาด (false positives) และเพิ่มความแม่นยำในการตรวจจับ anomaly ได้ โดยเฉพาะอย่างยิ่งเมื่อใช้กับข้อมูลที่มีลักษณะ time-series แบบหลายตัวแปร (multivariate time-series) เช่นใน SWaT dataset

เมื่อพิจารณาเปรียบเทียบกับวิธี anomaly detection อื่น ๆ ที่นิยม เช่น Autoencoder และ Isolation Forest จะเห็นประเด็นดังนี้:

1. Autoencoder มักใช้สำหรับการบีบอัดและกู้คืนข้อมูล (reconstruction-based anomaly detection) และสามารถตรวจจับความผิดปกติเมื่อค่าที่กู้คืนแตกต่างจากข้อมูลจริง แต่ข้อจำกัดคือ Autoencoder อาจไม่สามารถสะท้อนความสัมพันธ์ตามเวลา (temporal dependency) ได้อย่างเต็มที่ โดยเฉพาะในระบบ ICS ที่ sensor และ actuator มีความเชื่อมโยงหลายขั้นตอน
2. Isolation Forest เหมาะกับการตรวจหาค่าผิดปกติ (outliers) ในข้อมูลเชิง tabular และทำงานได้ดีโดยไม่ต้องใช้การฝึกซ้ำที่ซับซ้อน แต่ข้อจำกัดคือไม่สามารถเรียนรู้ความสัมพันธ์เชิงเวลาและความเชื่อมโยงระหว่างหลายตัวแปรได้ลึกพอ ซึ่งเป็นลักษณะสำคัญของข้อมูลในระบบ ICS
3. CNN-LSTM จึงเป็นแนวทางที่ “มีโอกาสตอบโจทย์ได้ดีกว่า” เพราะ CNN สามารถดึงคุณลักษณะเชิงพื้นที่ระยะสั้น (local features) จาก time-series ขณะที่ LSTM สามารถเรียนรู้ลำดับข้อมูลระยะยาวได้ เมื่อรวมกันจึงมีศักยภาพที่จะตรวจจับความผิดปกติแบบ collective anomaly ได้ดีกว่าโมเดลทั่วไป ซึ่งเหมาะสมกับปัญหาของ SWaT dataset ที่มีทั้งข้อมูล multivariate, sequential และ anomalous attack patterns

2.6.4 Dense Layer และ Fully Connected Layer

หลังจากที่ CNN และ LSTM ทำหน้าที่สกัดคุณลักษณะและจับความสัมพันธ์เชิงเวลาแล้ว ผลลัพธ์ที่ได้จะถูกส่งต่อไปยัง Dense Layer หรือ Fully Connected Layer เพื่อทำการรวมข้อมูลและสร้างการตัดสินใจสุดท้าย Dense Layer ทำงานโดยเชื่อมต่อทุกนิวรอนกับนิวรอนในชั้นถัดไป ช่วยให้โมเดลสามารถเรียนรู้การรวมคุณลักษณะที่ซับซ้อนและสร้างการจำแนก (classification) ระหว่างเหตุการณ์ปกติและเหตุการณ์ผิดปกติได้อย่างมีประสิทธิภาพ ในโครงการนี้ Dense Layer ทำหน้าที่แปลง representation ที่ได้จาก CNN-LSTM ให้กลายเป็นค่าความน่าจะเป็น (probability) ของ class เช่น Normal (0) และ Anomaly (1)

2.6.5 Activation Functions

ฟังก์ชันกระตุ้น (Activation Functions) เป็นองค์ประกอบสำคัญของ Dense Layer ซึ่งใช้ในการกำหนดเส้นแบ่งเชิงเส้นหรือไม่เชิงเส้น เช่น:

1. ReLU (Rectified Linear Unit): ใช้ในชั้นซ่อน (hidden layers) ของ CNN และ LSTM เพื่อลดปัญหา vanishing gradient และเร่งการเรียนรู้
2. Sigmoid: เหมาะกับการจำแนกแบบ binary anomaly detection เนื่องจากผลลัพธ์อยู่ระหว่าง 0–1
3. Softmax: ใช้สำหรับ multi-class anomaly detection เมื่อมีความผิดปกติหลายประเภท

2.6.6 Regularization Layers (Dropout และ Batch Normalization)

เพื่อป้องกันการเกิด overfitting จากข้อมูลที่มีความซับซ้อนสูง เช่น SWaT dataset การเพิ่ม Regularization Layers มีความสำคัญมาก ได้แก่:

1. Dropout Layer: ทำการสุ่มปิดบางนิวรอนระหว่างการฝึก เพื่อป้องกันไม่ให้โมเดลจดจำข้อมูลมากเกินไป แต่ช่วยให้เกิดการเรียนรู้ที่ครอบคลุมมากขึ้น
2. Batch Normalization: ช่วยปรับค่าการกระจายของข้อมูลในแต่ละชั้นให้อยู่ในช่วงที่เหมาะสม ทำให้การฝึกโมเดลเสถียรและรวดเร็วขึ้น

2.7 Correlation Matrix

Correlation Matrix เป็นเครื่องมือเชิงสถิติที่ใช้วิเคราะห์ความสัมพันธ์ระหว่างตัวแปรหลายตัวพร้อมกัน โดยจะแสดงค่า Correlation Coefficient ที่มีค่าตั้งแต่ -1 ถึง 1

1. ค่า 1 หมายถึงความสัมพันธ์เชิงบวกอย่างสมบูรณ์ (เมื่อค่าของตัวแปรหนึ่งเพิ่ม อีกตัวก็เพิ่มตาม)
2. ค่า -1 หมายถึงความสัมพันธ์เชิงลบอย่างสมบูรณ์ (เมื่อค่าของตัวแปรหนึ่งเพิ่ม อีกตัวจะลดลง)
3. ค่า 0 หมายถึงไม่มีความสัมพันธ์

สำหรับ SWaT dataset การสร้าง Correlation Matrix มีความสำคัญในหลายประเด็น ได้แก่

1. การทำความเข้าใจความสัมพันธ์ของ Sensor และ Actuator: เช่น ระดับน้ำ (Level) ควรมีความสัมพันธ์กับการทำงานของปั๊ม (Pump) และวาล์ว (Valve)
2. การตรวจหาความผิดปกติ (Anomaly): หาก correlation ระหว่าง sensor และ actuator ไม่เป็นไปตามปกติ เช่น ค่า Flow ไม่สัมพันธ์กับการเปิดปิด Valve อาจบ่งชี้ถึงการโจมตีหรือความผิดปกติในกระบวนการ
3. Feature Selection: การวิเคราะห์ความสัมพันธ์ช่วยคัดเลือกตัวแปรที่สำคัญ และลด multicollinearity เพื่อลด noise ก่อนนำเข้าสู่โมเดล Machine Learning

การใช้ Correlation Matrix จึงเป็นขั้นตอนสำคัญใน Exploratory Data Analysis (EDA) ของโครงการนี้ เพื่อทำให้การออกแบบและฝึกสอนโมเดล anomaly detection มีความถูกต้องและแม่นยำมากขึ้น

2.8 Model Evaluation

การประเมินผลลัพธ์ของโมเดลมีความสำคัญอย่างยิ่ง เนื่องจากเป็นวิธีการตรวจสอบความถูกต้องและความน่าเชื่อถือของโมเดลในการตรวจจับความผิดปกติจากข้อมูลจริง โดยเฉพาะอย่างยิ่งในกรณีของ SWaT dataset ซึ่งมีลักษณะข้อมูล ไม่สมดุล (imbalanced data) กล่าวคือ ข้อมูลเหตุการณ์ปกติ (Normal) มีจำนวนมากกว่าข้อมูลเหตุการณ์ผิดปกติ (Anomaly) อย่างมาก การใช้ตัวชี้วัด (Evaluation Metrics) ที่เหมาะสมจึงมีบทบาทสำคัญในการสะท้อนศักยภาพของโมเดลได้อย่างครบถ้วน
ตัวชี้วัดที่ใช้ประกอบการประเมินในโครงการนี้ ได้แก่:

1. Accuracy
   เป็นสัดส่วนของจำนวนข้อมูลทั้งหมดที่โมเดลสามารถจำแนกได้ถูกต้อง แต่เนื่องจากข้อมูลมีความไม่สมดุล ค่าความแม่นยำ (accuracy) เพียงอย่างเดียวอาจไม่สะท้อนศักยภาพของโมเดลได้อย่างแท้จริง เพราะโมเดลที่ทำนายว่าข้อมูลเป็น “ปกติ” ตลอดเวลาอาจได้ค่า accuracy สูง แต่ไม่สามารถตรวจจับ anomaly ได้เลย

2. Precision
   เป็นตัวชี้วัดความถูกต้องของการทำนาย anomaly หมายถึง ในจำนวนทั้งหมดที่โมเดลทำนายว่า “ผิดปกติ” มีสัดส่วนเท่าใดที่เป็น anomaly จริง Precision ที่สูงบ่งชี้ว่าโมเดลสามารถลดการแจ้งเตือนผิดพลาด (False Positives) ได้

3. Recall (Sensitivity)
   เป็นตัวชี้วัดความสามารถของโมเดลในการตรวจจับ anomaly ได้ครบถ้วน หมายถึงในจำนวน anomaly ทั้งหมดที่มีอยู่โมเดลสามารถตรวจพบได้กี่กรณี Recall จึงมีความสำคัญอย่างยิ่งในงาน ICS/OT anomaly detection เพราะการพลาดการตรวจจับ anomaly เพียงเล็กน้อยอาจก่อให้เกิดผลกระทบต่อความปลอดภัยและการดำเนินงานของระบบ

4. F1-score
   เป็นค่าเฉลี่ยเชิงฮาร์โมนิก (Harmonic Mean) ระหว่าง Precision และ Recall ทำให้สะท้อนสมดุลระหว่างการลดการแจ้งเตือนผิดพลาดและการเพิ่มอัตราการตรวจจับ anomaly ได้อย่างเหมาะสม โดยเฉพาะอย่างยิ่งใน dataset ที่มี class imbalance

5. Confusion Matrix
   เป็นตารางที่แสดงการจำแนกผลลัพธ์ของโมเดลอย่างละเอียดในรูปของ True Positive (TP), True Negative (TN), False Positive (FP) และ False Negative (FN) ช่วยให้สามารถวิเคราะห์ข้อผิดพลาดของโมเดลได้เชิงลึก และชี้ชัดว่าปัญหาของโมเดลอยู่ที่การตรวจจับ anomaly ไม่เพียงพอ (Recall ต่ำ) หรือการแจ้งเตือนผิดพลาดสูง (Precision ต่ำ)

6. ROC-AUC (Receiver Operating Characteristic – Area Under Curve)
   เป็นการประเมินสมรรถนะของโมเดลโดยไม่ขึ้นกับ threshold โดยใช้ค่า True Positive Rate และ False Positive Rate มาวิเคราะห์เส้นโค้ง (ROC Curve) ค่า AUC ที่สูงใกล้ 1 บ่งชี้ว่าโมเดลสามารถจำแนก anomaly ออกจาก normal ได้อย่างมีประสิทธิภาพ

7. PR-AUC (Precision-Recall Area Under Curve)
   เป็นการวัดสมรรถนะที่เหมาะสมอย่างยิ่งสำหรับข้อมูลที่ไม่สมดุล (imbalanced dataset) โดยแสดงความสัมพันธ์ระหว่าง Precision และ Recall ตลอดทุก threshold ค่าที่สูงใกล้ 1 แสดงว่าโมเดลมีความสามารถในการตรวจจับ anomaly ได้ดีแม้ในสภาพที่ anomaly มีสัดส่วนค่อนข้างน้อย

2.9 การศึกษางานวิจัยที่เกี่ยวข้อง (Literature Review)

2.9.1 Isolation Forest on SWaT 2015

งานวิจัยนี้มีวัตถุประสงค์เพื่อทดสอบประสิทธิภาพของอัลกอริทึม Isolation Forest ในการตรวจจับความผิดปกติบนชุดข้อมูล SWaT testbed dataset ซึ่งถือเป็นมาตรฐานสำคัญด้านความปลอดภัยของระบบควบคุมอุตสาหกรรม (ICS/SCADA) วิธีการนี้จัดอยู่ในกลุ่ม unsupervised learning โดยใช้หลักการว่าข้อมูลผิดปกติ (anomaly) จะสามารถถูก “แยกออก” จากข้อมูลปกติได้ง่ายกว่า และจึงใช้จำนวนการ split ที่น้อยกว่าใน tree-based structure

ผลการทดลองพบว่า Isolation Forest สามารถตรวจจับ anomaly ได้ในระดับที่น่าพอใจ โดยมีข้อได้เปรียบคือ ความเร็วในการทำงานและการใช้ทรัพยากรต่ำ ซึ่งทำให้เหมาะสมกับระบบที่ต้องการประมวลผล

เบื้องต้น อย่างไรก็ตาม ข้อจำกัดสำคัญคือไม่สามารถจับความสัมพันธ์เชิงลึกระหว่างตัวแปรหลายมิติ และไม่สามารถสะท้อนความเชื่อมโยงเชิงเวลา (temporal dependency) ได้อย่างเพียงพอ

ในเชิงปริมาณ งานวิจัยรายงานว่า Isolation Forest บน SWaT dataset มี Accuracy ≈ 0.89 (≥ 89%) แต่ผลลัพธ์สะท้อนถึงความไม่สมดุลของคลาสอย่างชัดเจน ดังนี้:

1. Normal class: Precision = 0.97, Recall = 0.08, F1-score = 0.15
2. Attack class: Precision = 0.89, Recall = 1.00, F1-score = 0.94
3. Macro average F1 = 0.54
4. Weighted average F1 ≈ 0.84

แม้ว่าวิธีนี้สามารถตรวจจับ เหตุการณ์โจมตี (Attack) ได้อย่างมีประสิทธิภาพ (Recall สูงถึง 1.00) แต่มีข้อจำกัดชัดเจนในด้านการตรวจจับ เหตุการณ์ปกติ (Normal) ที่มี Recall ต่ำมาก ทำให้เกิด False Positive Rate ค่อนข้างสูง และไม่เหมาะสมกับการประยุกต์ใช้ในสภาพแวดล้อม ICS ที่ซับซ้อน เมื่อเปรียบเทียบกับวิธีการแบบ deep learning ที่สามารถใช้ประโยชน์จากโครงสร้างข้อมูลเชิงเวลาได้ดีกว่า [5].

2.9.2 Anomaly Detection for Industrial Control System Based on Autoencoder Neural Network

งานวิจัยนี้มีโจทย์หลักในการตรวจจับความผิดปกติในระบบควบคุมอุตสาหกรรม (ICS) โดยเฉพาะข้อมูลจากเซนเซอร์และแอกชูเอเตอร์ที่มีความซับซ้อนสูง นักวิจัยได้พัฒนา Composite Autoencoder Model ที่สามารถทำทั้งการทำนาย (prediction) และการสร้างข้อมูลกลับ (reconstruction) ไปพร้อมกัน เพื่อเพิ่มความสามารถในการเรียนรู้รูปแบบปกติของข้อมูล และตรวจจับ anomaly ผ่านค่า reconstruction error เมื่อข้อมูลผิดปกติถูกป้อนเข้าโมเดล Autoencoder จะไม่สามารถสร้างข้อมูลที่ใกล้เคียงกับข้อมูลจริงได้ ส่งผลให้เกิด error สูงและสามารถจัดเป็น anomaly ได้อย่างมีประสิทธิภาพ

ผลการทดลองบน SWaT dataset แสดงให้เห็นว่าโมเดล Autoencoder ที่ปรับปรุงนี้สามารถบรรลุค่า Recall ประมาณ 88.5

อย่างไรก็ตาม งานวิจัยยังชี้ให้เห็นข้อจำกัดบางประการ ได้แก่ การใช้พลังการประมวลผลสูงกว่าวิธีเชิงสถิติ เช่น Isolation Forest และความเสี่ยงต่อการเกิด overfitting หากขาดการออกแบบ regularization ที่เหมาะสม นอกจากนี้ โมเดลยังมีความไวต่อ noise ในข้อมูลจริง ซึ่งอาจทำให้เกิด False Positive ได้มากกว่าที่คาดการณ์ [16].

2.9.3 AI Training: Time Series Anomaly Detection

งานนี้นำเสนอการสร้างระบบ anomaly detection สำหรับข้อมูลอนุกรมเวลา (time-series) ในสภาพแวดล้อมอุตสาหกรรม โดยเน้นการทดลองฝึกโมเดลหลายประเภทเพื่อเปรียบเทียบแนวทางที่แตกต่างกัน ได้แก่ Isolation Forest, Autoencoder, และ LSTM/Hybrid models เพื่อสำรวจประสิทธิภาพในการตรวจจับความผิดปกติจากข้อมูลที่ซับซ้อน

แนวคิดหลักคือการใช้โมเดลหลากหลายรูปแบบ

1. Isolation Forest ทำงานในลักษณะ unsupervised โดยอาศัยหลักการแยกข้อมูล anomaly ที่ถูก “split” ได้ง่าย จึงรวดเร็วและใช้ทรัพยากรต่ำ
2. Autoencoder ใช้แนวทางการเรียนรู้ pattern ปกติของข้อมูลและตรวจจับ anomaly ผ่านค่า reconstruction error ทำให้เหมาะกับข้อมูลที่มีโครงสร้างซับซ้อน
3. LSTM และ Hybrid Models ถูกออกแบบมาเพื่อจับลักษณะเชิงเวลา (temporal dependency) ของข้อมูล ทำให้มีศักยภาพมากขึ้นในการระบุ anomaly ที่เกิดจากพฤติกรรมแบบ sequence

แม้ว่า notebook ไม่ได้รายงานค่าตัวชี้วัดเชิงปริมาณ เช่น Accuracy หรือ F1-score อย่างละเอียด แต่ผลการทดลองแสดงให้เห็นภาพรวมว่า

- Isolation Forest มีความรวดเร็วแต่ความแม่นยำต่ำกว่าโมเดลเชิงลึก
- Autoencoder ทำงานได้ดีกว่าหากข้อมูลมีรูปแบบชัดเจน
- LSTM/Hybrid models มีแนวโน้มให้ผลดีที่สุดในการตรวจจับ anomaly ที่ขึ้นกับลำดับเวลา

จุดแข็งของงานนี้คือการทำหน้าที่เป็น comparative study ที่แสดงภาพรวมของวิธีดั้งเดิม (traditional methods) และวิธีการเชิงลึก (deep learning methods) ว่ามีบทบาทที่ต่างกันตามเงื่อนไขการใช้งานจริง อย่างไรก็ตาม ข้อจำกัดคือการทดลองยังอยู่ในระดับต้นแบบ (prototype) โดยไม่ได้มีการปรับแต่ง hyperparameters หรือ fine-tuning โมเดลเชิงลึกอย่างเข้มข้น ทำให้ผลลัพธ์อาจไม่สามารถ generalize ได้กับทุก dataset [3].

2.9.4 LSTM-based Sequence-to-Sequence Model for ICS Anomaly Detectio

งานนี้นำเสนอการพัฒนาระบบ anomaly detection สำหรับระบบควบคุมอุตสาหกรรม (ICS) โดยใช้ LSTM-based Sequence-to-Sequence (seq2seq) Neural Network ซึ่งถูกออกแบบมาเพื่อเรียนรู้พฤติกรรมปกติของข้อมูลแบบลำดับเวลา (time-series) และตรวจจับ anomaly ผ่านค่า prediction error หรือ reconstruction error

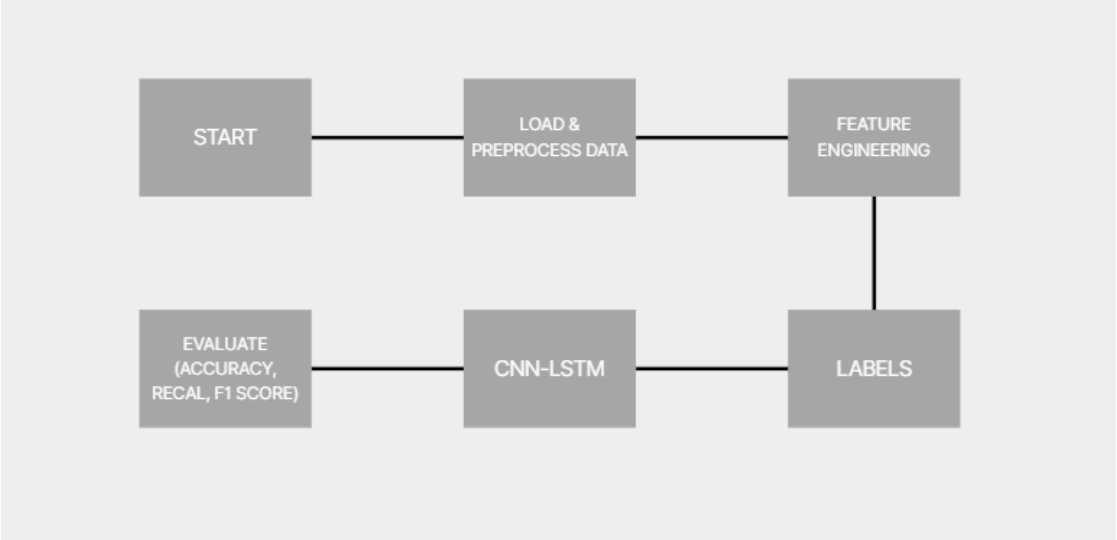
แนวคิดหลักคือการฝึกโมเดล seq2seq บนข้อมูลปกติ (normal logs) ของชุดข้อมูล SWaT testbed โดยโมเดลจะพยายามทำนายค่าลำดับเวลาถัดไป หากข้อมูลที่ป้อนเข้ามามีความผิดปกติ โมเดลจะทำนายได้ไม่แม่นยำ ส่งผลให้เกิด error สูง และสามารถระบุว่าเป็น anomaly

ผลการทดลองแสดงว่าโมเดลสามารถตรวจจับการโจมตีได้ 29 จาก 36 scenarios บน SWaT dataset ซึ่งถือเป็นผลลัพธ์ที่น่าพอใจ โดยเฉพาะอย่างยิ่งเมื่อเทียบกับวิธีการแบบดั้งเดิมที่ไม่ได้ใช้ sequence modeling อย่างไรก็ตาม งานไม่ได้รายงานค่าเชิงปริมาณ เช่น Accuracy หรือ F1-score อย่างละเอียด แต่เน้นไปที่การยืนยันว่า seq2seq LSTM มีความสามารถในการจับ temporal dependency ได้ดีกว่าวิธี unsupervised แบบทั่วไป เช่น Isolation Forest

จุดแข็งของงานนี้คือการแสดงให้เห็นว่า LSTM-based seq2seq สามารถเรียนรู้ลักษณะ multivariate time-series ได้ดี และมีศักยภาพในการตรวจจับพฤติกรรมซับซ้อนในระบบ ICS โดยไม่ต้องใช้ข้อมูลที่มี label จำนวนมาก ข้อจำกัดคือโมเดลมีความซับซ้อน ต้องใช้เวลาฝึกนาน และยังมีกรณีโจมตีบางประเภทที่ตรวจจับไม่ได้ ซึ่งอาจมาจากปัญหาการตั้ง threshold และ false negative ที่ยังคงเกิดขึ้น [8].

บทที่ 3
โครงสร้างของโครงงาน

3.1 โครงสร้างทางสถาปัตยกรรม



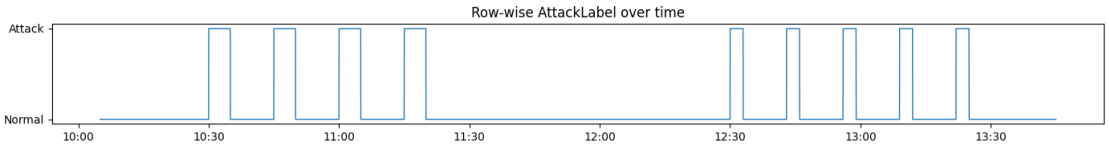
รูปที่ 3.1: System Architecture

3.2 การเตรียมข้อมูล (Data Preparation)

เพื่อให้เห็นลักษณะของข้อมูลจริง ภาพนี้เป็นตัวอย่างการแสดงผลข้อมูลจาก SWaT dataset ซึ่งบันทึกค่าของเซนเซอร์และแอกชูเอเตอร์ในรูปแบบ time-series การแสดงผลในรูปแบบตารางและกราฟช่วยให้เห็นทั้งโครงสร้าง dataset และการเปลี่ยนแปลงของค่าที่เกี่ยวข้องกับเหตุการณ์ผิดปกติ (Anomaly)

t_stamp	P1_STATE	LIT101.Pv	FIT101.Pv	MV101.Status	P101.Status	P102.Status	P2_STATE	FIT201.Pv	AIT201.Pv	AIT202.Pv	AIT203.Pv	MV201.Status	P201.Status	P202.Status	P203.Status	P204.Status	P205.Status	P206.Status	P207.Status
06/ธ.ค./2019 10:05:15	3	662.979065	0	1	2	1	2	2.32441616	34.734684	8.180467	134.042557	2	2	1	2	1	1	1	1
06/ธ.ค./2019 10:05:16	3	663.3716	0	1	2	1	2	2.32121229	34.734684	8.176942	134.042557	2	2	1	2	1	1	1	1
06/ธ.ค./2019 10:05:17	3	663.5286	0	1	2	1	2	2.31864929	34.9589844	8.175981	134.247635	2	2	1	2	1	1	1	1
06/ธ.ค./2019 10:05:18	3	663.8034	0	1	2	1	2	2.31493282	35.0551147	8.172776	134.247635	2	2	1	2	1	1	1	1
06/ธ.ค./2019 10:05:19	3	663.999634	0	1	2	1	2	2.313395	35.0551147	8.172135	134.247635	2	2	1	2	1	1	1	1
06/ธ.ค./2019 10:05:20	3	664.0782	0	1	2	1	2	2.31121659	35.0551147	8.17149448	134.247635	2	2	1	2	1	1	1	1
06/ธ.ค./2019 10:05:21	3	664.3529	0	1	2	1	2	2.31108832	35.0551147	8.17149448	134.247635	2	2	1	2	1	1	1	1
06/ธ.ค./2019 10:05:22	3	664.666931	0	1	2	1	2	2.309935	35.0551147	8.17149448	134.247635	2	2	1	2	1	1	1	1
06/ธ.ค./2019 10:05:23	3	664.9417	0	1	2	1	2	2.309935	35.0551147	8.17149448	134.247635	2	2	1	2	1	1	1	1
06/ธ.ค./2019 10:05:24	3	664.980957	0	1	2	1	2	2.31313872	35.0551147	8.17149448	134.247635	2	2	1	2	1	1	1	1
06/ธ.ค./2019 10:05:25	3	665.648254	0	1	2	1	2	2.31531739	35.0551147	8.174378	134.247635	2	2	1	2	1	1	1	1
06/ธ.ค./2019 10:05:26	3	665.923035	0	1	2	1	2	2.31941819	35.0551147	8.174378	134.247635	2	2	1	2	1	1	1	1
06/ธ.ค./2019 10:05:27	3	666.237061	0	1	2	1	2	2.32134056	35.0551147	8.176301	134.247635	2	2	1	2	1	1	1	1
06/ธ.ค./2019 10:05:28	3	666.708069	0	1	2	1	2	2.32364726	34.79877	8.176301	134.401428	2	2	1	2	1	1	1	1
06/ธ.ค./2019 10:05:29	3	667.3361	0	1	2	1	2	2.32364726	34.79877	8.17886448	134.427063	2	2	1	2	1	1	1	1
06/ธ.ค./2019 10:05:30	3	667.650146	0	1	2	1	2	2.325185	34.79877	8.17886448	134.503967	2	2	1	2	1	1	1	1
06/ธ.ค./2019 10:05:31	3	667.9249	0	1	2	1	2	2.325185	34.79877	8.17886448	134.503967	2	2	1	2	1	1	1	1
06/ธ.ค./2019 10:05:32	3	668.0034	0	1	2	1	2	2.325185	34.79877	8.17886448	134.760315	2	2	1	2	1	1	1	1
06/ธ.ค./2019 10:05:33	3	667.8857	0	1	2	1	2	2.325185	34.79877	8.17886448	134.760315	2	2	1	2	1	1	1	1
06/ธ.ค./2019 10:05:34	3	667.8072	0	1	2	1	2	2.32441616	34.60651	8.17886448	134.760315	2	2	1	2	1	1	1	1
06/ธ.ค./2019 10:05:35	3	667.846436	0	1	2	1	2	2.32441616	34.60651	8.17886448	134.760315	2	2	1	2	1	1	1	1
06/ธ.ค./2019 10:05:36	3	667.650146	0	1	2	1	2	2.32441616	34.60651	8.177903	135.1192	2	2	1	2	1	1	1	1
06/ธ.ค./2019 10:05:37	3	667.493164	0	1	2	1	2	2.32762	34.60651	8.175981	135.1192	2	2	1	2	1	1	1	1
06/ธ.ค./2019 10:05:38	3	667.493164	0	1	2	1	2	2.32902956	34.60651	8.173097	135.1192	2	2	1	2	1	1	1	1
06/ธ.ค./2019 10:05:39	3	667.493164	0	1	2	1	2	2.32954216	34.38221	8.172456	135.1192	2	2	1	2	1	1	1	1
06/ธ.ค./2019 10:05:40	3	667.8072	0	1	2	1	2	2.330311	34.38221	8.172456	135.247375	2	2	1	2	1	1	1	1
06/ธ.ค./2019 10:05:41	3	668.238953	0	1	2	1	2	2.330311	34.38221	8.170534	135.555	2	2	1	2	1	1	1	1
06/ธ.ค./2019 10:05:42	3	668.709961	0	1	2	1	2	2.32890153	34.38221	8.169893	135.555	2	2	1	2	1	1	1	1
06/ธ.ค./2019 10:05:43	3	669.2595	0	1	2	1	2	2.32582569	34.0938225	8.168932	135.555	2	2	1	2	1	1	1	1

รูปที่ 3.2: ตัวอย่าง SWaT dataset จาก itrust SUTD



รูปที่ 3.3: ตัวอย่างแสดงช่วงเวลาที่โดนโจมตีภายใน dataset

3.2.1 การโหลดและอ่านชุดข้อมูล (Data Loading)

ข้อมูลที่ใช้คือ SWaT Dataset ซึ่งเก็บค่าการทำงานของเซนเซอร์และแอกชูเอเตอร์ในลักษณะ time-series การโหลดข้อมูลใช้เครื่องมืออย่าง Pandas ในการอ่านไฟล์ .csv และจัดเก็บให้อยู่ในรูป DataFrame เพื่อความสะดวกต่อการวิเคราะห์และ preprocessing

3.2.2 การจัดการค่าที่หายไป (Missing Value Handling)

ข้อมูลที่ได้จากระบบ OT/ICS มักมีปัญหาค่าที่หายไป (missing values) อันเนื่องมาจากปัญหาของ sensor หรือการบันทึกข้อมูล การจัดการอาจทำได้หลายวิธี เช่น

1. การลบข้อมูลแถวที่หายไป (listwise deletion)
2. การแทนค่าด้วยสถิติ เช่น ค่าเฉลี่ย (mean) หรือค่ามัธยฐาน (median)
3. การแทนค่าด้วยการคำนวณจากข้อมูลรอบข้าง เช่น Interpolation เหมาะกับ time-series

3.2.3 การแยกประเภทคุณลักษณะ (Continuous / Discrete Features)

ชุดข้อมูลมักประกอบด้วยคุณลักษณะหลายประเภท เช่น

1. Continuous features: ค่าจากเซนเซอร์ เช่น ความดัน อัตราการไหล ค่าการนำไฟฟ้า
2. Discrete features: ค่าสถานะของแอกชูเอเตอร์ เช่น การเปิด/ปิดวาล์ว และการทำงานของปั๊ม การแยกประเภทนี้สำคัญ เนื่องจากขั้นตอน preprocessing จะแตกต่างกัน เช่น continuous ต้อง scaling แต่ discrete อาจใช้ encoding

3.2.4 การปรับสเกลข้อมูล (Scaling and Normalization)

ค่าของเซนเซอร์บางตัวอาจอยู่ในช่วงที่ต่างกันมาก เช่น อุณหภูมิ (0–100) กับค่าแรงดัน (0–10,000) หากไม่ปรับสเกล อาจทำให้โมเดลให้ความสำคัญกับตัวแปรที่มีค่ามากเกินไป โดยใช้วิธี

Min-Max Scaling (ปรับให้อยู่ในช่วง [0,1])

3.2.5 การแบ่งชุดข้อมูล (Training / Testing Split)

การแบ่งชุดข้อมูลใช้หลักการ time-based split เพื่อป้องกัน data leakage โดยแบ่งเป็น

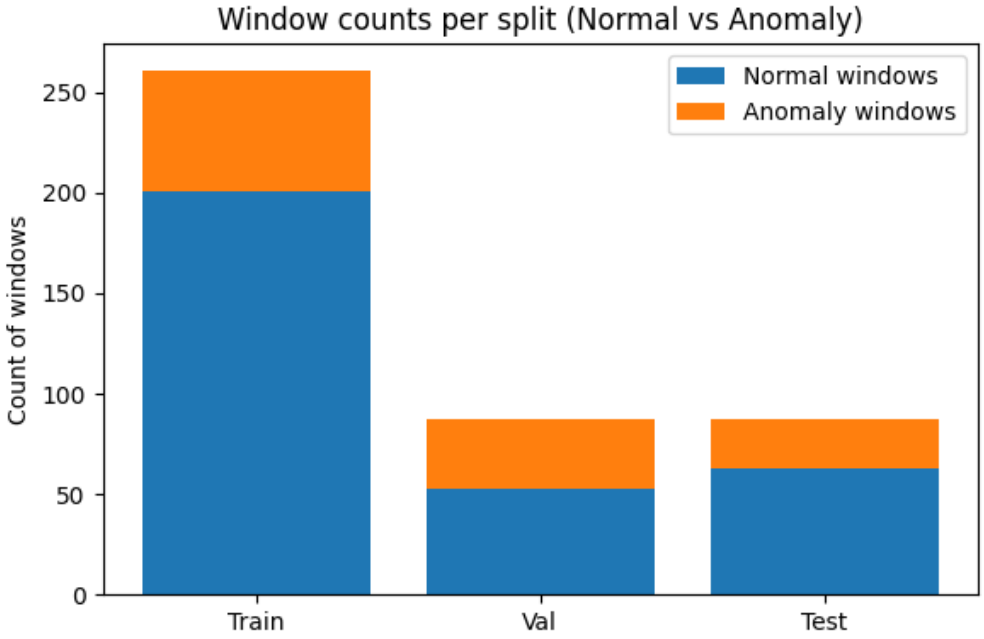
1. Training set (ใช้ฝึกโมเดล)
2. Validation set (ใช้ปรับ hyperparameter และ early stopping)

3. Test set (ใช้ทดสอบประสิทธิภาพจริง)

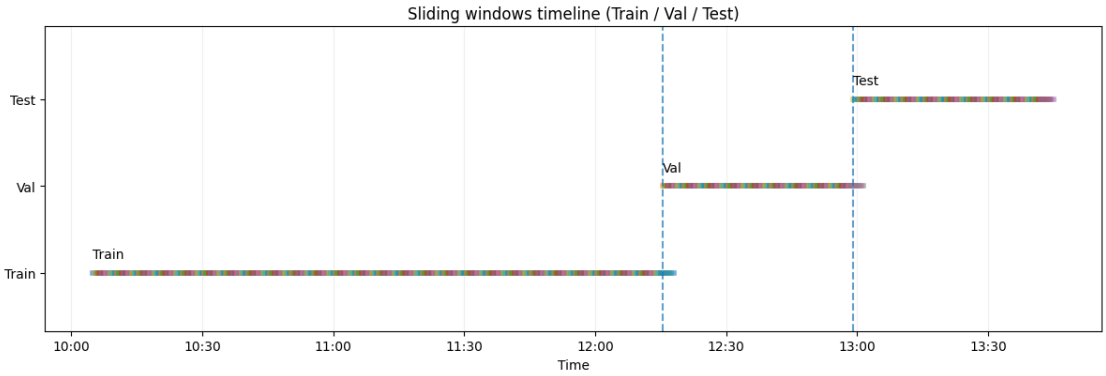
นอกจากนี้ยังใช้ Sliding Window เพื่อสร้าง sequences (window และ step) เพื่อเก็บ temporal dependency

3.3 การทำ Feature Engineering

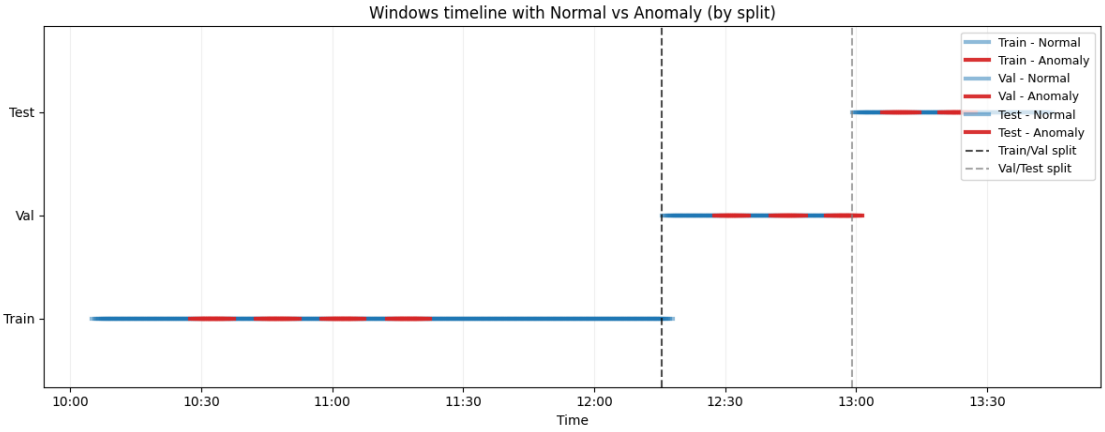
การใช้ Sliding Window ช่วยแปลงข้อมูล time-series ให้อยู่ในรูปแบบที่โมเดลสามารถเรียนรู้ pattern ได้ โดยแต่ละ window แทนค่าลำดับข้อมูลในช่วงเวลา เช่น 180 วินาที และเลื่อนด้วย step 30 วินาที และ Correlation Heatmap เพื่อวิเคราะห์ความสัมพันธ์ระหว่าง features ซึ่งใช้เป็นแนวทางในการเลือกหรือตัดคุณลักษณะที่ซ้ำซ้อนออก



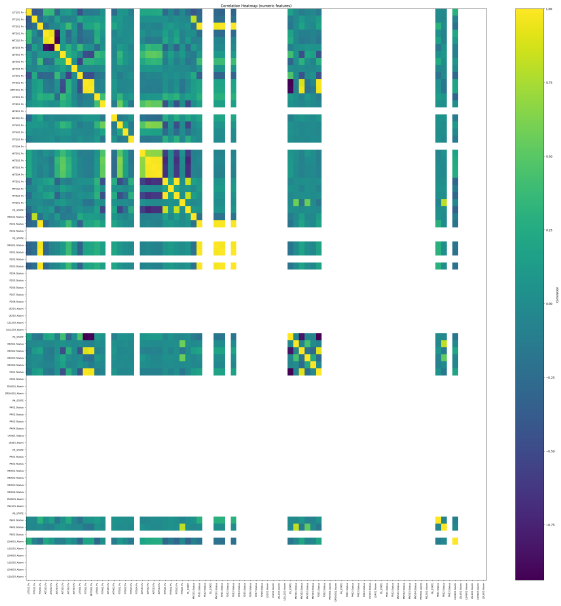
รูปที่ 3.4: จำนวนหน้าต่างต่อช่อง (ปกติ vs ผิดปกติ)



รูปที่ 3.5: ไทม์ไลน์แบบหน้าต่างเลื่อน (สำหรับ Train / Val / Test)



รูปที่ 3.6: ไทม์ไลน์ของหน้าต่างที่แสดงข้อมูล ปกติ เทียบกับ ผิดปกติ (แยกตามการแบ่งข้อมูล)



รูปที่ 3.7: แผนที่ความร้อนของค่าสหสัมพันธ์ (สำหรับตัวแปรเชิงตัวเลข)

3.3.1 การเลือกคุณลักษณะที่เกี่ยวข้อง (Feature Selection)

เพื่อให้การตรวจจับ anomaly มีประสิทธิภาพสูง ใช้วิธีเลือกคุณลักษณะ เช่น

1. Correlation Matrix เพื่อตรวจหาความสัมพันธ์ระหว่าง features
2. Mutual Information เพื่อตรวจสอบความสำคัญของแต่ละตัวแปร

3.3.2 การใช้ Sliding Window (Time-series Transformation)

ข้อมูล time-series จาก ICS ถูกแปลงให้อยู่ในรูป Window-based Input เพื่อสร้าง sequence ที่โมเดล CNN/LSTM สามารถเรียนรู้ได้

3.3.3 การสร้างคุณลักษณะใหม่ (Derived Features)

สามารถสร้าง feature เพิ่มเติมจากข้อมูลดิบ เช่น

1. ค่า moving average ของเซนเซอร์
2. ค่าความแตกต่างระหว่างเวลา (Δt)
3. อัตราการเปลี่ยนแปลง (derivatives)

สิ่งเหล่านี้ช่วยให้โมเดลจับความผิดปกติได้ดีกว่าการใช้ข้อมูลดิบเพียงอย่างเดียว

3.3.4 การตรวจสอบความสัมพันธ์ของคุณลักษณะ (Correlation Analysis)

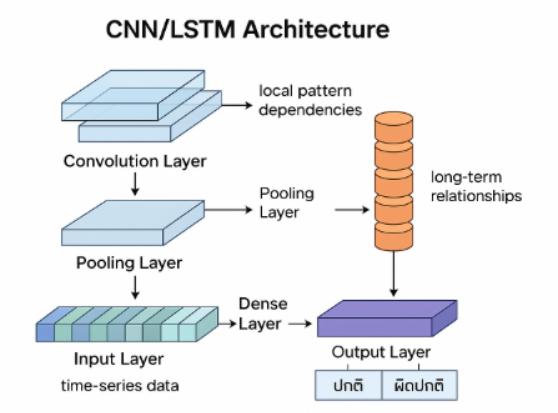
เพื่อตรวจสอบ multicollinearity หาก features มีความสัมพันธ์สูงเกินไปอาจเลือกตัดออกเพื่อลด redundancy

3.4 การออกแบบและพัฒนาโมเดล (Model Design and Development)

3.4.1 การเลือกโมเดลต้นแบบ (Prototype Model)

เลือกใช้ CNN และ LSTM เนื่องจากเหมาะกับการเรียนรู้ทั้ง local patterns (CNN) และ temporal dependency (LSTM)

3.4.2 สถาปัตยกรรมของโมเดล (Model Architecture)



รูปที่ 3.8: CNN/LSTM Architecture

1. Input Layer: รับข้อมูล time-series ที่ผ่าน preprocessing แล้ว
2. Convolutional Layer: ดึง pattern ที่ซ่อนอยู่จากข้อมูล เช่น ความเปลี่ยนแปลงของ sensor
3. Pooling Layer: ลดมิติและ noise ทำให้โมเดล generalize ได้ดีขึ้น
4. LSTM Layer : จับความสัมพันธ์ของข้อมูลตามเวลา

5. Fully Connected Layer: รวม feature ที่สกัดมาเพื่อใช้ในการจำแนก anomaly
6. Output Layer: ใช้ softmax/sigmoid เพื่อตัดสินว่าเป็น “ปกติ” หรือ “ผิดปกติ”

3.4.3 การตั้งค่า Hyperparameters

กำหนดค่าเช่น learning rate, batch size, จำนวน epoch, จำนวน filter และ kernel size ใน CNN การเลือกค่าเหล่านี้มีผลโดยตรงต่อความแม่นยำและความเร็วของการเรียนรู้

3.4.4 เครื่องมือและ Framework ที่ใช้ (เช่น TensorFlow / PyTorch)

เช่น TensorFlow, PyTorch, Scikit-learn ใช้ร่วมกับ Google Colab (GPU) เพื่อประมวลผลได้เร็วขึ้น

3.5 การทดสอบและประเมินผล (Experiment and Evaluation)

3.5.1 วิธีการแบ่งชุดข้อมูลสำหรับทดสอบ (Train/Test Strategy)

ใช้ Hold-out Validation และ Walk-forward Validation เพื่อเลียนแบบสถานการณ์จริงใน time-series

3.5.2 ตัวชี้วัดประสิทธิภาพ (Evaluation Metrics)

1. Accuracy: วัดความถูกต้องโดยรวม
2. Precision, Recall, F1-score: วัดความสามารถในการตรวจจับ anomaly
3. Detection Rate (DR), False Alarm Rate (FAR) : เน้นการวัดอัตราการตรวจจับและอัตราการแจ้งเตือนผิดพลาด
4. Confusion Matrix, ROC-AUC, PR-AUC : วิเคราะห์เชิงลึกถึงประสิทธิภาพของโมเดล

3.6 การประยุกต์ใช้งานและข้อจำกัด (Application, Integration & Limitations)

3.6.1 ศักยภาพในการนำไปใช้จริง (Practical Applications)

โมเดลที่พัฒนาสามารถนำไปใช้ในโรงงานอุตสาหกรรมจริงเพื่อเฝ้าระวังและตรวจจับพฤติกรรมผิดปกติของเซนเซอร์หรืออุปกรณ์ เช่น การโจมตีที่ปั๊มน้ำหรือวาล์ว โดยสามารถ ต่อยอดสู่ระบบจริงได้ดังนี้:

1. Integration เข้ากับ SCADA System: โมเดล anomaly detection สามารถทำงานเป็น service ภายนอกที่เชื่อมกับ SCADA ผ่าน API หรือ OPC (OLE for Process Control) โดยดึงข้อมูลจาก historian หรือ real-time tags ของเซนเซอร์/แอกชูเอเตอร์มาใช้ตรวจจับความผิดปกติ
2. การเชื่อมกับ PLC: โมเดลสามารถรับค่าจาก PLC (ผ่านโปรโตคอลเช่น Modbus/TCP หรือ EtherNet/IP) เพื่อตรวจสอบพฤติกรรมของอุปกรณ์แบบ real-time และสามารถส่ง “Alert” หรือ “Flag” กลับไปที่ PLC/SCADA เพื่อให้ระบบทำ action เช่น การหยุดปั๊ม การปิดวาล์ว หรือแจ้งเตือนผู้ปฏิบัติการ

3. Real-time Dashboard: พัฒนา dashboard ที่เชื่อมกับระบบ SCADA แสดงผลลัพธ์ anomaly detection เช่น timeline ของเหตุการณ์ผิดปกติ, อัตราการแจ้งเตือน, และสถิติการทำงาน เพื่อให้ผู้ปฏิบัติการตัดสินใจได้อย่างทันท่วงที
4. การทำงานร่วมกับระบบความปลอดภัยเดิม (Defense-in-Depth): prototype สามารถเป็น “เสริมชั้น” การป้องกัน (additional defense layer) ที่ทำงานร่วมกับ firewall, IDS/IPS ในเครือข่าย ICS เพื่อเพิ่มระดับการตรวจสอบจากระดับเครือข่ายลงสู่ระดับ process data

3.6.2 ข้อจำกัดของโครงงาน (Limitations)

1. ใช้ข้อมูลจำลอง (SWaT Dataset) ที่แม้จะใกล้เคียงโรงงานจริง แต่ไม่ครอบคลุมรูปแบบการโจมตีทั้งหมด
2. โมเดลยังไม่ผ่านการทดสอบบนสภาพแวดล้อม real-time streaming ที่มี delay/latency และ noise จาก field device จริง
3. ความแม่นยำอาจลดลงหาก dataset ที่ใช้ในอนาคตมีคุณลักษณะแตกต่างจาก SWaT (domain shift)
4. ข้อจำกัดเชิง integration เช่น ความเข้ากันได้ของโปรโตคอล (protocol compatibility) และการจัดการ real-time constraints

3.6.3 แนวทางการพัฒนาในอนาคต (Future Work)

1. พัฒนาต่อยอดจากโมเดล CNN-LSTM โดยเพิ่ม Attention Mechanisms หรือปรับใช้ Transformer-based Architectures เพื่อให้สามารถจับความสัมพันธ์ที่ซับซ้อนและยาวนานระหว่างเซนเซอร์ได้ดียิ่งขึ้น
2. พัฒนาการเชื่อมต่อกับ ระบบ SCADA/PLC โดยตรง (real-time interface) เพื่อลด latency ของการตรวจจับและตอบสนอง
3. สร้างระบบแจ้งเตือนอัจฉริยะ (Smart Alarm) ที่ลด false positive และเชื่อมกับระบบ Incident Response ของโรงงาน
4. ปรับใช้ร่วมกับระบบ Security Information and Event Management (SIEM) ขององค์กร เพื่อรวม anomaly logs เข้ากับ event logs ด้าน IT/OT อื่นๆ
5. ขยายการทดสอบกับ datasets อื่นๆ เช่น WADI หรือ ICS datasets ที่หลากหลาย เพื่อทดสอบการ generalization

บทที่ 4
การประเมินระบบ

4.1 วัตถุประสงค์การทดสอบ (Objective)

การประเมินผลของโครงงานนี้มีจุดประสงค์เพื่อวิเคราะห์ประสิทธิภาพของโมเดล CNN-LSTM ที่พัฒนาขึ้นสำหรับการตรวจจับความผิดปกติ (Anomaly Detection) ใน SWaT dataset โดยมีวัตถุประสงค์สำคัญดังนี้:

1. ตรวจสอบความสามารถของโมเดลในการจำแนกสถานะ ปกติ (Normal) และ ผิดปกติ (Anomaly) จากข้อมูล time-series ของ sensor และ actuator
2. ประเมินประสิทธิภาพตามตัวชี้วัดมาตรฐาน ได้แก่ Precision, Recall, F1-score, Area Under Curve (ROC-AUC, PR-AUC) เพื่อวัดความแม่นยำเชิงสถิติ
3. วิเคราะห์ Detection Delay และ Latency เพื่อพิจารณาความเหมาะสมในการใช้งานเชิงปฏิบัติการแบบ real-time
4. เปรียบเทียบผลลัพธ์กับโมเดล baseline เช่น CNN เดี่ยว, LSTM เดี่ยว, Autoencoder เพื่อยืนยันความเหมาะสมของ CNN-LSTM

4.2 ข้อกำหนดการทดสอบ (Requirements)

4.2.1 ข้อกำหนดเชิงฟังก์ชัน (Functional Requirements)

1. โมเดลต้องสามารถจำแนกข้อมูล Normal vs Anomaly ได้อย่างถูกต้อง
2. ระบบต้องสามารถ แจ้งเตือน (Alert System) ได้ทั้งแบบ batch processing และ real-time detection
3. รองรับการนำเข้าข้อมูลจาก sensor/actuator ที่มีโครงสร้างตาม SWaT dataset
4. มีการจัดเก็บและนำเสนอผลลัพธ์ในรูปแบบ visualization และ log files (เช่น confusion matrix, error curves)

4.2.2 ข้อกำหนดไม่เชิงฟังก์ชัน (Non-Functional Requirements)

1. Accuracy Benchmark: ค่าคะแนน F1-score ≥ 0.9
2. ค่า Precision ≥ 0.8 เพื่อจำกัดจำนวนการแจ้งเตือนผิด (False Positive) ให้อยู่ในระดับยอมรับได้
3. ค่า Recall ≥ 0.9 เพื่อให้มั่นใจว่าโมเดลสามารถตรวจจับเหตุการณ์ผิดปกติได้ครบถ้วน (High Detection Coverage)
4. Latency: การทำนายผลแต่ละ window ต้องใช้เวลา ≤ 20 ms เพื่อรองรับ real-time operation
5. Scalability: รองรับจำนวน sensor อย่างน้อย 50 ตัว และ batch size ≥ 64
6. Usability: ผลลัพธ์ต้องสามารถแสดงในรูป ตารางและกราฟ ที่เข้าใจง่ายต่อผู้ปฏิบัติการ

4.2.3 ข้อกำหนดด้านชุดข้อมูล (Dataset Requirements)

1. ใช้ SWaT dataset โดยแบ่งเป็น Normal logs สำหรับการฝึกโมเดล และ Normal + Attack logs สำหรับการทดสอบ
2. การแบ่งข้อมูลต้องเป็น Training / Validation / Testing sets โดยพิจารณาลำดับเวลาเพื่อป้องกัน data leakage
3. ต้องทำการ Preprocessing เช่น normalization, sliding window, และการแทนค่าข้อมูลที่หายไป (imputation)

4.3 กลยุทธ์การทดสอบ (Testing Strategy)

4.3.1 ประเภทของการทดสอบ (Types of Testing)

1. Functional Testing
   - ตรวจสอบความถูกต้องในการจำแนก anomaly ใน test set
2. Performance Testing
   - ประเมินผลด้วย Precision, Recall, F1-score, ROC-AUC และ PR-AUC
   - วัด latency และ throughput ของระบบ
3. Comparison Testing
   - เปรียบเทียบกับ baseline models ได้แก่ LSTM, CNN
   - วิเคราะห์ผลลัพธ์เพื่อยืนยันข้อดีของโมเดล CNN-LSTM

4.4 ผลการทดสอบเบื้องต้น (Preliminary Results)

4.4.1 ข้อมูลชุดทดลอง

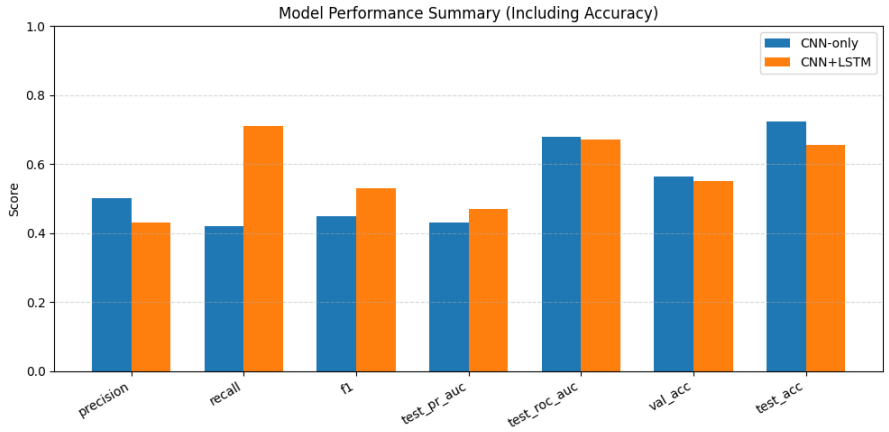
ใช้ข้อมูลจากชุด SWaT A6 ที่ผ่าน preprocessing และ oversampling แล้ว

1. ขนาดหน้าต่างเวลา (Window Length): 180 samples ($\approx$ 5 นาที)
2. ระยะเลื่อนหน้าต่าง (Step): 30 samples ($\approx$ 30 วินาที)
3. อัตราส่วนคลาสหลัง oversampling: anomaly $\approx$ 35%
4. Split: Train 60%, Validation 20% และ Test 20%
5. ชุด Validation ใช้ในการเลือก threshold (recall $\geq$ 0.7)

4.4.2 ผลการประเมินเบื้องต้น

Model	VAL (PR/ROC)	TEST (PR/ROC)	Accuracy (VAL/TEST)	Precision	Recall	F1
CNN-only	0.609 / 0.643	0.434 / 0.686	0.563 / 0.724	0.50	0.42	0.46
CNN+LSTM	0.496 / 0.616	0.471 / 0.675	0.551 / 0.655	0.43	0.71	0.53

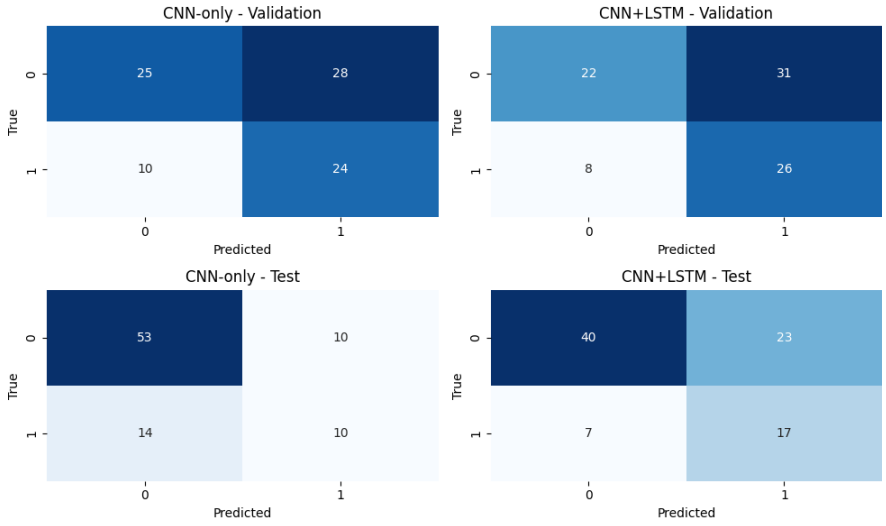
ตารางที่ 4.1: ตารางแสดงผลการประเมินเบื้องต้น



รูปที่ 4.1: ผลการประเมินเบื้องต้น

Confusion Matrix (Validation / Test):

1. CNN-only: [[25,28], [10, 24]], [[53, 10], [14, 10]]
2. CNN+LSTM: [[22, 31], [8, 26]], [[40, 23], [7, 17]]



รูปที่ 4.2: Confusion Matrix (Val/Test)

4.5 อภิปรายผลการทดสอบเบื้องต้น (Discussion of Preliminary Results)

ภาพรวมของผลการทดสอบ

1. โมเดล CNN+LSTM มีค่า Recall สูงกว่า CNN-only อย่างมีนัยสำคัญ (0.708 vs 0.417)
2. ค่า Accuracy ของ CNN-only อยู่ที่ 0.563 (Validation) และ 0.724 (Test) ในขณะที่ CNN+LSTM มีค่า 0.551 และ 0.655 ตามลำดับ แสดงว่าโมเดลทั้งสองสามารถจำแนกข้อมูลได้ในระดับหนึ่ง แต่ CNN-only มีความแม่นยำโดยรวมสูงกว่า เนื่องจากคาดเดาคลาส “ปกติ” ได้ดี
3. ค่า ROC-AUC ของทั้งสองโมเดลอยู่ในช่วง 0.67–0.69 แสดงถึงความสามารถในการจำแนกคลาสในระดับน่าพอใจ

อย่างไรก็ตาม โมเดล CNN+LSTM มีแนวโน้มตรวจจับ anomaly ได้ครบถ้วนกว่า แต่เกิดการแจ้งเตือนผิด (False Positive) มากกว่า

จุดเด่นของโมเดล CNN+LSTM

1. โครงสร้าง BiLSTM ช่วยให้โมเดลสามารถเรียนรู้ลำดับเวลา (Temporal Dependency) ของสัญญาณจาก Sensor ได้ดีกว่า CNN-only
2. ให้ค่า Recall สูง แสดงถึงความสามารถในการครอบคลุมเหตุการณ์ผิดปกติที่เกิดต่อเนื่องในช่วงเวลา
3. เหมาะกับการใช้งานในระบบตรวจจับแบบ Real-time ที่ให้ความสำคัญกับ “การไม่พลาดเหตุการณ์ผิดปกติ”
4. การเพิ่ม LSTM ยังช่วยลดความไวต่อ Noise บางประเภท เนื่องจากสามารถอ้างอิงบริบทของข้อมูลก่อนหน้าได้

จุดที่ต้องปรับปรุง

1. ค่า Precision ยังต่ำ (0.4–0.5) แสดงถึงการแจ้งเตือนผิด (false positive) ในระดับสูง
2. พบแนวโน้ม overfitting จากผลที่ validation ดีกว่า test ส่งผลให้ควรเพิ่ม regularization หรือ dropout
3. ความไม่สมดุลของข้อมูล (class imbalance) ยังคงเป็นสาเหตุหลักที่ทำให้โมเดล bias ต่อคลาสปกติ
4. CNN+LSTM มี Accuracy โดยรวมต่ำกว่า CNN-only เนื่องจากทำนาย anomaly มากเกินไป สะท้อนถึงความจำเป็นในการปรับสมดุลระหว่าง Precision และ Recall

แนวทางปรับปรุงในอนาคต

1. ทดลองใช้ Focal Loss หรือ Class Weighting เพื่อเพิ่มความแม่นยำของการจำแนก anomaly
2. ปรับ Dropout / L2 Regularization เพื่อควบคุม Overfitting
3. เพิ่ม Attention Mechanism หรือ Temporal Convolution เพื่อช่วยให้โมเดลเลือกโฟกัส feature สำคัญ

4. ทดสอบโมเดลบนกระบวนการอื่นใน SWaT (เช่น P1–P6) เพื่อประเมินความสามารถในการ generalize
5. ปรับ threshold การตัดสินใจให้เหมาะสมกับวัตถุประสงค์เชิงปฏิบัติ เช่น การใช้งาน real-time monitoring

บรรณานุกรม

[1] A causality-inspired approach for anomaly detection in a water treatment testbed. *PMC*, 2022.

[2] Cyberattack detection on swat plant industrial control systems using machine learning, No Date.

[3] AB109316. Ai training: Ts anomaly detection. Kaggle, 2024.

[4] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, and X. Zheng. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pages 265–283, 2016.

[5] Abidzar16. Isolation forest on swat 2015. Kaggle, 2022.

[6] Victor Ambonati. Unsupervised anomaly detection. Kaggle, 2017.

[7] DZone. Building ai-driven anomaly detection model to secure industrial automation, 2025.

[8] B. Kim, M. A. Alawami, E. Kim, S. Oh, J. Park, and H. Kim. Anomaly detection for industrial control systems using sequence-to-sequence neural networks. *arXiv preprint arXiv:1911.04831*, 2019.

[9] Bedeuro Kim, Mohsen Ali Alawami, Eunsoo Kim, Sanghak Oh, Jeongyong Park, and Hyoungshick Kim. A comparative study of time series anomaly detection models for industrial control systems. *PubMed*, 2023.

[10] S. Kiranyaz, T. Ince, O. Abdeljaber, O. Avci, and M. Gabbouj. 1d convolutional neural networks and applications: A survey. *Mechanical Systems and Signal Processing*, 151:107398, 2021.

[11] M. M. Macas and collaborators. An attention-based deep generative model for anomaly detection in industrial control systems. GitHub, 2023.

[12] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, and É. Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[13] ArXiv Preprint. An attention-based deep generative model for anomaly detection in industrial control systems. arXiv, 2024.

[14] Nathan Shone, Tran Nguyen Ngoc, Vu Dinh Phai, and Qiu Shi. A deep learning

approach to network intrusion detection. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2(1):41–50, 2018.

[15] Joshua Swords. Time series anomaly detection. Kaggle, 2023.

[16] Jian Zhou, Chen Li, et al. Anomaly detection for industrial control system based on autoencoder neural network. *ResearchGate*, 2020.