การวิเคราะห์แรงบิดเพื่อตรวจจับความผิดปกติวาล์วที่ใช้มอเตอร์ด้วยเทคนิคการเรียนรู้เครื่อง

ฐาปน สิทธิเลิศพิศาล

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
บัณฑิตศึกษา สาขาเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีไทย-ญี่ปุ่น
ปีการศึกษา 2568

TORQUE ANALYSIS FOR FAULT DETECTION IN MOTOR OPERATED VALVE USING MACHINE LEARNING TECHNIQUES

Thapana Sitthilertpisarn

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science Program in Information Technology
Graduate Studies
Thai-Nichi Institute of Technology
Academic Year 2025

หัวข้อวิทยานิพนธ์	การวิเคราะห์แรงบิดเพื่อตรวจจับความผิดปกติวาล์วที่ใช้มอเตอร์ด้วยเทคนิคการเรียนรู้เครื่อง
โดย	ฐาปน สิทธิเลิศพิศาล
สาขาวิชา	เทคโนโลยีสารสนเทศ
อาจารย์ที่ปรึกษาวิทยานิพนธ์	ดร.ณปภัช วิชัยดิษฐ์

บัณฑิตศึกษา สถาบันเทคโนโลยีไทย-ญี่ปุ่น อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

.....รองอธิการบดีฝ่ายวิชาการ
(รองศาสตราจารย์ ดร.วรากร ศรีเชวงทรัพย์)
วันที่.....เดือน.....พ.ศ.....

คณะกรรมการสอบวิทยานิพนธ์

.....ประธานกรรมการ
(รองศาสตราจารย์ ดร.กันต์พงษ์ วรรัตน์ปัญญา)

.....กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.ฐิติพร เลิศรัตน์เดชากุล)

.....กรรมการ
(ดร.ศรายุทธ นนท์ศิริ)

.....อาจารย์ที่ปรึกษาวิทยานิพนธ์
(ดร.ณปภัช วิชัยดิษฐ์)

ฐาปน สิทธิเลิศพิศาล : การวิเคราะห์แรงบิดเพื่อตรวจจับความผิดปกติวาล์วที่ใช้มอเตอร์ด้วยเทคนิคการเรียนรู้เครื่อง. อาจารย์ที่ปรึกษา : ดร.ณปภัช วิชัยดิษฐ์, 67 หน้า.

ในอุตสาหกรรมกระบวนการ การเกิดการหยุดเดินเครื่อง (Shutdown) โดยไม่คาดคิดส่งผลให้เกิดความเสียหายทั้งด้านการผลิตและต้นทุนการดำเนินงาน Motor-Operated Valve (MOV) เป็นหนึ่งในอุปกรณ์ที่อาจเป็นสาเหตุของการเกิด Unplanned Shutdown โดยชิ้นส่วนที่มีแนวโน้มก่อให้เกิดปัญหามากที่สุดส่วนหนึ่งคือ Drive Bush ซึ่งเป็นชิ้นส่วนที่เกิดการสึกหรอจากการใช้งาน หากสามารถคาดการณ์ระดับการสึกหรอได้ล่วงหน้า จะช่วยให้สามารถวางแผนบำรุงรักษาเชิงคาดการณ์และลดโอกาสการหยุดเดินเครื่องที่ไม่คาดคิดได้

งานวิจัยนี้พัฒนาแบบจำลองการเรียนรู้ของเครื่องสำหรับจำแนกระดับการสึกหรอของ Drive Bush ในมอเตอร์ขับวาล์ว โดยแบ่งระดับการสึกหรอออกเป็น 4 ระดับ โดยใช้ข้อมูลโปรไฟล์แรงบิด–ตำแหน่ง 1,600 ตัวอย่าง ภายใต้สภาวะการทดสอบเดียวกันแต่จำลองระดับการสึกหรอ ข้อมูลถูกแบ่งออกเป็นชุดฝึกสอน ชุดตรวจสอบความถูกต้อง และชุดทดสอบ สำหรับการประเมินแบบ Train–Test Split และเพิ่มเติมด้วยการทำ K-Fold Cross Validation เพื่อประเมินความคงที่ของแบบจำลอง

การศึกษาครั้งนี้เปรียบเทียบโมเดลแบบรวมกลุ่ม ensemble จำนวน 5 แบบ ได้แก่ Random Forest, XGBoost, AdaBoost, LightGBM และ CatBoost ภายใต้การแบ่งข้อมูลแบบ Train–Test Split โดย XGBoost และ LightGBM ให้ผลลัพธ์ดีที่สุด มีค่าความถูกต้อง (Accuracy) ได้ 0.9875 ค่าความแม่นยำ (Precision) 0.9881 ค่า Recall 0.9875 และค่า F1-Score 0.9878 และผลจาก K-Fold Cross Validation ยังยืนยันประสิทธิภาพสูงอย่างสม่ำเสมอของทั้งสองโมเดลในการจำแนกระดับการสึกหรอของ Drive Bush ขณะที่ AdaBoost ให้ผลการทำนายต่ำที่สุดเมื่อเทียบกับแบบจำลองอื่น

ผลงานวิจัยนี้ชี้ให้เห็นว่าแบบจำลอง โดยเฉพาะ XGBoost และ LightGBM มีศักยภาพสูงในการตรวจจับความผิดปกติของโปรไฟล์แรงบิดใน Motor Operated Valve และสามารถนำไปประยุกต์ใช้เป็นส่วนหนึ่งของระบบบำรุงรักษาเชิงคาดการณ์ได้ในอนาคต ทั้งนี้ การวิจัยต่อไปควรขยายการเก็บข้อมูลจากหน้างานจริงและจากวาล์วขนาดอื่นๆ รวมถึงสภาวะการเดินเครื่องที่หลากหลายมากขึ้น เพื่อเพิ่มความสามารถในการทั่วไปของแบบจำลองสำหรับการใช้งานในโรงงานอุตสาหกรรมจริง

บัณฑิตศึกษา ลายมือชื่อนักศึกษา .....

สาขาวิชา เทคโนโลยีสารสนเทศ ลายมือชื่ออาจารย์ที่ปรึกษา .....

ปีการศึกษา 2568

THAPANA SITTHILERTPISARN : TORQUE ANALYSIS FOR FAULT DETECTION IN MOTOR OPERATED VALVE USING MACHINE LEARNING TECHNIQUES. ADVISOR : DR.NAPAPHAT VICHAIDIS, 67 PP.

In processed industries, unexpected shutdowns specifically on motor operated valves can cause significant losses in both production output and operating costs. These devices trigger problems like unplanned shutdowns because of the gradual wear of the drive bush during operation. If the level of drive-bush wear can be predicted in advance, it is possible to plan predictive maintenance activities thereby reducing unplanned shutdowns.

This research develops machine learning models to classify four drive-bush wear levels in MOVs using 1,600 torque–position profiles collected under controlled, systematically simulated wear conditions. Model performance was evaluated using train–test split (training, validation and test sets) and K-fold cross-validation to assess robustness and stability.

This study compared five ensemble models: Random Forest, XGBoost, AdaBoost, LightGBM, and CatBoost. Under the train–test split, XGBoost and LightGBM achieved the best results accuracy 0.9875, precision 0.9881, recall 0.9875, F1-score 0.9878. K-fold cross-validation further validated the consistently high performance of both models in classifying drive-bush wear levels, whereas AdaBoost exhibited the lowest predictive performance.

The findings indicate both XGBoost and LightGBM have high potential for detecting abnormalities in torque profiles of MOV drive bushes and can be applied as part of a predictive maintenance system in the future. For further research, it is recommended to extend data collection to real field conditions; to valves of different sizes and operating conditions, in order to improve the generalization capability of the models for practical deployment in industrial plants.

Graduate Studies Student's Signature.....

Field of Study Information Technology Advisor's Signature.....

Academic Year 2025

กิตติกรรมประกาศ

ในการจัดทำวิทยานิพนธ์ฉบับนี้ ทางผู้วิจัยได้รับความกรุณาจาก ดร.ณปภัช วิชัยดิษฐ์ ในการเป็นอาจารย์ที่ปรึกษาวิทยานิพนธ์ และ ดร.ศรายุทธ นนท์ศิริ ที่ให้คำแนะนำและแนวทางที่มีคุณค่า อันเป็นปัจจัยสำคัญที่ทำให้งานวิจัยฉบับนี้สำเร็จลุล่วงได้ด้วยดี ทางผู้วิจัยรู้สึกซาบซึ้งและขอกล่าวขอบพระคุณมา ณ ที่นี้

นอกจากนี้ ผู้วิจัยยังได้รับคำชี้แนะที่เป็นประโยชน์จากคณะกรรมการสอบวิทยานิพนธ์ ได้แก่ รองศาสตราจารย์ ดร.กันต์พงษ์ วรรัตน์ปัญญา และผู้ช่วยศาสตราจารย์ ดร.ฐิติพร เลิศรัตน์เดชากุล ที่ได้กรุณาให้เกียรติเป็นคณะกรรมการสอบ และให้คำแนะนำอันก่อให้เกิดประโยชน์อย่างยิ่งต่องานวิจัย ผู้วิจัยขอแสดงความขอบพระคุณมา ณ ที่นี้

ผู้วิจัยขอขอบพระคุณ คุณสายสุดา นักศึกษาระดับปริญญาโท คณะเทคโนโลยีสารสนเทศ ที่ได้ช่วยเหลือในการสร้างโปรแกรมสำหรับการฝึกสอนโมเดล (Training Code) อันเป็นส่วนสำคัญที่ทำให้งานวิจัยประสบความสำเร็จ

งานวิจัยฉบับนี้ได้รับการสนับสนุนทุนในการวิจัยจาก Dr.Noriaki Kano สำหรับเป็นค่าใช้จ่ายในการดำเนินโครงการ ทำให้สามารถดำเนินงานวิจัยนี้สำเร็จลุล่วงได้อย่างสมบูรณ์ จึงกราบขอพระคุณมา ณ โอกาสนี้

ขอขอบพระคุณ บริษัท โอเอเค คอร์ปอเรชั่น (ประเทศไทย) จำกัด และห้างหุ้นส่วนจำกัด เอสซีเค แอนด์ พาร์ท ที่ได้ให้การสนับสนุนด้านกระบวนการวิจัย สถานที่ รวมทั้งคำปรึกษาที่เป็นประโยชน์ต่อการดำเนินงาน

ผู้วิจัยขอแสดงความขอบคุณ คุณพันธ์วิทย์ กังแฮ นักศึกษาระดับปริญญาโท คณะเทคโนโลยีสารสนเทศ ที่ได้ร่วมศึกษาและจบการศึกษาที่เดียวกันอีกครั้ง

ท้ายที่สุด ผู้วิจัยขอขอบพระคุณ ครอบครัวและบุคคลอันเป็นที่รัก ที่คอยสนับสนุนและเป็นกำลังใจอันยิ่งใหญ่ เป็นแรงผลักดันให้ผู้วิจัยสามารถดำเนินงานวิจัยจนสำเร็จลุล่วง

ทางผู้วิจัยหวังอย่างยิ่งว่าวิทยานิพนธ์ฉบับนี้จะเป็นประโยชน์แก่ผู้ที่สนใจในอุปกรณ์ Motor Operated Valve รวมถึงสนใจด้านปัญญาประดิษฐ์และการเรียนรู้เครื่อง เพื่อนำไปประยุกต์ใช้งานในอุตสาหกรรมหรือในอุปกรณ์อื่นๆ อันจะช่วยส่งเสริมการพัฒนาเทคโนโลยีและอุตสาหกรรมของประเทศให้มีความก้าวหน้ายิ่งขึ้น

ฐาปน สิทธิเลิศพิศาล

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ง
บทคัดย่อภาษาอังกฤษ ..... จ
กิตติกรรมประกาศ ..... ฉ
สารบัญ ..... ช
สารบัญตาราง ..... ฌ
สารบัญรูป ..... ญ

บทที่

1 บทนำ 1
1.1 ความสำคัญและที่มา 1
1.2 วัตถุประสงค์ 2
1.3 ขอบเขตงานวิจัย 2
1.4 ประโยชน์ที่คาดว่าจะได้รับ 2

2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง 3
2.1 อุตสาหกรรมอินเตอร์เน็ตประสานสรรพสิ่ง (Internet of Things : IOT) 3
2.2 แฝดดิจิทัล (Digital Twin) 4
2.3 หัวขับ (Actuator) 4
2.4 การสึกหรอ (Wear) 8
2.5 การซ่อมบำรุง (Maintenance) 10
2.6 ปัญญาประดิษฐ์และการเรียนรู้ของเครื่อง 12
2.7 ต้นไม้ตัดสินใจ (Decision Tree) 13
2.8 ป่าสุ่ม (Random Forest) 15
2.9 Gradient Boosting Trees 17
2.10 XGBoost 18
2.11 Adaptive Boosting (Adaboost) 20
2.12 LightGBM 22
2.13 Categorical Boosting (CatBoost) 24

สารบัญ (ต่อ)

บทที่		หน้า
3	วิธีดำเนินการวิจัย	27
	3.1 การจัดหาชุดข้อมูล	28
	3.2 การเตรียมข้อมูล	29
	3.3 การประเมินประสิทธิภาพของโมเดล	35
	3.4 การเปรียบเทียบประสิทธิภาพ	37
4	ผลการวิจัย	38
5	สรุปและอภิปราย	41
	5.1 การเปรียบเทียบประสิทธิภาพ	41
	5.2 ข้อจำกัดในงานวิจัย	46
	5.3 ข้อเสนอแนะในงานวิจัย	47
บรรณานุกรม		48
ภาคผนวก		52
	ภาคผนวก ก. รายละเอียดโมเดล	53
ประวัติย่อผู้วิจัย		67

สารบัญตาราง

ตาราง หน้า

2.1 Related Work 25
3.1 การแบ่งชุดข้อมูลออกเป็นชุดฝึกสอนและชุดทดสอบ 31
3.2 การแบ่งชุดข้อมูลแบบทำ K-Fold Cross Validation 31
3.3 สัดส่วนข้อมูลในแต่ละรอบของ K-Fold Cross Validation 32
3.4 ระดับความสึกหรอ 32
3.5 ระดับความหนาที่เหลืออยู่ 34
3.6 ค่า Hyperparameter 35
4.1 ผลการประเมินโมเดลแบบ Train Validation Test Split 38
4.2 ผลการประเมินโมเดลโดยมีการทำ K-fold Cross Validation 39

สารบัญรูป

รูป		หน้า
2.1	การปฏิวัติอุตสาหกรรมจากอุตสาหกรรม 1.0 จนถึง อุตสาหกรรม 4.0	3
2.2	หัวขับวาล์วไฟฟ้ายี่ห้อ “Rotork” รุ่นต่างๆ	5
2.3	หัวขับวาล์วชนิดลมและไฮดรอลิกแบบต่างๆ ยี่ห้อ “Rotork”	5
2.4	หัวขับวาล์วชนิดใช้แก๊ส	6
2.5	Manual Gate Valve	6
2.6	วาล์วทำการติดตั้งหัวขับไฟฟ้ายี่ห้อ “Rotork” (มอเตอร์ควบคุมวาล์ว)	6
2.7	บูชขับวาล์วขัดกับก้านวาล์ว	7
2.8	กราฟเปรียบเทียบค่าใช้จ่ายการบำรุงรักษาแบบต่างๆ	11
2.9	โครงสร้างต้นไม้ตัดสินใจ	14
2.10	การสุ่มตัวอย่างแบบ Bootstrapped	16
2.11	โครงสร้างการทำงานของ Random Forest	16
2.12	การทำงานของโมเดล Gradient Boosting Trees	17
2.13	การทำงานของโมเดล XGBoost	18
2.14	เปรียบเทียบการแตกใบของโมเดล LightGBM	24
3.1	ขั้นตอนการดำเนินงานวิจัย	27
3.2	มอเตอร์ควบคุมวาล์วที่ใช้ทดลอง	28
3.3	ค่าที่ได้จากหัวขับไฟฟ้า	29
3.4	ตารางที่ใช้ในการกรอกข้อมูล	29
3.5	ภาพตัดเกลียวของ Bush และ Stem Valve	33
3.6	Confusion Matrix	36
4.1	Confusion Matrix XGBoost	38
4.2	Confusion Matrix XGBoost โดยมีการทำ K-fold Cross Validation	39
5.1	Confusion Matrix AdaBoost ทั้ง Train Test Split และ K-Fold	42
5.2	Torque Profile แบบปกติ	43
5.3	Torque Profile แบบสึกหรอเล็กน้อย	44
5.4	Torque Profile แบบสึกหรอปานกลาง	44
5.5	Torque Profile แบบสึกหรอรุนแรง	45

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มา

ในอุตสาหกรรมที่เกี่ยวข้องกับการควบคุมของไหลภายในท่อ เช่น น้ำ น้ำมัน ก๊าซ หรือของเหลวอื่นๆ จะใช้วาล์วเป็นตัวควบคุมในการเปิด-ปิด ของไหลในท่อโดยใช้คนหมุน ซึ่งในอุตสาหกรรมผลิตน้ำประปา ผลิตไฟฟ้า โรงกลั่นน้ำมัน ฯลฯ ใช่ท่อขนาดใหญ่จึงทำให้มีวาล์วมีขนาดใหญ่ตามไปด้วย ทำให้ต้องใช้อุปกรณ์ในการหมุนเปิด-ปิด วาล์ว ซึ่งเรียกว่าหัวขับ (Actuator) ซึ่งจะมีจะมีหลายแบบจะมีการใช้งานที่แตกต่างกันไปตามแต่ลักษณะงาน อีกทั้งยังมีการเชื่อมต่อกับระบบสั่งงานระยะไกลในการควบคุมหัวขับไฟฟ้า (Electric Actuator) จากห้องควบคุม โดยส่วนใหญ่ใช้หัวขับไฟฟ้า ในการสั่งงานเปิด-ปิดทั่วไป ไม่ต้องการระบบฉุกเฉินในการสั่งการ วาล์วที่ถูกติดตั้งหัวขับไฟฟ้าแล้ว เราจะเรียกว่ามอเตอร์ควบคุมวาล์ว (Motor Operated Valve : MOV) การเปิด-ปิดเป็นประจำทำให้เกิดการสึกหรอจากชิ้นส่วนของวาล์วและหัวขับไฟฟ้า ทำให้ต้องมีการซ่อมบำรุงเพื่อป้องกันไม่ให้เกิดความเสียหายจากการใช้งาน ซึ่งบ่อยครั้งมีความเสียหายอย่างไม่ได้คาดคิดทำให้เกิดความเสียหายกับอุปกรณ์และการเสียโอกาสในการผลิต ทำให้ต้องเสียค่าปรับจากรอในการรับหรือจ่ายน้ำมัน และบางกรณีที่ทำให้เกิดการปนเปื้อนของผลิตภัณฑ์ ทำให้เกิดความเสียหายในการปนเปื้อนมหาศาล ดังนั้นอุปกรณ์จึงต้องมีการซ่อมบำรุง

โดยการซ่อมบำรุงมีเป้าหมายทำให้อุปกรณ์มีสภาพที่พร้อมใช้งานอยู่เสมอ ซึ่งจะครอบคลุมไปถึงการซ่อมแซมอุปกรณ์ด้วย เพื่อหลีกเลี่ยงปัญหาการผลิตหรือการบริการ เป็นไปตามที่โรงงานผู้ผลิตให้คำแนะนำ ซึ่งโดยส่วนใหญ่มักใช้เวลาในการใช้งานกำหนดรอบเวลาเป็นซ่อมบำรุงเชิงป้องกัน (Preventive Maintenance) ในบางครั้งเป็นการซ่อมบำรุงที่ทำตามรอบเวลาซึ่งทำให้มีค่าใช้จ่ายที่อาจไม่จำเป็นต้องจ่าย หรือบางครั้งเกิดความเสียหายของเครื่องจักรก่อนการซ่อมบำรุง ทำให้เกิดความเสียหายต่อการผลิตและยังทำให้เสียโอกาสอีกด้วย ในการซ่อมบำรุงเครื่องจักรและอุปกรณ์ต่างๆ

ในปัจจุบันมีการนำเทคโนโลยีปัญญาประดิษฐ์ (Artificial Intelligence : AI) และการเรียนรู้ของเครื่อง (Machine Learning : ML) เข้ามาประยุกต์ใช้ในการซ่อมบำรุงกันอย่างแพร่หลาย สามารถนำเข้าช่วยในการซ่อมบำรุงได้อย่างมีประสิทธิภาพ โดยทำการเก็บข้อมูลต่างๆ จากอุปกรณ์และใช้การเรียนรู้ของเครื่องสร้างโมเดลเพื่อใช้ในการตรวจจับความผิดปกติของอุปกรณ์ เพื่อทำการคาดการณ์การซ่อมบำรุง ก่อนระยะเวลาที่ต้องทำการซ่อมบำรุงตาม โดยจะเรียกการบำรุงรักษาเชิงคาดการณ์ [1]

1.2 วัตถุประสงค์

1.2.1 เพื่อสร้างโมเดลในการตรวจจับความผิดปกติของอุปกรณ์มอเตอร์ควบคุมวาล์ว

1.2.2 เพื่อทำการเปรียบเทียบประสิทธิภาพและหาโมเดลที่เหมาะสม

1.3 ขอบเขตงานวิจัย

1.3.1 ข้อมูลที่นำมาใช้ในงานวิจัยนี้เป็นอุปกรณ์มอเตอร์ควบคุมวาล์วยี่ห้อ Rotork รุ่น IQ12 F10A และประตูน้ำตามมาตรา API 600 [2], ขนาด 6” Class 150 ยี่ห้อ “KJS”

1.3.2 เปรียบเทียบประสิทธิภาพของโมเดลการบำรุงรักษาเชิงคาดการณ์ของมอเตอร์ควบคุมวาล์ว โดยใช้เทคนิคดังนี้

1) LightGBM
2) AdaBoost
3) XGBoost
4) CatBoost
5) Random Forest

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1.4.1 สามารถลดค่าใช้จ่ายในการซ่อมบำรุงและความเสียหายของอุปกรณ์ได้

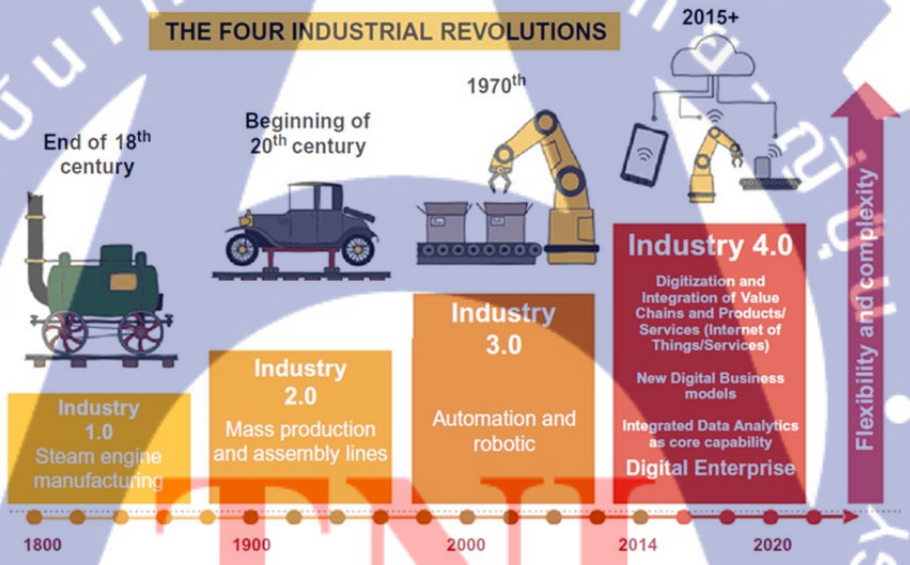
1.4.2 สามารถคาดการณ์การซ่อมบำรุงของตัวอุปกรณ์ได้

1.4.3 โมเดลที่มีประสิทธิภาพที่สุดในการตรวจจับความผิดปกติของอุปกรณ์มอเตอร์ควบคุมวาล์ว

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 อุตสาหกรรมอินเตอร์เน็ตประสานสรรพสิ่ง (Internet of Things : IOT)

ปัจจุบันเทคโนโลยีการสื่อสารและอินเตอร์เน็ตมีความเร็วสูงในการรับส่งข้อมูลด้วยเทคโนโลยี ส่งผลให้เกิดการปฏิวัติอุตสาหกรรมครั้งที่ 4 เข้าสู่ระบบการผลิตและแปรรูปสินค้าด้วยระบบอัตโนมัติ เป็นการเชื่อมต่อเครื่องจักรอุตสาหกรรมเข้ากับระบบอินเตอร์เน็ต โดยเครื่องจักรมีเซ็นเซอร์ในการตรวจวัดและส่งข้อมูลแจ้งเตือน ติดต่อสื่อสารกันเองระหว่างเครื่องจักร (Machine-to-Machine: M2M) หรือเก็บข้อมูลเพื่อวิเคราะห์ในการแก้ปัญหาได้อย่างแม่นยำและมีประสิทธิภาพ ส่งผลให้สามารถคาดการการณ์ซ่อมบำรุงรวมทั้งสามารถนำข้อมูลมาปรับปรุงระบบการผลิตได้อย่างมีประสิทธิภาพสูงสุด [3], [4]



รูปที่ 2.1 การปฏิวัติอุตสาหกรรมจากอุตสาหกรรม 1.0 จนถึง อุตสาหกรรม 4.0 [3]

จากรูปที่ 2.1 เป็นวิวัฒนาการของอุตสาหกรรมจากการเริ่มใช้เครื่องจักรไอน้ำแทนการใช้แรงงานมนุษย์ ในยุคปฏิวัติอุตสาหกรรมครั้งที่ 1 ต่อมามีการผลิตแบบสายพานช่วยเพิ่มประสิทธิภาพและลดเวลาในการผลิตเป็นยุคปฏิวัติอุตสาหกรรมครั้งที่ 2 เมื่อเข้าสู่ยุคคอมพิวเตอร์มีการนำระบบอัตโนมัติและคอมพิวเตอร์เข้าควบคุมการผลิตเป็นการปฏิวัติอุตสาหกรรมครั้งที่ 3 การเข้ามาของอินเตอร์เน็ตความเร็วสูงทำให้สามารถส่งข้อมูลขนาดใหญ่ทำได้รวดเร็ว พร้อมกับการพัฒนาเทคโนโลยีขั้นสูง เช่น อินเตอร์เน็ตประสานสรรพสิ่ง ปัญญาประดิษฐ์และการเรียนรู้ของเครื่อง ทำให้สามารถวิเคราะห์ข้อมูลแบบเรียลไทม์

ทำให้เพิ่มประสิทธิภาพในการผลิตและสามารถปรับปรุงการผลิตได้อย่างแม่นยำ เป็นปฏิวัติอุตสาหกรรมครั้งที่ 4 [5]

2.2 แฝดดิจิทัล (Digital Twin)

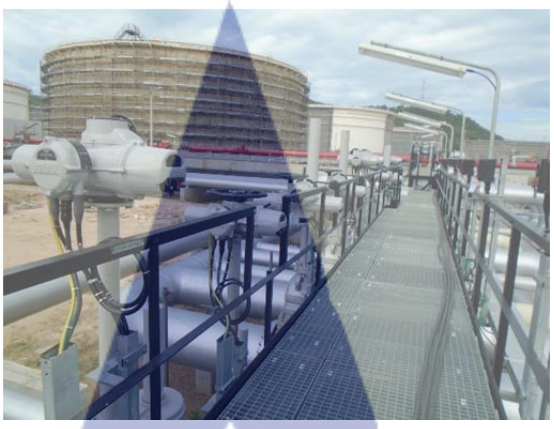
คือ แบบจำลองเสมือนของวัตถุทางกายภาพที่ถูกสร้างขึ้นจากการบูรณาการระหว่างวัตถุจริง (Physical Object) และแบบจำลองเสมือน (Virtual Model) ผ่านเทคโนโลยี Internet of Things (IoT) โดยใช้เทคโนโลยีหลายอย่าง เช่น ปัญญาประดิษฐ์ อินเตอร์เน็ตผสานสรรพสิ่ง คลาวด์คอมพิวติ้ง และเทคโนโลยีอื่นๆ เข้าด้วยกัน พร้อมทั้งเซ็นเซอร์ มิดเดิลแวร์ ซอฟต์แวร์ เพื่อเก็บข้อมูลแล้วส่งไปที่แบบจำลองฝาแฝดให้สามารถแสดงรายละเอียดและคุณสมบัติเทียบเท่าอุปกรณ์จริง และให้ข้อมูลเกี่ยวกับสิ่งที่อาจเกิดขึ้นในอนาคตโดยอาศัยหลักการวิเคราะห์ข้อมูล โดยสามารถนำมาประยุกต์ใช้ได้กับอุปกรณ์ในอุตสาหกรรม จำลองการผลิตของกระบวนการผลิต แม้แต่งานในด้านบริการ เช่น การขนส่งสินค้า แวร์เฮาส์ ตลอดถึงทางการแพทย์ก็สามารถจำลองการรักษาของผู้ป่วยในการจำลองอวัยวะเพื่อวินิจฉัยและรักษาได้แม่นยำ [6]

คำว่า “แฝดดิจิทัล” ถูกนำมาใช้ครั้งแรกในปี พ.ศ. 2546 โดย ดร.ไมเคิล กรีฟส์ (Dr. Michael Grieves) และถูกบันทึกในเอกสาร White Paper จนกลายเป็นพื้นฐานสำหรับพัฒนาแนวคิดแฝดดิจิทัล และเป็นที่รู้จักกันอย่างแพร่หลายในปี พ.ศ. 2555 โดยนาซ่า ได้เผยแพร่เอกสารเรื่อง “The Digital Twin Paradigm for Future NASA and U.S. Air Force Vehicles" ทำให้นิยามของแฝดดิจิทัลชัดเจนยิ่งขึ้น [7] จนมีการนำไปพัฒนาและใช้งานในปัจจุบัน

2.3 หัวขับ (Actuator)

เป็นอุปกรณ์ที่ติดตั้งบนวาล์วมีหน้าที่การรับคำสั่งในการควบคุมวาล์ว โดยทำการแปลงพลังงานจากรูปแบบหนึ่งเป็นการเคลื่อนที่หรือแรงทางกล โดยทั่วไปมีหลายประเภทของ Actuator ตามประเภทของพลังงานที่ใช้และลักษณะของการเคลื่อนที่ที่สร้างขึ้น ได้แก่ [8]

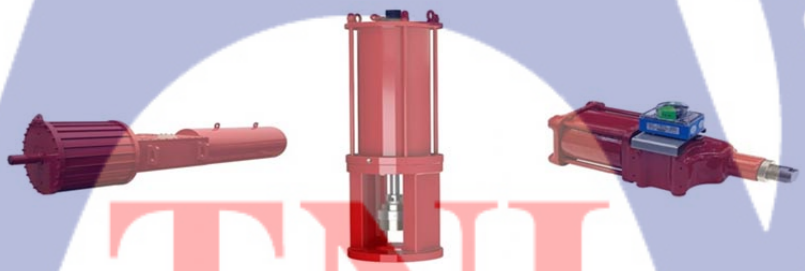
1. หัวขับวาล์วชนิดใช้ไฟฟ้า (Electric Actuators) ใช้พลังงานไฟฟ้าผ่านมอเตอร์เพื่อสร้างแรงบิด (Torque) และแรงกระทำในแนวแกน (Thrust) ในการเปิด-ปิดวาล์ว สามารถใช้ได้กับวาล์วชนิด Multi Turn และ Quarter Turn



รูปที่ 2.2 หัวขับวาล์วไฟฟ้ายี่ห้อ “Rotork” รุ่นต่างๆ

2. หัวขับวาล์วชนิดไฮดรอกลิค (Hydraulic Actuators) ใช้ของไหลประเภทน้ำมันไฮดรอลิกภายใต้แรงดัน เพื่อสร้างแรงบิดหรือแรงดึงที่สามารถเอาชนะแรงต้านของวาล์วสำหรับการเปิด-ปิดวาล์ว ตามรูปที่ 2.2

3. หัวขับวาล์วชนิดลม (Pneumatic Actuators) ใช้ลมภายใต้แรงดันเพื่อสร้างแรงบิดหรือแรงดึงที่สามารถเอาชนะแรงต้านของวาล์วสำหรับการเปิด-ปิดวาล์ว ตามรูปที่ 2.3



รูปที่ 2.3 หัวขับวาล์วชนิดลมและไฮดรอลิกแบบต่างๆ ยี่ห้อ “Rotork”

4. Gas Overoil Actuator ใช้ Gas ที่เป็นของไหลในท่อภายใต้แรงดัน เพื่อสร้างแรงบิดที่สามารถเอาชนะแรงต้านของวาล์วสำหรับการเปิด-ปิดวาล์วทำงานร่วมกับ Spring นิยมใช้กับวาล์วชนิด Quarter Turn ทำหน้าบล็อคก๊าซในท่อในกรณีมีก๊าซรั่วระหว่างสถานี โดยเมื่อตรวจจับได้ว่าแรงดันในท่อก๊าซน้อยกว่าค่าที่กำหนด ตามรูปที่ 2.3

รูปที่ 2.4 หัวขับวาล์วชนิดใช้แก๊ส

ซึ่งแต่ละประเภทของ Actuator ที่ใช้งานจะขึ้นอยู่กับฟังก์ชันการใช้งานและแหล่งพลังงานที่จะนำมาใช้ในการสร้างแรง ในที่นี้จะกล่าวถึงแต่ หัวขับไฟฟ้า ที่ทำการติดตั้งพร้อมวาล์วจะเรียกว่ามอเตอร์ควบคุมวาล์ว (Motor Operated Valve : MOV)

รูปที่ 2.5 Manual Gate Valve

รูปที่ 2.6 วาล์วทำการติดตั้งหัวขับไฟฟ้ายี่ห้อ “Rotork” (มอเตอร์ควบคุมวาล์ว)

จากรูปที่ 2.5 เป็นรูป Gate Valve ทั่วไปที่ยังไม่ได้ติดตั้ง หัวขับไฟฟ้า ซึ่งจะทำการเปิด-ปิด โดยใช้พวงมาลัย รูปที่ 2.6 เป็นรูปมอเตอร์ควบคุมวาล์ว ซึ่งหัวขับไฟฟ้าที่ได้ทำการติดตั้งกับเกิดวาล์ว การทำงานของมอเตอร์ควบคุมวาล์วสามารถสั่งงานที่ตัวหัวขับวาล์วและสามารถจะควบคุมการทำงานจากระยะไกลได้อีกด้วยผ่านระบบควบคุมแบบกระจายศูนย์ (Distributed Control System: DCS) หรือผ่านคอนโทรลเลอร์ สามารถกำหนดฟังก์ชันการควบคุมระบบวาล์วในกระบวนการรับ-จ่ายผลิตภัณฑ์ได้ ในกรณีที่เกิดไฟฟ้าดับ สามารถเปิด-ปิดวาล์วได้ด้วยมาลัย (Handwheel) เพื่อให้ระบบทำงานต่อเนื่องและปลอดภัย

มอเตอร์ควบคุมวาล์วจำเป็นต้องได้รับการบำรุงรักษาอย่างสม่ำเสมอ โดยต้องมีการเติมจาระบีหล่อลื่นที่ก้านวาล์ว เพื่อช่วยลดแรงเสียดทานและป้องกันการสึกหรอของอุปกรณ์ หากไม่ได้ทำการหล่อลื่นก้านวาล์วอาจแห้งและเกิดสนิม ส่งผลให้หัวขับไฟฟ้าต้องใช้แรงบิดสูงกว่าปกติในการทำงาน

โดยทั่วไปมอเตอร์ควบคุมวาล์วจะได้รับการซ่อมบำรุงเชิงป้องกัน (Preventive Maintenance) ทุกปี และทำการ Overhaul ทุก 5 ปี อย่างไรก็ตาม อาจมีบางชิ้นส่วนที่เสียหายก่อนถึงรอบการซ่อมบำรุง ในกรณีที่อุปกรณ์บางชิ้นของหัวขับไฟฟ้าเกิดความเสียหาย ระบบยังสามารถเปิด-ปิดวาล์วได้โดยใช้มาลัย (Handwheel) เป็นตัวช่วย แต่หาก บูชขับวาล์วเสียหาย จะทำให้ไม่สามารถเปิด-ปิดวาล์วได้ ส่งผลกระทบโดยตรงต่อการทำงานของระบบ Drive Bush ถือเป็นชิ้นส่วนสำคัญ เนื่องจากเป็นจุดสัมผัสโดยตรงกับก้านวาล์วในการถ่ายทอดแรงบิดเพื่อเปิด-ปิดวาล์ว

ตามมาตรฐานการออกแบบ บูชขับวาล์วจะใช้วัสดุที่อ่อนกว่าก้านวาล์ว เพื่อให้เป็นจุดที่เกิดการสึกหรอแทนก้านวาล์ว เพราะการเปลี่ยนบูชขับวาล์ว ทำได้ง่ายและมีความยุ่งยากน้อยกว่าการเปลี่ยนก้านวาล์ว ซึ่งช่วยลดต้นทุนและเวลาที่ใช้ในการบำรุงรักษาระบบโดยรวม

รูปที่ 2.7 บูชขับวาล์วขัดกับก้านวาล์ว

ความเสียหายจากการสึกหรอของบูชขับวาล์ว ทำให้เกลียวในตัวไม่สัมผัสกับเกลียวของก้านวาล์วทำให้ไม่สามารถหมุนวาล์วเปิด-ปิดได้ ในบางกรณีทำให้เกิดการขัดกันของบูชขับวาล์ว และ ก้านวาล์ว ซึ่งจะทำให้เกิดความเสียหายจากการที่บูชขับวาล์ว มีเกลียวที่บางลงทำให้มีการสัมผัสกับเกลียวของวาล์วหรือการขัดกันของบูชขับวาล์ว ทำให้ไม่สามารถเปิด-ปิดวาล์วได้ทำให้เกิดการปนเปื้อนของผลิตภัณฑ์ ทำให้มาสามารถการซื้อ-ขายผลิตภัณฑ์ทางเรือได้ ทำให้โดนปรับจากความล่าช้า

จากรูปที่ 2.7 บูชขับวาล์วขัดกับก้านวาล์วทำให้ไม่สามารถเปิด-ปิดวาล์วได้ ขัดกับก้านวาล์วทำให้ไม่สามารถเปิด-ปิดวาล์วได้ เนื่องจากมีสิ่งสกปรกหรือละอองโลหะเสียดสีกับก้านวาล์วจนเกิดรอยเมื่อใช้งานต่อเนื่อง บูชขับวาล์วซึ่งทำจาก Aluminum Bronze ซึ่งเป็นวัสดุที่อ่อนกว่าก้านวาล์วจะถูกเสียดสีและสึกหรอ ส่งผลให้เศษโลหะจากบูชขับวาล์วหลุดออกและสะสมอยู่ในร่องเกลียวของก้านวาล์วเมื่อเศษโลหะสะสมมากขึ้น จะทำให้บูชขับวาล์วและก้านวาล์วขัดกัน ส่งผลให้แรงบิดที่ใช้ในการเปิด-ปิดวาล์วเพิ่มขึ้น หรืออาจทำให้วาล์วติดขัดจนไม่สามารถทำงานได้ตามปกติ

เนื่องจากเป็นอุปกรณ์ที่มีความสำคัญ จึงมีการทำงานวิจัยที่นำข้อมูลการตรวจวัดค่าแรงดันจากวาล์วใต้น้ำซึ่งเป็นหัวขับชนิดไฮดรอกลิก มาค่าความผิดปกติจากการใช้งานโดยใช้เทคนิคการเรียนรู้ของเครื่องแบบ Support Vector Machines (SVM) Feed-Forward Neural Networks Classification Trees และ Complex-Valued Neural Network มาเปรียบเทียบประสิทธิภาพ [9]

วาล์วไฟฟ้าพลังงานนิวเคลียร์รุ่นใหม่ที่นำไปใช้งาน มักจะมีเฉพาะข้อมูลปกติแต่ไม่มีข้อมูลที่ผิดพลาดในการทำงานในสภาวะปกติ ทำให้มีการศึกษาน้อยมากเกี่ยวกับการตรวจจับความผิดพลาดของหัวขับไฟฟ้าในโรงไฟฟ้านิวเคลียร์ที่ไม่มีความผิดปกติในการใช้งานเลย จึงมีการนำเทคนิค Deep Auto Encoder (DAE) และ Weighted Deep Support Vector Data Description (WDSVD) มาใช้ในการฝึกฝนทำให้เพิ่มประสิทธิภาพเมื่อเทียบกับ โมเดลดังเดมอย่าง Support Vector Machines [10]

2.4 การสึกหรอ (Wear)

การสึกหรอเป็นรูปแบบหนึ่งของความเสียหายที่เกิดขึ้นกับชิ้นส่วนเครื่องจักรที่มีการเคลื่อนที่หรือเสียดสีระหว่างกันอยู่เสมอ ผลที่เกิดจากการสึกหรออาจทำให้ชิ้นส่วนเสียหายและมีค่าใช้จ่ายในการซ่อมแซมหรือเปลี่ยนใหม่สูง การศึกษาเกี่ยวกับการสึกหรอครอบคลุมความรู้ในหลายสาขา เช่น การหล่อลื่นและการลดแรงเสียดทาน ซึ่งเราจะเรียกรวมๆ ว่า “ไทรโบโลยี” (Tribology) ตามมาตรฐานอุตสาหกรรมของเยอรมัน (DIN 50320) การสึกหรอถูกจำแนกออกเป็น 4 ประเภทใหญ่ๆ ดังนี้

1. กลไกการสึกหรอแบบยึดติด (Adhesive Wear) เกิดขึ้นเมื่อพื้นผิวของชิ้นส่วนสัมผัสและเสียดสีกัน โดยเฉพาะบริเวณที่มีจุดยอดหรือพื้นผิวที่ขรุขระ ซึ่งบางครั้งอาจทำให้พื้นผิวเหล่านั้นติดกัน เมื่อชิ้นส่วนเคลื่อนที่ พื้นผิวบางส่วนอาจหลุดหรือแตกออก ทำให้เกิดการสึกหรอ เช่นเดียวกับกรณีของน็อตกับสกรูที่มีการเสียดสีขณะหมุน การสึกหรอแบบนี้สามารถทำให้เกิดความเสียหายที่ชิ้นส่วนได้

2. การสึกหรอแบบขูดขีด (Abrasive Wear) เกิดขึ้นเมื่อมีวัสดุหรืออนุภาคที่แข็งกว่าผิวชิ้นส่วนมาขูดหรือขัดเอาเนื้อวัสดุออกไป ซึ่งทำให้ชิ้นส่วนสึกหรอได้ ตัวอย่างเช่น ฝุ่นละออง เม็ดทราย หรือเศษโลหะที่ตกค้างในระบบ สามารถทำหน้าที่เหมือนกระดาษทราย ค่อยๆ ขัดชิ้นส่วนจนทำให้เกิดความเสียหาย

3. การสึกหรอจากการล้าตัวที่ผิวหน้าชิ้นงาน (Surface Fatigue Wear) เกิดจากการที่พื้นผิวของชิ้นงานต้องรับแรงกดและแรงดึงสลับกันเป็นรอบๆ อย่างต่อเนื่อง การกระทำซ้ำๆ เหล่านี้ทำให้วัสดุบริเวณผิวหน้าชิ้นงานเกิดการล้าและอาจเกิดรอยแตกหรือหลุมบนพื้นผิวได้ เมื่อมีเศษโลหะหรืออนุภาคเล็กๆ ถูกบดและแทรกอยู่ในบริเวณนี้ก็ยิ่งเร่งให้เกิดการสึกหรอได้เร็วขึ้น ตัวอย่างที่พบได้บ่อยคือการสึกหรอของเฟืองที่ต้องรับแรงสลับกัน

4. กลไกการสึกหรอแบบปฏิกิริยาไทรโบเคมี (Tribochemical Reaction) เป็นการสึกหรอที่เกิดจากการขัดสี (Tribo) และปฏิกิริยาเคมี (Reaction) โดยเฉพาะปฏิกิริยาออกซิเดชั่น และยังมีกลไกการสึกหรอแบบยึดติด แบบขูดขีด ร่วมกันอยู่ด้วย ได้แก่ โซ่ที่ไม่มีการหล่อลื่นโดยบริเวณข้อโซ่จะมีการสึกหรอแบบยึดติดและมีขนาดเล็กลง เมื่อใช้งานมีการขัดสีเกิดเศษเหล็กและบริเวณใช้งานเกิดความร้อนความชื้นทำให้ผงเหล็กเกิดปฏิกิริยาเคมีเป็นสนิม ผงเหล็กที่มีความแข็งเปราะทำให้เกิดการสึกหรอแบบขูดขีด [11]

เนื่องจาก บูชขับวาล์ว ทำจาก Aluminum Bronze และ Stem ของวาล์วทำจาก Alloy 410 Stainless Steel การทำงานรหว่าง 2 วัสดุนี้ทำให้เกิดการสึกหรอแบบยึดติด โดย Aluminum Bronze เป็นวัสดุที่อ่อนกว่าจึงเกิดการฉีกตัวขาดออกไป โดยสามารถคำนวณได้ตามหลักการของ Archard และ Rabinowicz [12]

$$V = k_{adh} \cdot \frac{Fs}{\sigma_0} \qquad (2.1)$$

V = ปริมาณการสึกหรอ

k_{adh} = ค่าคงที่ของการสึกหรอแบบ Adhesive

F = แรงกดที่ใช้

S = ระยะที่เคลื่อนที่

σ_0 = ความแข็งของวัสดุที่นุ่มกว่า

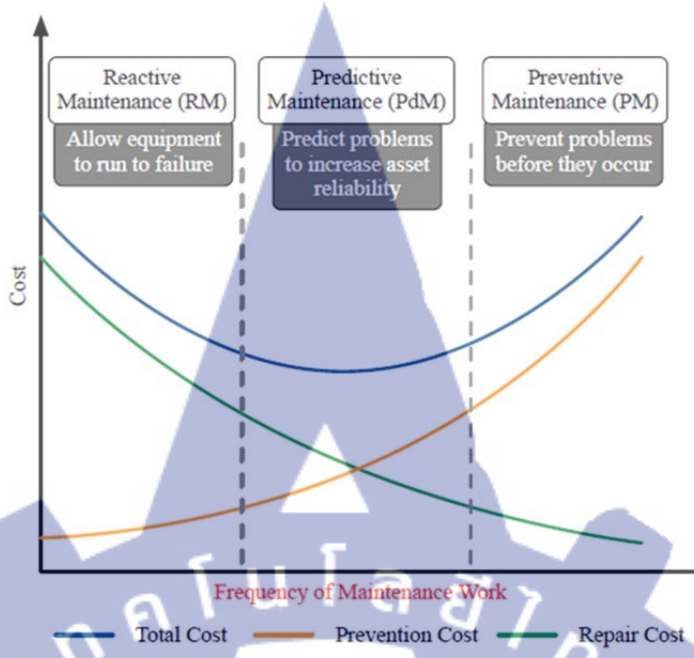
2.5 การซ่อมบำรุง (Maintenance)

การดำเนินการทางเทคนิคและการจัดการเพื่อเครื่องจักรและอุปกรณ์ที่ใช้ในงานอุตสาหกรรมสามารถใช้งานได้ตามต้องการ [13] โดยจะเป็นไปตามที่โรงงานผู้ผลิตให้คำแนะนำ ซึ่งโดยส่วนใหญ่มักจะใช้เวลาในการใช้งานกำหนดรอบเวลาซ่อมบำรุง ในบางครั้งเป็นการซ่อมบำรุงที่ทำตามรอบเวลาซึ่งทำให้มีค่าใช้จ่ายที่อาจไม่จำเป็นต้องจ่าย หรือบางครั้งเกิดความเสียหายของเครื่องจักรก่อนการซ่อมบำรุง ทำให้เกิดความเสียหายต่อการผลิตและยังทำให้เสียโอกาสอีกด้วย ในการซ่อมบำรุงเครื่องจักรและอุปกรณ์ต่างๆ จะมีหลากหลายแบบ โดยมีรายละเอียดดังนี้

1. การบำรุงรักษาเชิงรับ (Reactive Maintenance: RM) เป็นการรอให้เครื่องจักรและอุปกรณ์เกิดความเสียหายก่อน ถึงทำการซ่อมแซมให้กลับมาใช้ได้ใหม่ ทำให้ค่าเสียโอกาสและค่าซ่อมสูง

2. การบำรุงรักษาเชิงป้องกัน (Preventive Maintenance: PM) คือการซ่อมบำรุงเชิงป้องกันก่อนการเกิดความเสียหายของอุปกรณ์ ช่วงเวลาในการซ่อมบำรุงจะขึ้นอยู่กับระยะเวลาในการใช้งาน ทำให้การผลิตเป็นไปได้อย่างต่อเนื่องพร้อมทั้งวางแผนระยะยาวได้ การการซ่อมบำรุงเชิงป้องกันช่วยลดค่าใช้จ่ายในการซ่อมแซม และยังช่วยลดข้อผิดพลาดที่อาจจะเกิดขึ้นจนส่งผลกระทบต่อกระบวนการผลิต แต่ก็มีข้อเสียด้วยคือต้องเวลาให้ครบรอบในการซ่อมบำรุงเชิงป้องกัน ในบางอุปกรณ์ช้าเกินไป หรือเร็วเกิน อีกทั้งยังไม่สามารถป้องกันความเสียหายแบบเฉียบพลันได้ด้วย

3. การบำรุงรักษาเชิงคาดการณ์ (Predictive Maintenance: PdM) ระบบการบำรุงรักษาเชิงคาดการณ์ เป็นการคาดการณ์ความเสียหายของเครื่องจักรและอุปกรณ์ต่างๆ ภายในเครื่องจักร โดยใช้ข้อมูลในอดีต มีการวัดค่าพารามิเตอร์ที่สามารถนำมาวิเคราะห์ผลและความเสียหายของเครื่องจักรผ่านเครื่องมือต่างๆ การรวมทั้งการใช้ปัญญาประดิษฐ์และการเรียนรู้ของเครื่องช่วยวิเคราะห์ปัญหา ในการคาดการณ์ระยะเวลาที่เครื่องจักรจะเสียหายหรือเสื่อมสภาพที่เกิดจากการใช้งาน และสามารถจัดการวางแผนการซ่อมบำรุงเครื่องจักรได้ก่อนเวลาที่เครื่องจักรจะเกิดความเสียหายหนัก อาจจะส่งผลทำให้ต้องหยุดการผลิตกะทันหันขึ้น ซึ่งระบบนี้จะช่วยลดค่าใช้จ่ายให้กับเจ้าของกิจการในการซ่อมบำรุงแบบกะทันหันได้



รูปที่ 2.8 กราฟเปรียบเทียบค่าใช้จ่ายการบำรุงรักษาแบบต่างๆ [14]

จากรูปที่ 2.8 เป็นการเปรียบเทียบค่าใช้จ่ายการบำรุงรักษาแบบต่างๆ จะเห็นได้ว่า Reactive Maintenance จะใช้จนอุปกรณ์พังถึงจะทำการซ่อมบำรุง ไม่มีการวางแผนทำให้เกิดค่าใช้จ่ายในการซ่อมบำรุงและเสียโอกาสในการใช้งานเป็นค่าใช้จ่ายที่สูง ในส่วนของ Preventive Maintenance จะพบว่ายังอุปกรณ์ยังสามารถใช้งานต่อได้อีกระยะแต่มีการซ่อมบำรุงไปก่อน จึงเสียโอกาสที่จะใช้งานอุปกรณ์ต่อแต่ยังมีค่าใช้จ่ายน้อยกว่าแบบ Reactive Maintenance ในแบบ Predictive Maintenance จะเป็นแบบที่คุ้มค่าในการใช้งานที่สุดทำให้สามารถลดต้นทุนในการซ่อมบำรุง อีกทั้งยังช่วยลดการสูเสียโอกาสในการผลิตจากการที่อุปกรณ์มีปัญหาและไม่สามารถซ่อมแซมได้ เนื่องจากอะไหล่บางชิ้นมีระยะเวลาในการสั่งสินค้านานต้องรออะไหล่ อีกทั้งยังมีราคาแพงในการนำสินค้าดังกล่าวเพื่อให้ทันต่อการซ่อมแซม ในบางกรณีอาจจะมีค่าปรับเนื่องจากไม่สามารถผลิตสินค้าได้หรือไม่สามารถส่งผลิตภัณฑ์ได้

ดังนั้นหากสามารถคาดการณ์ความเสื่อมสภาพของอุปกรณ์ทำให้สามารถใช้งานได้สูงสุดของวงจรการใช้งาน และสั่งอะไหล่ล่วงหน้ามารอการเปลี่ยนอะไหล่ ทำให้ลดการสูญเสียโอกาสในการผลิตและอุปกรณ์ทำงานอย่างเต็มประสิทธิภาพ จะทำให้เกิดเป็นกำไรสูงสุดต่อองค์กร

2.6 ปัญญาประดิษฐ์และการเรียนรู้ของเครื่อง

ปัญญาประดิษฐ์ ความสามารถของเครื่องจักรในการเลียนแบบความฉลาดของมนุษย์ในการคิด วิเคราะห์และตัดสินใจ โดยการผสมผสานศาสตร์แขนงต่างๆ เข้าด้วยกัน ได้แก่ คณิตศาสตร์ วิศวกรรมศาสตร์ ฟิสิกส์และวิทยาการคอมพิวเตอร์ [14]

ในปี พ.ศ. 2499 จอห์น แม็กคาร์ธี (John McCarthy) ได้นิยามคำว่า “ปัญญาประดิษฐ์“ (Artificial Intelligence: AI) ขึ้นเป็นครั้งแรกในงานประชุมวิชาการ โดยมีแนวคิดว่าเครื่องจักรสามารถเรียนรู้ความฉลาดได้และมันสามารถเรียนรู้ด้วยตนเองได้ พร้อมทั้งปรับตัวเองและสามารถแก้ไขด้วยตนเองได้โดยไม่ต้องมีมนุษย์มาควบคุม [15]

<u>ปัญญาประดิษฐ์สามารถแบ่งออกได้ 3 ประเภท</u>

1. ปัญญาประดิษฐ์แคบ (Narrow AI) ถูกออกแบบมาเพื่อให้ใช้งานเฉพาะด้านเท่านั้น ไม่สามารถทำงานหรือวิเคราะห์ ได้เกินขอบเขตที่ออกแบบไว้ ได้แก่ระบบแนะนำเนื้อหาใน ได้แก่ระบบแนะนำเนื้อหาใน ระบบจดจำใบหน้าและอื่นๆ
2. ปัญญาประดิษฐ์ทั่วไป (Artificial General Intelligence: AGI) มีความสามารถเรียนรู้และทำงานรวมทั้งการคิดและวิเคราะห์ใกล้เทียบเท่ามนุษย์
3. ปัญญาประดิษฐ์ขั้นสูง (Artificial Super Intelligence: ASI) ความสามารถเรียนรู้และทำงานมีรวมทั้งการคิด วิเคราะห์และตัดสินใจได้ดีกว่ามนุษย์ [16]

การเรียนรู้ของเครื่องเป็นส่วนหนึ่งของปัญญาประดิษฐ์ที่เลียนแบบความฉลาดของมนุษย์ โดยจุดเริ่มต้นในช่วงปลายปี พ.ศ. 2493-2503 อาเธอร์ ซามูเอล (Arthur Samual) ได้คิดค้นโปรแกรมหมากฮอสแบบเรียนรู้ด้วยตนเอง โดยใช้วิธีการเรียนรู้แบบเสริมแรง (Reinforcement Learning) จนโปรแกรมเล่นได้เก่งกว่าผู้สร้างมันเอง ในช่วงเวลาเดียวกัน ริชาร์ด เบลแมน (Richard Bellman) ได้วางรากฐานทางคณิตศาสตร์ในการเรียนรู้แบบเสริมแรงขึ้น ในเวลา คาร์ล สไตน์บุช (Karl Steinbuch) ได้คิดค้น “Learning Metrix” เป็นอุปกรณ์ที่มีระบบหน่วยความจำแบบเชื่อมโยง (Associative Memory) ที่สามารถเรียนรู้และจำรูปแบบข้อมูลได้ ซึ่งเป็นพื้นฐานที่สำคัญในการพัฒนเครือข่ายประสาทเทียม (Neural Network) แฟรงค์ โรเซนแบลตต์ (Frank Rosenblatt) ได้สร้างพื้นฐานของระบบเครือข่ายประสาทเทียมและสร้างคอมพิวเตอร์ที่ใช้เครือข่ายประสาทเทียมร่วมกับ ชาร์ลส์ ไวท์แมน (Charles Wightman) ได้แก่ Mark I Perception

ในปี พ.ศ. 2523 เครือข่ายประสาทเทียมกลับมามีความนิยมอีกครั้ง โดยเจฟฟรีย์ ฮินตัน (Geoffrey Hinton) และ เทอร์รี่ เซย์นอฟสกี้ (Terry Sejnowski) โดยเจฟฟรีย์ ฮินตัน ได้รับการยกย่องว่าเป็นบิดาแห่ง Deep Learning [17]

การเรียนรู้ของเครื่อง (Machine Learning) เป็นสาขาหนึ่งของปัญญาประดิษฐ์พัฒนาอัลกอริธึม เพื่อให้เครื่องสามารถเรียนจากข้อมูลและปรับปรุงประสิทธิภาพด้วยตนเอง โดยไม่ต้องมีมนุษย์มาควบคุม สามารถจำแนกได้เป็น 3 ประเภท

1. การเรียนรู้แบบมีผู้สอน (Supervised Learning) เป็นการเรียนรู้พื้นฐานของการเรียนรู้ของเครื่อง มีการบ่งบอกหรือมีป้ายกำกับในการเรียนรู้ โดยมีบางส่วนได้ระบุคำตอบล่วงหน้าอยู่แล้ว จากนั้นเครื่องจักรจะได้รับข้อมูลชุดใหม่เพื่อให้อัลกอริธึมมาทำการวิเคราะห์ฝึกฝน (Validation Dataset) เพื่อให้ได้คำตอบที่ถูกต้องจากข้อมูลที่มีป้ายกำกับ สามารถจำแนกได้เป็น 2 แบบคือ

1.1 Classification: ใช้ในการแก้ปัญหาในการพยากรณ์ผลลัพธ์แบบหมวดหมู่ เช่น การจำแนกสัตว์ การจำแนกรูปทรงต่างๆ เป็นต้น

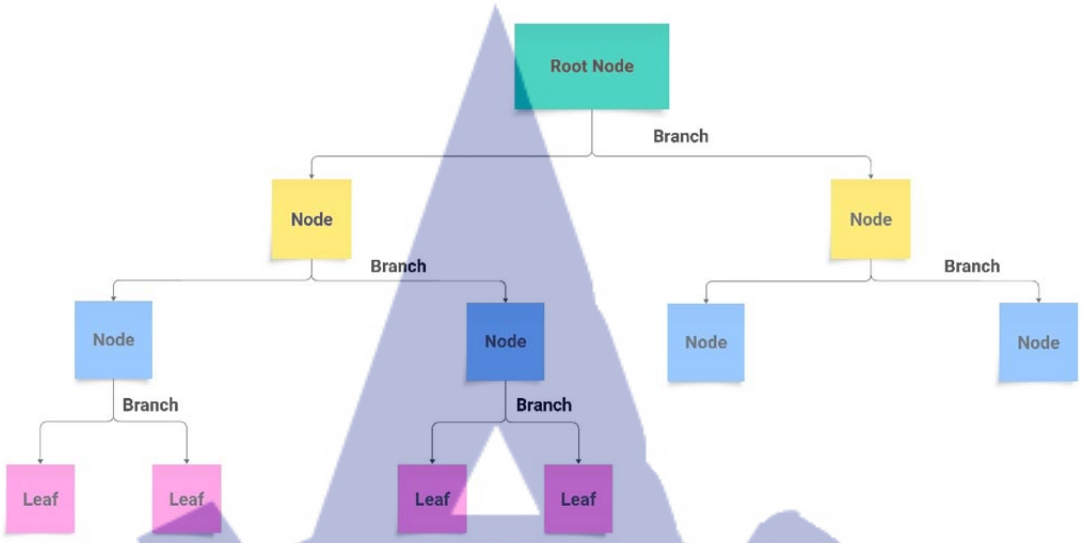
1.2 Regression: ใช้ในการแก้ปัญหาในการพยากรณ์ผลลัพธ์แบบต่อเนื่อง โดยจะพยายามหาความสัมพันธ์ระหว่างตัวแปรป้อนเข้าไปและคำตอบ เพื่อสร้างสมการในการพยากรณ์ค่าที่ต่อเนื่อง เช่น การทำนายราคาที่ดิน

2. Unsupervised Learning เป็นการเรียนรู้แบบไม่มีการบ่งบอกหรือป้ายกำกับใดๆที่คำตอบล่วงหน้าให้ ซึ่งเครื่องจักรจะต้องจำแนกข้อมูลต่างๆด้วยตัวเองจะทำงานซับซ้อนกว่าแบบ Supervised learning เช่นการจัดกลุ่ม (Clustering) หรือการลดมิติ (Dimensionality Reduction)

3. Reinforcement Learning เป็นการเรียนรู้จากการทดลองและตอบสนองโดยจะให้รางวัลเมื่อตอบถูกและมีการลงโทษเมื่อตอบผิด เพื่อเป็นพัฒนาการตัดสินใจ เช่นหุ่นยนต์ที่เล่นเกมส์

2.7 ต้นไม้ตัดสินใจ (Decision Tree)

เป็นเทคนิคการเรียนรู้ของเครื่องที่ได้รับความนิยมในการจำแนกประเภท (Classification) และการถดถอย (Regression) โดยใช้โครงสร้างของต้นไม้เพื่อแสดงการตัดสินใจตามคุณลักษณะข้อมูล โดยโมเดลนี้มีความสามารถในการจัดการข้อมูลที่ซับซ้อนมีผลลัพธ์ที่เข้าใจง่าย แนวคิดนี้ถูกนำเสนอครั้งแรกโดย รอส ควินแลน (Ross Quinlan) ในปีพ.ศ. 2527 ซึ่งพัฒนาอัลกอริธึม Iterative Dichotomies 3 (ID3) และ C4.5 เป็นที่นิยมใช้ในการจำแนกประเภท [18]



รูปที่ 2.9 โครงสร้างต้นไม้ตัดสินใจ

ตามรูปที่ 2.9 เป็นโครงสร้างของต้นไม้ตัดสินใจประกอบด้วย

1. โหนดราก (Root Node) เป็นจุดเริ่มต้นของต้นไม้ที่แสดงคุณลักษณะแรกในการแบ่งข้อมูล
2. โหนด (Node) โหนดที่ใช้เกณฑ์การแบ่งข้อมูลตามคุณสมบัติ
3. กิ่ง (Branch) เส้นเชื่อมระหว่างโหนดที่แสดงถึงการตัดสินใจ
4. โหนดใบ (Leaf Node) เป็นผลลัพธ์ของการจำแนกประเภทหรือค่าตัวแปรเป้าหมาย

เทคนิคการเรียนรู้ต้นไม้ตัดสินจะใช้ Entropy คือความไม่แน่นนอนของข้อมูลจะใช้เป็นตัววัดในการแยกโหนดหรือไม่แยกโหนด และยังใช้ Information Gain มาคำนวณด้วย โดยใช้สมการดังนี้ [19]

$$H = -\sum_{i=1}^{n} p_i \log_2 p_i \quad (2.2)$$

$$IG = H(T) - H(A) \quad (2.3)$$

$H(T)$ = เอนโทรปีทั้งหมด

P_i = ความน่าจะเป็นที่จะเกิดขึ้นในแต่ละคลาส i

IG = Information gain

n = จำนวนเหตุการณ์

$H(A)$ = เอนโทรปีที่จะพิจารณา

โดยค่า Entropy เท่ากับศูนย์จุดห้าหมายถึงมีค่าความเจือปนสูงสุด เมื่อมีค่าเท่ากับศูนย์ หมายถึงไม่มีการเจือปนเลย โดยจะคำนวณค่า Information Gain ในการแบ่ง Node เพื่อหาค่าสูงสุดการแบ่ง Node ว่าใช้ข้อมูลใดให้ผลในการให้ข้อมูลดีกว่ากันโดยจะมีการทำซ้ำเพื่อหาคุณลักษณะหรือตัวแปรที่เหมาะสมในทุกๆ Node

นอกจากนั้นยังสามารถใช้ค่า Gini Impurity ซึ่งเป็นดัชนีชี้วัดการเจือปนของข้อมูลในการแยกโหนดได้เช่นกัน โดยใช้สมการดังนี้ [20]

$$\text{Gini Impurity} = \sum_i p_i (1 - p_i) \tag{2.4}$$

เมื่อค่า p_i เป็นสัดส่วนของข้อมูลคลาส i จากข้อมูลทั้งหมดในหมวดนั้น ดังนั้น $\sum_i p_i = 1$ โดยถ้าค่า Gini Impurity เป็นศูนย์ คือไม่มีการเจือปนอีกแล้วจึงไม่สามารถแยกโหนดออกไปได้อีก แต่ถ้ามีค่าเท่ากับ 0.5 เหมือนถึงการเจือปนสูงสุดของการจำแนกข้อมูลสองคลาส

เนื่องจากการใช้งานต้นไม้ตัดสินใจมักจะพบปัญหาการจำแนกข้อมูลมากเกินไปและเมื่อนำโมเดลมาทดสอบพบว่าประสิทธิภาพจากข้อมูลทดสอบต่ำกว่าข้อมูลฝึกฝน จนทำให้เกิดการ Overfitting ของโมเดลของต้นไม้ตัดสินใจ เราสามารถแก้ปัญหา Overfitting ได้ ดังนี้

1. Ealy Stopping เป็นการบังคับให้ต้นไม้หยุดโตก่อนที่โครงสร้างของต้นไม้จะซับซ้อนจนเกินไปมีวิธีดังต่อไปนี้

    1.1 ทำการกำหนดค่าความลึกของต้นไม้

    1.2 กำหนดการแบ่งโหนดเพิ่มจากการแตกข้อมูลที่น้อยออกจากกัน โดยไปกำหนดจำนวนข้อมูลที่มีมากพอในการแตกโหนด

    1.3 กำหนดการไม่มีการแต่งโหนดเพิ่ม ถ้าไม่มีการลดการปนเปื้อนอย่างมีนัยยะสำคัญ

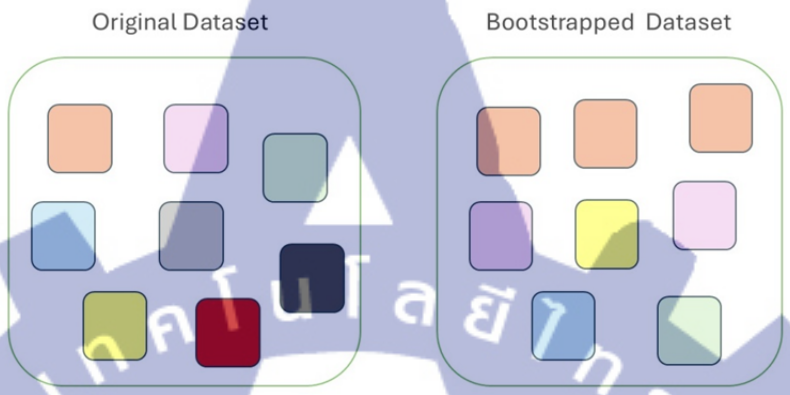
2. การตัดทอนหลังจากต้นไม้สมบูรณ์ (Post Pruning) เป็นการตัดการเปรียบเทียบ

2.8 ป่าสุ่ม (Random Forest)

เทคนิคป่าสุ่มถูกนำเสนอครั้งแรกในปี พ.ศ. 2544 โดย ลีโอ ไบรแมน (Leo Breiman) เป็นหนึ่งในเทคนิค Ensemble Learning คือการการนำคุณลักษณะต่างๆของข้อมูลชุดเดียวกัน แต่นำมาเรียนรู้ซ้ำหลายๆ ครั้ง แล้วนำผลลัพธ์ของแต่ละโมเดลมาโหวตว่าคลาสไหนถูกเลือกมากที่สุดที่ได้ [21]

โดยป่าสุ่มคือการใช้ต้นไม้ตัดสินหลายๆต้นมาทำการเรียนรู้ข้อมูลจากการสุ่มข้อมูล (Bagging : Bootstrap Aggregating) ซึ่งจะถูกแบ่งออกเป็น 2 ส่วน คือ

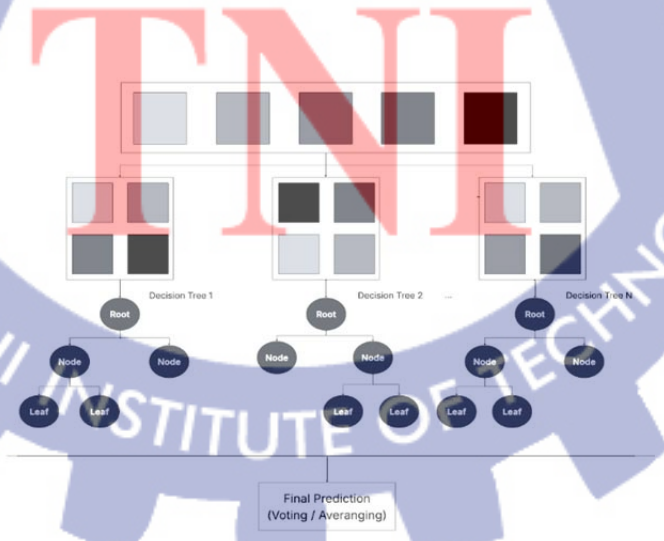
1. Bootstrap เป็นสร้างชุดข้อมูลใหม่จากชุดข้อมูลเดิม โดยสุ่มข้อมูลจากชุดข้อมูลเดิมแต่ข้อมูลเดิมก็ยังอยู่ (Bootstrap Sampling) ซึ่งจะได้ข้อมูลชุดใหม่ที่ใช้เทรนนิ่งจะต่างกันเล็กน้อยและสามารถสุ่มเลือกเฉพาะบางฟีเจอร์ในชุดข้อมูลมาเรียนรู้หรือการเทรนนิ่งซ้ำ โดยต้นไม้แต่ละต้นมีอิสระต่อกันในเทรนนิ่งและผลลัพธ์ทั้งหมดจะถูกนำกลับไปเป็นข้อมูลในชุดข้อมูลเดิมอีกที [22]



รูปที่ 2.10 การสุ่มตัวอย่างแบบ Bootstrapped

จากรูปที่ 2.10 กล่องด้านซ้ายเป็นชุดข้อมูลเริ่มต้นมีจุดข้อมูลอยู่แปดจุดไม่ซ้ำกันเลย กล่องทางด้านขวาเป็นการสุ่มจุดข้อมูลที่ทำการสุ่มมา จะพบว่าจุดข้อมูลบางตัวมีซ้ำกัน บางตัวไม่มีเลย

2. Aggregate นำผลลัพธ์ที่ผ่านการเทรนนิ่งแล้วจากข้อมูลการ Bootstrap ที่มีพารามิเตอร์ต่างกันที่เรียกว่าจำนวนต้นไม้ตัดสินใจ นำผลลัพธ์ทั้งหมดมาช่วยกันตัดสินว่าผลลัพธ์เป็นอย่างไร โดยการจำแนกข้อมูลจะนำมาโหวต แต่ถ้าเป็นแบบถอดถอยจะนำค่าเฉลี่ยมาทำนายค่า

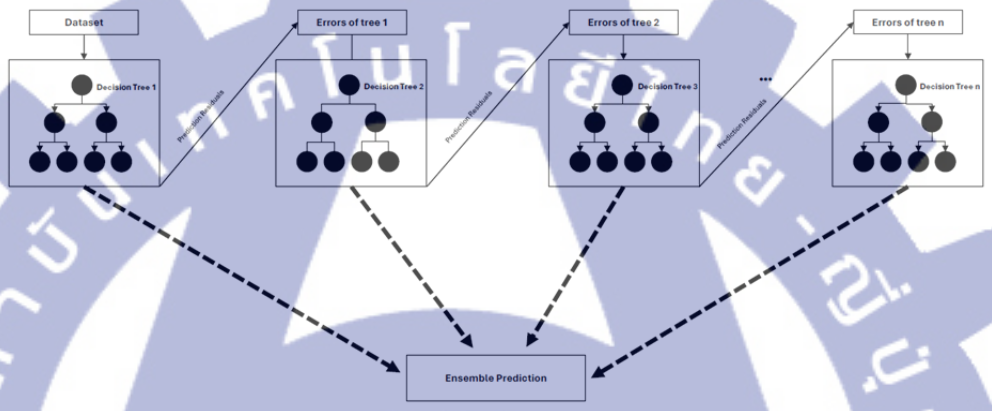


รูปที่ 2.11 โครงสร้างการทำงานของ Random Forest

จากรูปที่ 2.11 เป็นโครงสร้างโมเดลของ Random Forest ที่นำต้นไม้หลายๆ ต้น มาใช้ในการฝึกฝนและจากข้อมูลแบบ Bootstrap แล้วนำผลลัพธ์จากต้นไม้ในแต่ละต้นมา Voting กันสำหรับงานชนิด Classification และนำผลลัพธ์จากต้นไม้แต่ละต้นมาหาค่าเฉลี่ยสำหรับงานชนิด Regression

2.9 Gradient Boosting Trees

เป็นหนึ่งในเทคนิค Ensemble Learning โดยการนำต้นไม้ตัดสินใจขนาดเล็กมาทำการทำนายค่าเป็นกระบวนการวนซ้ำ (Iterative Process) โดยการเพิ่ม Weak Learner ด้วยการถ่วงน้ำหนักในแต่ละข้อมูล ทำให้การทำนายถูกต้องแล้วจะไม่ให้ความสำคัญ แต่จะสนใจกับค่าที่ทำนายผิดเพื่อปรับปรุงโมเดล



รูปที่ 2.12 การทำงานของโมเดล Gradient Boosting Trees

จากรูปที่ 2.12 โมเดล Gradient Boosting จะนำข้อมูลมาถ่วงน้ำหนักมาฝึกฝนด้วย Weak Model แล้วมาดูผลลัพธ์ที่ทำนายไม่ถูกต้อง นำกลับมาปรับปรุงโมเดลในจุดที่ยังไม่เข้าใจในแต่ละรอบ โดยการลดน้ำหนักในผลลัพธ์ที่ทำนายถูก และเพิ่มน้ำหนักในผลลัพธ์ที่ทำนายผิด เพื่อใช้ไปฝึกใน Weak Learner โมเดลถัดไป และนำค่าที่ได้ทั้งหมดมารวมกัน

โดยโมเดลจะมีทำนายค่าเฉลี่ยของตัวแปรที่จะนำมาใช้ในการทำนาย หลังจากโมเดลเริ่มทำนายแล้ว จะทำการหาค่าความผิดพลาดในการทำนาย (Residuals) ตามสมการนี้

$$y - \hat{y} \tag{2.5}$$

y = ค่าจริง

$\hat{y}$ = ค่าที่ทำนาย

จากสมการข้างต้นนำมาเขียนใหม่ในอนุพันธ์เชิงย่อยลบของฟังก์ชันการสูญเสียได้ดังนี้

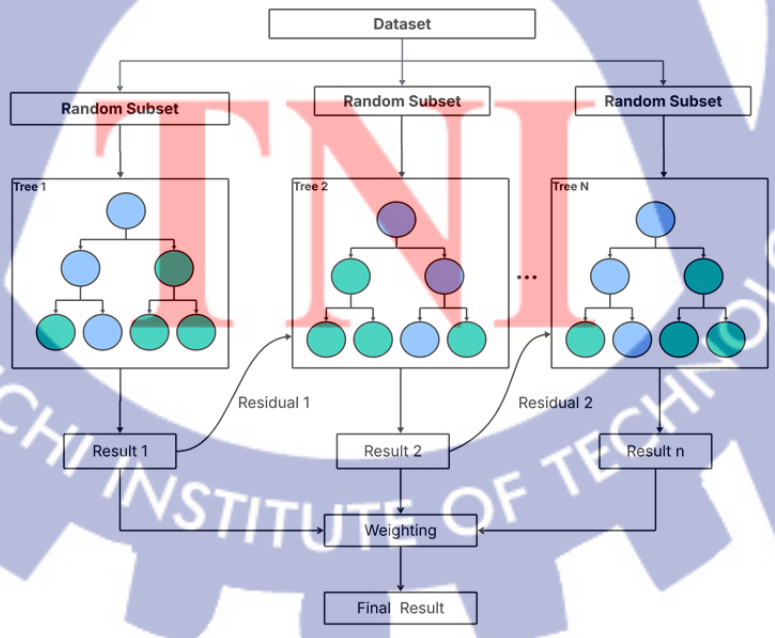
$$y - \hat{y} = -\frac{\partial L}{\partial \hat{y}} \tag{2.6}$$

ค่าที่ได้จากการคำนวณนี้จะถูกนำไปใช้ในการคำนวณในต้นไม้ตัดสินใจ ต้นไม้ตัดสินใจต้นใหม่และถูกทำซ้ำจนได้ค่าที่ถูกต้อง

การเรียนของเครื่องด้วยโมเดลนี้ให้ค่าความแม่นยำสูงเมื่อเทียบกับโมเดลป่าสุ่มแต่อาจจะเกิด Overfitting ได้ง่าย ดังนั้นต้องมีการใช้ Regularization และ Early Stop อีกทั้งยังใช้เวลานานในการฝึกโมเดล

2.10 XGBoost

เป็นหนึ่งในเทคนิค Ensemble Learning อยู่ในกลุ่มการเรียนรู้แบบ Boosting โดยมีความแตกต่างจาก Gradient Boost Tree ที่ถูกออกแบบมาให้รองรับข้อมูลขนาดใหญ่และการประมวลผลแบบขนาน มีการคำนวณเลือกจุดแบ่ง (Split point) ทำการกำหนดฟังก์ชันเป้าหมาย (Objective Function) ตามสมการระหว่างค่าความผิดพลาด (Loss Function) และค่าการปรับแต่ง (Regularized Objective) [23] โดยมีโครงสร้างการทำงานตามรูปที่ 2.13



รูปที่ 2.13 การทำงานของโมเดล XGBoost

โดย XGBoost มีสมการการตัดสินใจการเรียนรู้ของโมเดลตามสมการดังนี้

$$L(\phi) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum k\Omega(f_k) \quad (2.7)$$

$L(\phi)$ = ค่า Objective Function (Loss + Regularization)

$l(y_i, \hat{y}_i)$ = ค่าความผิดพลาดระหว่างค่าจริงและค่าที่ทำนาย

$\Omega(f_k)$ = ค่า Regularization ของ Decision Trees Function

โดยมีสมการค่าการปรับแต่ง (Regularized Objective) ช่วยป้องกัน Overfit ของโมเดลดังนี้

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda\|\omega\|^2 \quad (2.8)$$

T = ต้นไม้ที่ถูกเพิ่มไปในการเรียนรู้แต่ละรอบ

ω = ค่าถ่วงน้ำหนักของใบไม้ตัดสินใจ

γ และ λ = ค่าพารามิเตอร์ในการควบคุม Regularization

สมการ เลือกจากนั้นโมเดลจะทำนายผลโดยใช้ต้นไม้หลายๆ ต้น มารวมกัน โดยแต่ละต้นจะมีผลรวมตามสมการนี้

$$\hat{y}_i = \phi(x_i) = \sum_{k=1}^{K} f_k(x_i), f_k \in F \quad (2.9)$$

$\hat{y}_i$ = ค่าที่ทำนายได้

K = เซ็ทหรือกลุ่มทั้งหมดที่ใช้สร้างโมเดล

F = เซ็ทหรือกลุ่มทั้งหมดที่ใช้สร้างโมเดล

จากนั้นโมเดลจะทำการเรียนรู้แบบเพิ่มต้นไม้ทีละต้นเพื่อปรับปรุงการทำนายของต้นไม้ก่อนหน้าตามสมการอนุกรมเทย์เลอร์ดังนี้

$$L^{t} \approx \sum_{i=1}^{n}\left[g_i f_t(x_i) + \frac{1}{2} h_i f_t(x_i)^2 \right] + \Omega(f_t) \quad = \quad (2.10)$$

g_i = ฟังก์ชั่นสูญเสีย Gradient (อนุพันธ์อันดับหนึ่ง)

h_i = ฟังก์ชั่นสูญเสีย Hessian (อนุพันธ์อันดับสอง)

โมเดลจะทำการแบ่ง split ที่ดีที่สุดและสร้างต้นไม้ใหม่ โดยการพิจารณาแบ่งจุดที่เป็นไปได้ด้วย Greedy ตามสมการด้านนี้

$$\tau_{split} = \frac{1}{2}\left[\frac{(\sum_{i\in I_L} gi)^2}{\sum_{i_{\in I_L}} h_i + \lambda} + \frac{(\sum_{i\in I_R} g_i)^2}{\sum_{i_{\in I_R}} h_i + \lambda} - \frac{(\sum_{i\in} g_i)^2}{\sum_{i_{\in}} h_i + \lambda} \right] - \gamma \quad (2.11)$$

I_L และ I_R = กลุ่มข้อมูลที่แยกซ้ายและขวา

โดยค่าที่ให้ τ_{split} มากที่สุดจะถูกเลือกเป็นจุดแบ่งที่ดีที่สุด หรือจะเป็นจุดที่ทำการหยุดตามเงื่อนไขกำหนด ผลลัพธ์ที่ได้จากต้นไม้จะนำมาปรับปรุงโมเดลโดยรวมทันที โดยการนำค่าทำนายต้นไม้ต้นใหม่เพิ่มในค่าต้นไม้ปัจจุบัน โดยใส่ตัวคูณอัตราการเรียนรู้ (Learning Rate) แล้วคำนวณค่าผิดพลาดในการทำนาย (Residual) จะทำให้ได้ จะทำให้ได้ค่าความคาดเคลื่อนจริงกับค่าทำนายล่าสุด และนำค่าความผิดพลาดล่าสุดไปฝึกต้นไม้ต้นถัดไป จนค่าฟังก์ชันการสูญเสียไม่ลดลงแล้วหรือครบตามรอบฝึกที่กำหนด

โมเดลนี้จะสามารถคำนวณแบบคู่ขนานและใช้ Histogram-Base Split Diagram ช่วยลดจำนวนการเปรียบเทียบในทุกฟีเจอร์ทำให้ใช้เวลาลดลง

2.11 Adaptive Boosting (Adaboost)

เป็นหนึ่งในเทคนิค Ensemble Learning อยู่ในกลุ่มการเรียนรู้แบบ Boosting ถูกคิดค้นโดยโยอาฟ ฟรอยด์ (Yoav Freund) และ โรเบิร์ตอี. ชาไพร์ (Robert E. Schapire) ในปีพ.ศ. 2540 โดยการสร้าง Weak Learner ให้กลายเป็น Strong Learner ทำให้ประสิทธิภาพในการคาดการณ์ดีขึ้น เริ่มต้นโมเดลสร้างให้น้ำหนักเท่ากันในทุกข้อมูลตามสมการนี้ โดยเป็นการจำแนกข้อมูลสองชนิด [23]

$$D_1(i) = \frac{1}{m} \text{ for } i = \{1,...,m\} \quad (2.12)$$

$D_1(i)$ = ค่าถ่วงน้ำหนักเริ่มต้นให้ทุกจุดมีค่าเท่ากัน

m = จำนวนข้อมูลที่อยู่ในโดเมน

Weak Learner หลายตัวต่อเนื่องกันมาทำการฝึกฝนในการทำนายผลลัพธ์ทำนายค่าบวกหนึ่งหรือลบหนึ่ง โดยแต่ละรอบจะเลือกโมเดลที่มีการจำแนกข้อผิดพลาดต่ำที่สุดที่โดยมีการปรับแต่งน้ำหนักผิดพลาดในแต่ละรอบ โดยนั้นไปที่ตัวอย่างที่ทำนายผิดรอบก่อน กำหนดโมเดลตามฟังก์ชันนี้

$$h_t = X \rightarrow \{-1,1\} \quad (2.13)$$

ทำการคำนวณน้ำหนักผิดพลาด (Weight Error) ในแต่ละรอบการการฝึกฝนและทำนายโดยใช้สมการนี้

$$\in_t = \sum_{i=1}^{m} D_t(i).I(h_t(x_i \neq y_i) \quad (2.14)$$

$D_t(i)$ = ค่าน้ำหนักของ Data Point ที่ I ในรอบ t

h_t = ผลลัพธ์ที่ได้จากการทำนาย

y_i = ค่าผลจริง

$I(h_t(x_i \neq y_i)$ = ค่า Indicator Function ซึ่งมีค่าดังนี้

เท่ากับ 1 ถ้าผลการทำนายผิด ($h_t(x_i \neq y_i)$

เท่ากับ 0 ถ้าการทำนายถูก ($h_t(x_i = y_i)$

เลือกค่าน้ำหนักที่ดีที่สุดจาก Weak Learner หลายๆ ตัว โดยค่าน้ำหนักผิดพลาดจะคำนวณจากข้อมูลทุกตัวที่ทำนายผิด ไม่ปล่อยละเลยค่าความผิดพลาด หลังจากการทำนายรอบแรกเสร็จสิ้น ค่าของน้ำหนักจะเพิ่มขึ้นสำหรับตัวอย่างที่คาดการณ์ผิด และลดค่าน้ำหนักตัวอย่างที่คาดการณ์ถูก คำนวณค่าน้ำหนักตามสมการที่กำหนด

$$\alpha_t = \frac{1}{2}\ln\left(\frac{1-\epsilon_y}{\varepsilon_t}\right) \quad (2.15)$$

α_t = ค่าน้ำหนัก (Weight) หรือ Voting Power ที่กำหนดให้กับโมเดลย่อย (classifier) ตัวที่ t ในการโหวตสุดท้ายของ AdaBoost

ε_t = อัตราค่าความผิดพลาด (Error Rate) ของโมเดลย่อยตัวรอบ t

โมเดลทำการอัปเดตน้ำหนักตัวอย่างข้อมูลใหม่ โดยเพิ่มน้ำหนักให้กับตัวอย่างที่ทำนายผิดและลดน้ำหนักตัวอย่างที่ทำนายถูกตามสมการดังต่อไปนี้

$$D_{t+1}(i) = \frac{D_t(i).e^{-\alpha_i y_i h_t(x_i)}}{Z_t} \quad (2.16)$$

$D_t(i)$ = น้ำหนักตัวที่ I อย่างรอบที่ t

$h_t(x\_i)$ = คำตอบ Classifier ในรอบ t

Z_t = ค่าปกติมาตรฐาน (Normalization Factor) ให้ค่าผลรวมน้ำหนักเท่ากับ 1

หลังจากปรับปรุงโมเดลในทุกรอบแล้ว จะนำค่าน้ำหนัก α_t ในแต่ละรอบมารวมกันมาทำการโหวตและตัดสินผลในครั้งสุดท้ายตามสมการ

$$H(x) = \text{sign}\left(\sum_{t=1}^{T}\alpha_t h_t(x)\right) \quad (2.17)$$

H(x) = ผลลัพธ์สุดท้ายของการรวมข้อมูลจาก Classifier

α_t = ค่าน้ำหนัก (Weight) หรือ Voting Power จากการคำนวณในแต่ละรอบ

$h_t(x)$ = ผลการทำนายในแต่ละรอบ

2.12 LightGBM

โมเดลนี้ถูกพัฒนาโดยบริษัท Microsoft ใช้เทคนิคพิเศษที่เรียกว่า Gradient-Based One-Side Sample (GOSS) ทำการเลือกค่า Gradient ที่มีค่าสูงคือเป็นตัวที่ทำนายผิดพลาดจะนำข้อมูลส่วนนี้มาใช้ ในส่วน Gradient ต่ำเป็นส่วนที่ทำนายถูกอยู่แล้วจะถูกตัดออกบางส่วน ทำให้ลดจำนวนตัวอย่างในการเรียนรู้จึงใช้เวลาน้อยลง มีการคำนวณตามสมการดังนี้ [24]

$$\text{Gain} = \frac{1}{n}\left[\left(\sum_{x_i \in A_l} g_i + \frac{1-a}{b}\sum_{x_i \in B_l} g_i\right)^2 \frac{1}{n_{jl}(d)} + \left(\sum_{x_i \in A_r} g_i + \frac{1-a}{b}\sum_{x_i \in B_l} g_i\right)\right] \quad (2.18)$$

A = ข้อมูลที่ Gradient สูง เลือกทั้งหมด

B = ข้อมูลที่ Gradient ต่ำ สุ่มเลือก

a = อัตราข้อมูลที่เลือกจากกลุ่ม Gradient สูง

b = อัตราข้อมูลที่เลือกจากกลุ่ม Gradient ต่ำ

เมื่อแบ่งโหนดสองข้างซ้ายและขวาแล้วจะเรียกว่าใบโหนด (Leaf Node) จะกำหนดค่าของใบด้วยสมการดังต่อไปนี้

$$\omega^* = \frac{\sum_{i \in leaf} g_i}{\sum_{i \in leaf} h_i + \lambda} \quad (2.19)$$

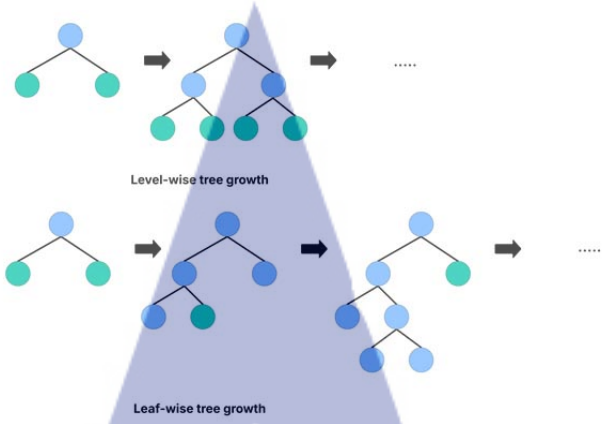
ω^* = ค่าน้ำหนักใบดีที่สุด

g_i, h_i = ค่า Gradient และค่า Hessian ที่

γ = ควบคุมความซับซ้อน (Regularization)

จากนั้นจะทำการทำการอัปเดทคำทำนายก่อนหน้า โดยใส่ตัวคูณอัตราการเรียนรู้ (Learning Rate) แล้วคำนวณค่าผิดพลาดในการทำนาย (Residual) จะทำให้ได้ค่าความคาดเคลื่อนจริงกับค่าทำนายล่าสุด และนำค่าความผิดพลาดล่าสุดไปฝึกต้นไม้ต้นถัดไป

นอกจากนั้นโมเดลจะทำการลดลดจำนวนลง Exclusive Feature Bundling (EFB) โดยถ้า Feature ไม่มีข้อมูลในแถวเดียวกัน จะจับ Feature เข้าด้วยกันเพื่อประหยัดเวลา



รูปที่ 2.14 เปรียบเทียบการแตกใบของโมเดล LightGBM

จากรูปที่ 2.14 เป็นการเปรียบเทียบการแตกใบของการทำงานของโมเดล LightGBM กับ XBGBoost (แบบ Level Wise) โดย XGBoost จะขยายต้นไม้แบบทีละระดับก่อน ในส่วนของ LightGBM จะขยายแบบ Leaf-Wise Growth คือเลือกแตกใบที่ดีที่สุดก่อน คือใบที่มีค่า Max Delta Loss สูงที่สุด

2.13 Categorical Boosting (CatBoost)

เป็นการเรียนรู้เครื่องประเภท Gradient Boosting พัฒนาโดย Yandex ในปีพ.ศ. 2560 เป็นโมเดลที่มีความสามารถในการจัดการกับข้อมูลประเภทหมวดหมู่ได้อย่างมีประสิทธิภาพ

โดยโมเดลจะทำการแปลงข้อมูล Categorical ให้เป็นตัวเลขโดยใช้ เทคนิค Ordered Target Encoding จะคำนึงถึงลำดับข้อมูลในการะทำการเพื่อป้องกันข้อมูลรั่วไหล (Data Leakage) โดยสร้างค่าเฉลี่ยของข้อมูลใหม่เพื่อนำไปใช้งาน และยังช่วยในการลด Bias ในการฝึกฝนจากโมลเดล Gradient Boost ทั่วไป นอกจากนั้นโมลเดลยังสามารถทำการจับคู่ความสัมพันธ์ของ feature อัตโนมัติ การสร้างต้นไม้ในโมเดลนี้เป็นแบบ Oblivious Trees เป็นต้นไม้ตัดสินใจที่สมมาตร (Symmetric Decision Tree) ที่ทุกโหนดใช้เกณฑ์ในการแบ่งแบบเดียวกัน จากนั้นโมเดลจะทำการฝึกฝนโดยใช้ Gradient Boosting Algorithm สร้างต้นไม้ทีละต้น แต่ละต้นจะเรียนรู้ความผิดพลาดจากต้นไม้ก่อนหน้าและทำนายผลลัพธ์สุดท้าย แต่ในโมเดลนี้จะสามารถนำมาคำนวณในแบบขนานได้รวดเร็วขึ้นและลด Overfitting

โมเดลสามารถจัดการกับ Miss Value อัตโนมัติ พร้อมทั้งทำการคำนวณแบบขนาน ซึ่งจะทำให้โมเดลทำงานได้เร็วขึ้น

ตารางที่ 2.1 Related Work

ลำดับ	อุปกรณ์	คุณสมบัติ	อัลกอริทึม	ประสิทธิภาพ	อ้างอิง
1	Electric Actuator	- Stress Signals	- DAE - WDSVVD - SVM	- DAE-WDSVVD 98.4% Accuracy	[10]
2	Pump System	- Vibration Signals - Fault Conditions	- SVM with HMM - KNN, ANN, DT - Linear Regression - Logistic Regression - Naïve Bayes - Linear Discriminant	- SVM- HMM 98.7% Accuracy	[25]
3	Subsea Valve	- Pressure Hydraulic Signal - Command	- AdaBoost - RF , DT (C5.0) - kNN , GBM - SVM , NNet - FNN	- AdaBoost F1score 0.112, 0.99 AUC	[26]
4	Hydraulic Actuator	- Hydraulic Pressure Signal	- ANN	- Errors in Histogram. Minimal Gradient Value of 9.2956×10^{-7}	[27]
5	Line Start-Permanent Magnet Synchronous Motors	- Current Signals - Torque Signal - Speed Sensor	- RF, DT (CART) - Naïve Bayes - LR, SVM - Linear Ridge Classifier	- Random Forest 98.8% Accuracy	[28]
6	Solenoid Valve	- Current Signals	- LSTM-CNN - ODE-CNN - Hyperbolic-CNN, FFN - kNN, RT, SVM - 1D Transformer Networks - Gradient Boosting	- LSTM-CNN ODE-CNN and Hyperbolic-CNN 100% Accuracy	[29]

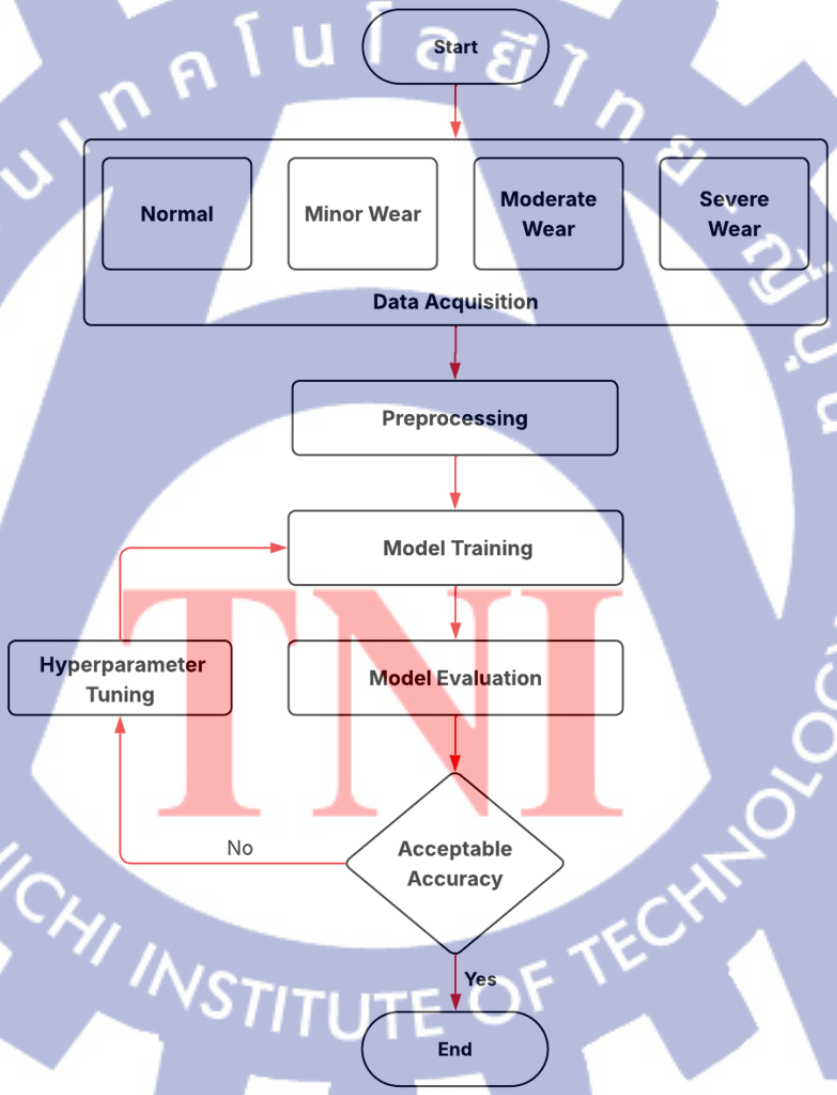
ตารางที่ 2.1 Related Work (ต่อ)

ลำดับ	อุปกรณ์	คุณสมบัติ	อัลกอริทึม	ประสิทธิภาพ	อ้างอิง
7	Virtual Sensor	- Pressure Signal - Level Signal - Pump Setpoint Signal - Valve Position Signal	- MLR - MLP - LSTM - DeepLSTM	- DeepLSTM MAE = 0.39 RMSE = 0.19 R^2 = 0.9997	[30]
8	Servo-Motor	- Vibration Data - Temperature Data - Auditory Signals - Rotational Speed - Misalignment Measurements - Torque & Load Measurements	- KNN - SVM - DNN - LSTM Autoencoder	- LSTM 98% Accuracy	[31]
9	Blowout Preventer Valves	- Hydraulic Pressure Signal - Fault Signal	- Weibull Distribution Model - ARIMA	Not Specify.	[32]

จากการทบทวนงานวิจัยในตารางที่ 2.1 พบว่างานวิจัยเดิมแตกต่างจากงานวิจัยนี้ด้านชนิดอุปกรณ์ที่ศึกษา แม้บางอุปกรณ์คล้ายกันแต่ชนิดวาล์วและหัวขับที่ใช้ไม่เหมือนกันรวมถึงการใช้โมเดลที่ซับซ้อนกว่าและตัวชี้วัดความผิดปกติที่ต่างกัน ดังนั้นงานวิจัยนี้จึงมุ่งศึกษาค่าความผิดปกติของทอร์คในการเปิด-ปิดวาล์วแต่ละครั้ง เพื่อประยุกต์ใช้กับอุปกรณ์ปัจจุบันได้อย่างมีประสิทธิภาพ

บทที่ 3
วิธีดำเนินการวิจัย

ขั้นตอนการดำเนินงานมีทั้งหมดทั้งหมด 6 กระบวนการได้แก่ ทำกาดทดลองเพื่อเก็บข้อมูลจากอุปกรณ์ ทำการจัดการข้อมูล ทำการสร้างโมเดลจากการเรียนรู้ของเครื่อง ทำการประเมินโมเดลจนกว่าจะได้ค่าที่มีความแม่นยำเป็นที่ยอมรับได้ หากยังไม่ได้จะทำการปรับจูน Hyperparameter ต่างๆ เพื่อนำให้โมเดลเรียนรู้ใหม่ จากนั้นทำการเปรียบเทียบประสิทธิภาพในแต่ละโมเดล เป็นไปตามขั้นตอนตามตามรูปที่ 3.1



รูปที่ 3.1 ขั้นตอนการดำเนินงานวิจัย

3.1 การจัดหาชุดข้อมูล

1. ทำการทดสอบ เปิด-ปิด มอเตอร์ควบคุมวาล์วพร้อมเก็บค่าทอร์คเทียบกับตำแหน่งในการเปิดและปิด มอเตอร์ควบคุมวาล์ว ตามรูปที่ 3.2 โดยจะทำการเปิด-ปิดอย่างละ 200 รอบ จากนั้นทำการถอดหัวขับไฟฟ้าออกจากวาล์วและทำการถอด Drive Bush ออกจากหัวขับไฟฟ้าและนำไปกลึงเพื่อทำให้เสมือนมีการสึกหรอเพิ่มเป็นการจำลองระดับการสึกหรอในระดับถัดไป โดยจะทำไปจนครบทั้งหมด 4 ระดับ

2. ทำการดึงข้อมูลจากโปรแกรมสำหรับมอเตอร์ควบคุมวาล์ว ตามรูปที่ 3.3 และบันทึกในรูปแบบไฟล์ Excel ตามรูปที่ 3.4

3. ทำการแบ่งข้อมูลออกเป็น 4 ชุด ได้แก่

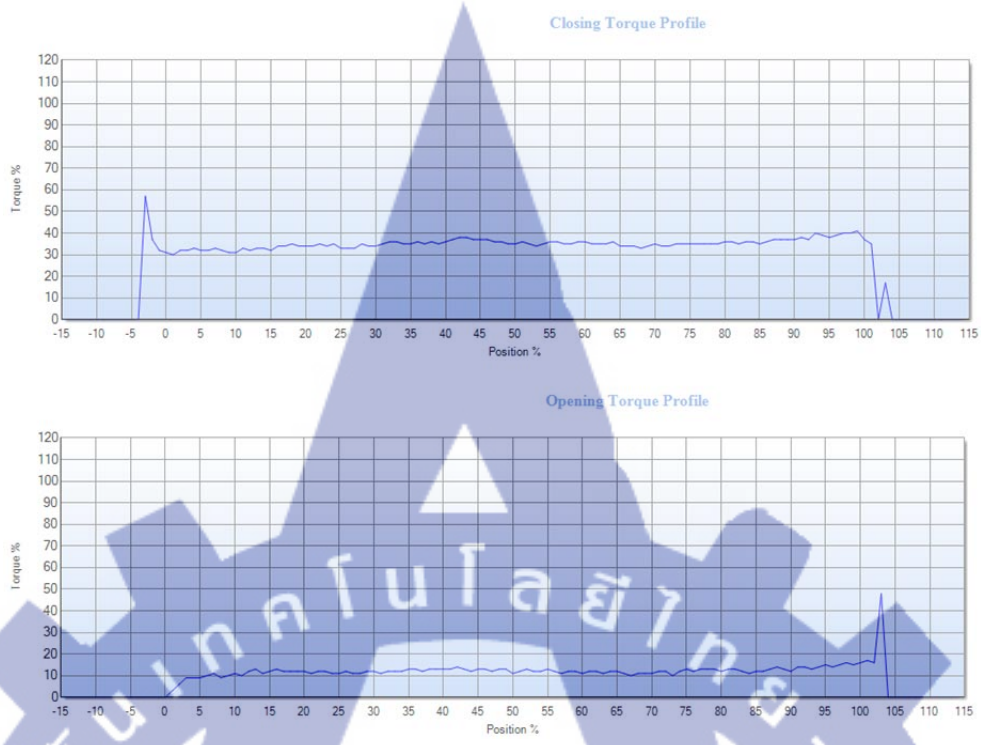
3.1 ชุดข้อมูลมอเตอร์ควบคุมวาล์วที่ใช้งานปกติ

3.2 ชุดข้อมูลสำหรับมอเตอร์ควบคุมวาล์วเกิดการสึกหรอเล็กน้อย

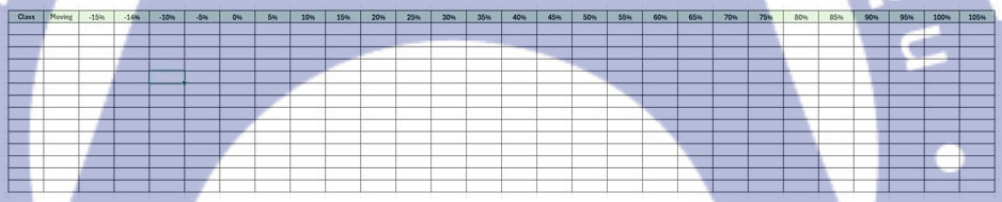
3.3 ชุดข้อมูลสำหรับมอเตอร์ควบคุมวาล์วเกิดการสึกหรอปานกลาง

3.4 ชุดข้อมูลสำหรับมอเตอร์ควบคุมวาล์วเกิดการสึกหรอรุนแรง

รูปที่ 3.2 มอเตอร์ควบคุมวาล์วที่ใช้ทดลอง



รูปที่ 3.3 ค่าที่ได้จากหัวขับไฟฟ้า



รูปที่ 3.4 ตารางที่ใช้ในการกรอกข้อมูล

3.2 การเตรียมข้อมูล

ในขั้นตอนการเตรียมข้อมูล โดยจะทำการเก็บเปอร์เซ็นต์ค่าทอร์คอยู่ระหว่าง 0% ถึง 100% เทียบกับตำแหน่งการเปิด-ปิดวาล์ว ตั้งแต่ -15% ถึง 105% ทั้งหมดจำนวน 45 คอลัมน์ต่อรอบการเปิดหรือปิด ในการเก็บข้อมูลโดยทั่วไปค่าทอร์คจะสูงในช่วงการปิดวาล์วที่ตำแหน่ง -5% ถึง 5% โดยหลังจากเปิดแล้วค่าทอร์คที่ใช้จะคงที่และจะสูงขึ้นอีกทีตอนเปิดสุดสำหรับตอนเปิดสุดในแต่ละคลาสนอกจากเก็บทุก 5% ของตำแหน่งการเปิด-ปิดวาล์ว ในกรณีที่ค่าที่สูงผิดปกติในบางตำแหน่งจะทำการเก็บค่าเพิ่มโดยการเก็บค่าที่สูงผิดปกติในบางตำแหน่งนี้ ในแต่ละคลาสจะเกิดในคนละตำแหน่งกันทำให้ค่าที่ได้จากเก็บที่ระดับปกติจะเป็นค่าเริ่มต้น เมื่อเก็บข้อมูลไปทีละคลาสทำให้ค่าที่ผิดปกติจากคลาสอื่นๆ ไม่ได้บันทึก

ในคลาสก่อนหน้าทำให้ต้องมีการจัดการนำข้อมูลจากคลาสก่อนหน้า จึงต้องทำข้อมูลให้เท่ากันในทุกคลาส เพื่อการเรียนรู้ของเครื่อง โดยการทำอินเตอร์โพเลชันแบบเชิงเส้นเป็นช่วง (Piecewise-Linear Interpolation) เพื่อประมาณและเติมเต็มค่าที่ขาดหายไป ตามสมการดังนี้

$$\Delta = \frac{T_R - T_L}{q+1}, \hat{T}_{i+m} = T_L + m\Delta \quad (3.1)$$

T_R = ค่าจริงด้านขวา

T_L = ค่าจริงด้านซ้าย

g = จำนวนจุดที่หายต่อเนื่อง

m = ลำดับช่องว่าง

จากนั้นประมวลผลข้อมูลด้วยวิธี Window Average Normalization เพื่อให้ข้อมูลมีความเรียบและอยู่ในสเกลเดียวกัน ดังสมการด้านล่าง

$$\mu_i = \frac{1}{2k+1} \sum_{j=-k}^{k} T_{i+j} \quad (3.2)$$

μ_i = ค่าเฉลี่ยเชิงหน้าต่างที่ตำแหน่ง i

i = ตำแหน่ง

j = ลำดับภายในหน้าต่าง

k = ครึ่งความกว้างหน้าต่าง

$$N_i = \frac{T_i - \mu}{\mu} \quad (3.3)$$

N_i = ค่าเบี่ยงเบนมาตรฐานที่ตำแหน่ง i

T_i = ค่าที่ทอร์คที่ตำแหน่ง i

โดยจะนำค่าที่คำนวณได้จากการทำอินเตอร์โพเลชันแบบเชิงเส้นเป็นช่วงๆ จากสมการที่ 3.1 มาคำนวณค่าเฉลี่ยกับทั้งช่วงเพื่อเปรียบเทียบกับค่าเพื่อดูว่าค่าที่ได้ไม่มากหรือน้อยเกินไปตามสมการที่ 3.2 จากนั้นนำมาคำนวณแยกทีละตัวตามสมการที่ 3.3 โดยไม่ให้เกิน $\leq$ 10% จากค่าเฉลี่ย เนื่องจากช่วงที่ขาดส่วนใหญ่จะเป็นตอนที่ค่าทอร์คจะใกล้เคียงกัน หากเกินกว่า > 10% จะไปนำค่าจริงที่ได้มาใส่

จากนั้นทำการแบ่งชุดข้อมูลออกเป็นชุดฝึก ชุดตรวจสอบ และชุดทดสอบ สำหรับใช้ในการพัฒนาและประเมินประสิทธิภาพของแบบจำลองการเรียนรู้ของเครื่องแบบการแบ่งชุดข้อมูลออกเป็นชุดฝึกสอนและชุดทดสอบ (Train Validation Test Split)

1. ข้อมูลสำหรับฝึกฝน: คิดเป็น 80% ของข้อมูลทั้งหมดจะเท่ากับ 1,280 ตัวอย่าง
2. ข้อมูลสำหรับปรับปรุงพารามิเตอร์: คิดเป็น 10% ของข้อมูลทั้งหมดจะเท่ากับ 160 ตัวอย่าง
3. ข้อมูลสำหรับทดลอง: 10% ของข้อมูลทั้งหมดจะเท่ากับ 160 ตัวอย่าง

ตารางที่ 3.1 การแบ่งชุดข้อมูลออกเป็นชุดฝึกสอนและชุดทดสอบ

ประเภทชุดข้อมูล	สัดส่วน (%)	จำนวนตัวอย่าง
Train	80%	1,280
Validation	10%	160
Test	10%	160
จำนวนตัวอย่างทั้งหมด		1,600

ในกรณีมีการทำ K-Fold Cross-Validation จะทำการโดยแบ่งชุดข้อมูลทั้งหมดออกเป็น 2 ส่วน ได้แก่ ชุดฝึกสอน-ตรวจสอบความถูกต้อง (Training-Validation Set) จำนวน 90% และชุดทดสอบ (Test Set) จำนวน 10% จากนั้นจึงนำชุด Training-Validation มาทำการประเมินด้วยวิธี Stratified K-Fold Cross-Validation โดยแบ่งออกเป็น 5 ส่วนเท่าๆ กัน (5-Fold) พร้อมทั้งสุ่มสลับลำดับข้อมูล (Shuffle = True, Random State = 42) และวนซ้ำ 5 ครั้ง โดยแต่ละครั้งใช้ 1 ส่วนเป็นชุดตรวจสอบความถูกต้องและใช้ 4 ส่วนที่เหลือเป็นชุดฝึกสอน

ตารางที่ 3.2 การแบ่งชุดข้อมูลแบบทำ K-Fold Cross Validation

ประเภทชุดข้อมูล	สัดส่วน (%)	จำนวนตัวอย่าง
Training & Validation	90%	1,440
Test	10%	160
จำนวนตัวอย่างทั้งหมด		1,600

ตารางที่ 3.3 สัดส่วนข้อมูลในแต่ละรอบของ K-Fold Cross Validation

ประเภทชุดข้อมูล	สัดส่วน (%)	จำนวนตัวอย่าง
Training (4 Folds จากชุด Training-Validation)	72%	1,152
Validation (1 Fold จากชุด Training-Validation)	18%	288
Test	10%	160
จำนวนตัวอย่างทั้งหมด		1,600

การแบ่งระดับความสึกของ Drive Bush จะแบ่งออกเป็น 4 ระดับ ได้แก่

(1) ระดับปกติ

(2) ระดับการสึกหรอเล็กน้อย (Minor Wear)

(3) ระดับการสึกหรอปานกลาง (Moderate Wear)

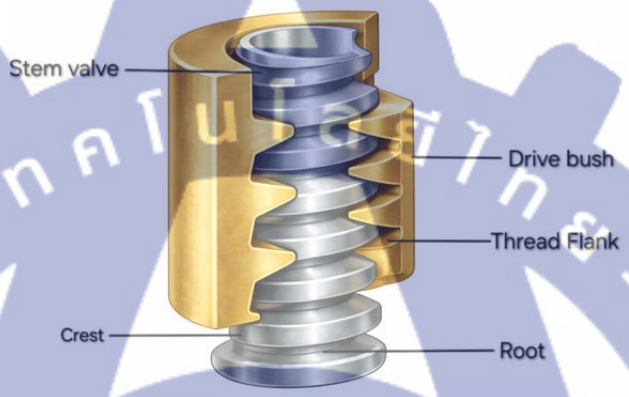
(4) ระดับการสึกหรอรุนแรง (Severe Waer)

ตามมาตรฐานของหัวขับไฟฟ้า EN-15714-2 [33] และวาล์วตามมาตรา API 600 เนื่องจากระดับการสึกหรอของ Drive Bush ไม่ได้เระบุเอาไว้ในมาตรฐานดังกล่าว โดยก้านของวาล์วมีขนาด เส้นผ่านศูนย์กลางขนาด 1 นิ้ว มีเกลียว 5 เกลียวต่อนิ้ว เป็นไปตามมาตรฐาน B 1.5 A Screw Threads [34] แต่มาตรฐานดังกล่าวไม่ได้ระบุค่าความสึกหรอในการใช้งาน แต่มีการระบุค่าสูงสุดและต่ำสุดของเกลียว พร้อมทั้ง ค่าความคลาดเคลื่อนที่ยอมรับได้ สำหรับการผลิตและการยอมรับชิ้นงานใหม่ ดังนั้นงานวิจัยนี้จึงใช้มาตรฐานดังกล่าวสำหรับกำหนดระดับการสึกหรอของ Drive Bush โดยเฉพาะเกลียวใน (Internal Thread) โดยมีช่วงการใช้งานเริ่มต้นไม่ต่ำกว่า 22.86 มิลลิเมตร และสูงสุดที่ 22.96 โดยมีค่าความคลาดเคลื่อนที่ยอมรับได้ ± 0.102 มม. จึงจะใช้ค่าดังกล่าวเป็นสภาวะปกติ (Normal) ชิ้นงานใหม่ยังไม่ถูกใช้งานและใช้เป็นจุดตั้งต้นในการแบ่งระดับความสึกหรอ โดยกำหนดระดับความสึกหรอตามตารางที่ 3.4

ตารางที่ 3.4 ระดับความสึกหรอ

Class	Max. Internal Pitch Diameter	Deviation from Standard Tolerance
Normal	22.96 mm.	±0.102 mm.
Minor Wear	23.26 mm.	+ 0.30 mm.
Moderate Wear	23.56 mm.	+0.60 mm.
Severe Wear	23.86 mm.	+0.90 mm.

การเลือก Pitch Diameter เป็นตัวแทนความสึกหรอ เนื่องจากจะบ่งชี้ถึงการประกบกันของปีกเกลียว (Flank) ซึ่งจะเป็นพื้นที่ในการถ่ายแรงแกนในกลไกการส่งกำลังมากกว่ายอดเกลียว (Crest) หรือท้องเกลียว (Root) เมื่อ Drive Bush สึกจริง การสึกมักเกิดที่ผิวปีกเกลียว ตามรูปที่ 3.5 ทำให้ค่า Pitch Diameter ของเกลียวในเพิ่มขึ้น ทำให้เกลียวหลวมขึ้น เมื่อ Drive Bush สึกที่ปีกเกลียว จะทำให้จุดสัมผัสเหลือแคบลง แรงเดิมจึงถูกกดลงบนพื้นที่ที่เล็กลงทำให้ฝืดขึ้น และเมื่อกลับทิศจะมีระยะฟรีก่อนที่ปีกเกลียวจะมาชนกัน จึงเกิดการกระชากและแรงบิดพุ่งเป็นช่วงๆ โดยจะส่งผลให้วัสดุที่อ่อนกว่า คือ Aluminum Bronze ซึ่งอ่อนกว่าสึกออกมาเศษโลหะไปติดอยู่ในร่องเกลียว



รูปที่ 3.5 ภาพตัดเกลียวของ Bush และ Stem Valve

จากรูปที่ 3.5 แสดงให้เห็นการประกบกันปีกเกลียว (Thread Flank) ของ Stem และ Drive Bush ซึ่งจะเป็นส่วนในการรับแรง และแสดงส่วนประกอบของเกลียว

ดังนั้นการแบ่งระดับความสึกหรอเป็น 4 ระดับ และการกำหนดให้เพิ่มทีละ 0.3 มม. ในการกลึง Internal Pitch Diameter ตามลำดับ จะส่งผลให้เพิ่มช่องว่างที่ระดับ Pitch และความหนาลดลง เพื่อให้เห็นความสัมพันธ์ระหว่างความหนามาตรฐานกับความสูงของเกลียวมาตรฐานสำหรับ Stem ขนาด 1 นิ้ว (25.4 มม.) ชนิด 5 เกลียวนิ้ว (TPI) โดยจะคำนวณจาก ระยะ Pitch ต่อเกลียว จะได้ ค่า Pitch เป็น 5.08 มม. จากจะคำนวณความหนามาตรฐาน (t_0) จากสมการ

$$t_0 = \frac{P}{2} \tag{3.4}$$

$$t_0 = 2.54 \text{ มม.}$$

แล้วนำมาคำนวณหาค่าความหนาคงเหลือจากสมการ

$$t_{remaining} = t_0 - t_{wear} \quad (3.5)$$

$t_{remaining}$ = ความหนาที่เหลือ

t_{wear} = ระยะที่สึกออก

โดยค่าที่ได้จากการคำนวณในแต่ละระดับความสึกหรอจะได้ค่าตามตารางที่ 3.5

ตารางที่ 3.5 ระดับความหนาที่เหลืออยู่

Class	ระยะที่สึก (mm.)	ความหนาที่เหลือ (mm.)	% การสึกหรอ
Normal	0.00	2.54	0.00
Minor	0.30	2.24	11.81
Moderate	0.60	1.94	23.62
Severe	0.90	1.64	35.43

หมายเหตุ : ค่าความหนาที่เหลืออยู่และเปอร์เซ็นต์การสึกที่ได้จากการคำนวณเพื่อให้เห็นการสึกหรอ

จากตารางที่ 3.5 จะเห็นว่าค่าเปอร์เซ็นต์การสึกหรอที่เกิดขึ้นที่ต่างกันจะส่งผลต่อความหนาและการรับแรงที่ปีกเกลียว ซึ่งเมื่อความหนาของปีกเกลียวลดลงจากการสึกหรอ พื้นที่รับแรงตามแนวปีกเกลียวจะลดลง ส่งผลให้ความเค้นเฉลี่ยเพิ่มขึ้น แม้แรงตามแกนที่กระทำจะเท่าเดิม ส่งผลให้ความสามารถในการรับแรงของชุดเกลียวลดลง และเพิ่มความเสี่ยงต่อการเกิดความเสียหายแบบเกลียวรูด (Thread Stripping) โดยค่าความแต่งต่างในการสึกแต่คลาสมากพอที่จะแยกเป็นระดับการสึกหรอได้

การจำกัดค่าที่จำลองไว้สูงสุด 0.9 mm. เนื่องจากการกลึงออกมากกว่านี้จะทำให้เกิดปัญหาในการรับแรงของ drive bush อาจจะทำให้รูดและทำให้ไม่สามารถที่จะเก็บข้อมูลได้ครบ

งานวิจัยนี้จะทำการสร้างโมเดลการเรียนรู้เครื่อง เพื่อให้สามารถตรวจจับได้ว่าค่าทอร์คที่ได้รับมีความผิดปกติหรือไม่ โดยใช้เทคนิคดังต่อไปนี้

(1) เทคนิค LightGBM

(2) เทคนิค AdaBoost

(3) เทคนิค XGBoost

(4) เทคนิค CatBoost

(5) เทคนิค Random Forest

สำหรับแต่ละโมเดล จะทำการประเมินแบบ การแบ่งชุดข้อมูลออกเป็นชุดฝึกสอนและชุดทดสอบ (Train Validation Test Split) และแบบ K-Fold Cross-Validation โดยปรับแต่งค่าไฮเปอร์พารามิเตอร์ตามที่แสดงในตารางที่ 3.3 ทั้ง 2 แบบ

ตารางที่ 3.6 ค่า Hyperparameter

Algorithms	Hyperparameter
Random Forest	Number of n_estimators = 10 Max depth = 3
XGBoost	Number of n_estimators = 10 Learning rate = 0.001 Max depth = 3
LightGBM	Number of n_estimators = 10 Learning rate = 0.001
AdaBoost	Number of n_estimators = 10 Learning rate = 0.001
CatBoost	Number of n_estimators = 10 Learning rate = 0.001 Max depth = 3

3.3 การประเมินประสิทธิภาพของโมเดล

เมื่อได้โมเดลมาแล้วทำการทดสอบเพื่อประเมินประสิทธิภาพ ในการตรวจจับค่าทอร์คที่ผิดปกติโดยจะใช้การประเมินดังนี้

1. ค่าความแม่นยำ (Accuracy) เนื่องจากการงานวิจัยนี้ต้องการตรวจจับความผิดปกติของ Drive Bush ดังนั้นจึงต้องทำการดูว่าโมเดลคาดการณ์ความผิดปกติของ Drive Bush ความแม่นยำเพียงใด

$$\text{Accuracy} = \frac{TP + TN}{(TP + TN + FP + FN} \quad (3.6)$$

True Positive (TP): โมเดลทำนายว่า Drive-Bush สึกหรอ และสึกหรอจริง

True Negative (TN): โมเดลทำนายว่า Drive-Bush ปกติ และปกติจริง

False Positive (FP): โมเดลทำนายว่า Drive-Bush สึกหรอ แต่จริงๆ ปกติ

False Negative (FN): โมเดลทำนายว่า Drive-Bush ปกติ แต่จริงๆ สึกหรอ

2. ค่าความเที่ยงตรง (Precision) เพื่อต้องการตรวจเช็คค่าความถูกต้องที่โมเดลคาดการณ์ว่าค่าที่คาดการณ์นั้นผิดปกติจริง เพื่อลดการคาดการณ์ผิดพลาดทำให้เข้าใจว่าอุปกรณ์มีปัญหา

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3.7}$$

3. ค่าความไว (Recall) เป็นค่าที่ใช้ในการดูการคาดการณ์ของโมเดลมีความถูกต้องได้ดีขนาดไหน ยิ่งสูงแสดงว่าคาดการณ์ค่าทอร์คได้ถูกต้องได้ครบถ้วน

$$\text{Recall} = \frac{TP}{TP + FP} \tag{3.8}$$

4. ค่า F1 Score เป็นค่าที่ใช้วัดประสิทธิภาพของโมเดล เพื่อดูว่าโมเดลใดมีประสิทธิภาพเพียงใด

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.9}$$

5. ค่า Confusion Metrix เป็นตารางการเปรียบเทียบการทำนายการจำแนกสถานะของ Drive Bush ว่าเป็นปกติหรือมีการสึกหรอ โดยแสดงจำนวนที่ถูกต้องและไม่ถูกต้องในแต่ละคลาสว่าตรงตามป้ายกำกับมากน้อยเท่าไร เพื่อใช้ในการเปรียบเทียบประสิทธิภาพของโมเดลดังรูปที่ 3.5

		Predicted	
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN

รูปที่ 3.6 Confusion Matrix

3.6 การเปรียบเทียบประสิทธิภาพ

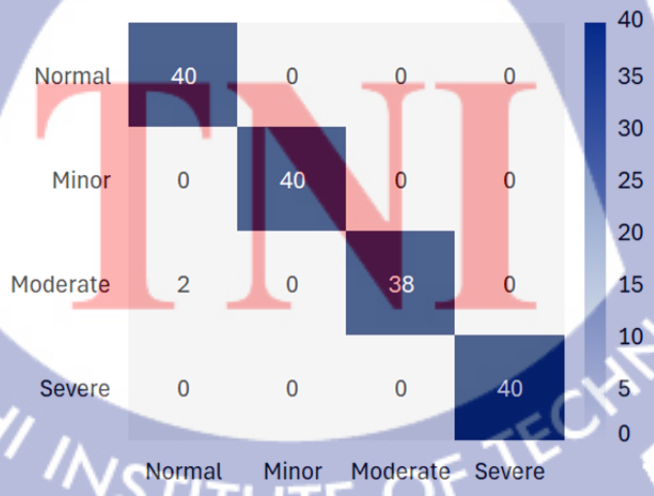
เพื่อดูว่าโมเดลใดมีประสิทธิภาพของโมเดลการเรียนรู้ของเครื่องในการทำนายระดับความสึกหรอของ Drive Bush ของมอเตอร์ขับวาล์ว โดยพิจารณาทั้งกรณีที่ฝึกโมเดลโดยใช้เทคนิค Train Validation Test Split และการใช้เทคนิค K-Fold Cross-Validation เพื่อศึกษาผลของการใช้เทคนิคดังกล่าวต่อสมรรถนะของโมเดล และใช้เป็นเกณฑ์ในการเลือกโมเดลที่เหมาะสมสำหรับการนำไปใช้งานจริง

บทที่ 4
ผลการวิจัย

จากการฝึกฝนและทดสอบทั้ง 5 เทคนิคด้วยชุดข้อมูลค่าทอร์คการเปิด-ปิด ในแต่ละตำแหน่งของมอเตอร์ขับวาล์วตามตารางที่ 4.1 แบบการแบ่งชุดข้อมูลออกเป็นชุดฝึกสอนและชุดทดสอบ (Train Validation Test Split) ผลการประเมินโมเดล มีโมเดล XGBoost และ LightGBM มีประสิทธิภาพสูงสุด มีค่า Accuracy 0.9875, ค่า Precision 0.9881, ค่า Recall อยู่ที่ 0.9875 และค่า F1-score อยู่ที่ 0.9878 โดยโมเดลอื่นๆ ก็มีประสิทธิภาพสูงยกเว้นโมเดล AdaBoost ที่ให้ค่าต่างๆ น้อยกว่า โดยค่า Accuracy 0.7375, ค่า Precision 0.5989, ค่า Recall อยู่ที่ 0.7375 และค่า F1-score อยู่ที่ 0.6610

ตารางที่ 4.1 ผลการประเมินโมเดลแบบ Train Validation Test Split

Model	Accuracy	Precision	Recall	F1-score
RF	0.9688	0.9722	0.9688	0.9705
XGBoost	0.9875	0.9881	0.9875	0.9878
LightGBM	0.9875	0.9881	0.9875	0.9878
AdaBoost	0.7375	0.5989	0.7375	0.6610
CatBoost	0.9812	0.9826	0.9812	0.9819



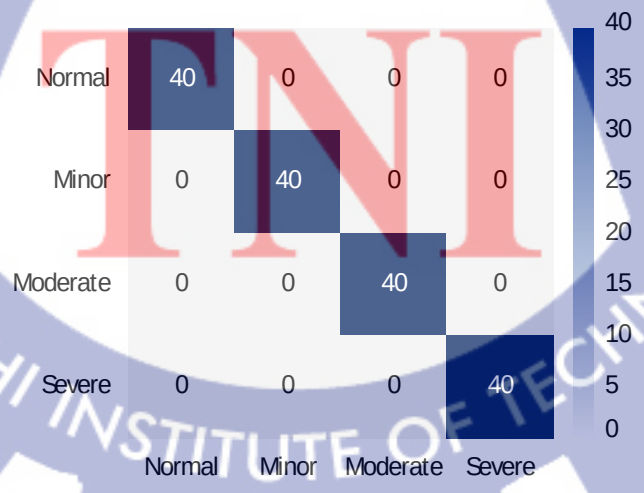
รูปที่ 4.1 Confusion Matrix XGBoost

จากรูปที่ 4.1 Confusion Matrix XGBoost แสดงให้เห็นถึงความแม่นยำในการจำแนกประเภทที่สูงของแบบจำลองในสภาวะการสึกหรอทั้งสี่แบบ คลาสที่แสดงถึงแบบปกติ การสึกหรอเล็กน้อย และการสึกหรอรุนแรง ได้รับการคาดการณ์ด้วยความแม่นยำ 100% (มี 40 ตัวอย่างจากทั้งหมด 40 ตัวอย่างที่จำแนกประเภทได้อย่างถูกต้องสำหรับแต่ละคลาส) คลาสการสึกหรอปานกลางมีความแม่นยำ 95% โดยมี 38 ตัวอย่างจากทั้งหมด 40 ตัวอย่างที่จำแนกประเภทได้อย่างถูกต้อง และอีก 2 ตัวอย่างที่จำแนกประเภทผิดพลาดว่าเป็นปกติ

ผลการฝึกฝนและทดสอบทั้ง 5 เทคนิคด้วยชุดข้อมูลค่าทอร์คการเปิด-ปิด ในแต่ละตำแหน่งของมอเตอร์ขับวาล์วแบบใช้ K-Fold Cross Validation ซึ่งโมเดล XGBoost และ LightGBM มีประสิทธิภาพดีที่สุดและดีกว่าโมเดลที่แบบแบ่งชุดข้อมูลออกเป็นชุดฝึกสอนและชุดทดสอบ โดยมีผลตามตารางที่ 4.2 ด้านล่าง

ตารางที่ 4.2 ผลการประเมินโมเดลโดยมีการทำ K-fold Cross Validation

Model	Accuracy	Precision	Recall	F1-score
RF	0.9188	0.9277	0.9188	0.9232
XGBoost	1.000	1.000	1.000	1.000
LightGBM	1.000	1.000	1.000	1.000
AdaBoost	0.7375	0.5989	0.7375	0.6610
CatBoost	0.9812	0.9826	0.9812	0.9578



รูปที่ 4.2 Confusion Matrix XGBoost โดยมีการทำ K-fold Cross Validation

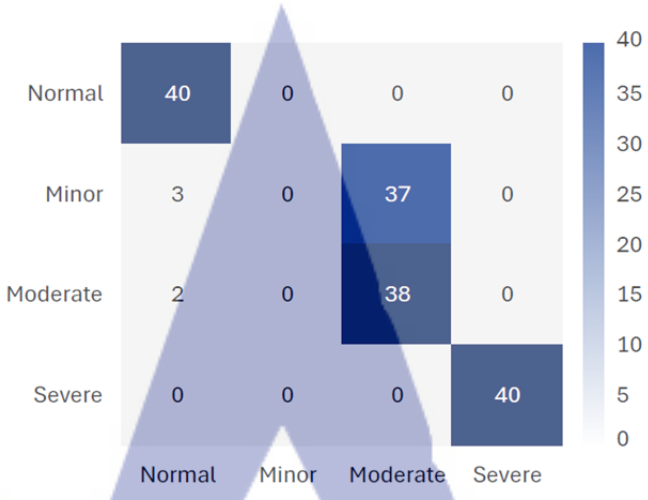
จากรูปที่ 4.2 แสดง Confusion Matrix ของ XGBoost ที่ทำ K-fold Cross-Validation สามารถจำแนกระดับการสึกหรอของ Drive Bush ได้ถูกต้องทุกคลาส ซึ่งมีประสิทธิภาพสูงกว่าแบบการแบ่งชุดข้อมูลออกเป็นชุดฝึกสอนและชุดทดสอบ (Train Validation Test Split)

บทที่ 5
สรุปและอภิปราย

5.1 การเปรียบเทียบประสิทธิภาพ

จากผลการทดลองเปรียบเทียบประสิทธิภาพของแบบจำลองทั้งห้าแบบ ได้แก่ Random Forest, XGBoost, LightGBM, CatBoost และ AdaBoost โดยประเมินทั้งจากการแบ่งชุดข้อมูลออกเป็นชุดฝึกสอนและชุดทดสอบ (Train Validation Test Split) และแบบทำ K-fold Cross-Validation พบว่าแบบจำลอง XGBoost และ LightGBM ให้ผลการทำนายสูงที่สุดอย่างต่อเนื่องสำหรับการประเมินแบบ Train-Test Split โมเดล XGBoost และ LightGBM ให้ค่าความถูกต้อง (Accuracy) 0.9875 ความแม่นยำ (Precision) 0.9881 การตรวจพบ (Recall) 0.9875 และค่า F1-score 0.9878 บนชุดทดสอบ ขณะที่เมื่อประเมินด้วยวิธี K-fold Cross Validation และนำโมเดลที่เรียนรู้จากชุด Train + Validation ทั้งหมดไปทดสอบกับชุดทดสอบเดิม พบว่าค่าความถูกต้อง ความแม่นยำ การตรวจพบ และค่า F1-score ของทั้งสองโมเดลมีค่าเท่ากับ 1.0000 แสดงให้เห็นว่าโมเดล XGBoost และ LightGBM สามารถจำแนกระดับการสึกหรอของ Drive Bush ได้ถูกต้องทุกกรณีในชุดข้อมูลทดสอบ และมีประสิทธิภาพสูงกว่าวิธีประเมินแบบ Train Validation Test Split เพียงอย่างเดียว

สำหรับ CatBoost พบว่ามีประสิทธิภาพรองลงมาเล็กน้อย ทั้งแบบการแบ่งชุดข้อมูลออกเป็นชุดฝึกสอนและชุดทดสอบ (Train Validation Test Split) และแบบทำ K-fold Cross-Validation มีค่าการทดสอบเท่ากันโดยค่า Accuracy และ Recall 0.9812, Precision 0.9826 และ F1-score 0.9819 AdaBoost ให้ผลการทำนายต่ำที่สุดอย่างชัดเจน ทั้งในกรณีการแบ่งชุดข้อมูลออกเป็น Train Validation Test Split และการทำ K-fold Cross Validation โดยมีค่าความถูกต้อง (Accuracy) และค่า F1-score บนชุดทดสอบเพียงประมาณ 0.7375 และ 0.6610 ตามลำดับ หากดูจาก Confusion Matrix ของ AdaBoost ที่เท่ากันของ Train Test Validation Split และ K-fold Cross Validation ตามรูปที่ 5.1



รูปที่ 5.1 Confusion Matrix AdaBoost ทั้ง Train Test Split และ K-Fold

จากรูปที่ 5.1 มีค่าการทำทายค่าระดับการสึกหรอได้ดังนี้

ปกติ : ตัวอย่าง 40 ตัวอย่าง ทายถูก 40 ตัวอย่าง

สึกหรอเล็กน้อย : ตัวอย่าง 40 ตัวอย่าง ทายไม่ถูกเลย

: ทายเป็นสึกหรอปานกลาง 37 ตัวอย่าง

: ทายเป็นปกติ 2 ตัวอย่าง

สึกหรอปานกลาง : ตัวอย่าง 40 ตัวอย่าง ทายถูก 38 ตัวอย่าง

: ทายว่าปกติ 2 ตัวอย่าง

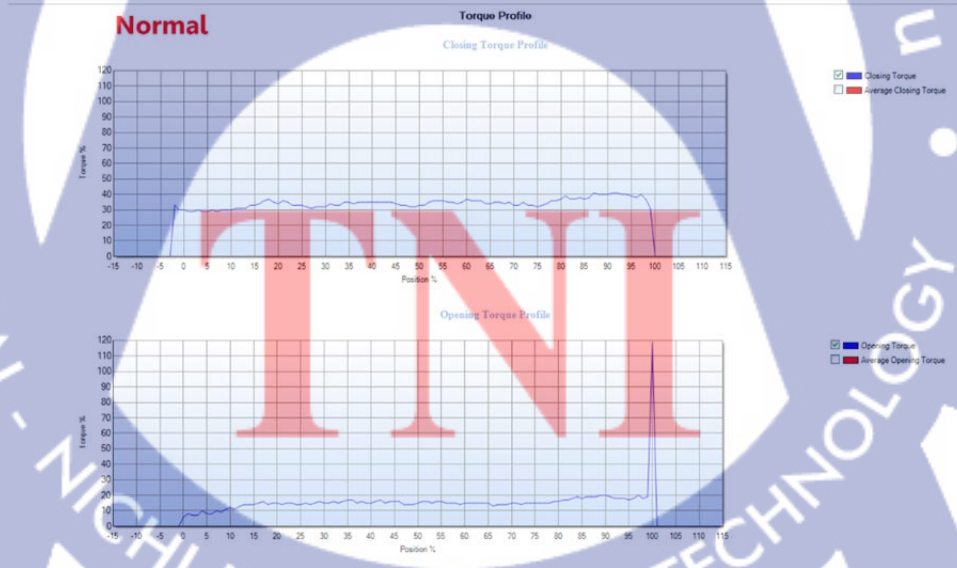
สึกหรอรุนแรง : ตัวอย่าง 40 ตัวอย่าง ทายถูก 40 ตัวอย่าง

จากผลการทำนายใน Confusion Matrix พบว่าโมเดล AdaBoost สามารถจำแนกแบบปกติและการสึกหรอรุนแรงได้ถูกต้องทั้งหมด อย่างไรก็ตาม โมเดลไม่สามารถจำแนกระดับสึกหรอเล็กน้อยได้เลย โดยตัวอย่างระดับสึกหรอเล็กน้อย ทั้งหมดถูกทำนายผิดเป็นระดับสึกหรอปานกลาง จำนวน 37 ตัวอย่าง และเป็นระดับปกติ จำนวน 3 ตัวอย่าง สะท้อนว่า AdaBoost ไม่สามารถแยกความแตกต่างระหว่างระดับการสึกหรอเล็กน้อย กับ ระดับสึกหรอปานกลาง ได้อย่างมีประสิทธิภาพ เนื่องจากรูปแบบค่าแรงบิดตามตำแหน่ง (torque profile) ของการสึกหรอระดับปานกลางมีความทับซ้อนกันสูง และความแตกต่างของสัญญาณมักเกิดขึ้นเพียงบางช่วงตำแหน่งเท่านั้น

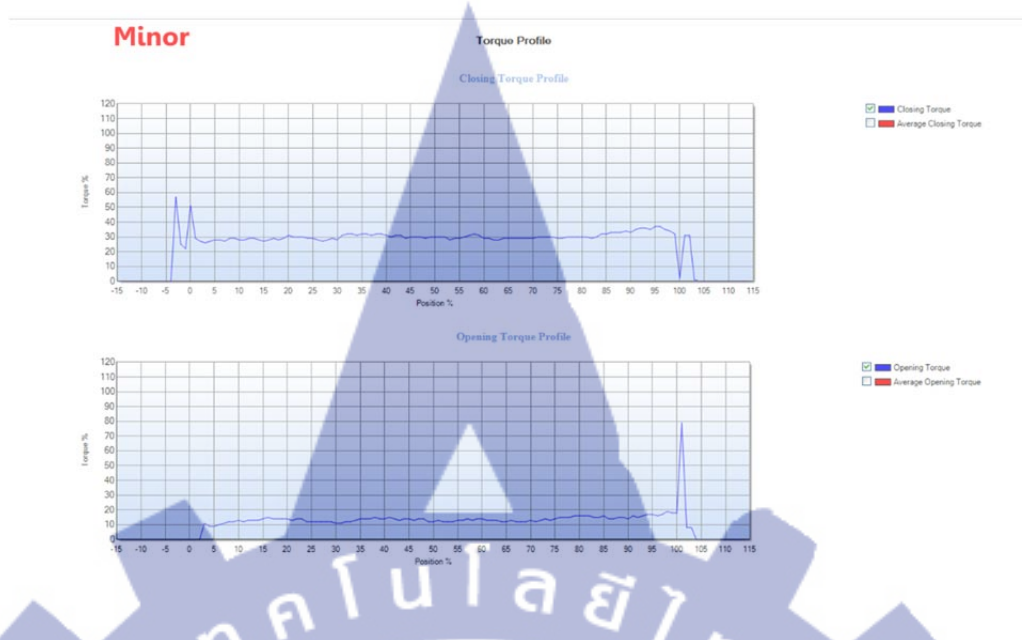
ข้อจำกัดดังกล่าวสัมพันธ์กับกลไกของ AdaBoost ที่โดยทั่วไปใช้ตัวเรียนรู้ฐานเป็นต้นไม้ตื้น (มักลึกเพียง 1 ชั้น) ทำให้การเรียนรู้มีลักษณะพิจารณาคุณลักษณะทีละมิติ และต้องอาศัยจำนวนรอบการ Boosting มากพอเพื่อสะสมเป็นเส้นแบ่งที่ซับซ้อนขึ้น แต่ในการทดลองนี้กำหนดค่า n_estimators =

10 และ Learning_rate = 0.001 ซึ่งเป็นการตั้งค่าโดยใช้เงื่อนไขเดียวกัน เพื่อความเท่าเทียมกันในการเปรียบเทียบกับโมเดลอื่น ส่งผลให้ AdaBoost มีความสามารถในการเรียนรู้ (Model Capacity) ต่ำและมีแนวโน้มเกิด Underfitting โดยเฉพาะเมื่อต้องจำแนกข้อมูลที่มีฟีเจอร์จำนวน 45 คอลัมน์ และมีความแตกต่างแบบละเอียดระหว่างคลาส ส่งผลให้ผลลัพธ์การทำนายออกมาตามที่แสดงในตารางที่ 4.1 และ 4.2

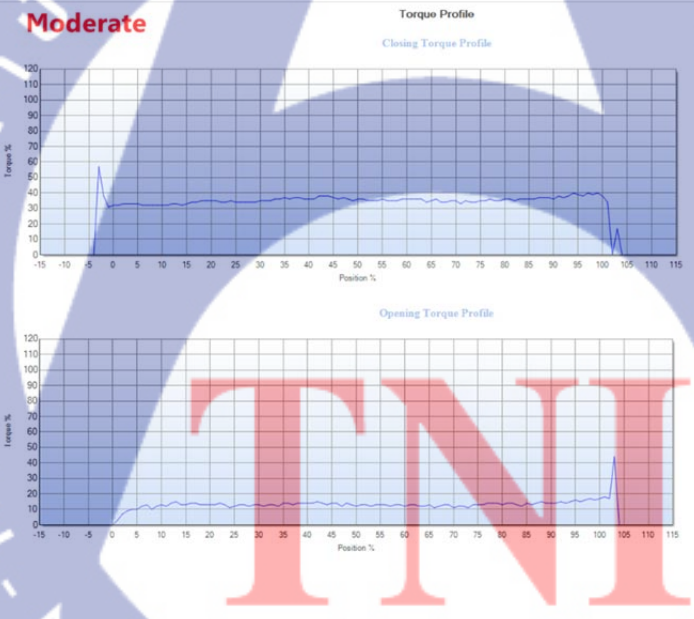
เมื่อเปรียบเทียบกับโมเดล XGBoost, LightGBM และ CatBoost ซึ่งอยู่ในกลุ่ม Gradient Boosting Decision Trees (GBDT) จะพบว่าโมเดลกลุ่มนี้สามารถสร้างต้นไม้ที่เรียนรู้ “เงื่อนไขร่วม” จากหลายคุณลักษณะได้ดีกว่า จึงเหมาะกับข้อมูล Torque Profile ที่ต้องพิจารณาหลายช่วงตำแหน่งพร้อมกัน ตัวอย่างเช่น Torque Profile ของการเปิดวาล์วในช่วงตำแหน่ง 0-10% ของคลาสปกติ และระดับความสึกหรอเล็กน้อย มีค่าใกล้เคียงกันมาก แต่เริ่มเห็นความแตกต่างชัดเจนในช่วง 95-105% ตามรูปที่ 5.2 Torque Profile แบบปกติ และภาพที่ 5.3 Torque Profile แบบสึกหรอเล็กน้อยทำให้โมเดลกลุ่ม Gradient Boosting Trees สามารถใช้ข้อมูลจากหลายช่วงตำแหน่งร่วมกันเพื่อจำแนกคลาสได้อย่างมีประสิทธิภาพ ในขณะที่ AdaBoost ภายใต้โครงสร้างต้นไม้ที่ตื้นและการตั้งค่าพารามิเตอร์ดังกล่าวไม่สามารถดึงความแตกต่างที่ละเอียดนี้ออกมาได้ จึงเกิดความคลาดเคลื่อนในการจำแนก Minor ไปเป็น Moderate เป็นส่วนใหญ่



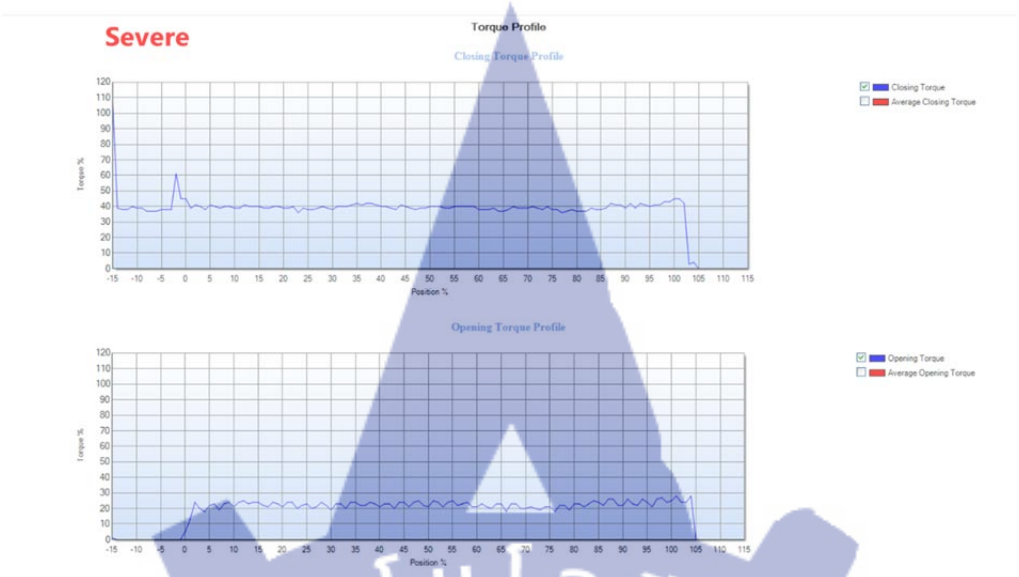
รูปที่ 5.2 Torque Profile แบบปกติ



รูปที่ 5.3 Torque Profile แบบสึกหรอเล็กน้อย



รูปที่ 5.4 Torque Profile แบบสึกหรอปานกลาง



รูปที่ 5.5 Torque Profile แบบสึกหรอรุนแรง

เมื่อเปรียบเทียบผลจาก K-Fold Cross-Validation กับผลจากการแบ่งชุดข้อมูลแบบ Train-Test เพียงครั้งเดียว พบว่าแนวโน้มของผลลัพธ์มีความสอดคล้องกัน คือ XGBoost และ LightGBM ยังคงเป็นโมเดลที่มีประสิทธิภาพสูงที่สุด รองลงมาคือ CatBoost, Random Forest และ AdaBoost ตามลำดับ ความใกล้เคียงระหว่างค่าเฉลี่ยจาก K-Fold Cross Validation กับผลบนชุดทดสอบจริงบ่งชี้ว่า ผลการประเมินไม่ได้ขึ้นกับการสุ่มแบ่งข้อมูลครั้งเดียว แต่มีความน่าเชื่อถือและมีเสถียรภาพสูง

เนื่องจากผลการทดลองที่ได้ในแต่ระดับความสึกหรอจะค่าที่สูงผิดปกติเป็นบางช่วงแตกต่างกันทำให้กลายเป็นคุณลักษณะที่เด่นชัดในแต่ละระดับความสึกหรอตามรูปที่ 5.2 ถึงรูปที่ 5.5 ค่า Torque Profile ที่ระดับความสึกหรอต่างๆ

เมื่อเปรียบเทียบงานวิจัยนี้กับงานวิจัยก่อนหน้าเรื่องการตรวจจับความผิดปกติของแรงบิดของตัวขับวาล์วไฟฟ้าในโรงไฟฟ้านิวเคลียร์ด้วยวิธี DAE-WDSVDD [10] จะเห็นว่าทั้งสองงานทำการวิเคราะห์ความผิดปกติของระบบ Electric Valve Actuator เพื่อเพิ่มความปลอดภัยและความน่าเชื่อถือของการทำงาน โดยมีวัตถุประสงค์เหมือนกันคือการตรวจจับสัญญาณที่สะท้อนการเสื่อมสภาพหรือความผิดปกติล่วงหน้าเพื่อลดความเสียหายและค่าใช้จ่ายการหยุดเดินเครื่อง อย่างไรก็ตามความแตกต่างอยู่ที่ชนิดข้อมูลที่ใช้และการเรียนรู้ของโมเดล โดยข้อจำกัดแตกต่างกันออกไป งาน DAE-WDSVDD ถูกออกแบบมาสำหรับอุปกรณ์ที่มีชุดข้อมูลมีแต่ข้อมูลปกติเป็นหลัก และแทบไม่มีข้อมูลผิดปกติ จึงใช้การเรียนรู้แบบ Unsupervised แบบ One-Class โดยฝึก Deep Autoencoder (DAE) ด้วยข้อมูลปกติให้บีบอัดและสร้างกลับ แล้วใช้ Reconstruction Error เป็นตัวแทนความแตกต่างจากรูปแบบปกติ ก่อนนำความผิดพลาดที่ได้ไปสร้างขอบเขตความเป็นปกติด้วย Weighted Deep SVDD (WDSVDD) ซึ่งมองว่าข้อมูลปกติควรกระจุกอยู่

ใกล้ศูนย์กลางของ Hypersphere และวัดระดับความผิดปกติจากระยะห่างจากศูนย์กลาง ทำให้เหมาะอย่างยิ่งกับงานที่ไม่สามารถเก็บ Fault ได้ครบถ้วน และรายงานความแม่นยำชุดทดสอบอยู่ที่ 98.4%

ในส่วนของข้อมูล งานก่อนหน้านี้ยังใช้การประยุกต์เซ็นเซอร์เสริมเพื่อให้ได้ข้อมูลที่หลากหลายในการดูแรงที่กระทำต่อก้านวาล์ว ได้แก่ Strain Gauge Bridge Circuit และ Laser Displacement Sensor เพื่อช่วยคำนวณแรงที่เกิดกับก้านวาล์วและช่วยจับอาการผิดปกติหลายแบบ วิธีนี้ทำให้เห็นอาการเปลี่ยนไปของชิ้นส่วนได้ละเอียดขึ้น แต่ต้องทำการการติดตั้ง การสอบเทียบ และความเสี่ยงจากความคลาดเคลื่อนของเซ็นเซอร์หรือสภาพแวดล้อมหน้างาน ตรงกันข้าม งานวิจัยของเราเลือกใช้ข้อมูลแรงบิดเทียบกับเปอร์เซ็นต์ตำแหน่งการเปิด-ปิด (Torque vs Position) ที่ดึงได้จากหัวขับไฟฟ้าโดยตรง ไม่ต้องติดตั้งอุปกรณ์เสริม ทำให้มีความได้เปรียบเชิงการนำไปใช้จริง (Deployability) และการขยายผลในโรงงานหลายจุด แต่เราใช้กรอบ Supervised Learning ที่มีการแบ่งคลาสชัดเจน จึงตอบโจทย์เชิงซ่อมบำรุงได้ละเอียดกว่า เพราะสามารถชี้ระดับความเสื่อมสภาพเพื่อวางแผนในการซ่อมบำรุงได้ดีกว่า เช่น แยกปกติ สึกน้อย สึกปานกลาง และ สึกรุนแรง ขณะเดียวกันโมเดลดารเรียนรู้แบบ Supervised ก็มีความท้าทายเฉพาะตัว ได้แก่

1. คุณภาพฉลาก (Label Quality) และความสอดคล้องของเกณฑ์แบ่งคลาส
2. ความคล้ายกันสูงของคลาสที่ติดกัน เช่น ระดับปกติกับระดับสึกหรอเล็กน้อยที่ Torque Profile ใกล้เคียงจนทำให้โมเดลสับสนได้ง่าย
3. ความแปรปรวนจากสภาวะการทำงานจริง เช่น แรงดัน อัตราการไหล อุณหภูมิ และสภาพการหล่อลื่น ที่ทำให้รูปทรง Torque Profile เปลี่ยนได้แม้ไม่ได้เกิดการสึกหรอจริง ซึ่งบังคับให้เราต้องออกแบบการทำ Normalization และ Feature Engineering และการประเมินผลรายคลาสให้รัดกุมกว่างาน One-Class

โดยภาพรวม งาน DAE-WDSVDD เด่นด้านตรวจจับได้ แม้ไม่มี Fault Data และเหมาะเป็นระบบเฝ้าระวังระยะแรก ในขณะที่งานของวิจัยฉบับนี้เด่นด้านการจำแนกระดับเพื่อการตัดสินใจซ่อมบำรุง

5.2 ข้อจำกัดในงานวิจัย

ในงานวิจัยนี้ได้ทำการเก็บข้อมูลจากการใช้มอเตอร์ควบคุมวาล์วขนาด 6 นิ้ว Class 150 ในสภาวะบรรยากาศนั้น พบว่ามีข้อจำกัดของงานวิจัยดังนี้

1. ขอบเขตอุปกรณ์จำกัด เนื่องจากใช้ข้อมูลจากวาล์วขนาด 6 นิ้ว Class 150 และหัวขับวาล์วไฟฟ้าเพียงรุ่นเดียว
2. สภาวะทดลองจำกัด เนื่องทดลองภายใต้สภาวะไม่มีความดันและไม่มีการไหลของของไหลจริงในท่อ

3. ปริมาณข้อมูลมีน้อย โดยมีข้อมูลรวมน้อยเพียง 1,600 ตัวอย่าง 45 คอลัมน์ต่อตัวอย่าง จึงอาจยังไม่เพียงพอสำหรับใช้ในการ Training ทำให้ผลการทำนาย Over Fitting

4. ขอบเขตความผิดปกติที่ศึกษายังไม่ครอบคลุมอาการผิดปกติของอุปกรณ์ ได้แก่ อาการฝืดจากขาดสารหล่อลื่น การขัดกัน และสิ่งสกปรกในร่องเกลียว ซึ่งอาจช่วยจำแนกสภาวะซับซ้อนได้ดีขึ้น

5. การเปรียบเทียบแบบจำลองจำกัดอยู่เพียงกลุ่ม Ensemble จำนวน 5 แบบ คือ Random Forestม, XGBoost, LightGBM, CatBoost และ AdaBoost ยังไม่ครอบคลุมโมเดลประเภทอื่น

6. ข้อมูลที่นำมาใช้มีเพียงข้อมูลจากการทดลอง ไม่ใช่ข้อมูลที่มาจากการใช้งานจริง ทำให้ประสิทธิภาพไม่ครอบคลุมกับการใช้งานจริง

5.3 ข้อเสนอแนะในงานวิจัย

จากข้อจำกัดของงานวิจัยที่กล่าวไปก่อนหน้าทำให้ประสิทธิภาพของโมเดลในการนำไปใช้งานจริงไม่สามารถครอบคลุมกับมอเตอร์ควบคุมวาล์วได้ทั้งหมด จึงมีข้อเสนอแนะในการทำงานวิจัยต่อไปในอนาคตดังต่อไปนี้

1. ขยายขอบเขตอุปกรณ์โดยเก็บข้อมูลเพิ่มเติมจากวาล์วหลายขนาดและ Class อีกทั้งหัวขับวาล์วไฟฟ้าหลายๆ รุ่น เพื่อให้ประสิทธิภาพของโมเดลครอบคลุมการใช้งาน

2. ทำการเก็บข้อมูลจากมอเตอร์ขับวาล์วจริงจากหางานที่มีแรงดันและของไหลในท่อจริง เพื่อให้รูปแบบค่าทอร์คและตำแหน่งในการเปิด-ปิด สะท้อนพฤติกรรมการทำงานในสภาพจริง

3. เพิ่มปริมาณข้อมูลโดยเพิ่มจำนวนรอบในการเปิด-ปิด เพื่อให้โมเดลเรียนมีข้อมูลในการรู้เพิ่มขึ้นและความเสี่ยงในการเกิด Overfitting

4. ขยายขอบเขตความผิดปกติของ Drive Bush เพื่อให้จำแนกความผิดปกติอื่นๆ ที่ซับซ้อนและใกล้เคียงกันได้ดีขึ้น เช่น ขาดสารหล่อลื่น, มีสิ่งสกปรกอุดตันบริเวณร่องเกลียวและความเสียหายอื่นๆ

5. ขยายขอบเขตการใช้โมเดลที่หลายหลาก เนื่องจากการเก็บข้อมูลจากหน้างานจริงจะไม่สามารถระบุความปกติหรือผิดปกติได้ จึงไม่สามารถระบุ Label ได้ ดังนั้นต้องพัฒนาแนวทางโดยใช้การเรียนรู้แบบ ไฮบริดที่ใช้ Unsupervised ช่วยขยายสัญญาณความผิดปกติระยะเริ่มต้น แล้วให้ Supervised ทำหน้าที่ตีความเป็นระดับความสึกหรอ พร้อมกับการเก็บค่าอย่างอื่นเช่น ค่ารอบในการหมุนของ Drive Bush ค่าการทำงานของ Contactor อุปกรณ์ เพื่อเพิ่มทั้งความไวในการตรวจจับและความน่าเชื่อถือเมื่อใช้งานจริงในภาคอุตสาหกรรม


สถาบันเทคโนโลยีไทย-ญี่ปุ่น
THAI - NICHI INSTITUTE OF TECHNOLOGY
TNI

บรรณานุกรม

บรรณานุกรม

[1] A. Ucar et al., “Artificial intelligence for predictive maintenance applications : Key components, trustworthiness, and future trends,” *Applied Sciences,* vol. 14, no. 2, p. 898, January 2024.

[2] American Petroleum Institute (API), *Steel Gate Valves-Flanged and Butt-Welding Ends, Bolted Bonnets,* Washington, DC : Pearson Prentice Hall, 2015.

[3] A. Gilchrist, *Industry 4.0 : The Industrial Internet of Things,* Berkeley, CA : Apress, 2016.

[4] S. Munirathinam, *Industry 4.0 : Industrial Internet of Things (IIOT),* Amsterdam : Elsevier, 2020.

[5] A. ErdiL, “Industry 4.0 perception regarding to new developments and new trends of industries,” *European Journal of Science and Technology,* vol. 34, no. 28. pp. 228-240, November 2021.

[6] G. N. Schroeder et al., “A methodology for digital twin modeling and deployment for industry 4.0,” *Proceeding of IEEE,* vol. 109, no. 4, pp. 556-567, April 2021.

[7] A. Fuller et al., “Digital twin enabling technologies, challenges and open research,” *IEEE Access,* vol. 8, pp. 108,952-108,971, May 2020.

[8] P. L. Skousen, *Valve Handbook,* New York : McGraw-Hill, 2012.

[9] M. Atif et al., “Valve health identification using sensors and machine learning methods,” *International Conference on Information and Communication Technology Convergence,* ICIT 2020, Ghent, Belgium, September 14-18, 2020, pp. 45-60.

[10] J. Peng et al., “Torque anomaly detection of nuclear power electric valve actuator based on DAE-WDSVVD,” *Journal of Physics : Conference Series,* vol. 21, no. 1, pp. 81-87, February 2022.

[11] B. Bhushan, *Introduction to Tribology in Tribology Series,* Chicheste : Wiley, 2013.

[12] V. Popov, “Generalized archard law of wear based on rabinowicz criterion of wear particle formation,” *Journal of Education Naresuan University*, vol. 17, no. 1, pp. 39-45, Mar 2019.

[13] R. Gouriveau et al., *From Prognostics and Health Systems Management to Predictive Maintenance 1 : Monitoring and Prognostics,* Newark : Wiley, 2016.

[14] G. F. Luger, *Artificial Intelligence : Structures and Strategies for Complex Problem Solving,* Boston : Pearson Prentice Hall, 2009.

[15] P. Karmakar et al., "Artificial intelligence," *International journal of advanced research in science communication and technology,* vol. 4, no. 2. pp. 79-87, September 2024.

[16] D. L. Poole and A. K. Mackworth, *Artificial Intelligence: Foundations of Computational Agents,* Boston : Cengage Learning, 2023.

[17] T. P. Trappenberg, *Fundamentals of Machine Learning,* Oxford : Butterworth-Heinemann, 2019.

[18] L. Rokach and O. Maimon, *Sound Design : The Expressive Power of Music Voice and Sound Effects in Cinema,* New York : Springer-Verlag, 2005.

[19] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal,* vol. 27, no. 3, pp. 379-423, July 1948.

[20] J. Han et al., *Data Mining : Concepts and Techniques in Morgan Kaufmann Series in Data Management Systems,* Boston : Elsevier/Morgan Kaufmann, 2012.

[21] S. Shalev and S. Ben, *Understanding Machine Learning: From Theory to Algorithms,* Cambridge, United Kingdom : Cambridge University Press, 2014.

[22] J. Korstanje, *Advanced Forecasting with Python,* Berkeley, CA : Apress, 2021.

[23] T. Chen and C. Guestrin, "XGBoost : A scalable tree boosting system," Proceedings of International Conference, *NEDSI 2016 Proceedings,* San Francisco, California, August 6-9, 2016, pp. 785-794.

[24] K. Guolin et al., "LightGBM : A highly efficient gradient boosting decision tree," *Advances in Neural Information Processing Systems, NIPS 2017 Long Beach,* CA, USA, December 21-24, 2017, pp. 3,146-3,154.

[25] N. Dutta et al., "Life cycle cost analysis of pumping system through machine learning and hidden markov model," *Processes,* vol. 11, no. 7, pp.451-474, July 2023.

[26] G. Wrat et al., "Neural network-enhanced internal leakage analysis for efficient fault detection in heavy machinery hydraulic actuator cylinders," *Journal of Mechanical Engineering Science,* vol. 239, no. 3, pp. 1,021-1,031, February 2025.

[27] J. C. Quiroz et al., "Fault detection of broken rotor bar in LS-PMSM using random forests," *Measurement,* vol. 116, pp. 273-280, February 2018.

[28] B. Uhrich et al., “Using differential equation inspired machine learning for valve faults prediction,” *International Conference on Industrial Informatics, INDIN 2023 Proceedings,* New Jersey, USA, July 3-5, 2023, pp. 1-8.

[29] R. González et al., “Assessment and deployment of a LSTM-based virtual sensor in an industrial process control loop,” *Neural Computing and Applications,* vol. 37, no. 17, pp. 10,507-10,519, June 2025.

[30] B. Suseela et al., “Intelligent fault diagnosis in industrial machinery : leveraging AI with LSTM Autoencoder for enhanced fault detection,” *Journal of Machine and Computing,* vol. 4, no. 4, pp. 931-942, October 2024.

[31] R. T. Yoder et al., “Smart monitoring and failure forecasting for in-line blowout preventer valves,” *Proceedings of the Offshore Technology Conference, NEDSI 2020 Proceedings,* New Jersey, USA, May 12-15, 2020, pp. 373-383.

[32] British Standards Institution (BSI). *Industrial Valves-Actuators-Part 2 : Electric Actuators for Industrial Valves-Basic Requirements,* Standard BS EN 15714-2:2009, London, November 2009.

[33] The American Society of Mechanical Engineers, *ACME Screw Threads,* Standard ASME B1.5, New York, USA., 1997.

ภาคผนวก

ภาคผนวก ก.

รายละเอียดโมเดล

Importing libraries for data handling, dataset splitting/scaling, multiple machine learning models and performance evaluation.

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split, KFold
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import (
    AdaBoostClassifier,
    RandomForestClassifier
)
from xgboost import XGBClassifier
from lightgbm import LGBMClassifier
from catboost import CatBoostClassifier
from imblearn.over_sampling import SMOTE
from sklearn.metrics import (
    accuracy_score,
    precision_score,
    recall_score,
    f1_score,
    confusion_matrix,
    classification_report
)
```

Reading data from hard drive

```
df = pd.read_excel("D:\Master Degree\Thesis\Test\งานพี่ต้อม Rev.2\Torque VS Position_Close+Open_Data_Training_24.8.2025.xlsx", sheet_name="Combination")
print("ข้อมูลตัวอย่าง:")
print(df.head())
```

Converting all column names to string type.

```
df.columns = df.columns.astype(str)
```

Preparing dataset and mapping classes.

```
label_mapping = {"Normal": 0, "Minor": 1, "Moderate": 2, "Sereve": 3}
df["Class"] = df["Class"].map(label_mapping)
```

Defining X as all numeric feature columns except column “Class”, and y as the target label from class column.

```
X = df.select_dtypes(include=[np.number]).drop(columns=["Class"])
y = df["Class"]
```

Converting column X to string type.

```
X.columns = X.columns.astype(str)
```

Separating feature X and Target y .

```
X = df.drop(columns=["Class"])
y = df["Class"]
```

Splitting dataset for train: validation: test, 80:10:10

```
X_temp, X_test, y_temp, y_test = train_test_split(
    X, y,
    test_size=0.1,
    random_state=42,
    stratify=y
)

X_train, X_val, y_train, y_val = train_test_split(
    X_temp, y_temp,
    test_size=1/9,
    random_state=42,
    stratify=y_temp
)
print(X_train.shape)
```

```
print(X_val.shape)
print(X_test.shape)
```

Standardizing the feature variables using StandardScaler fitted on the training set and applied to the training, validation, and test sets.

```
scaler = StandardScaler()
X_train = pd.DataFrame(scaler.fit_transform(X_train), columns=X.columns)
X_val = pd.DataFrame(scaler.transform(X_val), columns=X.columns)
X_test = pd.DataFrame(scaler.transform(X_test), columns=X.columns)
```

Creating model with adjustable Hyperparameters.

```
models = {
    "LightGBM": LGBMClassifier(n_estimators=10, learning_rate=0.001 ),
    "AdaBoost": AdaBoostClassifier(n_estimators=10, learning_rate=0.001),
    "XGBoost": n_estimators=10, learning_rate=0.001, max_depth=3),
    "CatBoost": CatBoostClassifier(n_estimators=10, learning_rate=0.001,
max_depth=3),
    "Random Forest": RandomForestClassifier(n_estimators=10, max_depth=3)
}
```

Training and model evaluation.

```
results = []

from sklearn.metrics import (
    accuracy_score, precision_score, recall_score, f1_score,
    confusion_matrix, classification_report
)
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
```

```
for name, model in models.items():
Training model.
    model.fit(X_train, y_train)

Predicted data testing
    y_pred = model.predict(X_test)

Calculation metrics.
    accuracy = accuracy_score(y_test, y_pred)
    precision = precision_score(y_test, y_pred, average='macro', zero_division=0)
    recall = recall_score(y_test, y_pred, average='macro', zero_division=0)
    f1 = f1_score(y_test, y_pred, average='macro', zero_division=0)

Results.
    results.append({
      "Model": name,
      "Accuracy": accuracy,
      "Precision": precision,
      "Recall": recall,
      "F1 Score": f1
    })

Show Confusion Matrix and Classification Report.
    cm = confusion_matrix(y_test, y_pred)
    print(f"\n=== {name} ===")
    print("Confusion Matrix:\n", cm)
    print("Classification Report:\n", classification_report(y_test, y_pred,
zero_division=0))
    plt.figure(figsize=(5, 4))
    sns.heatmap(cm, annot=True, fmt="d", cmap="Blues",
                xticklabels=np.unique(y_test),
```

```
                yticklabels=np.unique(y_test))
    plt.title(f"Confusion Matrix for {name}")
    plt.xlabel("Predicted Label")
    plt.ylabel("True Label")
    plt.tight_layout()
    plt.show()
```

Summary table results.

```
    results_df = pd.DataFrame(results)
    results_df = results_df.round(4)
    results_df = results_df.sort_values(by=["AUC", "F1 Score", "Accuracy"],
ascending=False)

    print("\n=== Summary of Model Performance ===")
    print(results_df.reset_index(drop=True))
```

Aggregating model metrics into a DataFrame and ranking models by F1-score and accuracy.

```
results_df = pd.DataFrame(results)
results_df["Accuracy"] = results_df["Accuracy"].round(4)
results_df["Precision"] = results_df["Precision"].round(4)
results_df["Recall"] = results_df["Recall"].round(4)
results_df["F1 Score"]  = results_df["F1 Score"].round(4)
results_df = results_df.sort_values(
    by=["F1 Score", "Accuracy"], ascending=False)

print("\n=== Summary of Model Performance ===")
print(results_df.reset_index(drop=True))
```

Importing libraries for data handling, dataset with K-fold cross validation, multiple machine learning models and performance evaluation.

```
import numpy as np
import pandas as pd

from sklearn.model_selection import train_test_split, StratifiedKFold
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import Pipeline
from sklearn.base import clone

from sklearn.ensemble import (
    AdaBoostClassifier,
    GradientBoostingClassifier,
    BaggingClassifier,
    RandomForestClassifier
)
from xgboost import XGBClassifier
from lightgbm import LGBMClassifier
from catboost import CatBoostClassifier

from sklearn.metrics import (
    accuracy_score,
    precision_score,
    recall_score,
    f1_score,
    confusion_matrix,
    classification_report
)

import seaborn as sns
import matplotlib.pyplot as plt
```

```
Reading data from hard drive.
file_path = "D:\Master Degree\Thesis\Test\งานพี่ต้อม Rev.2\Torque VS Position_Close+Open_Data_Training_27.11.2025_Modify.xlsx"
sheet_name = "Combination"

df = pd.read_excel("D:\Master Degree\Thesis\Test\งานพี่ต้อม Rev.2\Torque VS Position_Close+Open_Data_Training_27.11.2025_Modify.xlsx", sheet_name="Combination")
print("ข้อมูลตัวอย่าง:")
print(df.head())

Converting all column names to string type.
df.columns = df.columns.astype(str)

Preparing dataset and mapping classes.
label_mapping = {
    "Normal": 0,
    "Minor": 1,
    "Moderate": 2,
    "Sereve": 3
}
df["Class"] = df["Class"].map(label_mapping)

Defining X as all numeric feature columns except column “Class”, and y as the target label from class column.
X = df.select_dtypes(include=[np.number]).drop(columns=["Class"])
y = df["Class"]

Converting column X to string type.
X.columns = X.columns.astype(str)
```

```
Separating feature X and Target y.
X = df.drop(columns=["Class"])
y = df["Class"]

Splitting dataset for Train+Validation–Test Split (10% Hold-Out Test Set).
X_trainval, X_test, y_trainval, y_test = train_test_split(
    X, y,
    test_size=0.1,
    random_state=42,
    stratify=y
)

print("Train+Val shape:", X_trainval.shape)
print("Test shape     :", X_test.shape)

Creating model with adjustable Hyperparameters.
models = {
    "LightGBM": LGBMClassifier(
        n_estimators=10,
        learning_rate=0.001,
        random_state=42
    ),

    "AdaBoost": AdaBoostClassifier(
        n_estimators=10,
        learning_rate=0.001,
        random_state=42
    ),

    "XGBoost": XGBClassifier(
        n_estimators=10,
```

```
        learning_rate=0.001,
        max_depth=3,
        random_state=42
    ),

    "CatBoost": CatBoostClassifier(
        n_estimators=10,
        learning_rate=0.001,
        max_depth=3,
        random_state=42,
    ),

    "Random Forest": RandomForestClassifier(
        n_estimators=10,
        max_depth=3,
        random_state=42
    )
}

Configuration Stratified K-Fold.
n_splits = 5
skf = StratifiedKFold(
    n_splits=n_splits,
    shuffle=True,
    random_state=42
)
```

```
Training + Validation  K-Fold and separated testing Test set.
results = []

for name, clf in models.items():
    print("\n" + "=" * 70)
    print(f"Model: {name}")
    print("=" * 70)

    # Pipeline: StandardScaler + Model (ไม่มี SMOTE)
    base_pipeline = Pipeline(steps=[
        ("scaler", StandardScaler()),
        ("model", clf)
    ])

Cross-Validation (on Train+Val).
cv_acc, cv_pre, cv_rec, cv_f1 = [], [], [], []

    fold_id = 1
    for train_idx, val_idx in skf.split(X_trainval, y_trainval):
        X_tr = X_trainval.iloc[train_idx]
        y_tr = y_trainval.iloc[train_idx]
        X_val = X_trainval.iloc[val_idx]
        y_val = y_trainval.iloc[val_idx]

        pipe = clone(base_pipeline)
        pipe.fit(X_tr, y_tr)

        y_val_pred = pipe.predict(X_val)

        acc = accuracy_score(y_val, y_val_pred)
        pre = precision_score(y_val, y_val_pred, average="macro", zero_division=0)
```

```
        rec = recall_score(y_val, y_val_pred, average="macro", zero_division=0)
        f1  = f1_score(y_val, y_val_pred, average="macro", zero_division=0)

        cv_acc.append(acc)
        cv_pre.append(pre)
        cv_rec.append(rec)
        cv_f1.append(f1)

        print(f"[Fold {fold_id}] "
              f"Acc={acc:.4f}, Prec={pre:.4f}, Rec={rec:.4f}, F1={f1:.4f}")
        fold_id += 1

    cv_acc_mean = np.mean(cv_acc)
    cv_pre_mean = np.mean(cv_pre)
    cv_rec_mean = np.mean(cv_rec)
    cv_f1_mean  = np.mean(cv_f1)

    print("\n>> CV (mean over folds)")
    print(f"Accuracy = {cv_acc_mean:.4f}")
    print(f"Precision = {cv_pre_mean:.4f}")
    print(f"Recall    = {cv_rec_mean:.4f}")
    print(f"F1-score  = {cv_f1_mean:.4f}")
```

Model training on the Train+Validation set and evaluation on the Test set.

```
    final_pipe = clone(base_pipeline)
    final_pipe.fit(X_trainval, y_trainval)

    y_test_pred = final_pipe.predict(X_test)

    test_acc = accuracy_score(y_test, y_test_pred)
    test_pre = precision_score(y_test, y_test_pred, average="macro", zero_division=0)
```

```
test_rec = recall_score(y_test, y_test_pred, average="macro", zero_division=0)
test_f1  = f1_score(y_test, y_test_pred, average="macro", zero_division=0)

print("\n>>> Test set performance")
print(f"Accuracy = {test_acc:.4f}")
print(f"Precision = {test_pre:.4f}")
print(f"Recall    = {test_rec:.4f}")
print(f"F1-score  = {test_f1:.4f}")

cm = confusion_matrix(y_test, y_test_pred)
print("\nConfusion Matrix (Test):\n", cm)
print("\nClassification Report (Test):\n",
      classification_report(y_test, y_test_pred, zero_division=0))

plt.figure(figsize=(5, 4))
sns.heatmap(
    cm,
    annot=True,
    fmt="d",
    cmap="Blues",
    xticklabels=np.unique(y),
    yticklabels[np.unique(y)]
)
plt.title(f"Confusion Matrix (Test) - {name}")
plt.xlabel("Predicted label")
plt.ylabel("True label")
plt.tight_layout()
plt.show()

results.append({
    "Model": name,
```

```
        "CV_Accuracy": cv_acc_mean,
        "CV_Precision": cv_pre_mean,
        "CV_Recall": cv_rec_mean,
        "CV_F1": cv_f1_mean,
        "Test_Accuracy": test_acc,
        "Test_Precision": test_pre,
        "Test_Recall": test_rec,
        "Test_F1": test_f1
    })

Results.
results_df = pd.DataFrame(results).round(4)
results_df = results_df.sort_values(
    by=["Test_F1", "Test_Accuracy"],
    ascending=False
)

print("\n\n=== Summary of Model Performance (CV + Test) ===")
print(results_df.reset_in
```