การเปรียบเทียบเทคนิคการเรียนรู้ของเครื่องสำหรับการบำรุงรักษาเชิงคาดการณ์ของเครื่องวัดประสิทธิภาพหัวอ่าน HDD

สายสุดา นันต๊ะเหล็ก

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
บัณฑิตศึกษา สาขาเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีไทย-ญี่ปุ่น
ปีการศึกษา 2568

COMPARATIVE STUDY OF ENSEMBLE LEARNING FOR PREDICTIVE MAINTENANCE IN HDD MANUFACTURING

Saisuda Nantalek

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	การเปรียบเทียบเทคนิคการเรียนรู้ของเครื่องสำหรับการบำรุงรักษาเชิงคาดการณ์ของเครื่องวัดประสิทธิภาพหัวอ่าน HDD
โดย	สายสุดา นันต๊ะเหล็ก
สาขาวิชา	เทคโนโลยีสารสนเทศ
อาจารย์ที่ปรึกษาวิทยานิพนธ์	ดร.ศรายุทธ นนท์ศิริ

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.ณปภัช วิชัยดิษฐ์)

.....อาจารย์ที่ปรึกษาวิทยานิพนธ์
(ดร.ศรายุทธ นนท์ศิริ)

สายสุดา นันต๊ะเหล็ก : การเปรียบเทียบเทคนิคการเรียนรู้ของเครื่องสำหรับการบำรุงรักษาเชิงคาดการณ์ของเครื่องวัดประสิทธิภาพหัวอ่าน HDD. อาจารย์ที่ปรึกษา : ดร.ศรายุทธ นนท์ศิริ, 88 หน้า.

งานวิจัยฉบับนี้มุ่งพัฒนาแบบจำลองการบำรุงรักษาเชิงคาดการณ์สำหรับเครื่องทดสอบประสิทธิภาพหัวอ่านฮาร์ดดิสก์ โดยประยุกต์ใช้เทคนิคการเรียนรู้ของเครื่องเพื่อจำแนกสถานะการทำงานของเครื่องจักรจากข้อมูลเซ็นเซอร์เชิงอนุกรมเวลา เช่น อุณหภูมิของโมดูลและความเร็วรอบพัดลม (RPM) กระบวนการวิจัยครอบคลุมการเตรียมและปรับปรุงข้อมูล การจัดการค่าที่หายไปและค่าผิดปกติ การสร้างคุณลักษณะใหม่ และการแก้ไขปัญหาความไม่สมดุลของคลาสด้วยเทคนิค SMOTE จากนั้นพัฒนาและเปรียบเทียบแบบจำลอง 6 เทคนิค ได้แก่ XGBoost, LightGBM, CatBoost, AdaBoost, Gradient Boosting Tree และ Bagging ผลการวิจัยพบว่า CatBoost ให้ประสิทธิภาพสูงสุด โดยมีค่า Accuracy เท่ากับ 0.9830, F1-score เท่ากับ 0.9829 และ AUC เท่ากับ 0.9985 รองลงมาคือ LightGBM และ XGBoost ซึ่งแสดงให้เห็นถึงศักยภาพของโมเดลกลุ่ม Boosting ในการตรวจจับความผิดปกติได้อย่างแม่นยำ ลดการแจ้งเตือนผิดพลาด และสามารถนำไปประยุกต์ใช้ในระบบบำรุงรักษาเชิงคาดการณ์ในภาคอุตสาหกรรมได้อย่างมีประสิทธิภาพ

บัณฑิตศึกษา ลายมือชื่อนักศึกษา .....

สาขาวิชา เทคโนโลยีสารสนเทศ ลายมือชื่ออาจารย์ที่ปรึกษา .....

ปีการศึกษา 2568

SAISUDA NANTALEK : COMPARATIVE STUDY OF ENSEMBLE LEARNING FOR PREDICTIVE MAINTENANCE IN HDD MANUFACTURING. ADVISOR : DR.SARAYUT NONSIRI, 88 PP.

This research aims to develop a predictive maintenance model for hard disk drive (HDD) read-head performance testing equipment by applying machine learning techniques to classify machine operating conditions using time-series sensor data, such as module temperature and fan rotational speed (RPM). The research methodology includes data preparation and refinement, handling missing values and outliers, feature engineering, and addressing class imbalance using the Synthetic Minority Over-sampling Technique (SMOTE). Subsequently, six machine learning models are developed and compared: XGBoost, LightGBM, CatBoost, AdaBoost, Gradient Boosting Tree, and Bagging. The experimental results indicate that CatBoost achieves the highest performance, with an accuracy of 0.9830, an F1-score of 0.9829, and an AUC of 0.9985, followed by LightGBM and XGBoost. These findings demonstrate the strong potential of boosting-based models in accurately detecting anomalies, reducing false alarms, and effectively supporting the implementation of predictive maintenance systems in industrial applications.

Graduate Studies Student's Signature.....

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

Academic Year 2025

กิตติกรรมประกาศ

ผู้วิจัยขอกราบแสดงความขอบพระคุณอย่างสูงต่อ ดร.ศรายุทธ นนท์ศิริ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ผู้ซึ่งได้กรุณาให้คำแนะนำ ชี้แนะ และสนับสนุนการดำเนินงานวิจัยอย่างต่อเนื่อง ทั้งในด้านวิชาการและกระบวนการวิจัย ด้วยความเอาใจใส่ ความอดทน และความเมตตาเป็นอย่างยิ่ง การสนับสนุนจากท่านนับเป็นปัจจัยสำคัญที่ทำให้งานวิจัยฉบับนี้สามารถดำเนินการจนสำเร็จลุล่วงตามวัตถุประสงค์ที่กำหนดไว้

ผู้วิจัยขอขอบพระคุณ สถาบันเทคโนโลยีไทย–ญี่ปุ่น ที่ให้การสนับสนุนด้านทุนการศึกษา รวมถึงการจัดสภาพแวดล้อมทางวิชาการที่เอื้อต่อการเรียนรู้ การวิจัย และการพัฒนาศักยภาพทางวิชาชีพ ซึ่งมีส่วนสำคัญต่อการเสริมสร้างแนวคิดเชิงวิจัยและความก้าวหน้าทางวิชาการของผู้วิจัย นอกจากนี้ ผู้วิจัยขอขอบพระคุณ ทุนอุดหนุนงานวิจัยจาก Dr.Noriaki Kano ซึ่งมีส่วนสำคัญในการสนับสนุนทรัพยากรและส่งเสริมให้การดำเนินงานวิจัยเป็นไปอย่างมีประสิทธิภาพและสมบูรณ์ยิ่งขึ้น

ในโอกาสนี้ ผู้วิจัยขอแสดงความขอบพระคุณอย่างสูงต่อ คณะกรรมการสอบวิทยานิพนธ์ ได้แก่ รองศาสตราจารย์ ดร.กันต์พงษ์ วรรัตน์ปัญญา, ผู้ช่วยศาสตราจารย์ ดร.ฐิติพร เลิศรัตน์เดชากุล และผู้ช่วยศาสตราจารย์ ดร.ณปภัช วิชัยดิษฐ์ ที่ได้กรุณาให้เกียรติเป็นกรรมการสอบ พร้อมทั้งให้ข้อเสนอแนะอันทรงคุณค่า ซึ่งช่วยให้ผู้วิจัยสามารถปรับปรุงและพัฒนางานวิจัยให้มีความถูกต้อง สมบูรณ์ และมีคุณภาพทางวิชาการยิ่งขึ้น

ผู้วิจัยขอขอบพระคุณ บริษัท ซีเกท เทคโนโลยี (ประเทศไทย) จำกัด ที่ได้เอื้อเฟื้อข้อมูล สถานที่ และอุปกรณ์สำหรับการดำเนินงานวิจัยภาคปฏิบัติ ซึ่งมีบทบาทสำคัญอย่างยิ่งต่อความน่าเชื่อถือและประสิทธิภาพของผลการวิจัย

ท้ายที่สุดนี้ ผู้วิจัยขอแสดงความขอบพระคุณจากใจต่อ ครอบครัวและบุคคลอันเป็นที่รัก สำหรับความรัก ความเข้าใจ และการสนับสนุนทั้งทางร่างกายและจิตใจมาโดยตลอด ผู้วิจัยหวังเป็นอย่างยิ่งว่าวิทยานิพนธ์ฉบับนี้จะเป็นประโยชน์ต่อผู้ที่สนใจในด้านปัญญาประดิษฐ์และการเรียนรู้ของเครื่อง และสามารถนำองค์ความรู้ที่ได้ไปประยุกต์ใช้ในภาคอุตสาหกรรม รวมถึงการพัฒนาเทคโนโลยีของประเทศได้อย่างยั่งยืน

สายสุดา นันต๊ะเหล็ก

สารบัญ

สารบัญ (ต่อ)

สารบัญตาราง

สารบัญรูป

สารบัญรูป (ต่อ)

สารบัญรูป (ต่อ)

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของปัญหา

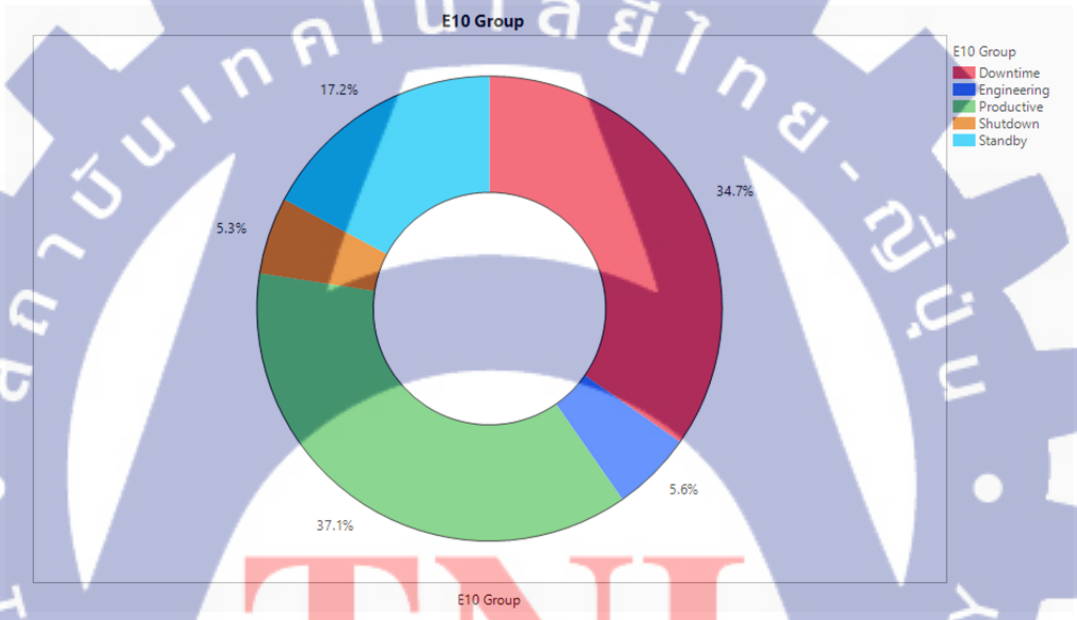
ในปัจจุบัน อุตสาหกรรมฮาร์ดดิสก์ไดรฟ์ (Hard Disk Drive: HDD) ยังคงมีบทบาทสำคัญต่อเศรษฐกิจของประเทศไทย จากความต้องการด้านการจัดเก็บข้อมูลที่เพิ่มสูงขึ้นอย่างต่อเนื่องในระดับโลก ส่งผลให้ผู้ผลิตจำเป็นต้องรักษามาตรฐานคุณภาพของผลิตภัณฑ์ในระดับสูงควบคู่กับการเพิ่มประสิทธิภาพของกระบวนการผลิต โดยเฉพาะกระบวนการตรวจวัดและประเมินสมรรถนะของหัวอ่าน-เขียน (Head Gimbal Assembly: HGA) ซึ่งต้องอาศัยเครื่องมือวิเคราะห์สมรรถนะขั้นสูง เช่น Guzik Signal Analyzer และ Read-Write Analyzer เพื่อรับรองคุณภาพของผลิตภัณฑ์ให้เป็นไปตามมาตรฐานสากล

อย่างไรก็ตาม การใช้งานเครื่องมือวิเคราะห์ดังกล่าวในสภาพแวดล้อมการผลิตจริงที่มีความแปรผันของอุณหภูมิ ความชื้น และรอบการทดสอบที่ยาวนาน โดยเฉพาะในสายการผลิต HGA ของบริษัท Seagate อาจทำให้เกิดการเสื่อมสภาพและความคลาดเคลื่อนของอุปกรณ์ ส่งผลให้เกิดการหยุดชะงักของเครื่องมือ ซึ่งส่งผลกระทบโดยตรงต่อประสิทธิภาพการผลิต จากการวิเคราะห์ข้อมูลประสิทธิผลโดยรวมของเครื่องจักร (Overall Equipment Effectiveness: OEE) ดังแสดงในรูปที่ 1.1 พบว่าเวลาที่เกิด Downtime มีสัดส่วนสูงถึง 34.7% ของกิจกรรมทั้งหมด ในขณะที่เวลาการผลิตจริง (Productive) มีสัดส่วน 37.1% และช่วงเวลาสแตนด์บาย (Standby) คิดเป็น 17.2% ข้อมูลดังกล่าวสะท้อนให้เห็นว่า Downtime เป็นหนึ่งในปัจจัยหลักที่ลดประสิทธิภาพของกระบวนการทดสอบ HDD อย่างมีนัยสำคัญ

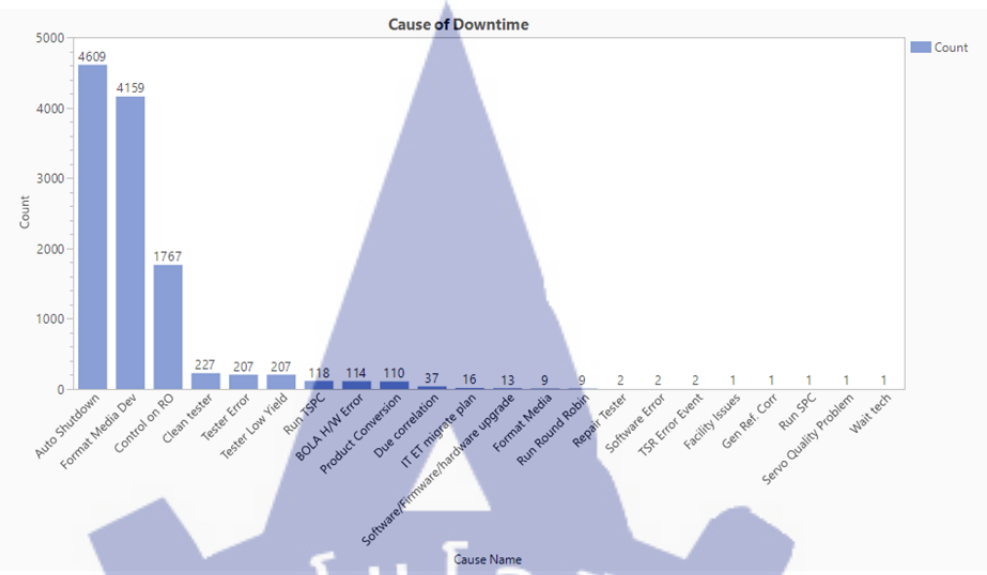
เมื่อพิจารณาสาเหตุของ Downtime ในเชิงรายละเอียดดังรูปที่ 1.2 พบว่าสาเหตุที่เกิดขึ้นบ่อยที่สุด ได้แก่ Auto Shutdown จำนวน 4,609 ครั้ง คิดเป็นสัดส่วนสูงสุดของเหตุขัดข้องทั้งหมด รองลงมาคือ Format Media Device จำนวน 4,159 ครั้ง และ Control on PO จำนวน 1,767 ครั้ง ขณะที่สาเหตุอื่น เช่น Clean Tester, Tester Error, Tester Low Yield, Run TSPC และ BOLA Hardware Error มีจำนวนการเกิดเหตุในระดับหลักร้อยครั้งต่อปี แสดงให้เห็นว่าปัญหา Downtime ส่วนใหญ่มีลักษณะการเกิดซ้ำ (Recurring Failures) และเกี่ยวข้องกับการทำงานของระบบและอุปกรณ์ทดสอบโดยตรง ซึ่งส่งผลให้เกิดการสูญเสียเวลาในการแก้ไขปัญหาและการซ่อมบำรุงเป็นจำนวนมาก

นอกจากนี้ จากการจำแนกสาเหตุของความผิดปกติตามความเกี่ยวข้องกับฮาร์ดแวร์ดังรูปที่ 1.3 พบว่าปัญหาที่เกี่ยวข้องกับฮาร์ดแวร์คิดเป็นสัดส่วน 24.7% ของความผิดปกติทั้งหมด ในขณะที่ปัญหาที่ไม่เกี่ยวข้องกับฮาร์ดแวร์มีสัดส่วน 75.3% แม้ปัญหาฮาร์ดแวร์จะมีสัดส่วนไม่ถึงครึ่งหนึ่งของเหตุขัดข้องทั้งหมด แต่ลักษณะของปัญหาดังกล่าวมักก่อให้เกิด Downtime ที่ยาวนาน ต้องใช้เวลาในการวิเคราะห์และซ่อมบำรุงสูง รวมถึงมีต้นทุนด้านอะไหล่และแรงงานที่มากกว่าปัญหาประเภทอื่น ส่งผลกระทบต่อประสิทธิภาพการผลิตโดยรวมอย่างมีนัยสำคัญ

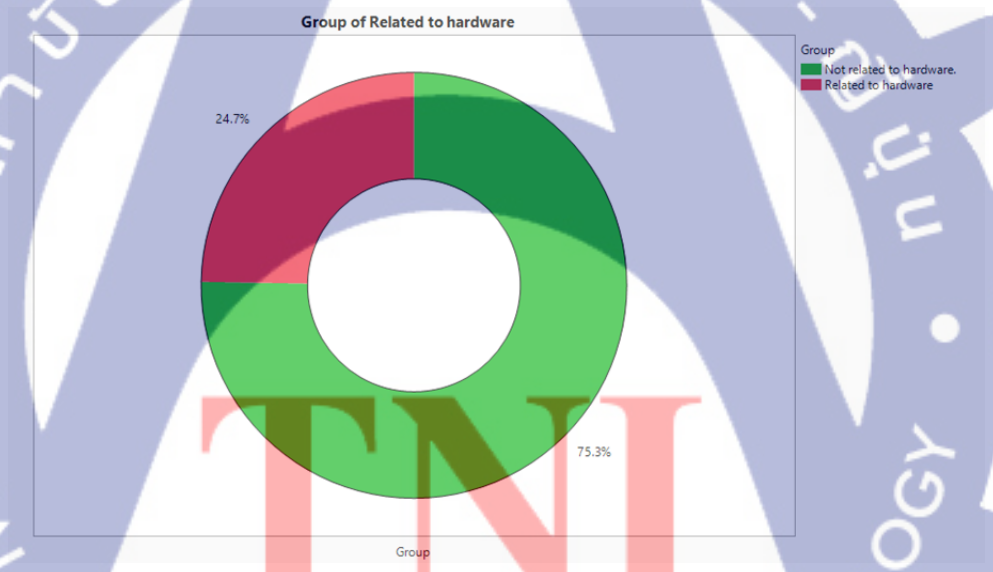
จากข้อมูลข้างต้น แสดงให้เห็นว่าการบำรุงรักษาเชิงป้องกันแบบดั้งเดิมยังไม่สามารถตอบโจทย์การลด Downtime ของอุปกรณ์ทดสอบที่มีความซับซ้อนและมีปริมาณข้อมูลจำนวนมากได้อย่างมีประสิทธิภาพ ดังนั้น แนวคิดการบำรุงรักษาเชิงคาดการณ์ (Predictive Maintenance) ซึ่งอาศัยการวิเคราะห์ข้อมูลจากเซ็นเซอร์และตัวแปรกระบวนการ เช่น อุณหภูมิ ความชื้น รอบการทดสอบ และสถานะการทำงานของอุปกรณ์ ร่วมกับเทคนิคการเรียนรู้ของเครื่อง (Machine Learning) เพื่อคาดการณ์ความผิดปกติล่วงหน้า จึงเป็นแนวทางที่มีศักยภาพในการลดการหยุดชะงัก เพิ่มความพร้อมใช้งานของเครื่องมือ และยกระดับประสิทธิภาพการผลิต งานวิจัยนี้จึงมุ่งเน้นการพัฒนาโมเดลการบำรุงรักษาเชิงคาดการณ์สำหรับอุปกรณ์ทดสอบ HDD โดยใช้ข้อมูลจริงจากกระบวนการผลิต HGA ของบริษัท Seagate เพื่อสนับสนุนการจัดการซ่อมบำรุงอย่างเป็นระบบและยั่งยืน



รูปที่ 1.1 การวิเคราะห์ประสิทธิผลโดยรวมของเครื่องจักร



รูปที่ 1.2 การวิเคราะห์ของสาเหตุการหยุดทำงาน



รูปที่ 1.3 การวิเคราะห์สัดส่วนของสาเหตุที่เกี่ยวข้องหรือไม่เกี่ยวข้องกับฮาร์ดแวร์

1.2 วัตถุประสงค์ของการวิจัย

1.2.1 เพื่อพัฒนาโมเดลบำรุงรักษาเชิงคาดการณ์สำหรับเครื่องควบคุมมาตรฐานและประเมินสุขภาพเครื่องวัดประสิทธิภาพหัวอ่าน HDD

1.2.2 เพื่อเปรียบเทียบเทคนิคการเรียนรู้ของเครื่องเพื่อคัดเลือกวิธีที่เหมาะสมที่สุดและช่วยลดต้นทุนด้านการบำรุงรักษา

1.3 ขอบเขตของงานวิจัย

1.3.1 ชุดข้อมูลได้มาจากเซนเซอร์อุณหภูมิของเครื่อง HTS ซึ่งประกอบไปด้วย Guzik Signal Analyzer GSA 6000 Series และ Read-Write Analyzer Systems 4000 TDMR Seriesเทคนิคการเรียนรู้ของเครื่องจักรที่ใช้ในการศึกษาประกอบด้วยโมเดลดังนี้

1) XGBoost
2) LightGBM
3) AdaBoost
4) CatBoost
5) GBT
6) Bagging

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1.4.1 โมเดลสำหรับการคาดการณ์ความเสียหายของเครื่องควบคุมมาตรฐานและประเมินประสิทธิภาพหัวอ่าน HDD

1.4.2 ได้เทคนิคการเรียนรู้ของเครื่องที่เหมาะสมกับชุดข้อมูล

1.4.3 เพื่อลดต้นทุนที่เกิดจากการซ่อมบำรุงอันเนื่องมาจากการรอจัดซื้ออะไหล่ที่มีระยะเวลานาน ลดการหยุดชะงักของกระบวนการผลิต และสนับสนุนการเพิ่มค่า Overall Equipment Effectiveness (OEE) ซึ่งเป็นเป้าหมายสำคัญของการเพิ่มประสิทธิภาพเครื่องจักร

1.5 แผนการดำเนินงาน

ตารางที่ 1.1 แผนการดำเนินงาน

ที่	รายการ	ระยะเวลาในการดำเนินงาน 2568												หมายเหตุ
		1	2	3	4	5	6	7	8	9	10	11	12	
1	ศึกษาเอกสารและงานวิจัยที่เกี่ยวข้อง													
2	เก็บรวบรวมและเตรียมข้อมูลจากเครื่องจักร													
3	การประมวลผลและทำความสะอาดข้อมูล													

ตารางที่ 1.1 แผนการดำเนินงาน (ต่อ)

ที่	รายการ	ระยะเวลาในการดำเนินงาน 2568												หมายเหตุ
		1	2	3	4	5	6	7	8	9	10	11	12	
4	พัฒนาและฝึกสอนโมเดล Machine Learning													
5	ประเมินและเปรียบเทียบประสิทธิภาพโมเดล													
6	เขียนรายงานผลการวิจัย													
7	จัดทำและส่งตีพิมพ์/นำเสนอผลงาน													

บทที่ 2
การทบทวนวรรณกรรม

2.1 วิวัฒนาการของอุตสาหกรรมฮาร์ดดิสก์ไดรฟ์

ฮาร์ดดิสก์ไดรฟ์ (HDD) ถือกำเนิดขึ้นครั้งแรกในปี ค.ศ. 1956 ในฐานะส่วนหนึ่งของระบบ IBM 305 RAMAC โดยใช้ชื่อหน่วยเก็บข้อมูลว่า IBM 350 Disk File ซึ่งนับเป็นอุปกรณ์เก็บข้อมูลเชิงพาณิชย์ตัวแรกที่สามารถเข้าถึงข้อมูลแบบสุ่ม (Random Access) ได้สำเร็จ ในยุคเริ่มต้นนั้น มันมีความจุเพียงประมาณ 3.75 เมกะไบต์ (MB) ซึ่งถูกจัดเก็บอยู่บนแผ่นดิสก์แม่เหล็กขนาดใหญ่ 24 นิ้ว จำนวนถึง 50 แผ่น การมาถึงของเทคโนโลยีนี้ถือเป็นการปฏิวัติครั้งสำคัญในประวัติศาสตร์การจัดเก็บข้อมูล คอมพิวเตอร์ [1]

ในปี ค.ศ. 1961 IBM ได้สร้างนวัตกรรมที่สำคัญอีกครั้งด้วยการเปิดตัว IBM 1301 ซึ่งได้นำเทคโนโลยี หัวอ่านแบบลอย (Flying Head) มาใช้งาน โดยอาศัยแรงดันอากาศในการพยุงหัวอ่านให้ลอยตัวเหนือผิวหน้าของดิสก์ การออกแบบนี้ช่วยลดการสัมผัสและลดการสึกหรอได้อย่างมีนัยสำคัญ ส่งผลให้ความเร็วและความน่าเชื่อถือของอุปกรณ์เพิ่มขึ้นอย่างมาก ซึ่งเป็นรากฐานสำคัญที่นำไปสู่การออกแบบ HDD ที่มีขนาดเล็กลงและมีประสิทธิภาพสูงขึ้นในเวลาต่อมา [2]

ในช่วงทศวรรษ 1980 ถึง 1990 HDD ได้เข้าสู่ตลาดคอมพิวเตอร์ส่วนบุคคลอย่างเต็มรูปแบบ ขนาดของแผ่นดิสก์ถูกลดลงอย่างต่อเนื่องจาก 5.25 นิ้ว จนกลายเป็นมาตรฐาน 3.5 นิ้ว และมีการพัฒนาอินเทอร์เฟซ (Interface) ที่สำคัญ เช่น Integrated Drive Electronics (IDE) และ Small Computer System Interface (SCSI) ในช่วงเวลานี้ ความหนาแน่นของข้อมูลได้พุ่งสูงขึ้นจากระดับเมกะบิตต่อตารางนิ้วไปสู่กิกะบิตต่อตารางนิ้ว ซึ่งเป็นผลให้ความจุของ HDD ขยายตัวอย่างรวดเร็วเพื่อตอบสนองต่อความต้องการการจัดเก็บข้อมูลที่เพิ่มขึ้นอย่างมหาศาลในยุคนั้น [3]

ในปัจจุบัน เทคโนโลยี HDD ได้พัฒนาไปสู่ขั้นสูงยิ่งขึ้น โดยใช้เทคโนโลยีหลักอย่าง Giant Magnetoresistance (GMR) และ Perpendicular Magnetic Recording (PMR) ขณะเดียวกันอุตสาหกรรมก็ยังคงวิจัยและพัฒนาเทคโนโลยีล้ำสมัยอย่างต่อเนื่อง เช่น Heat-Assisted Magnetic Recording (HAMR) และ Bit Patterned Media (BPM) เพื่อผลักดันความหนาแน่นของพื้นที่เก็บข้อมูล (Areal Density) ให้เกิน 2 เทระบิตต่อตารางนิ้ว ซึ่งจะช่วยให้ HDD สามารถมีความจุสูงเกิน 100 เทระไบต์ ได้ในอนาคต แม้ว่าตลาดจะมีการแข่งขันอย่างรุนแรงจาก SSD แต่ HDD ยังคงเป็นตัวเลือกที่สำคัญและคุ้มค่าที่สุดสำหรับงานที่ต้องการความจุสูงในราคาต้นทุนต่อหน่วยที่ต่ำ โดยเฉพาะอย่างยิ่งในศูนย์ข้อมูลและระบบคลาวด์ [4]

2.2 อุตสาหกรรมการผลิตฮาร์ดดิสก์ไดรฟ์

อุตสาหกรรมฮาร์ดดิสก์ไดรฟ์ (HDD) ยังคงมีความสำคัญอย่างมากในระบบจัดเก็บข้อมูลระดับองค์กรและศูนย์ข้อมูล แม้ว่า SSD จะเติบโตและได้รับความนิยมสูง แต่ HDD ยังคงได้เปรียบในด้านต้นทุนต่อเทระไบต์ (Cost Per TB) และความจุสูง เป็นเหตุผลให้บริษัทขนาดใหญ่ เช่นผู้ให้บริการคลาวด์ยังคงใช้ HDD ในการจัดเก็บข้อมูลแบบเย็น (cold storage) และสำหรับการสำรองข้อมูล โดยอ้างอิงจากบทวิเคราะห์แนวโน้มตลาด HDD [5]

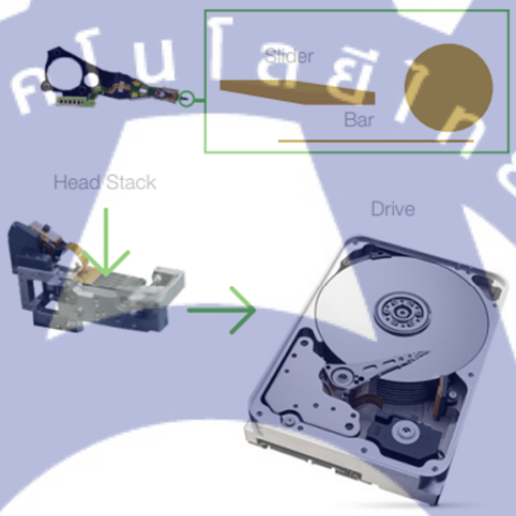
ในแง่เทคโนโลยีการอ่าน-เขียน (Read-Write) HDD มีการพัฒนาอย่างต่อเนื่องเพื่อเพิ่มความหนาแน่น (Areal Density) โดยหนึ่งในเทคโนโลยีเด่นคือ Heat-Assisted Magnetic Recording (HAMR) ซึ่งใช้เลเซอร์เพื่อให้จุดบนแผ่นจานร้อนขึ้นชั่วขณะ ทำให้สามารถเขียนข้อมูลบนเมีย (Media) ที่มี Coercivity สูงได้ ซึ่งช่วยให้เพิ่ม Density ได้มากขึ้น [6] นอกจากนี้คาดการณ์ว่า HDD ความจุสูง (50+ TB) จะสามารถพัฒนาได้ในอนาคตผ่านเทคโนโลยี EAMR (Energy-Assisted Magnetic Recording) [7]

อีกกลุ่มเทคโนโลยีที่สำคัญคือ Two-Dimensional Magnetic Recording (TDMR) ซึ่งเป็นเทคนิคการอ่านข้อมูลด้วยหัวอ่านหลายตัว (Multi-Read Heads) พร้อมกัน เพื่อช่วยลดสัญญาณรบกวนระหว่างแทร็ก (Inter-Track Interference) และเพิ่มความแม่นยำในการอ่าน ในระบบ TDMR มีการใช้กระบวนการประมวลผลสัญญาณขั้นสูงเพื่อแยกข้อมูลที่ซ้อนทับกันจากแทร็กที่อยู่ใกล้เคียง ลดความคลาดเคลื่อนของสัญญาณ และปรับปรุงความสามารถในการตรวจจับบิตข้อมูล นอกจากนี้ยังมีการใช้เทคนิคกำหนดรูปแบบรหัส (Constrained Coding) เพื่อหลีกเลี่ยงลักษณะลำดับบิตที่มีแนวโน้มก่อให้เกิดข้อผิดพลาด เช่น รูปแบบ Plus Isolation ซึ่งช่วยเพิ่มความเสถียรและความน่าเชื่อถือของระบบบันทึกข้อมูลภายใต้ความหนาแน่นการจัดเก็บที่สูงมาก [8]

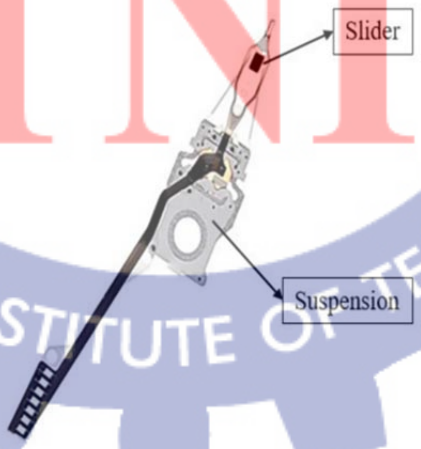
หัวอ่าน-เขียน (Read/Write Head) ถือเป็นจุดสำคัญทางฟิสิกส์และวิศวกรรม เนื่องจากระยะห่าง (Clearance) ระหว่าหัวและจาน (Head-Disk Spacing) อยู่ในระดับนาโนเมตร การควบคุมระยะห่างนี้เป็นหัวใจของความน่าเชื่อถือ (Reliability) เพราะหากลดต่ำเกินไป อาจเกิดปัญหา Head Crash, การเสียดสีระหว่างหัวกับจาน หรือการสึกหรอของชั้นหล่อลื่นได้ จากงานวิจัยได้ชี้ให้เห็นถึง Head Protrusion ที่อาจเพิ่มโอกาสสัมผัสกับจานเมื่อเกิดการเขียนข้อมูล [9] และอีกงานหนึ่งแสดงให้เห็นว่าเมื่อต้องการ Close-to-Contact Recording ความหนาแน่นของแทร็กสูงขึ้น ความใกล้ชิดของหัวกับจานเพิ่มขึ้นอย่างมาก ส่งผลให้การสึกหรอที่ Interface มากขึ้นด้วย [10] โมเดลความน่าเชื่อถืออื่นยังรวมผลของ Clearance ที่เปลี่ยนแปลงตามอุณหภูมิและเงื่อนไขสภาพแวดล้อมเข้าด้วย เพื่อทำนายอัตราการล้มเหลวของ HDD [11]

2.3 กระบวนการ Head Gimbal Assembly Process และบทบาทในคุณภาพ HDD

กระบวนการ Head Gimbal Assembly (HGA) หรือการประกอบชุดหัวอ่านของฮาร์ดดิสก์ (HDD) เป็นขั้นตอนสำคัญที่เปรียบได้กับการผ่าตัดที่ต้องใช้ความแม่นยำสูง โดยเป็นการนำ Slider ซึ่งมีหัวอ่าน-เขียนข้อมูลอยู่ มาประกอบเข้ากับ Flexure และ Suspension ดังรูปที่ 2.1 และ 2.2 ซึ่งทำหน้าที่เป็นแขนรองรับและเชื่อมต่อไฟฟ้า ขั้นตอนนี้ต้องอาศัยเทคโนโลยีระดับไมครอน ตั้งแต่การบัดกรีและการต่อสายไฟขนาดจิ๋ว ไปจนถึงการตรวจสอบด้วยแสงและไฟฟ้าอย่างเข้มงวด เพื่อให้แน่ใจว่าตำแหน่ง มุม และความสูงของหัวอ่าน (flying height) เป็นไปตามข้อกำหนดที่แม่นยำที่สุด เพราะความผิดพลาดเพียงเล็กน้อยก็อาจส่งผลต่อความสามารถในการอ่าน-เขียนข้อมูลที่แม่นยำของ HDD ได้ [12]



รูปที่ 2.1 ส่วนประกอบหลักของ HDD [13]



รูปที่ 2.2 กระบวนการ Head Gimbal Assembly (HGA) [14]

ความเที่ยงตรงและการควบคุมคุณภาพในแต่ละขั้นตอนของ HGA จึงมีอิทธิพลโดยตรงต่อประสิทธิภาพของ HDD ความเบี่ยงเบนเพียงเล็กน้อยของตำแหน่งหรือการปนเปื้อนอนุภาคสามารถทำให้เกิด Head Crash, Elevated Error Rates หรือการลดอายุการใช้งานของ HDD ได้ ดังนั้นผู้ผลิตจึงบูรณาการการวัดและระบบควบคุม (Metrology & Process Control) ในสายการผลิต HGA ตั้งแต่ระยะพัฒนาไปจนถึงการผลิตในปริมาณมากเพื่อรักษาคุณภาพของผลิตภัณฑ์

อีกมุมที่สำคัญคือ HGA ไม่ได้เป็นเพียงชิ้นส่วนทางกลไฟฟ้าธรรมดา แต่เป็นจุดรวมของเทคโนโลยีการผลิตขั้นสูงรวมถึงวัสดุที่ผ่านการเคลือบแบบบาง การต่อระดับนาโน และการประกอบที่ต้องการห้องสะอาดและอุปกรณ์อัตโนมัติที่มีความแม่นยำสูง ซึ่งทำให้การลงทุนด้านอุปกรณ์ ทรัพยากรบุคคล และการควบคุมกระบวนการเป็นปัจจัยสำคัญในการกำหนดต้นทุนต่อหน่วยและความสามารถในการแข่งขันของผู้ผลิต HDD ในระดับสากล

ในเชิงยุทธศาสตร์ การยกระดับกระบวนการ HGA ด้วยการนำวิธีการวิเคราะห์ข้อมูลและการควบคุมเชิงสถิติมาใช้ ร่วมกับการทดสอบเชิงไฟฟ้าและเชิงกลอย่างต่อเนื่อง จะช่วยลดข้อบกพร่อง เพิ่มความสม่ำเสมอของคุณภาพ และเร่งการปรับปรุงกระบวนการเมื่อเปลี่ยนรุ่นผลิตภัณฑ์ การบริหารจัดการห่วงโซ่อุปทานที่แนบชิดกับการพัฒนากระบวนการผลิต HGA ก็ยังเป็นกุญแจสำคัญในการรักษาคุณภาพในการผลิตจำนวนมาก ทั้งหมดนี้ชี้ชัดว่าการควบคุมและพัฒนา HGA ยังคงเป็นหัวใจสำคัญของคุณภาพ HDD สมัยใหม่

<u>กระบวนการผลิตและบทบาทของระบบทดสอบ</u>

ในกระบวนการผลิตฮาร์ดดิสก์ไดรฟ์ (HDD) ระบบทดสอบได้รับการจัดวางในหลายตำแหน่งของสายการผลิต เพื่อคัดกรองข้อบกพร่องตั้งแต่ต้นทางและเพิ่มอัตราการผลิตที่ผ่านมาตรฐาน โดยเริ่มตั้งแต่การตรวจสอบชิ้นส่วนที่เข้ามา (Incoming Component Test) จนถึงการทดสอบขั้นสุดท้าย (Final Test/Certify) ซึ่งทำให้สามารถตรวจจับปัญหาได้ตั้งแต่ยังอยู่ในขั้นตอนต้น ลดต้นทุนการซ่อมแซมหรือคัดทิ้งในขั้นตอนท้ายได้อย่างมาก [15] เส้นแบ่งของการวางเครื่องมือทดสอบอย่างเหมาะสมจึงมีผลอย่างมากต่อเวลาผลิตรวม (cycle time) และต้นทุนของโรงงาน พร้อมทั้งช่วยรักษาคุณภาพของผลิตภัณฑ์ให้เสถียร

การตรวจสอบคุณภาพมีบทบาทสำคัญอย่างยิ่ง เนื่องจาก HDD ใช้หัวอ่าน-เขียนที่มีขนาดนาโนเมตรและระยะบิน (Flying Height) ต่ำมาก ซึ่งทำให้ระบบมีความไวต่อความคลาดเคลื่อนทางกลและไฟฟ้า การทดสอบฟังก์ชันเช่นการอ่าน–เขียน (Read/Write Performance) และการทดสอบความทนทาน (Reliability Stress Test) จึงเป็นส่วนที่ไม่สามารถละเลยได้ เพื่อให้แน่ใจว่าอุปกรณ์จะมีประสิทธิภาพและความเสถียรตามมาตรฐานที่ลูกค้ากำหนด ยิ่งไปกว่านั้น การบูรณาการระหว่างกระบวนการผลิตกับการทดสอบช่วยให้กระบวนการทั้งระบบมีประสิทธิภาพยิ่งขึ้น

ความเชื่อมโยงระหว่างระบบทดสอบกับ Final Yield มีความชัดเจน การตรวจพบ Defect ตั้งแต่ต้นสายและการทดสอบที่มีประสิทธิภาพช่วยลดจำนวนชิ้นงานที่ต้องถูกคัดทิ้งในขั้นตอนท้าย ซึ่งมีต้นทุนสูงกว่า การนำผลการทดสอบมาใช้เป็น Feedback เพื่อปรับจูนกระบวนการประกอบและสอบเทียบจึงช่วยเพิ่มอัตราผลผลิตที่ผ่านมาตรฐาน (Yield) อย่างมีนัยสำคัญ โดยเฉพาะในสภาพที่ความหนาแน่นของบิตเพิ่มขึ้น เครื่องผลิตและเครื่องทดสอบต้องรองรับความแม่นยำสูง เช่น “การวางตำแหน่งหัวอ่านให้อยู่ในระยะคลาดเคลื่อนน้อยกว่า 3% ของระยะแทร็ก” ซึ่งเป็นการกำหนดมาตรฐานด้านความแม่นยำของเครื่องทดสอบในผลิตภัณฑ์ HDD

นอกจากนี้ ระบบทดสอบยังทำหน้าที่เป็นเครื่องมือในการติดตามเสถียรภาพของกระบวนการผลิต (Process Drift Monitoring) เครื่องจักรหรืออุปกรณ์ผลิตที่เริ่มเสื่อมสภาพอาจก่อให้เกิดข้อผิดพลาดซ้ำหรือเพิ่มอัตราการเสียหายได้ การทดสอบในลักษณะนี้ช่วยวินิจฉัยแนวโน้มของข้อผิดพลาดก่อนที่จะส่งผลกระทบต่อ Yield อย่างเป็นระบบ ดังนั้นการลงทุนในระบบทดสอบที่มีประสิทธิภาพ จึงไม่ใช่เพียงการประกันคุณภาพของผลิตภัณฑ์ แต่อีกทั้งเป็นกลไกเชิงกลยุทธ์ที่ช่วยรักษาประสิทธิภาพและความสามารถในการแข่งขันของโรงงานผลิต HDD

2.4 อุปกรณ์ทดสอบ HDD และปัญหาที่เกิดขึ้น

ในอุตสาหกรรมการผลิตฮาร์ดดิสก์ไดรฟ์ (HDD) เครื่องมือทดสอบมีบทบาทสำคัญในการตรวจสอบคุณภาพและความถูกต้องของสัญญาณอ่าน-เขียนในกระบวนการผลิต เครื่องมือที่ใช้อย่างแพร่หลาย ได้แก่ Guzik Signal Analyzer (GSA) 6000 Series ซึ่งรองรับการวิเคราะห์สัญญาณอ่าน-เขียนความละเอียดสูงด้วย Gigahertz Bandwidth (GHz) ช่วยตรวจสอบสัญญาณเชิงลึก เช่น Amplitude, Asymmetry, Transition Noise และความผิดปกติในระดับบิตได้อย่างแม่นยำ อีกทั้ง Read-Write Analyzer (RWA) 4000 TDMR Series ดังรูปที่ 2.3 ยังรองรับการทดสอบเทคโนโลยี Two-Dimensional Magnetic Recording (TDMR) ผ่านการวิเคราะห์สัญญาณหลายช่อง (Multi-Reader) การประเมิน Multi-Track Interference การตรวจสอบ Servo Tracking และการทำ Media Certification อุปกรณ์ทั้งสองจึงทำหน้าที่เป็นเครื่องมือวิจัยและควบคุมคุณภาพที่สำคัญในทั้ง R&D และสายการผลิต

รูปที่ 2.3 Guzik Signal Analyzer (GSA) 6000 Series

แม้อุปกรณ์ทดสอบจะมีความแม่นยำสูง แต่ยังคงพบปัญหาที่ส่งผลต่อกระบวนการผลิตอย่างมีนัยสำคัญ หนึ่งในความผิดปกติที่พบได้บ่อยคือ Auto Shutdown ซึ่งเกิดจากความร้อนสูงเกินค่าที่กำหนดหรือความผิดปกติของชุดควบคุมภายใน ทำให้อุปกรณ์หยุดทำงานกะทันหันระหว่างการทดสอบ นอกจากนี้ยังพบ Media Format Error ซึ่งมักเกี่ยวข้องกับความผิดปกติของสื่อแม่เหล็กหรือสัญญาณอ่าน-เขียนที่ผิดรูป และ PO Control Error ซึ่งสัมพันธ์กับวงจรควบคุมเสียงรบกวน (Positioning or Power Oscillation Control) ที่ทำให้การอ่านตำแหน่งแทร็กผิดพลาด ปัญหาเหล่านี้ลดความเชื่อถือได้ (Reliability) ของอุปกรณ์ทดสอบและเพิ่มอัตรางานเสียระหว่างกระบวนการตรวจสอบคุณภาพ

ผลกระทบเชิงเศรษฐกิจของความผิดปกติในอุปกรณ์ทดสอบมีทั้งทางตรงและทางอ้อม เมื่อเครื่องทดสอบ GSA หรือ RWA หยุดทำงาน โรงงานต้องเผชิญกับ Downtime ของสายการผลิตทันที เนื่องจากเครื่องทดสอบมักเป็นจุดคอขวด (Bottleneck) ในกระบวนการตรวจสอบ HDD ความล่าช้าเพียงไม่กี่นาทีสามารถสะสมเป็นชั่วโมงหรือวัน ส่งผลให้จำนวนชิ้นงานผ่านการทดสอบลดลงอย่างมีนัยสำคัญ นอกจากนี้ ความผิดปกติที่ไม่ได้รับการตรวจพบอาจทำให้ชิ้นงานคุณภาพต่ำถูกส่งต่อไปยังขั้นตอนถัดไป ส่งผลให้ต้นทุน Rework และ Scrap สูงขึ้น

ด้วยเหตุนี้ โรงงานหลายแห่งจึงเริ่มให้ความสำคัญกับระบบตรวจวิเคราะห์สภาพเครื่องมือทดสอบเชิงคาดการณ์ (Predictive Maintenance) เพื่อระบุสัญญาณความผิดปกติของอุปกรณ์ก่อนที่ความล้มเหลวจะเกิดขึ้นจริง โมเดลการเรียนรู้ของเครื่องสามารถใช้ข้อมูลเซนเซอร์ เช่น อุณหภูมิ แรงดันสัญญาณ และพฤติกรรมการอ่าน-เขียน เพื่อทำนายเหตุการณ์ผิดปกติ เช่น Auto Shutdown หรือ Media Format Error การทำนายล่วงหน้าเช่นนี้ช่วยลด Downtime และลดต้นทุนการบำรุงรักษา พร้อมทั้งเพิ่มความเสถียรของกระบวนการผลิต ซึ่งเป็นปัจจัยสำคัญต่อการแข่งขันของผู้ผลิต HDD ในระดับสากล

2.5 ปัจจัยที่ส่งผลต่อการเสื่อมสภาพของเครื่องทดสอบ

การเสื่อมสภาพของอุปกรณ์ทดสอบ (Test and Measurement Equipment) เป็นปัญหาสำคัญในอุตสาหกรรมเนื่องจากส่งผลต่อความแม่นยำ ความน่าเชื่อถือ และอายุการใช้งานของเครื่องมือ โดยหนึ่งใน ปัจจัยหลัก คือ อุณหภูมิ (Temperature) เมื่ออุปกรณ์ทำงานในสภาวะอุณหภูมิสูง ความเครียดทางความร้อน (Thermal Stress) จะเร่งปฏิกิริยาเสื่อมสภาพในชิ้นส่วนอิเล็กทรอนิกส์ภายใน เช่น การทดสอบแบบ Highly Accelerated Stress Test (HAST) ใช้อุณหภูมิสูงร่วมกับความชื้นเพื่อเร่งการเกิด Failure Mechanisms โดยอัตราเร่งถูกอธิบายโดยสมการ Arrhenius ซึ่งสัมพันธ์กับพลังงานการกระตุ้นทางอุณหภูมิ (Activation Energy) [16] นอกจากนี้การเปลี่ยนแปลงอุณหภูมิแบบรอบ (Thermal Cycling) ยังสามารถก่อให้เกิดความเมื่อยล้าทางวัสดุ (Material Fatigue) และแตกหักของบัดกรีได้ ซึ่งลดประสิทธิภาพและอายุของอุปกรณ์ได้อย่างมาก

ถัดมาความชื้น (Humidity) เป็นอีกปัจจัยสำคัญที่ส่งผลต่อการเสื่อมสภาพ เพราะความชื้นสูงอาจทำให้เกิดการกัดกร่อน (Corrosion) และการอิเล็กโตรเคมิคอลมิเกรชัน (Electrochemical Migration) ภายในองค์ประกอบอุปกรณ์ โดยเฉพาะอย่างยิ่งเมื่ออยู่ภายใต้ Bias ร่วมกับอุณหภูมิ (Temperature-Humidity-Bias Test) ซึ่งเป็นมาตรฐานทดสอบเร่งอายุ (Accelerated Life Test) สำหรับอุปกรณ์อิเล็กทรอนิกส์ [17] การวิจัยพบว่าการทดสอบ THB (Temperature-Humidity-Bias) ที่ค่า 85°C/85% RH เป็นจุดเริ่มต้นทั่วไปสำหรับการจำลองการกัดกร่อนในระยะยาว [18] ผลกระทบทางความชื้นจึงไม่เพียงแต่ส่งผลต่ออุปกรณ์ภายนอก แต่ยังสามารถส่งผลต่อวงจรภายใน ทำให้ประสิทธิภาพลดลงเมื่อเวลาผ่านไป รอบการทดสอบ (Testing Cycles or Cycling) เป็นปัจจัยอีกประการที่เร่งการเสื่อมสภาพของอุปกรณ์ เมื่ออุปกรณ์ถูกใช้งานและทดสอบซ้ำๆ หลายรอบ (เช่น การเปิด-ปิด, การ Ramp อุณหภูมิ, การสลับโหลด) จะเกิดแรงเครียดเชิงกลและความร้อนสะสมในชิ้นส่วน ทำให้วัสดุเกิด Fatigue และ Micro-Cracks ซึ่งสะสมจนทำให้เกิดความล้มเหลว (Failure) ในที่สุด แนวทางการประเมิน Degradation จากรอบการทดสอบมักใช้การทดสอบเร่งรอบ (Accelerated Cycling) ร่วมกับโมเดลทางฟิสิกส์ (Physics-of-Failure) หรือการวิเคราะห์ Stochastic Degradation Process เพื่อทำนายอายุที่เหลือ (Remaining Useful Life, RUL) [19]

ส่วนการลอยของเซนเซอร์ (Sensor Drift) เป็นปรากฏการณ์ที่ค่าเอาต์พุตของเซนเซอร์เปลี่ยนแปลงเมื่อเวลาผ่านไป แม้สภาวะแวดล้อมภายนอกจะไม่เปลี่ยนแปลง ซึ่งส่งผลต่อความถูกต้องของการวัดในระยะยาว ตัวอย่างในงานวิจัยของ MEMS Flow Sensor พบว่าที่อุณหภูมิสูง (เช่น 150°C) เซนเซอร์มี Drift สูงถึง ~3.15% หลังการทดสอบเร่งอายุ และระบบประมวลผลสัญญาณ (Signal Processing System) มีการเสื่อมสภาพอย่างชัดเจนมากขึ้น [20] นอกจากนี้ Drift อาจมาจากการเปลี่ยนแปลงพฤติกรรมทางเคมีหรือโครงสร้างของวัสดุภายในเซนเซอร์ ซึ่งหากไม่ถูกชดเชยหรือปรับเทียบใหม่ (Re-calibration) อุปกรณ์ทดสอบอาจให้ผลวัดที่คลาดเคลื่อน

สุดท้ายอายุของอุปกรณ์ (Component Aging) เป็นปัจจัยสั่งสำคัญที่สะสมผลกระทบจากปัจจัยอื่นๆ อย่างถาวร การเสื่อมสภาพเนื่องจาก Aging อาจเกิดจากหลายกลไก เช่น การออกซิเดชันของโลหะภายใน อัตราการอพยพของไอออน (Ion Migration) หรือการผุพังของวัสดุฉนวน เมื่อเวลาผ่านไป ค่าเหล่านี้ส่งผลให้ประสิทธิภาพการทำงานลดลงและความน่าเชื่อถือของอุปกรณ์ตกต่ำ งานวิเคราะห์ Reliability ของอุปกรณ์อิเล็กทรอนิกส์มักใช้แนวทาง Physics-of-Failure หรือ Accelerated Degradation Testing เพื่อทำนายอายุการใช้งานและวางแผนบำรุงรักษาเชิงคาดการณ์ (Predictive Maintenance) [21]

2.6 แนวคิดด้านความเชื่อถือได้ของอุปกรณ์

ความเชื่อถือได้ (Reliability) ของระบบหรืออุปกรณ์หมายถึงโอกาสที่องค์ประกอบนั้นทำงานได้โดยไม่มีข้อผิดพลาดภายใต้เงื่อนไขการทำงานที่กำหนดในช่วงเวลาหนึ่ง วิศวกรรมความเชื่อถือได้มุ่งวัดและจัดการกับปัจจัยที่อาจทำให้เกิดการล้มเหลว (Failure) โดยใช้ตัวชี้วัดสำคัญอย่าง MTBF (Mean Time Between Failures) และ MTTR (Mean Time To Repair) MTBF คือค่าเฉลี่ยของเวลาระหว่างความล้มเหลวของอุปกรณ์หนึ่งรอบถึงรอบถัดไป ส่วน MTTR คือเวลาเฉลี่ยที่ใช้ในการซ่อมแซมและนำระบบกลับมาใช้งานอีกครั้ง [21] ช่วยให้ผู้จัดการและวิศวกรประเมินประสิทธิภาพของแผนการบำรุงรักษา (maintenance) และความพร้อมใช้งาน (availability) ของอุปกรณ์

อัตราความล้มเหลว (Failure Rate, มักใช้สัญลักษณ์ λ) เป็นอีกแนวคิดพื้นฐานในวิศวกรรมความเชื่อถือได้ ซึ่งบ่งชี้ถึงจำนวนข้อผิดพลาดที่คาดว่าจะเกิดขึ้นต่อหน่วยเวลา อัตรานี้อาจเป็นค่าคงที่ในระบบบางประเภท (เช่น เมื่อสมมติ Exponential Distribution) หรืออาจแปรผันตามอายุของอุปกรณ์ (Non-Constant Failure Rate) เช่นในช่วง “Wear-Out” ของอุปกรณ์เมื่ออายุเพิ่มขึ้น [22] ค่า failure Rate นี้มีผลโดยตรงต่อ MTBF เมื่อ λ คงที่ จะมีความสัมพันธ์ผกผันกับ MTBF (MTBF ≈ $1/\lambda$) [23] การแจกแจง Weibull (Weibull Distribution) เป็นแบบจำลองที่นิยมมากในงาน Reliability เพราะมีความยืดหยุ่นสูง สามารถจับลักษณะอัตราความล้มเหลวที่เพิ่มขึ้น ลดลง หรือคงที่ได้ ขึ้นอยู่กับพารามิเตอร์รูปร่าง (Shape) และมาตราส่วน (Scale) ตัวอย่างเช่น ถ้าค่า Shape > 1 แสดงถึงอัตราล้มเหลวที่เพิ่มตามอายุ (Wear-Out) ซึ่งเหมาะสมกับอุปกรณ์ที่เสื่อมตามเวลา ในขณะที่ Shape = 1 จะสอดคล้องกับอัตราล้มเหลวคงที่ (Exponential Case) การประเมินพารามิเตอร์เหล่านี้สามารถใช้วิธี Maximum Likelihood Estimation (MLE) หรือ Least Squares ตามสถิติการล้มเหลวจริง เพื่อใช้ในการวางแผนบำรุงรักษา [24]

ความสัมพันธ์ระหว่าง Reliability Engineering กับ Predictive Maintenance (PdM) มีความสำคัญอย่างยิ่ง: การรู้พารามิเตอร์เช่น MTBF, Failure rate และการแจกแจง Weibull ช่วยให้สามารถทำนาย Remaining Useful Life (RUL) ของอุปกรณ์ ทั้งยังช่วยออกแบบเกณฑ์เงื่อนไขและเวลาแทรกแซง (intervention) ที่เหมาะสม เพื่อบำรุงรักษาก่อนเกิดความล้มเหลวอย่างมีประสิทธิภาพ ในงานวิจัยสมัยใหม่ ยังมีการใช้โมเดลเชิงลึก (Deep Models) ที่ฝึกให้ฟิตพารามิเตอร์ของ Weibull Distribution ร่วมกับข้อมูลล้มเหลวจริง เพื่อพยากรณ์เหตุการณ์ล้มเหลวและอายุที่เหลือได้อย่างแม่นยำยิ่งขึ้น [25] การรวมกันของการวัด Reliability แบบดั้งเดิมและ Predictive Maintenance แบบข้อมูลเชิงสถิติช่วยเพิ่มความพร้อมใช้งานลด Downtime และลดต้นทุนบำรุงรักษาอย่างมีนัยสำคัญ

2.7 แนวคิดการบำรุงรักษาเชิงคาดการณ์

การบำรุงรักษาในระบบอุตสาหกรรมสามารถแบ่งออกเป็นหลายรูปแบบตามลักษณะของการใช้งานและเป้าหมายการควบคุมความเสี่ยง โดย Reactive, Corrective, Preventive, Condition-Based และ Predictive Maintenance เป็นแนวทางหลักที่ถูกใช้อย่างแพร่หลายดังตารางที่ 2.1 โดย PdM ถูกมองว่าเป็นกลยุทธ์ขั้นสูงที่สุด เพราะสามารถตรวจสอบสภาวะอุปกรณ์แบบเรียลไทม์และทำนายการเสื่อมสภาพก่อนเกิดความล้มเหลวจริงผ่านข้อมูลจากเซ็นเซอร์และเทคโนโลยีวิเคราะห์ข้อมูลขั้นสูง ช่วยให้สามารถกำหนดเวลาซ่อมบำรุงได้แม่นยำ ลด Downtime และลดต้นทุนโดยรวมมากกว่าการบำรุงรักษาแบบกำหนดรอบเวลา (Preventive) ที่มีโอกาสเปลี่ยนชิ้นส่วนเร็วกว่าความจำเป็น ในบริบทของ HDD โดยเฉพาะอุปกรณ์ทดสอบ เช่น GSA และ RWA แนวคิด PdM มีความสำคัญยิ่งเนื่องจากสภาวะความผิดปกติของระบบทดสอบส่งผลโดยตรงต่อคุณภาพการตรวจสอบฮาร์ดดิสก์และต้นทุนกระบวนการผลิต [26]

งานวิจัยด้านอุตสาหกรรมอิเล็กทรอนิกส์และอุปกรณ์ทดสอบได้แสดงให้เห็นว่า PdM ให้ผลลัพธ์ที่มีประสิทธิภาพสูงในระบบที่มีข้อมูลลำดับเวลา (Time-Series) เช่น อุณหภูมิ ค่าการสั่นสะเทือน กระแสไฟฟ้า และพารามิเตอร์เชิงสภาวะของอุปกรณ์ โดยเทคนิคที่นิยมใช้ ได้แก่ LSTM, GRU, Random Forest, Gradient Boosting รวมถึงโมเดลเชิงลึกยุคใหม่อย่าง Transformer ซึ่งสามารถจับความสัมพันธ์ระยะยาวในข้อมูลและทำนายสถานะสุขภาพของอุปกรณ์ได้แม่นยำกว่าแบบดั้งเดิม สำหรับ HDD เอง ข้อมูล S.M.A.R.T. เช่นอัตรา Read Error, อุณหภูมิ และค่าการหมุนของ Spindle ถูกนำมาใช้เพื่อระบุแนวโน้มเสื่อมสภาพของดิสก์ หากพบแนวโน้ม Error สูงขึ้นร่วมกับอุณหภูมิที่ผิดปกติ จะเป็นตัวบ่งชี้ที่ชัดเจนถึงความเสี่ยงต่อการเสียในอนาคต [27] ในการประยุกต์ใช้กับเครื่องทดสอบ HDD เช่น RWA หรือ GSA เซ็นเซอร์ภายใน เช่น Thermal Sensor, Vibration Sensor และ Current Sensor สามารถใช้เป็นแหล่งข้อมูลหลักสำหรับ PdM เช่นเดียวกัน

ความสามารถในการตีความผลลัพธ์ของโมเดล PdM ถือเป็นประเด็นสำคัญในงานวิจัยยุคปัจจุบัน โดย Explainable AI (XAI) ได้ถูกนำมาใช้เพื่อช่วยให้ผู้เชี่ยวชาญเข้าใจว่าตัวแปรใดเป็นปัจจัยหลักที่ทำให้โมเดลคาดการณ์ว่าอุปกรณ์กำลังเสื่อมสภาพ ซึ่งเป็นประโยชน์ต่อการวางแผนซ่อมบำรุงและการตรวจสอบย้อนกลับ เช่น หาก XAI ระบุว่าอุณหภูมิของโมดูลทดสอบเฉพาะส่วนมีผลต่อการเพิ่มขึ้นของ Failure Rate ผู้ดูแลสามารถตรวจสอบเซ็นเซอร์หรือระบบระบายความร้อนของโมดูลนั้นได้โดยตรง [28] ประโยชน์ของ PdM ไม่เพียงช่วยเพิ่มความน่าเชื่อถือและยืดอายุการใช้งานของ HDD หรืออุปกรณ์ทดสอบเท่านั้น แต่ยังเพิ่มประสิทธิภาพด้านการจัดกำลังคน การลด downtime และการสนับสนุนความยั่งยืนผ่านการลดการเปลี่ยนอุปกรณ์ที่ไม่จำเป็น ทั้งยังมีแนวโน้มที่จะผสานเข้ากับระบบคลาวด์ การวิเคราะห์แบบเรียลไทม์ และ Digital Twin ของระบบจัดเก็บข้อมูลเพื่อเพิ่มความแม่นยำของแบบจำลองในอนาคต [29]

ตารางที่ 2.1 ตารางเปรียบเทียบประเภทการบำรุงรักษาเครื่องจักร

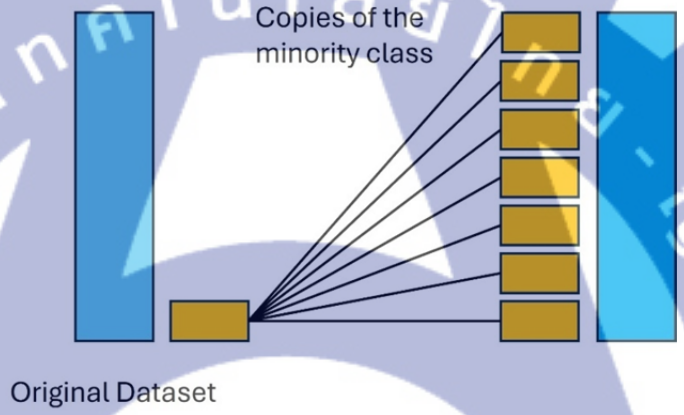
ประเภทการบำรุงรักษา	ลักษณะการทำงาน	ข้อดี	ข้อเสีย
Reactive Maintenance	ซ่อมเครื่องจักรเมื่อเสียโดยไม่วางแผน (Run-to-Failure)	ไม่ต้องวางแผนล่วงหน้า, ลงทุนน้อย	เสี่ยงต่อการหยุดการผลิตกะทันหัน, อาจเสียหายรุนแรง
Corrective Maintenance	ซ่อมเมื่อพบความผิดปกติหรือเครื่องหยุดทำงาน มีการตอบสนองบางส่วน	ลดผลกระทบต่อการผลิต, วางแผนซ่อมได้	ยังไม่สามารถป้องกัน Downtime ได้เต็มที่
Preventive Maintenance (PM)	ตรวจสอบหรือเปลี่ยนชิ้นส่วนตามรอบเวลา	ลดโอกาสเครื่องจักรเสียกะทันหัน, ป้องกัน Downtime	อาจเปลี่ยนอะไหล่ก่อนหมดอายุจริง
Predictive Maintenance (PdM)	ใช้เซ็นเซอร์, IoT, และ AI วิเคราะห์สภาพเครื่องจักร คาดการณ์อายุการใช้งานที่เหลือ (RUL)	ซ่อมเฉพาะเมื่อจำเป็น, ลด Downtime, คุ้มค่าต้นทุน	ต้องลงทุนระบบเซ็นเซอร์และ AI, ต้องมีข้อมูลเพียงพอ

2.8 การจัดการข้อมูลสำหรับการคาดการณ์ความล้มเหลว

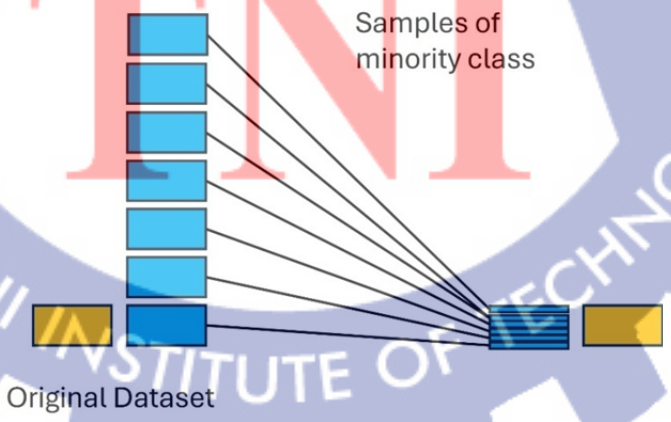
การจัดการข้อมูลสำหรับการคาดการณ์ความล้มเหลว (Predictive Maintenance: PdM) จำเป็นต้องพิจารณาโครงสร้างของข้อมูลเชิงเวลา (Time-Series Characteristics) อย่างรอบด้าน เนื่องจากข้อมูลเซนเซอร์มีความสัมพันธ์ตามลำดับเวลาและสะท้อนสภาวะการทำงานของเครื่องจักรและแนวโน้มความเสื่อมสภาพ การเตรียมข้อมูลจึงรวมถึงการจัดเรียงข้อมูลตามเวลา การสร้าง Time Window หรือ Sequence เพื่อจับพฤติกรรมก่อนเกิดความล้มเหลว และการสร้างคุณลักษณะเชิงเวลา เช่น ค่าเฉลี่ยเคลื่อนที่ ความแปรปรวน ความชันของสัญญาณ และตัวชี้วัดความเสถียร เพื่อเสริมความสามารถของโมเดลในการเรียนรู้รูปแบบความผิดปกติระยะเริ่มต้นได้อย่างแม่นยำ [30]

ขั้นตอนการทำความสะอาดข้อมูลและการจัดการข้อมูลขาดหายมีความสำคัญอย่างยิ่งในงาน PdM เนื่องจากข้อมูลจริงในโรงงานมักปนเปื้อนด้วย Noise ค่า Outlier หรือกรณีที่เซนเซอร์ส่งสัญญาณขาดช่วง การเตรียมข้อมูลจึงอาจต้องใช้วิธีกรองสัญญาณ เช่น Median Filter หรือ Kalman Filter การตรวจคัดกรองค่าผิดปกติด้วยกฎทางสถิติหรือข้อจำกัดทางกายภาพของระบบ ตลอดจนการเติมเต็มข้อมูลที่หายไปด้วยวิธี Interpolation หรือ Model-Based Imputation [31] ทั้งนี้ต้องคำนึงถึงพลวัตของอุปกรณ์เพื่อรักษารูปแบบความเสื่อมสภาพที่แท้จริง ไม่ให้ข้อมูลถูกปรับแก้จนผิดไปจากพฤติกรรมที่เกิดขึ้นในสภาพการทำงานจริง

ปัญหาคลาสไม่สมดุล (Class Imbalance) เป็นความท้าทายหลักในงาน PdM เนื่องจากข้อมูลสถานะปกติมีจำนวนมากกว่าเหตุการณ์ความล้มเหลวอย่างมาก วิธีการแก้ไขที่นิยม เช่น SMOTE (Synthetic Minority Oversampling Technique) ช่วยสังเคราะห์ข้อมูลคลาสส่วนน้อยเพื่อเพิ่มความสมดุลของชุดข้อมูล ส่งผลให้โมเดลตรวจจับสัญญาณผิดปกติได้ดีขึ้น อย่างไรก็ตาม ในบางกรณีสามารถใช้ Undersampling ร่วมด้วย โดยการลดจำนวนข้อมูลจากคลาสปกติลงอย่างมีหลักการ เช่น Random Undersampling หรือ Cluster-Based Undersampling เพื่อช่วยลดความลำเอียงของโมเดลในสถานการณ์ที่ข้อมูลปกติมีจำนวนล้นเกิน การผสาน Oversampling ดังรูปที่ 2.4 และ Undersampling ดังรูปที่ 2.5 หรือใช้วิธีการที่คำนึงถึงลักษณะของข้อมูลเชิงเวลา ช่วยให้ชุดข้อมูลมีโครงสร้างสมดุลและเหมาะสมต่อการเรียนรู้ของโมเดลคาดการณ์ความล้มเหลวมากยิ่งขึ้น



รูปที่ 2.4 การทำ Oversampling



รูปที่ 2.5 การทำ Undersampling

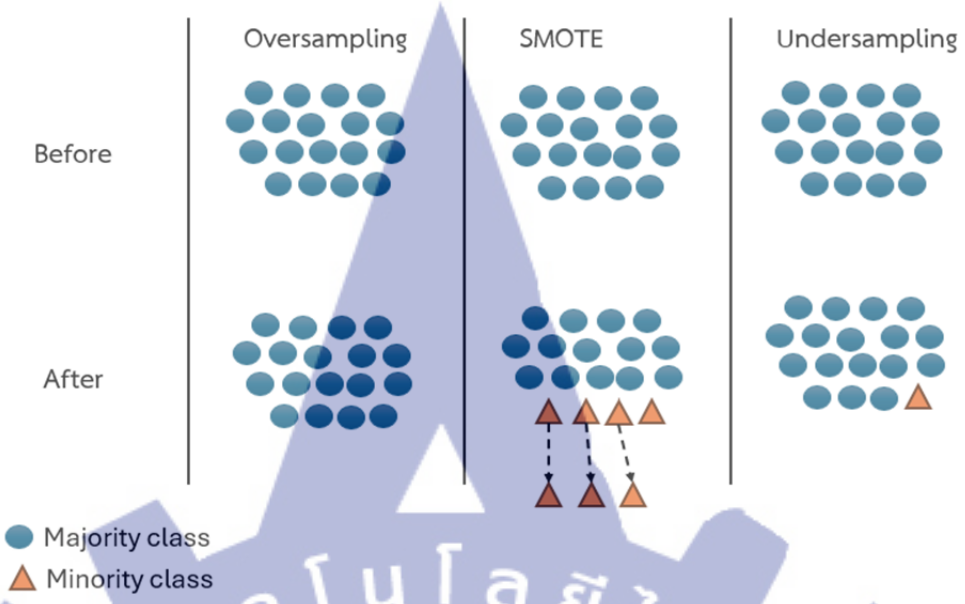
2.9 การจัดการข้อมูลไม่สมดุล

ข้อมูลไม่สมดุลเกิดขึ้นเมื่อจำนวนตัวอย่างของแต่ละคลาสในชุดข้อมูลแตกต่างกันอย่างมาก เช่น ในปัญหาการจำแนกประเภทข้อมูลทางการแพทย์ ข้อมูลผู้ป่วยที่ป่วยจริงอาจมีจำนวนน้อยกว่าผู้ป่วยปกติหลายเท่า หรือในระบบตรวจจับการฉ้อโกง ข้อมูลธุรกรรมที่เป็นการฉ้อโกงมีน้อยมากเมื่อเทียบกับธุรกรรมปกติ ปัญหานี้ทำให้โมเดลเรียนรู้ไม่สมดุล (Biased Learning) กล่าวคือ โมเดลจะ “เอนเอียง” ไปทำนายคลาสที่มีจำนวนมากกว่า ทำให้ประสิทธิภาพในการทำนายคลาสส่วนน้อยลดลงอย่างมาก แม้ว่าค่า Accuracy โดยรวมของโมเดลอาจดูสูง แต่ความสามารถในการตรวจจับเหตุการณ์สำคัญ (เช่น การฉ้อโกง หรือโรคที่เกิดขึ้น) จะต่ำ

หนึ่งในวิธีแก้ปัญหาที่นิยมใช้คือ SMOTE (Synthetic Minority Over-Sampling Technique) เทคนิคนี้เป็นการเพิ่มจำนวนตัวอย่างของคลาสส่วนน้อยโดยสร้างข้อมูลสังเคราะห์ขึ้นใหม่ กระบวนการทำงานของ SMOTE คือการเลือกตัวอย่างเดิมจากคลาสส่วนน้อยแล้วสุ่มเลือกจุดเพื่อนบ้าน (k-nearest Neighbors) จากนั้นสร้างตัวอย่างใหม่ระหว่างจุดข้อมูลเดิมและเพื่อนบ้าน (Interpolation) ทำให้ข้อมูลสังเคราะห์มีความหลากหลายและใกล้เคียงกับลักษณะของคลาสส่วนน้อยจริง ดังรูปที่ 2.6 เทคนิคนี้ช่วยปรับความสมดุลของชุดข้อมูล ทำให้โมเดลสามารถเรียนรู้คุณสมบัติของคลาสส่วนน้อยได้ดีขึ้น ลดความเอนเอียง และเพิ่มความแม่นยำในการทำนายโดยรวม [32]

นอกจาก SMOTE ยังมีวิธีอื่นที่ใช้จัดการข้อมูลไม่สมดุล เช่น Random Oversampling การคัดลอกตัวอย่างเดิมของคลาสส่วนน้อยซ้ำๆ Random Undersampling การลดจำนวนตัวอย่างของคลาสที่มากเกินไปเพื่อลดความไม่สมดุล, SMOTE-ENN หรือ SMOTE-Tomek ที่รวมการสร้างข้อมูลสังเคราะห์กับการลบตัวอย่างที่เป็น Noise หรือไม่เหมาะสมเพื่อให้ข้อมูลมีคุณภาพสูงขึ้น นอกจากนี้ยังมีแนวทาง Cost-Sensitive Learning ที่ปรับค่า Penalty ให้โมเดล “ใส่ใจ” กับคลาสส่วนน้อยมากขึ้น เช่น เพิ่มน้ำหนักของคลาสส่วนน้อยในการคำนวณ Loss Function [32]

การจัดการข้อมูลไม่สมดุลเป็นขั้นตอนสำคัญในงานด้าน Machine Learning โดยเฉพาะปัญหาที่คลาสส่วนน้อยมีความสำคัญสูง เช่น การตรวจจับโรค, การทำนายความล้มเหลวของอุปกรณ์, หรือการตรวจสอบการฉ้อโกง เพราะแม้โมเดลจะมี Accuracy สูง ถ้าไม่ปรับสมดุล ข้อมูลส่วนน้อยจะถูกมองข้าม ทำให้ระบบตัดสินใจผิดพลาด การใช้เทคนิคอย่าง SMOTE และวิธีอื่นๆ จึงช่วยให้โมเดลเรียนรู้ลักษณะของข้อมูลแต่ละคลาสอย่างสมดุล ส่งผลให้ผลลัพธ์การทำนายมีความแม่นยำและเชื่อถือได้มากขึ้น



รูปที่ 2.6 หลักการจัดการข้อมูลไม่สมดุล

2.10 ปัญญาประดิษฐ์และเทคนิคการเรียนรู้ของเครื่อง

ปัญญาประดิษฐ์ (Artificial Intelligence: AI) คือศาสตร์ที่มุ่งพัฒนาระบบอัจฉริยะ ซึ่งสามารถรับรู้ สังเคราะห์ข้อมูล วางแผน และตอบสนองอย่างอิสระ AI ไม่ได้จำกัดเพียงการเลียนแบบผลลัพธ์จากมนุษย์เท่านั้น แต่ยังเป็นเครื่องมือทางวิศวกรรมที่แก้ไขปัญหาซับซ้อนได้อย่างมีประสิทธิภาพ โดยเฉพาะเมื่อประยุกต์กับบริบทด้านการจัดการเครือข่าย พลังงาน หรือข้อมูลปริมาณมาก เทคนิคการเรียนรู้ของเครื่อง (Machine Learning: ML) ถือเป็นกลไกสำคัญที่ทำให้ AI สามารถเรียนรู้และปรับตัวจากข้อมูลจริงได้อย่างต่อเนื่อง โดยไม่ต้องพึ่งกฎที่เขียนไว้ล่วงหน้า Machine Learning ครอบคลุมหลายแนวทางที่เหมาะกับสถานการณ์ต่างกัน ได้แก่

(1) Supervised Learning ใช้ข้อมูลที่มีคำตอบกำกับเพื่อฝึกโมเดลให้ทำนายผลลัพธ์ เช่น การจำแนกหรือทำนายเชิงตัวเลข

(2) Unsupervised Learning ใช้ข้อมูลที่ไม่มีคำตอบเพื่อค้นหารูปแบบ เช่น การจัดกลุ่มหรือการลดมิติข้อมูล

(3) Reinforcement Learning เป็นการเรียนรู้ผ่านการทดลองในสภาพแวดล้อมโดยรับรางวัลหรือผลตอบแทนเป็นตัวชี้นำ

(4) Self Supervised Learning ซึ่งสร้างสัญญาณเรียนรู้จากข้อมูลเอง โดยไม่ต้องมีป้ายกำกับภายนอก

การวิจัยเชิงเครือข่ายในปัจจุบันได้วาง AI/ML เป็นกลไกสำคัญที่ช่วยให้ระบบเครือข่าย ฉลาดขึ้น เช่น การจัดการทรัพยากรแบบอัตโนมัติ การตัดสินใจแบบเรียลไทม์ และการวิเคราะห์พฤติกรรมของระบบอย่างต่อเนื่อง มีการนำ Deep Learning, CNN, RNN, GAN หรือ Transformer เข้ามาทดสอบประสิทธิภาพและสร้างความสามารถใหม่ๆ ให้กับระบบเครือข่ายยุคใหม่ [33]

AI/ML ไม่ได้จำกัดเพียงแค่การวิเคราะห์ข้อมูลทั่วไปเท่านั้น แต่ยังถูกบูรณาการลงในมาตรฐานเครือข่าย เช่น การใช้โมเดล ML เพื่อบีบอัดข้อมูลช่องสัญญาณ (CSI Compression) หรือตัดสินใจเรื่องการเข้าถึงช่องทางสื่อสารแบบอัตโนมัติ เป็นต้น แนวโน้มคือการเปลี่ยนจากการใช้ AI/ML บนคลาวด์ [34] ไปสู่การฝังตัวในโปรโตคอลเครือข่ายจริง (Native AI/ML) เพื่อเพิ่มประสิทธิภาพและความหน่วงต่ำในระบบจริง

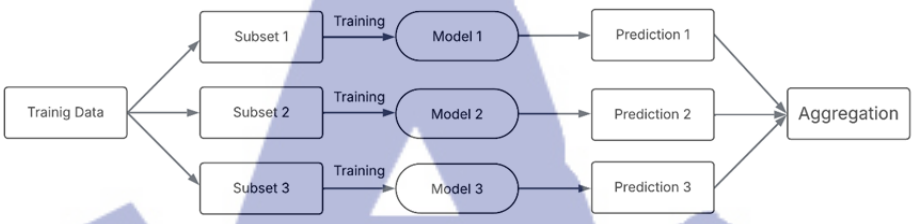
แม้ว่า AI/ML จะให้ผลลัพธ์ที่น่าประทับใจ แต่ก็ยังมีความท้าทายที่ต้องเผชิญ เช่นการขาดความโปร่งใสในกระบวนการตัดสินใจ (Explainability), ปัญหาอคติในข้อมูล (Bias), ความปลอดภัยข้อมูล และการไม่มีมาตรฐานกลางรองรับ รวมถึงข้อกำหนดด้านจริยธรรม แนวทางงานวิจัยในอนาคตจึงควรเน้นการพัฒนาโมเดลที่อธิบายได้ มีความยุติธรรม และได้รับการประเมินผ่านมาตรฐานสากลเพื่อสร้างความเชื่อมั่นทั้งทางเทคนิคและสังคม

2.11 Ensemble Learning

Ensemble Learning คือเทคนิคที่ผสานโมเดลทำนายหลายตัวเข้าด้วยกันเพื่อสร้างโมเดลที่แข็งแกร่งขึ้น (Strong Learner) จากการรวมผลของโมเดลที่มีความสามารถเฉพาะด้านหรือความแม่นยำระดับปานกลาง (Weak Learners) หลายตัว ซึ่งมักให้ประสิทธิภาพดีกว่าการใช้โมเดลเดี่ยวเพียงตัวเดียว เทคนิคนี้ช่วยลดปัญหา Overfitting และ Variance เนื่องจากความผิดพลาดของโมเดลแต่ละตัวจะถูกเฉลี่ยและชดเชยกัน ทำให้ผลลัพธ์มีความเสถียรและลดความลำเอียงที่อาจเกิดจากข้อมูลฝึก นอกจากนี้การรวมมุมมองและความเชี่ยวชาญที่แตกต่างกันของแต่ละโมเดลยังช่วยเพิ่มความแม่นยำและความน่าเชื่อถือของการทำนาย และโดยทั่วไป Ensemble Learning สามารถแบ่งออกเป็นสามกลุ่มหลักตามรูปแบบการรวมผลลัพธ์ของโมเดลย่อย ได้แก่ วิธีการเฉลี่ยผลแบบ Bagging วิธีการผสานลำดับความสัมพันธ์แบบ Boosting และวิธีการรวมโมเดลผ่านกฎการโหวตหรือการเรียนรู้เชิงผสม เช่น Stacking ซึ่งแต่ละแนวทางมีลักษณะเฉพาะที่เหมาะสมกับปัญหาและโครงสร้างข้อมูลที่แตกต่างกัน [35] Ensemble Learning สามารถแบ่งออกเป็น 3 ประเภทหลักตามวิธีการรวมโมเดลย่อยดังนี้

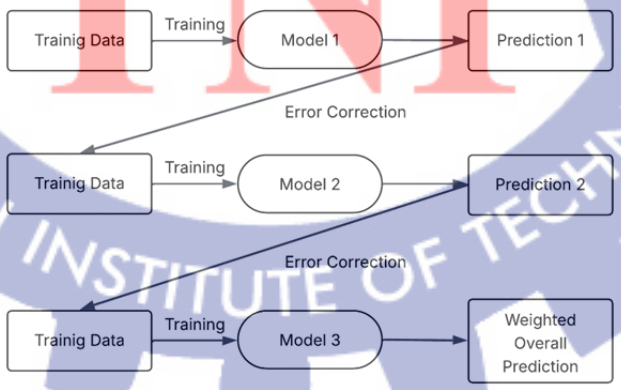
Bagging เป็นเทคนิคการทำ Ensemble Learning ที่มุ่งลดความแปรปรวนของโมเดลด้วยการฝึกโมเดลหลายตัวแบบขนานบนชุดข้อมูลย่อยที่ได้จากการสุ่มทดแทน (Bootstrap Sampling) จากชุดข้อมูลต้นฉบับ วิธีนี้ช่วยให้แต่ละโมเดลย่อยเรียนรู้รูปแบบที่แตกต่างกันและลดโอกาสเกิด Overfitting เมื่อรวมผลลัพธ์ด้วยการโหวตหรือการหาค่าเฉลี่ยจึงเกิดการคาดการณ์ที่เสถียรมากขึ้น โดยหนึ่งในโมเดล

ที่ได้รับความนิยมสูงสุดภายใต้แนวคิด Bagging คือ Random Forest ดังรูปที่ 2.7 ซึ่งสร้างต้นไม้ตัดสินใจจำนวนมากพร้อมการสุ่มตัวแปรในแต่ละขั้นตอน เพื่อเพิ่มความหลากหลายให้กับโมเดลและเพิ่มประสิทธิภาพในการจัดการข้อมูลที่ซับซ้อน เช่น งานวิเคราะห์เซนเซอร์และการคาดการณ์ความล้มเหลวของอุปกรณ์ในงานด้านอุตสาหกรรม [36]



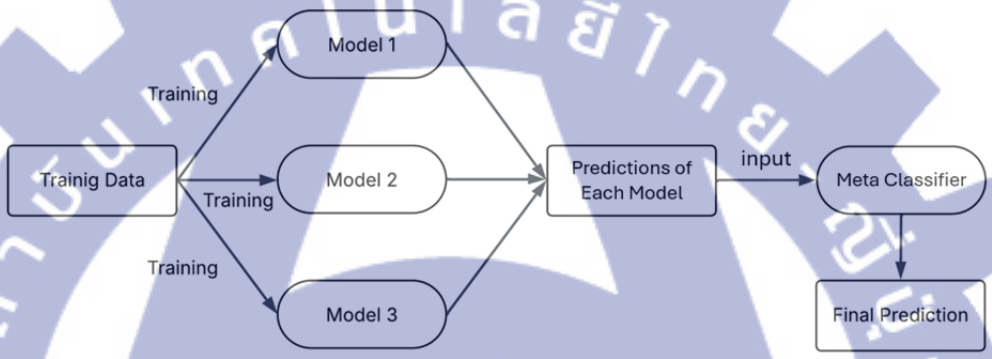
รูปที่ 2.7 เทคนิคการทำ Bagging

Boosting เป็นเทคนิคการทำ Ensemble Learning ที่มุ่งเพิ่มความแม่นยำของโมเดลโดยการสร้างโมเดลย่อยแบบลำดับต่อเนื่อง ซึ่งแต่ละโมเดลจะถูกปรับให้แก้ไขข้อผิดพลาดของโมเดลก่อนหน้า โดยให้ความสำคัญกับตัวอย่างที่ถูกทำนายผิด ทำให้สามารถลดความลำเอียงของโมเดล (Bias) และเพิ่มความสามารถในการจับรูปแบบที่ซับซ้อนในข้อมูลได้อย่างมีประสิทธิภาพ เทคนิคนี้ถูกใช้อย่างแพร่หลายในงานทำนายเชิงอุตสาหกรรมและการประมวลผลข้อมูลขนาดใหญ่ โดยมีตัวอย่างโมเดลที่เป็นที่รู้จัก ได้แก่ AdaBoost ซึ่งปรับน้ำหนักของตัวอย่างตามความยากง่ายในการทำนาย และ Gradient Boosting [37] ดังรูปที่ 2.8 ซึ่งใช้แนวคิดของการลดค่าความสูญเสียแบบขั้นบันได (Stage-Wise Optimization) เพื่อสร้างโมเดลที่แข็งแกร่งขึ้นทีละขั้น ทำให้ Boosting กลายเป็นเทคนิคสำคัญในงานคาดการณ์เชิงคุณภาพและเชิงปริมาณในหลายสาขา



รูปที่ 2.8 เทคนิคการทำ Boosting

Stacking เป็นเทคนิคการทำ Ensemble Learning ที่มีความซับซ้อนสูงกว่าแบบ Bagging และ Boosting โดยใช้โครงสร้างการเรียนรู้แบบหลายชั้นเพื่อผสานความสามารถของโมเดลหลากหลายประเภทเข้าด้วยกัน ขั้นแรก โมเดลพื้นฐาน (Base Learners) จะถูกฝึกด้วยชุดข้อมูลเดียวกันเพื่อสร้างมุมมองที่แตกต่างจากกัน เมื่อได้ผลการทำนายแล้ว ค่าทำนายเหล่านี้จะถูกนำไปเป็นอินพุตให้กับโมเดลขั้นสุดท้ายที่เรียกว่า Meta Learner ซึ่งมีหน้าที่เรียนรู้ความสัมพันธ์ระหว่างผลทำนายของโมเดลย่อยและสร้างผลลัพธ์สุดท้ายที่มีความแม่นยำมากขึ้น [38] ดังรูปที่ 2.9 เทคนิคนี้ได้รับการประยุกต์ใช้ในหลายงานด้านอุตสาหกรรมและการประมวลผลข้อมูล เช่น งานจำแนกประเภทสัญญาณ งานวิเคราะห์ข้อมูลเชิงความเชื่อมโยง และระบบคาดการณ์เชิงสถิติ เนื่องจากสามารถรวมข้อดีของโมเดลหลายรูปแบบเข้าด้วยกันและเพิ่มความสามารถในการทำนายได้อย่างมีนัยสำคัญ

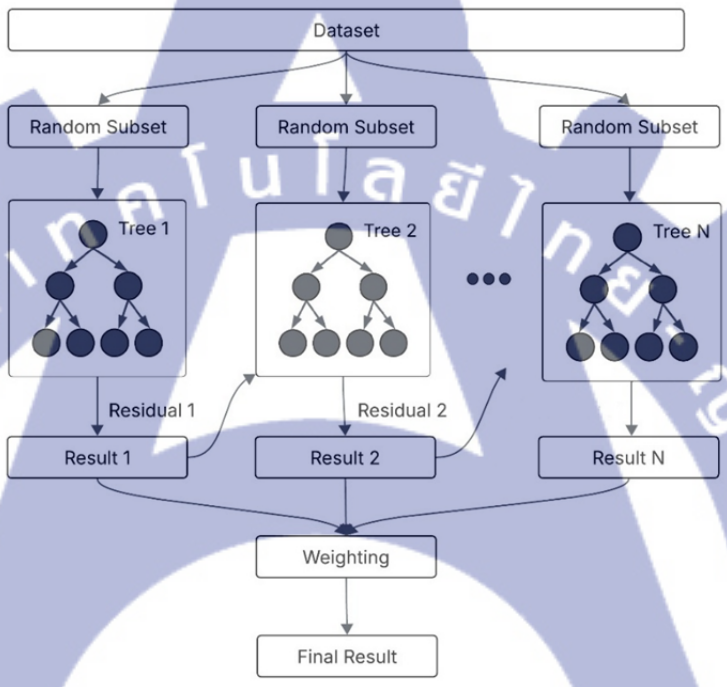


รูปที่ 2.9 เทคนิคการทำ Stacking

2.11.1 Extreme Gradient Boosting (XGBoost)

XGBoost เป็นอัลกอริทึมที่พัฒนาต่อยอดจากแนวคิดของ Gradient Boosting โดยใช้ต้นไม้ตัดสินใจเป็นตัวเรียนรู้หลักและมุ่งเน้นเพิ่มประสิทธิภาพทั้งด้านความเร็วและความแม่นยำ โมเดลจะสร้างต้นไม้หลายต้นแบบลำดับต่อเนื่อง โดยแต่ละต้นเรียนรู้จากความผิดพลาดหรือค่าคงเหลือ (Residual Error) ของต้นก่อนหน้า ทำให้การทำนายมีความแม่นยำมากขึ้นเรื่อยๆ เมื่อรอบการเรียนรู้เพิ่มขึ้น ผลการทำนายจากต้นไม้ทั้งหมดจะถูกนำมารวมกันผ่านการถ่วงน้ำหนัก เพื่อให้ต้นไม้ที่แม่นยำกว่ามีผลต่อผลลัพธ์สุดท้ายมากขึ้น [39] ดังรูปที่ 2.10 แนวคิดนี้ได้รับการนำไปใช้อย่างแพร่หลายในการวิเคราะห์ข้อมูลอุตสาหกรรมและงานคาดการณ์เชิงปริมาณซึ่งต้องการความรวดเร็วและความสามารถในการประมวลผลข้อมูลปริมาณมาก

ในเชิงคณิตศาสตร์ XGBoost ใช้ Objective Function ที่ประกอบด้วย Loss Function เพื่อประเมินความคลาดเคลื่อนของโมเดลและ Regularization Term เพื่อควบคุมความซับซ้อน ลดโอกาสเกิด Overfitting โดยจุดเด่นสำคัญคือการใช้ทั้ง Gradient และ Hessian ในการบ่งชี้ทิศทางและความโค้งของค่าความผิดพลาด ทำให้สามารถระบุจุดแบ่ง (Optimal Split) ที่เหมาะสมได้อย่างแม่นยำและมีประสิทธิภาพ วิธีนี้ช่วยให้การสร้างต้นไม้มีความรวดเร็วและเสถียร โดยเฉพาะเมื่อทำงานร่วมกับเทคนิคเสริม เช่น การประมวลผลแบบขนานและการจัดการค่าที่หายไปอัตโนมัติ



รูปที่ 2.10 หลักการทำงานของ Extreme Gradient Boosting

2.11.2 Light Gradient Boosting Machine (LightGBM)

LightGBM เป็นอัลกอริทึม Machine Learning ที่พัฒนาต่อยอดจาก Gradient Boosting โดยใช้ Decision Tree เป็นตัวเรียนรู้หลัก จุดเด่นของ LightGBM คือความรวดเร็วและการใช้หน่วยความจำน้อย ทำให้สามารถจัดการกับชุดข้อมูลขนาดใหญ่ได้อย่างมีประสิทธิภาพ หลักการคือการสร้างต้นไม้การตัดสินใจหลายๆ ต้น (Weak Learners) แบบต่อเนื่อง โดยแต่ละต้นจะเรียนรู้จากความผิดพลาด (Residual Error) ของต้นก่อนหน้า เพื่อค่อยๆ ปรับปรุงความแม่นยำ เมื่อสร้างครบแล้ว LightGBM จะนำผลการทำนายจากทุกต้นมารวมกันโดยใช้การถ่วงน้ำหนัก (Weighting) ทำให้ผลลัพธ์สุดท้ายมีความแม่นยำสูง [40] ดังรูปที่ 2.11

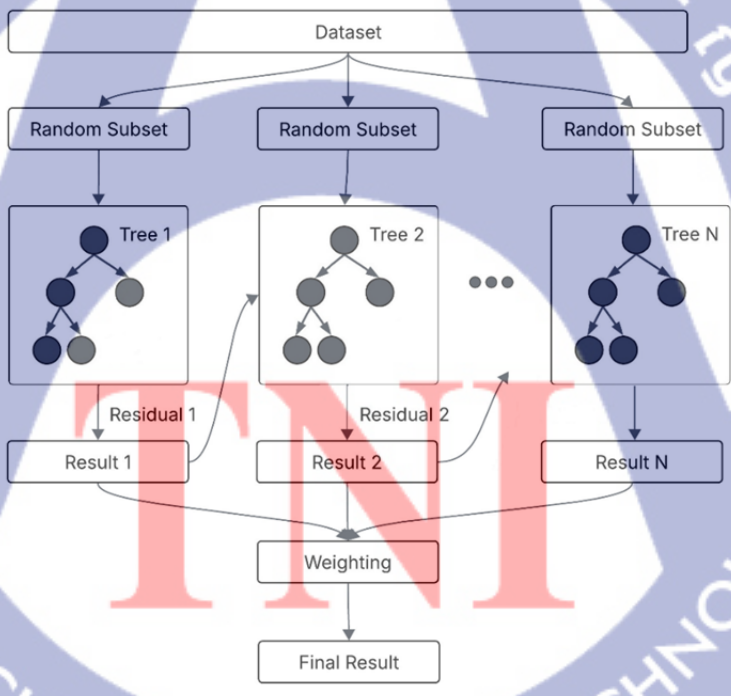
ในเชิงคณิตศาสตร์ LightGBM ใช้ Objective Function ที่ประกอบด้วย Loss Function สำหรับวัดความผิดพลาด และ Regularization Term สำหรับควบคุมความซับซ้อนของโมเดลเพื่อลดการ Overfitting สิ่งที่ทำให้ LightGBM แตกต่างคือการสร้างต้นไม้แบบ Leaf-Wise Growth โดยเลือกแตกกิ่งที่ใบ (Leaf) ที่ช่วยลดค่า Loss ได้มากที่สุด ไม่ใช่การขยายเต็มระดับ (Level-Wise) แบบ XGBoost จึงทำให้เรียนรู้ได้เร็วกว่าและแม่นยำกว่าในหลายกรณี อีกทั้งยังใช้เทคนิค Histogram-Based Algorithm, GOSS (Gradient-Based One-Side Sampling) และ EFB (Exclusive Feature Bundling) เพื่อเร่งความเร็วและลดหน่วยความจำ ทำให้ LightGBM เหมาะกับงานที่ต้องการทั้งความเร็วและประสิทธิภาพสูง [40]

ความต่างกับ XGBoost

(1) LightGBM ใช้ Leaf-wise Growth ส่วน XGBoost ใช้ Level-wise Growth

(2) LightGBM ใช้ Histogram-based Decision ลดเวลาและหน่วยความจำ

(3) LightGBM มักเร็วกว่าและเหมาะกับข้อมูลใหญ่ๆ มากกว่า

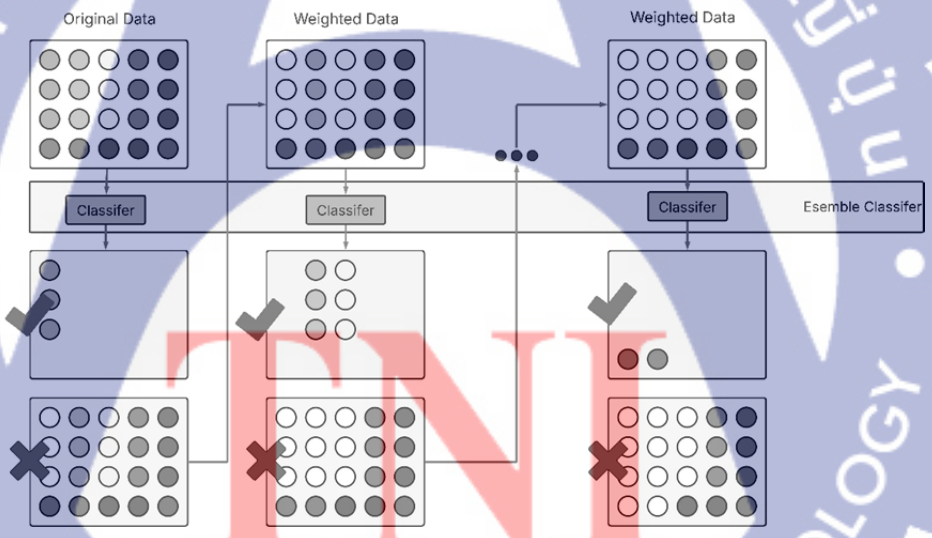


รูปที่ 2.11 หลักการทำงานของ Light Gradient Boosting Machine

2.11.3 Adaptive Boosting (AdaBoost)

AdaBoost เป็นอัลกอริทึม Machine Learning ในตระกูล Boosting ที่ใช้หลักการรวม Weak Learner หลายๆ ตัว (เช่น ต้นไม้การตัดสินใจตื้นๆ หรือ Decision Stump) เข้าด้วยกันเพื่อสร้างโมเดลที่แข็งแรงขึ้น (Strong Classifier) แนวคิดสำคัญคือการปรับน้ำหนักของข้อมูลในแต่ละรอบการเรียนรู้ โดยข้อมูลที่โมเดลก่อนหน้าทำนายผิดพลาดจะถูกให้น้ำหนักมากขึ้น ทำให้ Weak Learner รอบถัดไปโฟกัสไปที่การแก้ไขข้อผิดพลาดเหล่านั้น เมื่อฝึกครบหลายรอบแล้ว AdaBoost จะนำผลการทำนายจาก Weak Learner ทุกตัวมารวมกันโดยใช้การถ่วงน้ำหนัก (Weighting) เพื่อให้โมเดลที่มีความแม่นยำสูงกว่ามีอิทธิพลมากกว่าในผลลัพธ์สุดท้าย

ในเชิงคณิตศาสตร์ AdaBoost จะคำนวณค่าน้ำหนักของแต่ละ Weak Learner จากอัตราความผิดพลาด โดยค่าน้ำหนัก จากนั้นจะปรับปรุงน้ำหนักของข้อมูลใหม่ในแต่ละรอบเพื่อเพิ่มความสำคัญให้กับตัวอย่างที่ทำนายผิด ผลการทำนายสุดท้ายของ AdaBoost จะได้จากการรวมผลโหวตของ Weak Learner ทั้งหมดแบบถ่วงน้ำหนัก ส่งผลให้โมเดลมีความสามารถในการจัดหมวดหมู่ได้แม่นยำกว่าการใช้ Weak Learner เพียงตัวเดียวอย่างมีนัยสำคัญ [41] ดังรูปที่ 2.12



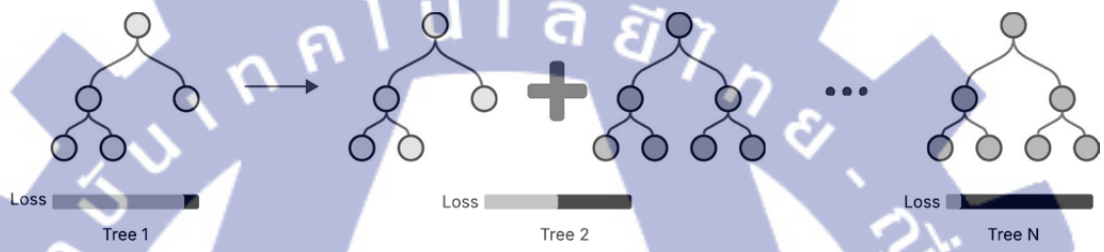
รูปที่ 2.12 หลักการทำงานของ Adaptive Boosting

2.11.4 Categorical Boosting (CatBoost)

CatBoost เป็นอัลกอริทึมในตระกูล Gradient Boosting ที่ใช้ต้นไม้ตัดสินใจหลายต้นต่อกันแบบลำดับ โดยแต่ละต้นทำหน้าที่แก้ไขความผิดพลาด (Loss) ที่ยังหลงเหลือจากต้นก่อนหน้า ทำให้ความแม่นยำของโมเดลเพิ่มขึ้นในทุกขั้นตอน จุดเด่นสำคัญคือเทคนิค Ordered Boosting ซึ่ง

ออกแบบมาเพื่อลดปัญหา Prediction Shift หรืออาการที่โมเดลทำนายได้ดีเฉพาะข้อมูลเทรนแต่ทำงานไม่ดีเมื่อเจอข้อมูลจริง การจัดลำดับข้อมูลลักษณะนี้ช่วยให้โมเดลมีความเสถียรและลดโอกาสเกิด overfitting ได้อย่างมีประสิทธิภาพ [42] ดังรูปที่ 2.13

นอกจากนี้ CatBoost ยังโดดเด่นในการจัดการข้อมูลเชิงหมวดหมู่ (Categorical Data) ได้โดยตรง เช่น สี ประเทศ เพศ หรือรหัสประเภทต่างๆ โดยไม่จำเป็นต้องทำ One-Hot Encoding เหมือนอัลกอริทึมทั่วไป โมเดลจะแปลงค่าหมวดหมู่ให้เป็นตัวเลขด้วยวิธีที่ลดความลำเอียงและลดความซับซ้อนของข้อมูลโดยอัตโนมัติ ส่งผลให้ CatBoost สามารถใช้งานได้ทั้งง่าย รวดเร็ว และให้ผลทำนายที่แม่นยำกว่า Gradient Boosting แบบดั้งเดิม โดยเฉพาะในงานที่มีข้อมูลหลากหลายและมีตัวแปรหมวดหมู่จำนวนมาก [42]

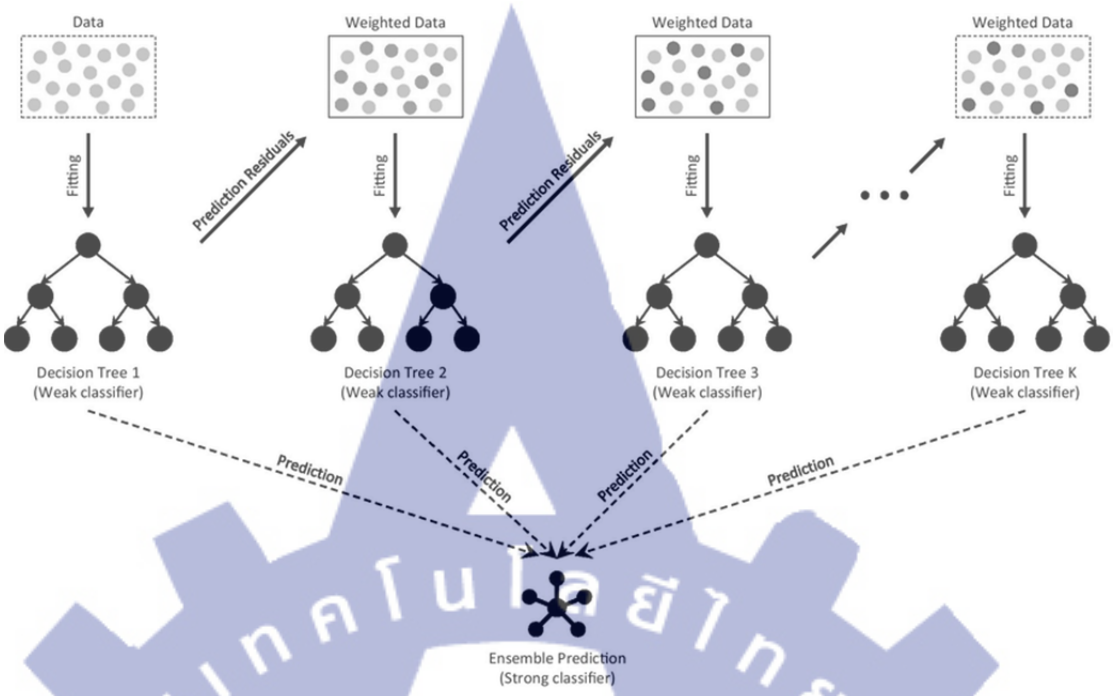


รูปที่ 2.13 หลักการทำงานของ Categorical Boosting

2.11.5 Gradient Boosting Tree (GBT)

GBT เป็นเทคนิค Ensemble Learning ที่นำแนวคิดการรวมโมเดลอ่อนแอ (Weak Learners) อย่างต้นไม้ตัดสินใจหลายต้นมาทำงานร่วมกันแบบลำดับ เพื่อสร้างโมเดลที่มีความแม่นยำสูงขึ้น โดยเริ่มจากการสร้างต้นไม้ต้นแรกซึ่งยังมีความคลาดเคลื่อนในการทำนาย จากนั้นโมเดลจะคำนวณค่า Residual ซึ่งสะท้อนถึงความผิดพลาดของต้นไม้ก่อนหน้า และนำข้อมูลนี้ไปใช้เป็นพื้นฐานในการสร้างต้นไม้ต้นถัดไป เพื่อให้ต้นไม้ใหม่มุ่งแก้ไขข้อผิดพลาดที่เกิดขึ้นในรอบก่อน กระบวนการนี้ทำซ้ำต่อเนื่อง ทำให้ต้นไม้แต่ละต้นไม่ได้ทำงานอย่างอิสระ แต่พัฒนาต่อจากกันเพื่อเพิ่มความแม่นยำอย่างค่อยเป็นค่อยไป

เมื่อสร้างต้นไม้ครบตามจำนวนรอบที่กำหนดแล้ว GBT จะนำผลการทำนายจากต้นไม้ทุกต้นมารวมกัน เช่น การเฉลี่ยหรือการรวมแบบถ่วงน้ำหนัก เพื่อให้ต้นไม้ที่แม่นยำกว่ามีอิทธิพลต่อผลลัพธ์สุดท้ายมากขึ้น ดังรูปที่ 2.14 วิธีการเรียนรู้แบบแก้ไขทีละขั้นนี้ช่วยให้โมเดลสามารถจับรูปแบบที่ซับซ้อนของข้อมูลได้ดีขึ้นและลดข้อจำกัดของการใช้ Decision Tree เพียงต้นเดียว จึงทำให้ GBT เป็นหนึ่งในอัลกอริทึมที่ได้รับความนิยมอย่างมากในงานทำนายเชิงอุตสาหกรรมและการวิเคราะห์ข้อมูลที่ต้องการความแม่นยำสูง

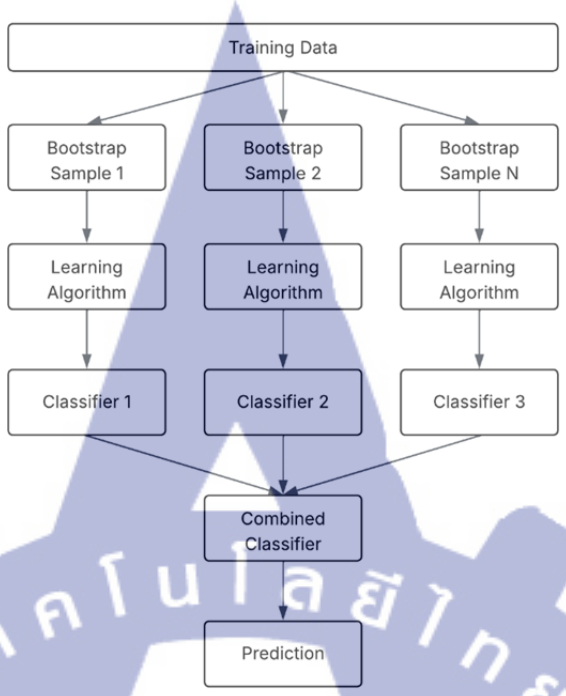


รูปที่ 2.14 หลักการทำงานของ Gradient Boosting Tree

2.11.6 Bootstrap Aggregating (Bagging)

Bagging เป็นเทคนิค Ensemble Learning ที่สร้างความหลากหลายของชุดข้อมูลโดยใช้วิธี Bootstrap Sampling ซึ่งเป็นการสุ่มตัวอย่างแบบมีการคืนกลับ (Sampling with Replacement) กล่าวคือ เมื่อข้อมูลตัวอย่างถูกเลือกแล้ว จะถูกนำกลับไปในชุดข้อมูลก่อนสุ่มครั้งถัดไป ทำให้ข้อมูลบางตัวอาจถูกเลือกซ้ำหลายครั้ง ขณะที่บางตัวอาจไม่ถูกเลือกเลย ขั้นตอนนี้ช่วยสร้างชุดข้อมูลย่อยหลายชุดจากข้อมูลฝึกชุดเดียว (Train Data) เพื่อนำไปฝึกโมเดลพื้นฐาน (Base Learners) เช่น Decision Tree ซึ่งแต่ละต้นจะได้รับข้อมูลต่างกันเล็กน้อย จึงเกิดความหลากหลายในการเรียนรู้

เมื่อฝึกโมเดลแต่ละตัวเสร็จแล้ว Bagging จะรวมผลการทำนายของโมเดลทั้งหมดเพื่อสร้างโมเดลสุดท้ายที่มีความเสถียรและแม่นยำมากขึ้น โดยใช้วิธีโหวตคะแนนเสียงข้างมาก (Majority Voting) สำหรับงาน Classification หรือการเฉลี่ยผลลัพธ์ (Averaging) สำหรับงาน Regression กระบวนการเรียนรู้แบบขนานและการรวมความเห็นของโมเดลหลายตัวนี้ช่วยลดปัญหา Overfitting ที่มักพบในโมเดลต้นไม้แบบเดี่ยว และเพิ่มความทนทานต่อสัญญาณรบกวนในข้อมูล ทำให้ Bagging เป็นเทคนิคที่ได้รับความนิยมอย่างมากในการพัฒนาระบบทำนายที่ต้องการความเสถียรสูง ดังรูปที่ 2.15



รูปที่ 2.15 หลักการทำงานของ Bootstrap Aggregating

2.12 ตัวชี้วัดประเมินประสิทธิภาพโมเดล

การประเมินประสิทธิภาพของโมเดลทำนายถือเป็นขั้นตอนสำคัญในการวิเคราะห์ผลลัพธ์ของอัลกอริทึม Machine Learning โดยเฉพาะในงานที่เกี่ยวกับการคาดการณ์ความล้มเหลว การจำแนกประเภท หรือการวิเคราะห์เหตุการณ์ผิดปกติ การเลือกตัวชี้วัดที่เหมาะสมช่วยให้สามารถประเมินความสามารถของโมเดลได้ครอบคลุมและเหมาะสมกับลักษณะข้อมูล โดยเฉพาะกรณีข้อมูลไม่สมดุล (Imbalanced Data) ซึ่งอาจทำให้ตัวชี้วัดบางประเภทเกิดความลวงได้ ตัวชี้วัดที่นิยมใช้มีดังนี้

Accuracy คือค่าความถูกต้องแสดงสัดส่วนของจำนวนข้อมูลที่โมเดลทำนายถูกต้องเมื่อเทียบกับจำนวนข้อมูลทั้งหมดดังสมการที่ 2.1 เป็นตัวชี้วัดพื้นฐานที่ใช้งานง่าย แต่มีข้อจำกัดสำคัญเมื่อข้อมูลไม่สมดุล เพราะแม้โมเดลทำนายเฉพาะคลาสใหญ่ก็ยังอาจได้ค่า Accuracy สูง จึงควรใช้ร่วมกับตัวชี้วัดอื่นเพื่อให้เห็นภาพที่ครบถ้วนยิ่งขึ้น

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.1)$$

Precision คือค่าความแม่นยำวัดสัดส่วนของข้อมูลที่โมเดลทำนายว่าเป็นคลาสบวก (Positive) แล้วถูกต้องจริงดังสมการที่ 2.2 เมื่อ Precision สูงหมายความว่าโมเดลมีความสามารถในการหลีกเลี่ยงการทำนายผิดประเภทแบบ False Positive เหมาะกับงานที่ต้องการลดการแจ้งเตือนผิด เช่น การคาดการณ์ความผิดปกติของอุปกรณ์ที่ไม่ต้องการแจ้งเตือนลวงจำนวนมาก

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2.2}$$

Recall (Sensitivity หรือ True Positive Rate) เป็นตัวชี้วัดนี้แสดงสัดส่วนของข้อมูล Positive จริงที่โมเดลสามารถตรวจพบได้ Recall สูงแสดงว่าโมเดลไม่ปล่อยให้เหตุการณ์สำคัญหลุดรอด (ลด FN) ดังสมการที่ 2.3 เหมาะกับงานด้านการตรวจจับความล้มเหลวที่ต้องการการแจ้งเตือนให้ครบถ้วนเพื่อลดความเสี่ยงในกระบวนการผลิต

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2.3}$$

F1-Score เป็นตัวชี้วัดที่ใช้ประเมินประสิทธิภาพโมเดลโดยพิจารณาทั้ง Precision และ Recall พร้อมกันผ่านการคำนวณค่าเฉลี่ยแบบฮาร์มอนิก (Harmonic Mean) จึงเป็นตัวชี้วัดที่สมดุลเมื่อจำเป็นต้องคำนึงถึงทั้งความสามารถในการหลีกเลี่ยงการแจ้งเตือนผิด (False Positive) และความสามารถในการตรวจจับเหตุการณ์จริง (True Positive) โดยเฉพาะอย่างยิ่งในกรณีที่ข้อมูลไม่สมดุล ซึ่งโมเดลอาจมีค่าที่ลวงจาก Precision หรือ Recall เพียงค่าเดียวได้ดังสมการที่ 2.4

$$\text{F1 - Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.4}$$

Specificity วัดสัดส่วนของข้อมูล Negative ที่โมเดลทำนายได้ถูกต้อง ตัวชี้วัดนี้ช่วยประเมินความสามารถของโมเดลในการหลีกเลี่ยงการแจ้งเตือนผิดพลาดจากกรณีที่ไม่มีเหตุการณ์เกิดขึ้น มีความสำคัญในระบบที่ต้องการความแม่นยำในคลาสปกติ เช่น การตรวจสอบสัญญาณที่ไม่ผิดรูปในอุปกรณ์ทดสอบ

Area Under the ROC Curve (AUC) เป็นตัวชี้วัดที่ประเมินประสิทธิภาพโมเดลโดยพิจารณาความสัมพันธ์ระหว่างค่า True Positive Rate (Sensitivity) และ False Positive Rate (1- Specificity) ค่าพื้นที่ใต้กราฟยิ่งสูงหมายถึงโมเดลสามารถแยกคลาสได้ดี เหมาะกับงานที่ข้อมูลไม่สมดุลหรือมีความซับซ้อนในการจำแนกดังสมการที่ 2.5

$$AUC = \frac{1}{2}\left(\left(\frac{TP}{TP + FN}\right)\right) + \left(\left(\frac{TN}{TN + FP}\right)\right) \quad (2.5)$$

Confusion Matrix เป็นตารางที่แสดงผลการทำนายโดยแบ่งออกเป็น 4 ส่วน ได้แก่ True Positive, True Negative, False Positive และ False Negative ซึ่งช่วยให้วิเคราะห์จุดอ่อนของโมเดลได้ชัดเจนดังรูปที่ 2.16 เช่น โมเดลทำนายคลาสใดผิดบ่อยหรือสับสนกับคลาสใดมากที่สุด จึงเป็นพื้นฐานสำคัญในการเลือกและปรับปรุงโมเดลให้เหมาะสมกับงานมากขึ้น

		Predicted: Positive	Predicted: Negative
Actual	Positive	TP	FN
	Negative	FP	TN

รูปที่ 2.16 ตาราง Confusion Matrix

2.13 สรุปงานวิจัยที่เกี่ยวข้อง

ในช่วงแรกของการบำรุงรักษาเชิงคาดการณ์ PdM มุ่งเน้นไปที่การประเมินอายุการใช้งานที่เหลือ (RUL) ของเครื่องจักรผ่านเทคนิคสถิติและโมเดล Regression แบบง่าย นักวิจัยทดลองใช้วิธี Linear Regression, Bayesian Regression และ Decision Forests บนชุดข้อมูลมาตรฐานอย่าง NASA C-MAPSS FD004 พบว่า Decision Forest Regression (DFR) สามารถคาดการณ์อายุการใช้งานได้ใกล้เคียงความจริง โดยมีค่า MAE ประมาณ 12.3 และ R^2 ประมาณ 0.96 การค้นพบนี้ทำให้ PdM เริ่มเป็นที่สนใจในโรงงานอุตสาหกรรมและการผลิต [43]

เมื่อเครื่องจักรและกระบวนการผลิตซับซ้อนขึ้น การใช้โมเดลเดี่ยวไม่สามารถตอบโจทย์ความแม่นยำได้ นักวิจัยจึงพัฒนา Hybrid Models เช่น การรวม Support Vector Regression (SVR) กับ Long Short-Term Memory (LSTM) เพื่อคาดการณ์ RUL แบบเสี่ยงต่ำ พร้อมปรับค่าพารามิเตอร์ด้วย

Grey Wolf Optimization (GWO-II) ผลลัพธ์จากชุดข้อมูล FD001 พบว่าค่า RMSE ของโมเดลผสมนี้อยู่ที่ 19.11 ซึ่งดีกว่าการใช้ SVR หรือ LSTM แยกกันอย่างชัดเจน ทำให้การวางแผนบำรุงรักษามีความแม่นยำและเชื่อถือได้มากขึ้น [44]

แนวทาง PdM ต่อมามุ่งเน้นไปที่การจำแนกประเภทความล้มเหลวของเครื่องจักร ระบบที่ใช้ Multilayer Perceptron (MLP) ฝึกด้วยข้อมูลแจ้งเตือนจาก ERP ของหุ่นยนต์บรรจุภัณฑ์ สามารถทำงานได้ แม่นยำถึง 91% และมี Mean Time to Failure (MTTF) ประมาณ 26 วัน จุดเด่นของวิธีนี้คือสามารถทำงานได้แม้ไม่มีข้อมูล IoT ทำให้โรงงานที่ยังไม่มีเซ็นเซอร์อัจฉริยะสามารถนำ PdM ไปใช้ได้ [45]

ในช่วงต่อมา การตรวจจับความผิดปกติ (Anomaly Detection) เริ่มเข้ามามีบทบาทเพื่อป้องกันปัญหาก่อนที่จะเกิดความเสียหายร้ายแรง ศูนย์ข้อมูล Tier-1 ใช้ Evolving Gaussian Fuzzy Classifier (eGFC) เพื่อตรวจจับความผิดพลาดจาก Log ของระบบ โดยได้ความแม่นยำถึง 92.48% และสามารถตีความผลลัพธ์ได้ง่าย ในอุตสาหกรรมทางทะเล มีการใช้ Ensemble ของ XGBoost, LSTM และ One-Class SVM เพื่อตรวจจับความเสียหายเริ่มต้นของแบริ่ง Crosshead แม้ว่าผล F1-score = 0.58 จะยังไม่สูงนัก แต่ก็ช่วยลดความเสี่ยงจากข้อมูล Time Series ที่มีปริมาณมากและสัญญาณรบกวนสูง

ปัจจุบัน PdM ครอบคลุมทั้งการคาดการณ์ RUL, การจำแนกความล้มเหลว และการตรวจจับความผิดปกติ ในโรงงานพิมพ์อุตสาหกรรม มีการนำ Machine Learning Classifiers เช่น Random Forest, XGBoost และ Logistic Regression มาประยุกต์ใช้กับข้อมูลกระบวนการผลิตและความล้มเหลวช่วงปี 2016-2018 โมเดล Random Forest สามารถทำนาย True Positives 28 รายได้อย่างถูกต้อง แม้จะมี False Positives 53 ราย และ False Negatives 24 รายก็ตาม การพัฒนานี้สะท้อนให้เห็นว่า PdM ในยุคปัจจุบันไม่เพียงช่วยลดเวลาหยุดทำงานและค่าใช้จ่าย แต่ยังช่วยให้การผลิตมีประสิทธิภาพสูงขึ้นและพร้อมรับมือกับความซับซ้อนของอุตสาหกรรมสมัยใหม่ [46]

ตารางที่ 2.2 Literature Review

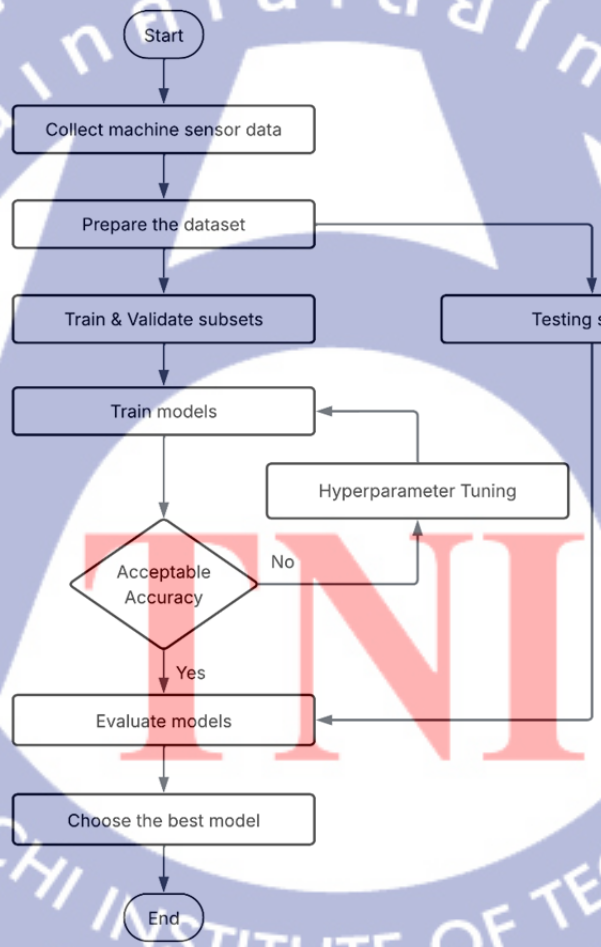
ลำดับที่	ปี	วัตถุประสงค์	เทคนิค	ชุดข้อมูลที่ใช้	ผลลัพธ์
1	[44], 2021	Risk-Averse RUL Estimation for PdM	Hybrid SVR + LSTM, GWO-II Optimization	NASA C-MAPSS FD001 (100 Engines, 21 Sensors	RMSE = 19.11 (vs. SVR = 24.61, LSTM = 20.16)
2	[45], 2020	Develop Predictive Maintenance System for Packaging Robots	MLP (ANN), MTTF Analysis, Poisson Dist, Back Propagation	Failure Notifications (157 Cases, 2017-2019)	Accuracy 91%, SD 7.25, MTTF ≈ 26.12 days

ตารางที่ 2.2 Literature Review (ต่อ)

ลำดับที่	ปี	วัตถุประสงค์	เทคนิค	ชุดข้อมูลที่ใช้	ผลลัพธ์
3	[19], 2020	Real-Time Anomaly Detection for PdM in Data Centers	Evolving Gaussian Fuzzy Classifier (eGFC) + Control Chart	Log Data from Tier-1 Bologna (1,436 Samples, 5 Features, 4 Classes)	Accuracy 92.48%, avg. 13.42 Fuzzy Rules
4	[47], 2020	Detect Early-Stage Crossheads Bearing Defects in Ships	XGBoost, LSTM, One-Class SVM, WPE, Ensemble Model	Sensor Data from 4 Ships, ~10GB, Time Series (1–10 min Sampling), ~30-day	F1-score 0.58, Precision 0.5, Recall 0.7
5	[46], 2019	Predict Machine Downtime Using ML	RF, XGBoost, LR, Feature Engineering, 3-fold CV	Printing press (2016–2018), Process & Downtime Data	RF deployed, TP=28, FP=53, FN=24
6	[43], 2019	Predict Remaining Useful Life (RUL) for PdM Using Feature Engineering	ML (LR, BLR, PR, NNR, BDTR, DFR)	C-MAPSS FD004 (NASA), 61,249 Records	Best: DFR (MAE ≈ 12.3, R^2 ≈ 0.96); NNR Worst in Most Setups

บทที่ 3
ขั้นตอนการดำเนินงาน

กรอบแนวคิดการวิจัยนี้เริ่มต้นจากการเก็บข้อมูลเซ็นเซอร์ที่ได้จากกระบวนการทดสอบ Head Gimbal Assembly (HGA) โดยอาศัยเครื่องทดสอบที่สามารถบันทึกค่าที่สะท้อนสภาวะจริงของอุปกรณ์ เช่น อุณหภูมิ (Temperature) และความเร็วรอบ (RPM) ข้อมูลที่ได้จะผ่านกระบวนการเตรียมชุดข้อมูล (Prepare the Dataset) เพื่อจัดรูปแบบ ทำความสะอาด และแบ่งออกเป็นชุดข้อมูลย่อยสำหรับฝึกตรวจสอบ (Train & Validate Subsets) และชุดข้อมูลทดสอบ (Testing Subset) ตามลำดับขั้นตอนในแผนภาพกระบวนการที่กำหนดดังรูปที่ 3.1



รูปที่ 3.1 ขั้นตอนการดำเนินงาน

เมื่อได้ชุดข้อมูลที่พร้อมใช้งาน โมเดลการเรียนรู้ของเครื่อง (Machine Learning Models) จะถูกฝึกด้วยชุดข้อมูลฝึกและตรวจสอบ เพื่อเรียนรู้รูปแบบความสัมพันธ์ระหว่างตัวแปรสภาพแวดล้อม ข้อมูลเซ็นเซอร์ และผลการทดสอบที่ระบุสถานะ Pass/Fail หากความแม่นยำที่ได้ยังไม่เป็นที่น่าพอใจ โมเดลจะถูกส่งเข้าสู่กระบวนการปรับค่าพารามิเตอร์ (Hyperparameter Tuning) ก่อนย้อนกลับมาฝึกใหม่ซ้ำจนกว่าจะได้ระดับความแม่นยำที่ยอมรับได้ (Acceptable Accuracy) ซึ่งสะท้อนเส้นทางการวนลูปใน Diagram

ในขั้นสุดท้าย โมเดลที่ผ่านเกณฑ์จะถูกประเมินด้วยชุดข้อมูลทดสอบ (Evaluate Models) เพื่อวัดความสามารถในการทำนายเหตุขัดข้องอย่างเป็นกลาง ก่อนนำไปเปรียบเทียบกับโมเดลอื่น รวมถึงโมเดลแบบ Ensemble เพื่อคัดเลือกโมเดลที่มีประสิทธิภาพสูงที่สุด (Choose the Best Model) กรอบแนวคิดทั้งหมดนี้จึงเชื่อมโยงตั้งแต่ข้อมูลสภาพแวดล้อม การประมวลผลเซ็นเซอร์ ไปสู่การสร้างโมเดลพยากรณ์ที่รองรับการบำรุงรักษาเชิงคาดการณ์ได้อย่างเป็นระบบ

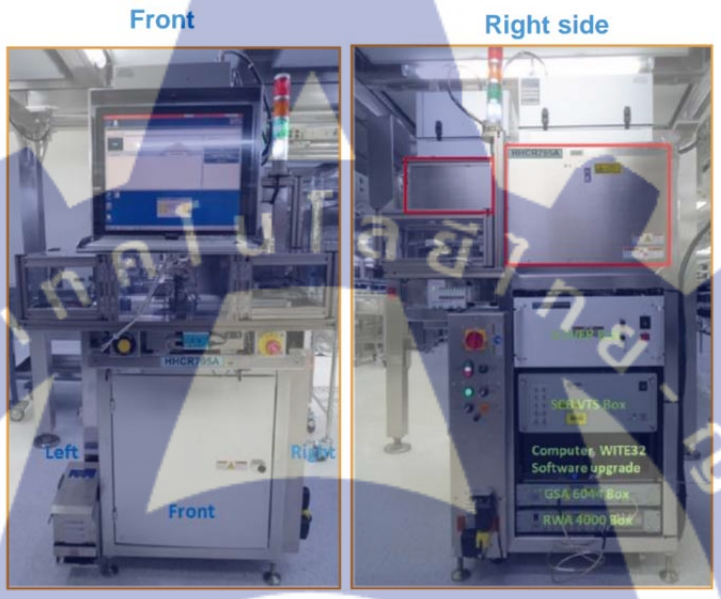
3.1 ข้อมูลที่ใช้ในการวิจัย

3.1.1 แหล่งที่มาและลักษณะข้อมูล

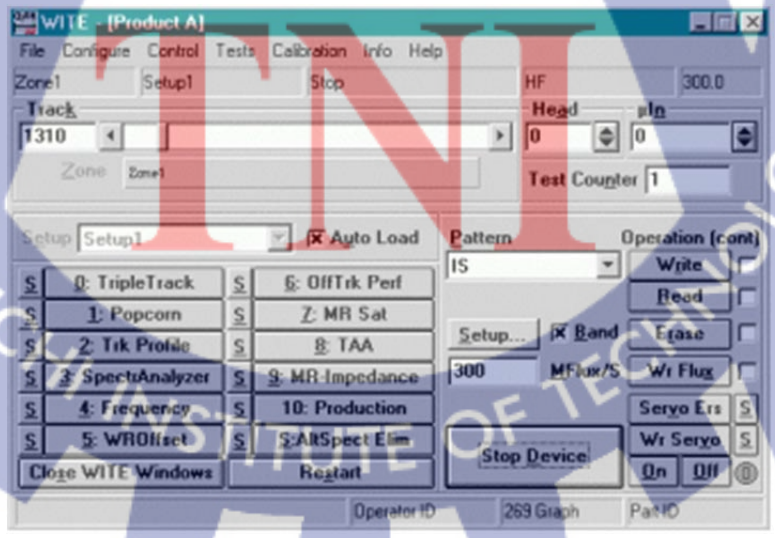
ข้อมูลหลักที่ใช้ในงานวิจัยนี้ได้มาจากระบบเครื่องทดสอบฮาร์ดดิสก์ ดังแสดงในรูปที่ 3.2 ซึ่งถูกควบคุมและติดตามการทำงานโดยโปรแกรม WITE32 Version 4.51B213 ดังแสดงในรูปที่ 3.4 โปรแกรมดังกล่าวทำหน้าที่บันทึกเหตุการณ์การทำงานของระบบ พารามิเตอร์ด้านสภาพแวดล้อม และสถานะของโมดูลและชิ้นส่วนต่างๆ ภายในเครื่องทดสอบในลักษณะ Real-Time โดยข้อมูลทั้งหมดจะถูกจัดเก็บในรูปแบบ Log File ดังแสดงในรูปที่ 3.3 การมีข้อมูล Log File จากระบบดังกล่าวทำให้สามารถตรวจสอบลำดับเหตุการณ์การทำงานของเครื่องทดสอบตั้งแต่การเริ่มต้นระบบ การตรวจสอบอุปกรณ์ การจ่ายไฟ การโหลดระบบทดสอบ ไปจนถึงการบันทึกค่าพารามิเตอร์ด้านสภาพแวดล้อม เช่น ค่าอุณหภูมิ และความเร็วรอบของพัดลมในแต่ละตำแหน่ง ข้อมูลเหล่านี้สามารถใช้ในการวิเคราะห์พฤติกรรมของระบบก่อนการเกิดความล้มเหลว (Failure) และนำไปสู่การสร้างแบบจำลองสำหรับการคาดการณ์การเสียของอุปกรณ์ได้อย่างแม่นยำมากยิ่งขึ้น

ข้อมูลด้านพารามิเตอร์สภาพแวดล้อมที่เครื่องทดสอบบันทึกไว้ใน Log ประกอบด้วยค่าอุณหภูมิของโมดูลต่างๆ เช่น Preamp, ADC, FPGA, PRML IC และอุณหภูมิบนบอร์ดย่อย (On-Board Sensor) จากตัวอย่าง Log พบว่าแต่ละโมดูลจะถูกบันทึกเป็นค่าแบบตัวเลขจริงพร้อมสถานะ เช่น “GSA6044AXI:0, Preamp #0:42.4844°C-OK” ซึ่งสะท้อนสภาวะการทำงาน ณ ขณะทดสอบจริง นอกจากนี้ ยังมีข้อมูลความเร็วพัดลม (Fan RPM) รวมถึง Box Fan หลายตำแหน่งที่อยู่ในช่วงประมาณ 0-15000 RPM ซึ่งเป็นตัวแปรสำคัญที่สามารถบ่งชี้สุขภาพของระบบระบายความร้อน หากมีความผิดปกติ เช่น ความเร็วตกลงกว่าปกติ มักสัมพันธ์กับการเกิด Fail ของอุปกรณ์ทดสอบในเวลาต่อมา

นอกเหนือจากข้อมูลสภาพแวดล้อมแล้ว Log File ยังบันทึกพารามิเตอร์ของกระบวนการ (Process Parameters) และผลการทดสอบ (Output/Result) เช่น OK/Alert!, การโหลดอุปกรณ์ (Device VTS Loaded) และการใช้หน่วยความจำของระบบ (Memory Usage) เช่น “Dynamic CRT heap: 9,920 blocks” ซึ่งช่วยสะท้อนภาระการทำงานของเครื่องทดสอบในแต่ละรอบ ทำให้สามารถระบุ Feature ที่สัมพันธ์กับความล้มเหลวของอุปกรณ์ได้อย่างชัดเจน และนำไปใช้สร้างแบบจำลอง Predictive Maintenance ได้อย่างมีประสิทธิภาพ



รูปที่ 3.2 เครื่องวัดคุณภาพสัญญาณของเครื่องอ่าน-เขียน



รูปที่ 3.3 โปรแกรม WITE32 Version 4.51B213

```
====== Log file for C:\WITE32_451B213\Wite.exe ===========
====== Started at: 11/14/2024,5:10:31
================================================


WITE32 ver. 4.51B213 (617853) 04/13/20;22:54

Servo shared


PCI power-up starts...
PCI power-up successfully finished.
PCI GSA presense checking starts...
PCI GSA presense checking successfully finished.
PCI system information getting starts...
PCI system information getting successfully finished.

Process DE0h, pass 1 memory usage:
Dynamic CRT heap:    9920 blocks,    2,232,129 bytes
Dynamic heaps:       4525 blocks,      565,253 bytes (excluding CRT)
Shared memory:        757 blocks,    3,196,416 bytes

05:13:42 GSA6044AXI:0 PCIe-HT bridge: Spare access count = 129
System monitors dump:
GSA6044AXI:0, Preamp #0: 42.4844 °C - OK
GSA6044AXI:0, Preamp #1: 42.6154 °C - OK
GSA6044AXI:0, Preamp #2: 48.8167 °C - OK
GSA6044AXI:0, Preamp #3: 48.8167 °C - OK
GSA6044AXI:0, ADC PA #0: 63.8125 °C - OK
GSA6044AXI:0, ADC ADC #0: 71.4375 °C - OK
GSA6044AXI:0, ADC PA #1: 63.8125 °C - OK
GSA6044AXI:0, ADC ADC #1: 72.5625 °C - OK
GSA6044AXI:0, ADC #0: 42.3 °C - OK
GSA6044AXI:0, ADC #1: 44.5 °C - OK
GSA6044AXI:0, AP FPGA #0: 42.3125 °C - OK
GSA6044AXI:0, AP FPGA #1: 45.75 °C - OK
GSA6044AXI:0, AP FPGA #2: 38.875 °C - OK
GSA6044AXI:0, AP FPGA #3: 44.4375 °C - OK
GSA6044AXI:0, HT FPGA: 42.625 °C - OK
GSA6044AXI:0, ADC fan #0: 1802.88 RPM - OK
GSA6044AXI:0, ADC fan #1: 2253.61 RPM - OK
GSA6044AXI:0, AP FPGA fan #0: 6008.15 RPM - OK
GSA6044AXI:0, AP FPGA fan #1: 6008.15 RPM - OK
GSA6044AXI:0, AP FPGA fan #2: 5435.94 RPM - OK
GSA6044AXI:0, AP FPGA fan #3: 5578.99 RPM - OK
GSA6044AXI:0, Box fan #0: 5056.18 RPM - OK
GSA6044AXI:0, Box fan #1: 4745.17 RPM - OK
GSA6044AXI:0, Box fan #2: 4720.28 RPM - OK
GSA6044AXI:0, Box fan #3: 4671.28 RPM - OK
GSA6044AXI:0, Box fan #4: 4856.12 RPM - OK
GSA6044AXI:0, Box fan #5: 4981.55 RPM - OK
GSA6044AXI:0, Box fan #6: 4745.17 RPM - OK
GSA6044AXI:0, Box fan #7: 4695.65 RPM - OK
GSA6044AXI:0, Box fan #8: 5009.28 RPM - OK
GSA6044AXI:0, Box fan #9: 4761.9 RPM - OK
MB4000:0, fan #1: 5300 RPM - OK
MB4000:0, fan #2: 5300 RPM - OK
MB4000:0, fan #3: 5100 RPM - OK
MB4000:0, fan #4: 5200 RPM - OK
MB4000:0, fan #5: 5200 RPM - OK
PG8G:0, PG FPGA: 51.9 °C - OK
PG8G:0, PG DAC: 49.3 °C - OK
CAI7000:0, On-board: 38.8125 °C - OK
CAI7000:0, FPGA: 41.625 °C - OK
CAI7000:0, PRML IC: 39.125 °C - OK
MFM4000:0, On-board: 51.5 °C - OK
Device VTS is loaded
```

รูปที่ 3.4 ตัวอย่างข้อมูลที่บันทึกอยู่ใน Log File

3.1.2 ลักษณะของชุดข้อมูล

จากการพิจารณาโครงสร้างและเนื้อหาของ Log File ที่ได้จากโปรแกรม WITE32 ดังตัวอย่างที่แสดงในรูปที่ 3.4 พบว่าชุดข้อมูลที่ใช้ในการวิจัยนี้มีลักษณะสำคัญดังต่อไปนี้

1) ชุดข้อมูลอยู่ในรูปแบบ Time-Series Log Data ข้อมูลถูกบันทึกตามลำดับเวลา โดยมี Timestamp กำกับในแต่ละเหตุการณ์ ทำให้สามารถติดตามลำดับการทำงานของเครื่องทดสอบตั้งแต่การเริ่มต้นระบบ การดำเนินกระบวนการทดสอบ ไปจนถึงการสิ้นสุดการทำงาน รวมถึงการวิเคราะห์พฤติกรรมของระบบก่อนการเกิดความล้มเหลวได้อย่างเป็นระบบ

2) ชุดข้อมูลมีลักษณะเป็น ข้อมูลผสมระหว่างการทำงานแบบต่อเนื่องและไม่ต่อเนื่อง (Hybrid Continuous–Discrete Operation) พารามิเตอร์ด้านสภาพแวดล้อม เช่น ค่าอุณหภูมิของโมดูลต่างๆ และความเร็วรอบของพัดลม ถูกบันทึกในลักษณะข้อมูลเชิงตัวเลขต่อเนื่อง (Continuous Data) ซึ่งสะท้อนสภาวะการทำงานของระบบแบบต่อเนื่อง ขณะที่ข้อมูลเหตุการณ์ เช่น การเริ่มต้นระบบ การตรวจสอบอุปกรณ์ การโหลดระบบทดสอบ และการรายงานสถานะการทำงาน ถูกบันทึกในรูปของเหตุการณ์ไม่ต่อเนื่อง (Discrete Events) ซึ่งเกิดขึ้นเฉพาะในช่วงเวลาที่ระบบมีการเปลี่ยนสถานะ

3) ชุดข้อมูลประกอบด้วย ข้อมูลเชิงข้อความและข้อมูลเชิงตัวเลขผสมกัน (Mixed-Type Data) ข้อมูลเชิงข้อความปรากฏในรูปของเหตุการณ์และข้อความสถานะการทำงาน เช่น การเริ่มต้นระบบหรือการสิ้นสุดกระบวนการ ขณะที่ข้อมูลเชิงตัวเลขประกอบด้วยค่าพารามิเตอร์ต่างๆ เช่น อุณหภูมิ ความเร็วรอบพัดลม และการใช้หน่วยความจำของระบบ ซึ่งจำเป็นต้องมีการแยกประเภทและแปลงข้อมูลให้อยู่ในรูปแบบเชิงโครงสร้างก่อนการวิเคราะห์

4) ข้อมูลด้านสภาพแวดล้อมเป็น ข้อมูลเชิงตัวเลขต่อเนื่อง (Continuous Numerical Data) จาก Log File พบว่ามีการบันทึกค่าอุณหภูมิของโมดูลต่างๆ เช่น Preamp, ADC, FPGA และ PRML IC รวมถึงค่าอุณหภูมิจากเซนเซอร์บนบอร์ดย่อย (On-Board Sensor) โดยค่าที่ได้เป็นเลขทศนิยมและมีหน่วยเป็นองศาเซลเซียส ซึ่งสะท้อนสภาวะการทำงานจริงของอุปกรณ์ในระหว่างการทดสอบ

5) ข้อมูลความเร็วรอบของพัดลมเป็น ตัวแปรเชิงกลไกที่มีช่วงค่ากว้างและมีความไวต่อความผิดปกติ ค่าความเร็วพัดลมจากหลายตำแหน่ง เช่น ADC Fan, AP FPGA Fan และ Box Fan มีช่วงค่าตั้งแต่หลักพันไปจนถึงหลายพันรอบต่อนาที ข้อมูลในลักษณะนี้สามารถใช้เป็นตัวบ่งชี้สุขภาพของระบบระบายความร้อน และมีความสัมพันธ์กับการเกิดความล้มเหลวของเครื่องทดสอบในภายหลัง

6) ชุดข้อมูลมี สถานะการทำงาน (Operational Status) กำกับร่วมกับค่าพารามิเตอร์ ค่าพารามิเตอร์แต่ละรายการมักถูกบันทึกร่วมกับสถานะ เช่น “OK” ซึ่งช่วยให้สามารถแยกแยะสภาวะการทำงานปกติและสภาวะผิดปกติของระบบได้ อย่างไรก็ตาม ข้อมูลที่แสดงถึงสถานะความล้มเหลวมีสัดส่วนค่อนข้างน้อยเมื่อเทียบกับข้อมูลสถานะปกติ ส่งผลให้ชุดข้อมูลมีลักษณะเป็น ข้อมูลไม่สมดุล (Imbalanced Dataset)

7) การบันทึกข้อมูลมีลักษณะ ไม่สม่ำเสมอของความถี่ (Irregular Sampling) การบันทึกข้อมูลขึ้นอยู่กับเหตุการณ์และการเรียกอ่านค่าจากระบบ ทำให้ช่วงเวลาระหว่างการบันทึกข้อมูลไม่คงที่ ซึ่งจำเป็นต้องมีการจัดรูปแบบ Timestamp และการปรับโครงสร้างข้อมูลก่อนนำไปใช้ในแบบจำลองเชิงพยากรณ์

8) ชุดข้อมูลมีโอกาสเกิด ค่าผิดปกติ (Outliers) และข้อมูลสูญหาย (Missing Values) เนื่องจากข้อมูลได้มาจากการทำงานจริงของเครื่องทดสอบ จึงอาจเกิดการเปลี่ยนแปลงค่าพารามิเตอร์อย่างฉับพลัน หรือการขาดหายของข้อมูลบางช่วงเวลา อันเนื่องมาจากสัญญาณรบกวน การรีสตาร์ตระบบ หรือการหยุดทำงานของโมดูลบางส่วน

ลักษณะของชุดข้อมูลดังกล่าวสะท้อนถึงลักษณะการทำงานของระบบอุตสาหกรรมจริงที่มีทั้งพฤติกรรมแบบต่อเนื่องและไม่ต่อเนื่อง และเป็นปัจจัยสำคัญในการกำหนดแนวทางการเตรียมข้อมูล การสร้างคุณลักษณะ และการพัฒนาแบบจำลองสำหรับการบำรุงรักษาเชิงพยากรณ์ ซึ่งจะอธิบายรายละเอียดในหัวข้อ 3.2 การเตรียมข้อมูล

3.2 การเตรียมข้อมูล

การเตรียมข้อมูลเป็นขั้นตอนสำคัญอย่างยิ่งสำหรับงานวิเคราะห์ข้อมูลและการสร้างแบบจำลองการเรียนรู้ของเครื่อง โดยเฉพาะอย่างยิ่งในสภาพแวดล้อมโรงงานอุตสาหกรรมที่ข้อมูลมักมีความไม่สมบูรณ์ มีสัญญาณรบกวน (Noise) และความคลาดเคลื่อนจากกระบวนการผลิตจริง หากไม่ได้มีการปรับปรุงคุณภาพข้อมูลอย่างเป็นระบบ อาจส่งผลให้โมเดลทำนายได้ไม่แม่นยำหรือเกิดการลำเอียง (Bias) กระบวนการเตรียมข้อมูลในงานวิจัยนี้จึงถูกออกแบบให้ครอบคลุมตั้งแต่การทำความสะอาดข้อมูล การจัดรูปแบบเวลา การจัดการค่าที่หายไป การตรวจจับและแก้ไขค่าผิดปกติ การสร้างคุณลักษณะใหม่ ตลอดจนการปรับสมดุลของชุดข้อมูล เพื่อให้ข้อมูลอยู่ในสภาพที่พร้อมที่สุดสำหรับการฝึกโมเดล

3.2.1 การทำความสะอาดข้อมูล

ข้อมูลจากเครื่องจักรในระบบทดสอบการอ่าน-เขียนของฮาร์ดดิสก์มักมีความคลาดเคลื่อนและความไม่สมบูรณ์จากปัจจัยต่างๆ เช่น ความผิดพลาดของสัญญาณเซนเซอร์ การรีเซ็ตอุปกรณ์ระหว่างการทดสอบ หรือข้อจำกัดของระบบบันทึกข้อมูลแบบต่อเนื่อง โดยเฉพาะข้อมูลจากเซนเซอร์อุณหภูมิและเซนเซอร์วัดความเร็วรอบการหมุนของพัดลม (Fan Speed หรือ RPM Sensors) ดังตารางที่ 3.1 ซึ่งใช้ติดตามสภาวะความร้อนและประสิทธิภาพของระบบระบายความร้อน ทั้งนี้ข้อมูลจากเซนเซอร์ดังกล่าวอาจปรากฏค่าที่ผิดรูปหรือไม่สมเหตุสมผลทางกายภาพ เช่น ค่าอุณหภูมิหรือความเร็วรอบที่เกินขอบเขตการทำงานจริง ค่าเป็นศูนย์ในช่วงเวลาที่อุปกรณ์ควรทำงานตามปกติ รวมถึงค่าติดลบซึ่งไม่สามารถเกิดขึ้นได้ในเชิงกายภาพ หากไม่ดำเนินการตรวจสอบและจัดการข้อมูลผิดปกติเหล่านี้อย่างเหมาะสม

อาจส่งผลกระทบต่อความถูกต้องของการวิเคราะห์ข้อมูลและประสิทธิภาพของแบบจำลองในขั้นตอนถัดไป

ตารางที่ 3.1 คำอธิบายการกำหนดค่าของเซนเซอร์ [48]

No.	Module	Component	Spec	Trigger	Function
1	CAI7000	On-board	0-120°C (±1-2°C)	70°C	เซนเซอร์วัดอุณหภูมิบริเวณ PCB บนโมดูล CAI7000 เพื่อตรวจสอบความร้อนสะสมและเสถียรภาพของวงจร
2	CAI7000	FPGA	0-120°C (±1-2°C)	70°C	ตรวจวัดอุณหภูมิ FPGA ซึ่งเป็นชิปประมวลผลความเร็วสูงเพื่อป้องกันความเสียหายจากความร้อน
3	CAI7000	PRML IC	0-120°C (±1-2°C)	70°C	ตรวจสอบอุณหภูมิชิประบบอ่านสัญญาณ PRML ซึ่งเป็นส่วนสำคัญของวงจรอ่านข้อมูล HDD
4	GSA6044AXI	Preamp #0	0-120°C (±1-2°C)	70°C	วัดอุณหภูมิภาคขยายสัญญาณ Preamp #0 ที่มีผลต่อความไวของสัญญาณก่อนผ่านเข้า ADC
5	GSA6044AXI	Preamp #1	0-120°C (±1-2°C)	70°C	ตรวจสอบระดับความร้อนใน Preamp #1 ป้องกัน thermal distortion
6	GSA6044AXI	Preamp #2	0-120°C (±1-2°C)	70°C	วัดความร้อนของ Preamp #2 ซึ่งมีผลโดยตรงต่อคุณภาพสัญญาณ
7	GSA6044AXI	Preamp #3	0-120°C (±1-2°C)	70°C	ตรวจสอบความร้อนเพื่อลด noise และป้องกันสัญญาณผิดเพี้ยน
8	GSA6044AXI	ADC #0	0-120°C (±1-2°C)	70°C	วัดอุณหภูมิ ADC ช่องที่ 0 เพื่อประเมินเสถียรภาพของการแปลงสัญญาณ
9	GSA6044AXI	ADC #1	0-120°C (±1-2°C)	70°C	วัดความร้อน ADC ช่องที่ 1 ลดผลกระทบต่อค่าที่แปลงจากอนาล็อกเป็นดิจิทัล
10	GSA6044AXI	ADC Section #0	0-120°C (±1-2°C)	85°C	ตรวจวัดส่วน ADC เสริมเพื่อควบคุมเสถียรภาพของระบบแปลงสัญญาณ
11	GSA6044AXI	ADC Section #1	0-120°C (±1-2°C)	85°C	ตรวจอุณหภูมิ ADC ส่วนที่สองเพิ่มการเฝ้าระวังอย่างละเอียด

ตารางที่ 3.1 คำอธิบายการกำหนดค่าของเซนเซอร์ (ต่อ) [48]

No.	Module	Component	Spec	Trigger	Function
12	GSA6044AXI	PA #0	0-120°C (±1-2°C)	85°C	วัดอุณหภูมิภาคขยายกำลัง PA #0 ป้องกันภาคขยายร้อนเกิน
13	GSA6044AXI	PA #1	0-120°C (±1-2°C)	85°C	ตรวจสอบอุณหภูมิ PA #1 ซึ่งมีโหลดสูงกว่าและถูกตั้งค่า Trigger สูงกว่า
14	GSA6044AXI	AP FPGA #0	0-120°C (±1-2°C)	70°C	วัดอุณหภูมิ FPGA ฝั่งประมวลผลเพื่อควบคุมการทำงานความเร็วสูง
15	GSA6044AXI	AP FPGA #1	0-120°C (±1-2°C)	70°C	ตรวจสอบ FPGA #1 ทำงานคู่กับ #0 เพื่อป้องกันความร้อนสะสม
16	GSA6044AXI	AP FPGA #2	0-120°C (±1-2°C)	70°C	เซนเซอร์สำหรับ FPGA ตัวที่ 2 บนบอร์ดเพื่อควบคุม thermal balance
17	GSA6044AXI	AP FPGA #3	0-120°C (±1-2°C)	70°C	ตรวจสอบความร้อนของ FPGA #3 ที่อยู่ในตำแหน่งโหลดต่ำกว่า
18	GSA6044AXI	HT FPGA	0-120°C (±1-2°C)	70°C	ตรวจวัด FPGA ความเร็วสูง (High-Throughput) ที่ critical ต่อความเร็วของระบบ
19	GSA6044AXI	MFM Section	0-120°C (±1-2°C)	70°C	ตรวจสอบอุณหภูมิบริเวณวงจรวิเคราะห์สัญญาณแม่เหล็ก (MFM)
20	GSA6044AXI	ADC Cooling Fan #0	0-15000 RPM/ ±5%	0-6000 RPM	วัดรอบพัดลมระบายความร้อน ADC ตัวที่ 0 เพื่อประเมินประสิทธิภาพการไหลเวียนอากาศของส่วน ADC
21	GSA6044AXI	ADC Cooling Fan #1	0-15000 RPM/ ±5%	0-6000 RPM	วัดการหมุนของพัดลมตัวที่ 1 ในบริเวณ ADC เพื่อป้องกันร้อนเกิน
22	GSA6044AXI	AP FPGA Fan #0	0-15000 RPM/ ±5%	0-15000 RPM	ตรวจสอบความเร็วพัดลม FPGA #0 รองรับภาระงานการประมวลผล
23	GSA6044AXI	AP FPGA Fan #1	0-15000 RPM/ ±5%	0-15000 RPM	ตรวจวัดความเร็วรอบพัดลมเพื่อรักษาเสถียรภาพความร้อนของ FPGA
24	GSA6044AXI	AP FPGA Fan #2	0-15000 RPM/ ±5%	0-15000 RPM	ควบคุมและตรวจสอบพัดลมตัวที่ 2 ในชุด FPGA

ตารางที่ 3.1 คำอธิบายการกำหนดค่าของเซนเซอร์ (ต่อ) [48]

No.	Module	Component	Spec	Trigger	Function
25	GSA6044AXI	AP FPGA Fan #3	0-15000 RPM/ ±5%	0-15000 RPM	เฝ้าระวังการหมุนพัดลมตัวที่ 3 ของ FPGA สำหรับงานประสิทธิภาพสูง
26	GSA6044AXI	Box Fan #0	0-15000 RPM/ ±5%	0-6000 RPM	ตรวจสอบพัดลมกล่องระบบ #0 ซึ่งใช้ระบายความร้อนให้ระบบทั้งหมด
27	GSA6044AXI	Box Fan #1	0-15000 RPM/ ±5%	0-6000 RPM	พัดลมกล่องระบบ #1 ให้การระบายอากาศหลักแก่ชุดทดสอบทั้งหมด ป้องกันความร้อนสะสม
28	GSA6044AXI	Box Fan #2	0-15000 RPM/ ±5%	0-6000 RPM	พัดลมกล่องระบบ #2 ให้การระบายอากาศหลักแก่ชุดทดสอบทั้งหมด ป้องกันความร้อนสะสม
29	GSA6044AXI	Box Fan #3	0-15000 RPM/ ±5%	0-6000 RPM	พัดลมกล่องระบบ #3 ให้การระบายอากาศหลักแก่ชุดทดสอบทั้งหมด ป้องกันความร้อนสะสม
30	GSA6044AXI	Box Fan #4	0-15000 RPM/ ±5%	0-6000 RPM	พัดลมกล่องระบบ #4 ให้การระบายอากาศหลักแก่ชุดทดสอบทั้งหมด ป้องกันความร้อนสะสม
31	GSA6044AXI	Box Fan #5	0-15000 RPM/ ±5%	0-6000 RPM	พัดลมกล่องระบบ #5 ให้การระบายอากาศหลักแก่ชุดทดสอบทั้งหมด ป้องกันความร้อนสะสม
32	GSA6044AXI	Box Fan #6	0-15000 RPM/ ±5%	0-6000 RPM	พัดลมกล่องระบบ #6 ให้การระบายอากาศหลักแก่ชุดทดสอบทั้งหมด ป้องกันความร้อนสะสม
33	GSA6044AXI	Box Fan #7	0-15000 RPM/ ±5%	0-6000 RPM	พัดลมกล่องระบบ #7 ให้การระบายอากาศหลักแก่ชุดทดสอบทั้งหมด ป้องกันความร้อนสะสม
34	GSA6044AXI	Box Fan #8	0-15000 RPM/ ±5%	0-6000 RPM	พัดลมกล่องระบบ #8 ให้การระบายอากาศหลักแก่ชุดทดสอบทั้งหมด ป้องกันความร้อนสะสม
35	GSA6044AXI	Box Fan #9	0-15000 RPM/ ±5%	0-6000 RPM	พัดลมกล่องระบบ #9 ให้การระบายอากาศหลักแก่ชุดทดสอบทั้งหมด ป้องกันความร้อนสะสม

ตารางที่ 3.1 คำอธิบายการกำหนดค่าของเซนเซอร์ (ต่อ) [48]

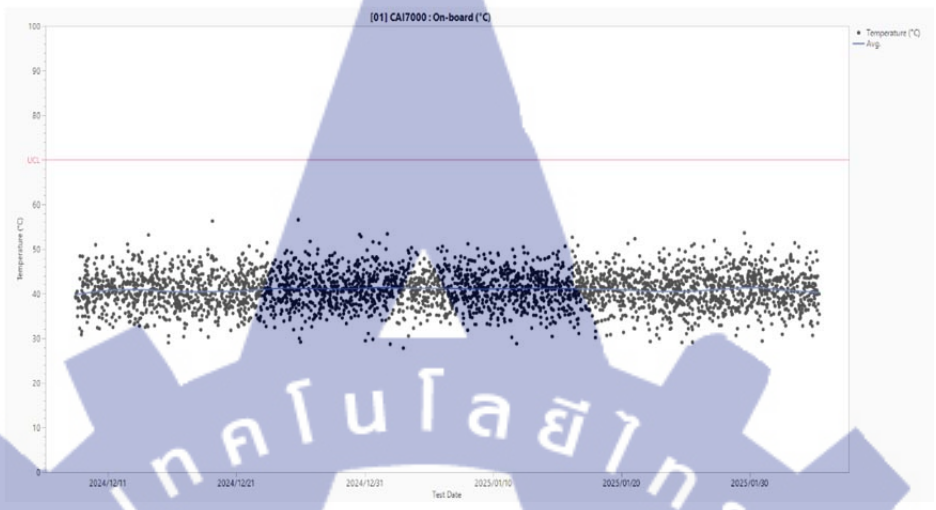
No.	Module	Component	Spec	Trigger	Function
36	PG6G	PG FPGA	0-120°C (±1-2°C)	70°C	ตรวจสอบ FPGA ที่ใช้สร้างสัญญาณทดสอบความเร็วสูง
37	PG6G	PG DAC	0-120°C (±1-2°C)	70°C	วัดอุณหภูมิ DAC เพื่อรักษาความเสถียรของสัญญาณทดสอบ
40	MB4000	System Fan #2	0-15000 RPM/ ±5%	0-6000 RPM	ตรวจวัดความเร็วรอบเพื่อรักษาปริมาณลมที่เย็นคงที่
41	MB4000	System Fan #3	0-15000 RPM/ ±5%	0-6000 RPM	เฝ้าระวังรอบพัดลมเพื่อป้องกัน overheating
42	MB4000	System Fan #4	0-15000 RPM/ ±5%	0-6000 RPM	ตรวจสอบการทำงานของพัดลมตัวที่ 4 ในระบบ
43	MB4000	System Fan #5	0-15000 RPM/ ±5%	0-6000 RPM	พัดลมหลักตัวที่ 5 สำหรับการระบายความร้อนแบบต่อเนื่อง

หมายเหตุ : ช่วงการวัด (Measurement Range) และความแม่นยำ (Accuracy) เป็นค่าประมาณหรือค่าที่มักระบุในสเปกทั่วไปของเซนเซอร์/โมดูลมักมีการเปลี่ยนแปลงขึ้นอยู่กับสภาพแวดล้อมจริง

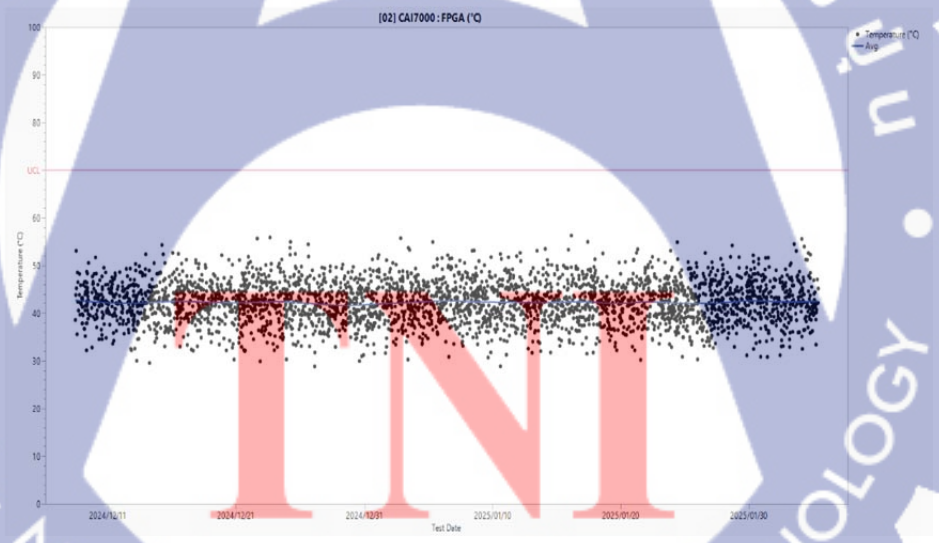
จากรูปที่ 3.5 ถึง 3.47 แสดงการกระจายตัวของข้อมูล (Data Distribution) ของเซนเซอร์แต่ละตัวในระบบทดสอบ ทั้งในกลุ่มเซนเซอร์อุณหภูมิและเซนเซอร์วัดความเร็วรอบการหมุนของพัดลม ซึ่งช่วยให้สามารถสังเกตลักษณะการกระจาย ความแปรปรวน และค่าผิดปกติ (Outliers) ที่เกิดขึ้นในข้อมูลจริงได้อย่างชัดเจน ตัวอย่างเช่น การกระจายตัวของค่าความเร็วรอบพัดลมที่มีการลดลงอย่างฉับพลันหรือมีค่าเป็นศูนย์ อันอาจบ่งชี้ถึงความผิดปกติของพัดลมหรือการขัดข้องของระบบบันทึกข้อมูล การวิเคราะห์รูปการกระจายตัวเหล่านี้จึงเป็นขั้นตอนสำคัญในการกำหนดเกณฑ์สำหรับการคัดกรองและทำความสะอาดข้อมูล

กระบวนการทำความสะอาดข้อมูลครอบคลุมการลบข้อมูลที่ซ้ำกัน (Duplicate Records) การกรองข้อมูลที่อยู่นอกช่วงเวลาการทดสอบจริง การคัดออกข้อมูลจากรอบการทดสอบที่ไม่สมบูรณ์ รวมถึงการตรวจสอบและจัดรูปแบบชนิดข้อมูล (Data Type) ให้ถูกต้องตามลักษณะของตัวแปร เช่น ค่าตัวเลขจากเซนเซอร์อุณหภูมิและความเร็วรอบพัดลม สถานะการทำงานของโมดูล และค่าลอจิกที่ใช้บ่งชี้เหตุการณ์ผิดปกติ การเตรียมข้อมูลในขั้นตอนนี้ช่วยให้ข้อมูลมีความสอดคล้อง ลดอิทธิพลของสัญญาณรบกวน และ

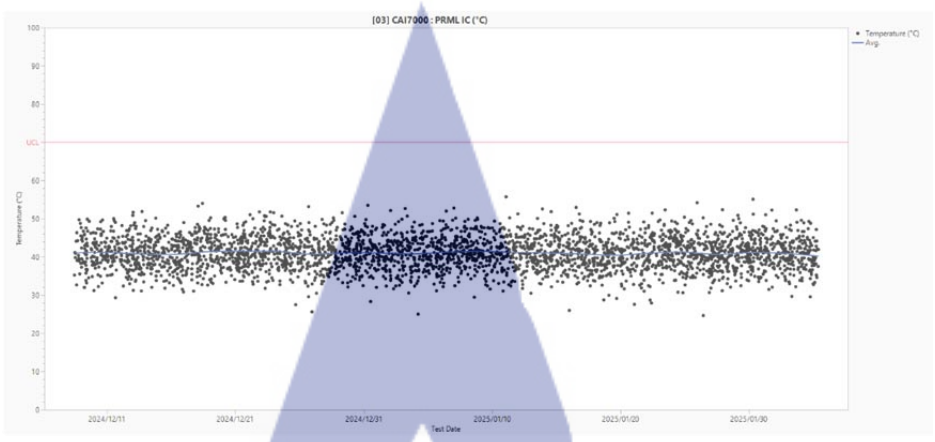
เพิ่มความน่าเชื่อถือของข้อมูลก่อนนำไปใช้ในการสร้างและประเมินแบบจำลองการเรียนรู้ของเครื่องในลำดับถัดไป



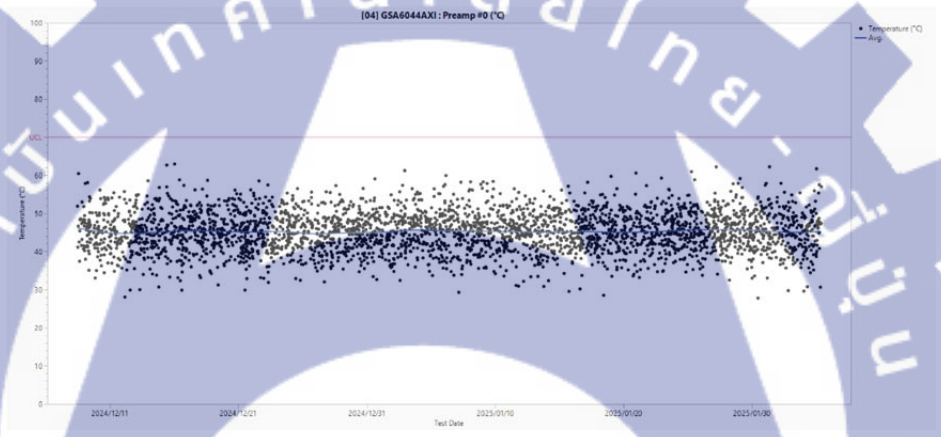
รูปที่ 3.5 การกระจายค่าความร้อนของโมดูล CAI7000 : On-board



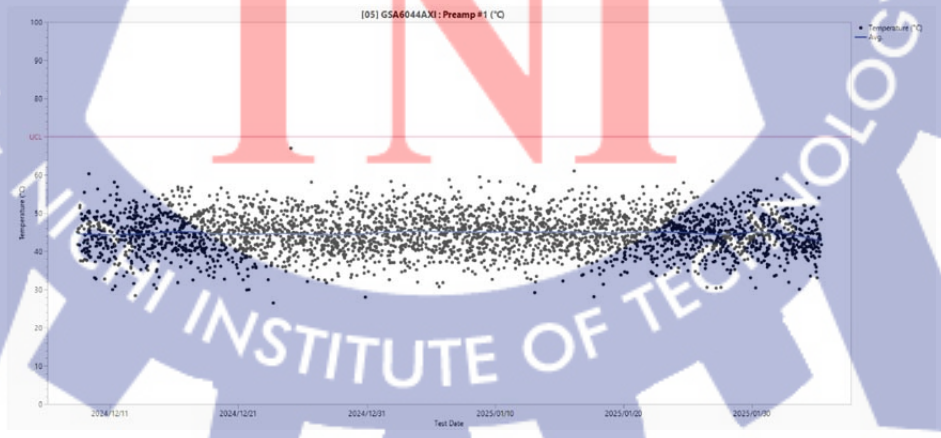
รูปที่ 3.6 การกระจายค่าความร้อนของโมดูล CAI7000 : FPGA



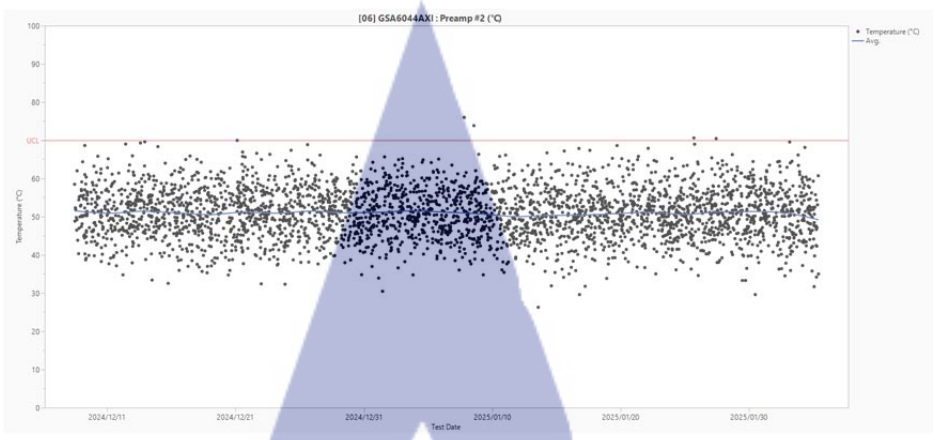
รูปที่ 3.7 การกระจายค่าความร้อนของโมดูล CAI7000 : PRML IC



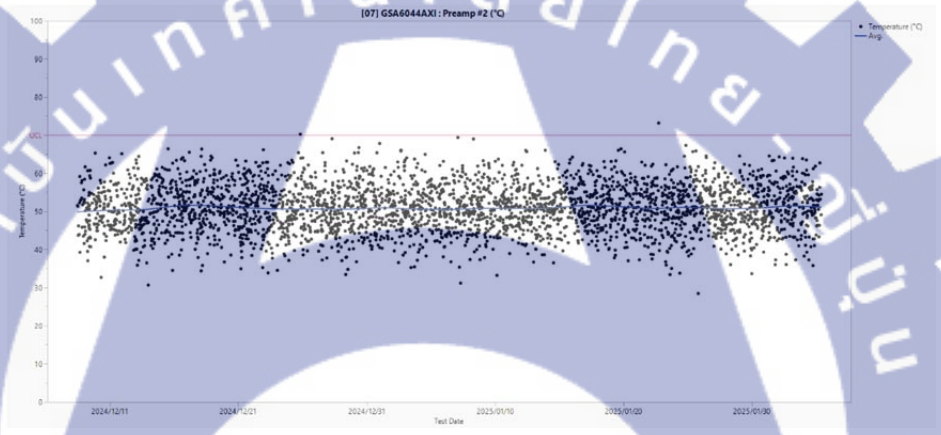
รูปที่ 3.8 การกระจายค่าความร้อนของโมดูล GSA6044AXI : Preamp #0



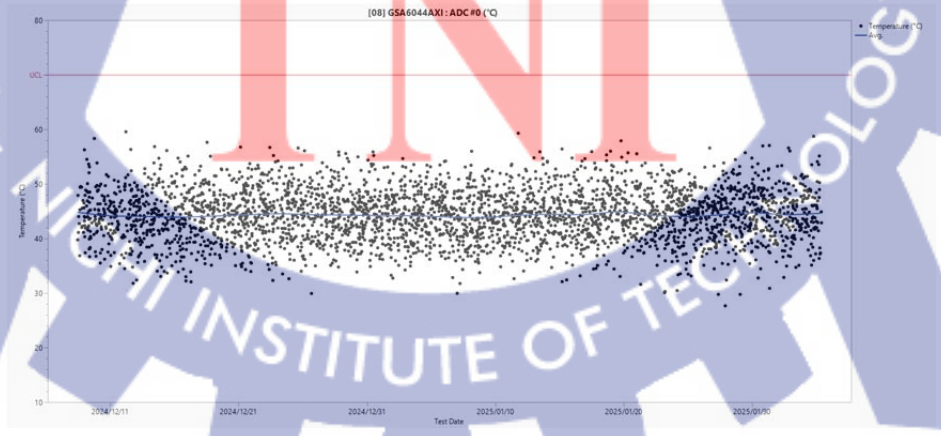
รูปที่ 3.9 การกระจายค่าความร้อนของโมดูล GSA6044AXI : Preamp #0



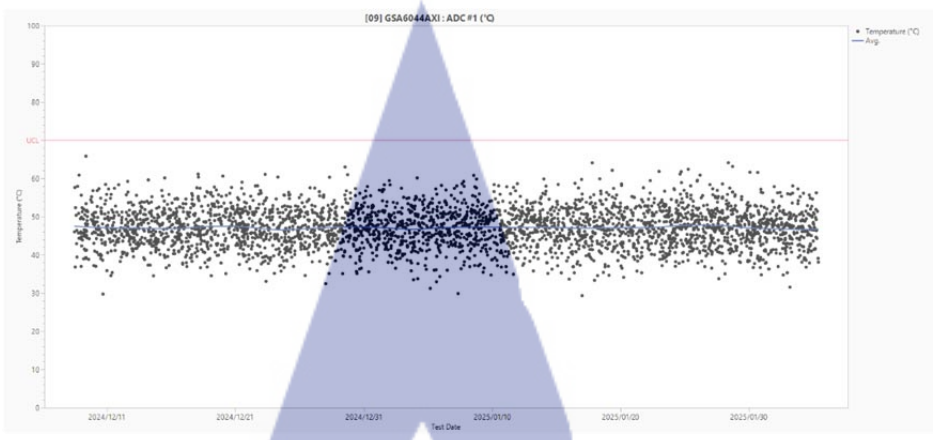
รูปที่ 3.10 การกระจายค่าความร้อนของโมดูล GSA6044AXI : Preamp #2



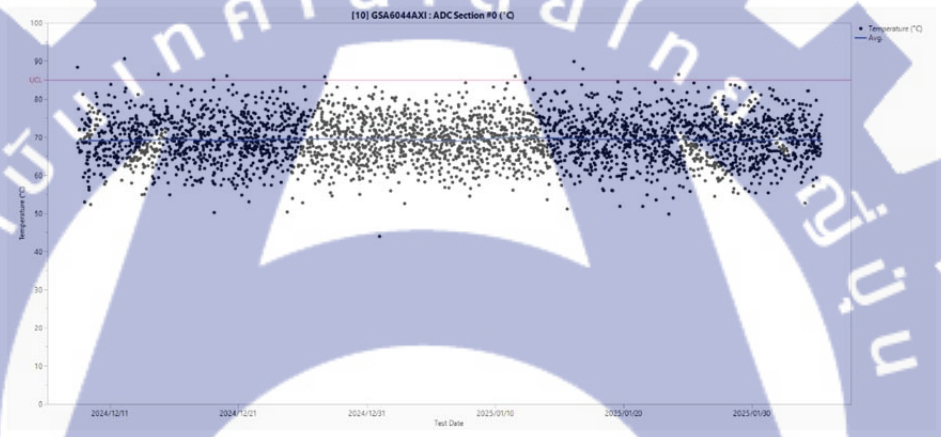
รูปที่ 3.11 การกระจายค่าความร้อนของโมดูล GSA6044AXI : Preamp #3



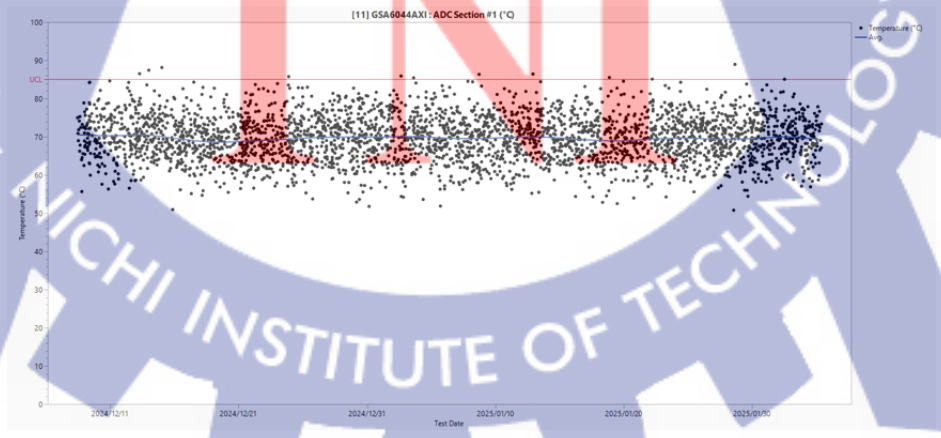
รูปที่ 3.12 การกระจายค่าความร้อนของโมดูล GSA6044AXI : ADC #0



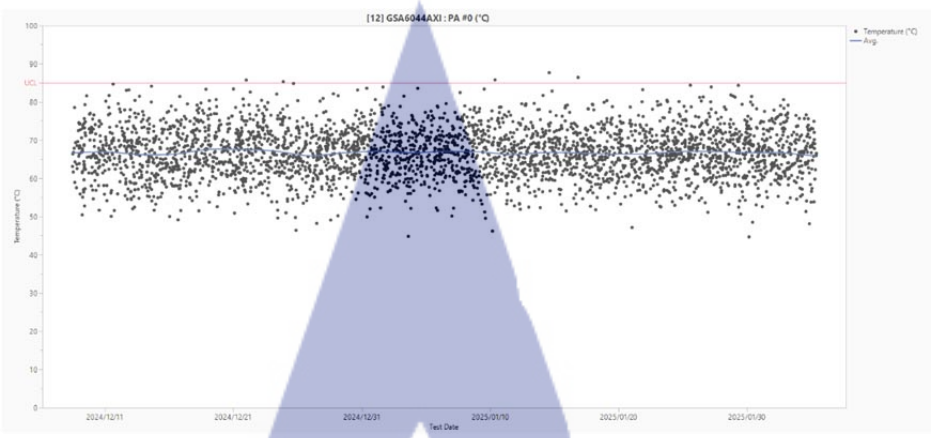
รูปที่ 3.13 การกระจายค่าความร้อนของโมดูล GSA6044AXI : ADC #1



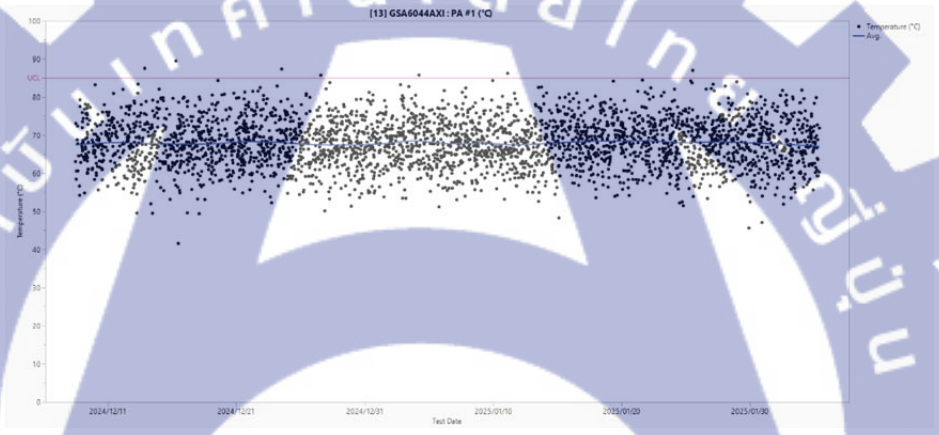
รูปที่ 3.14 การกระจายค่าความร้อนของโมดูล GSA6044AXI : ADC Section #0



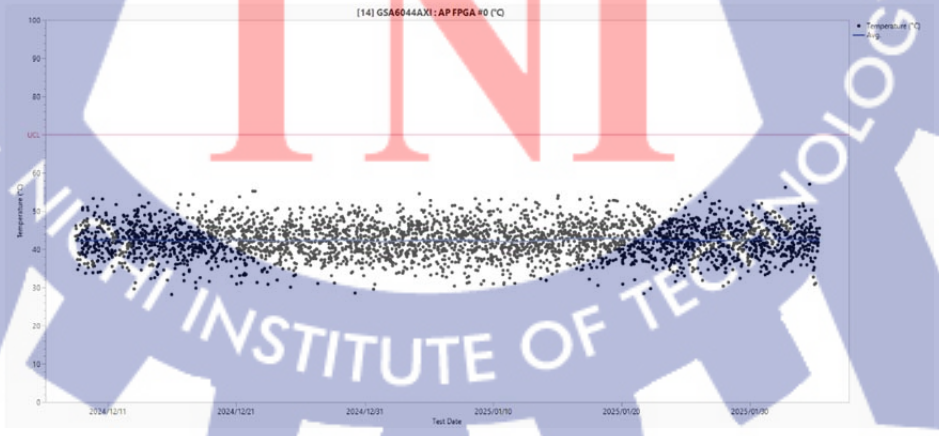
รูปที่ 3.15 การกระจายค่าความร้อนของโมดูล GSA6044AXI : ADC Section #1



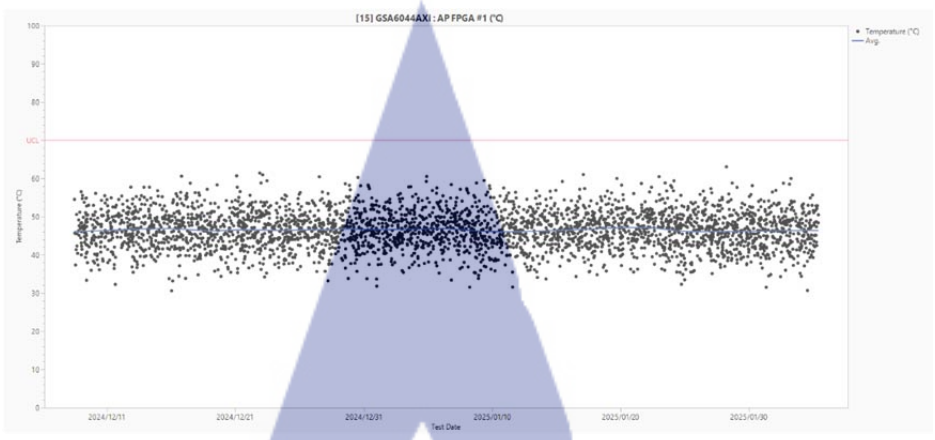
รูปที่ 3.16 การกระจายค่าความร้อนของโมดูล GSA6044AXI : PA #0



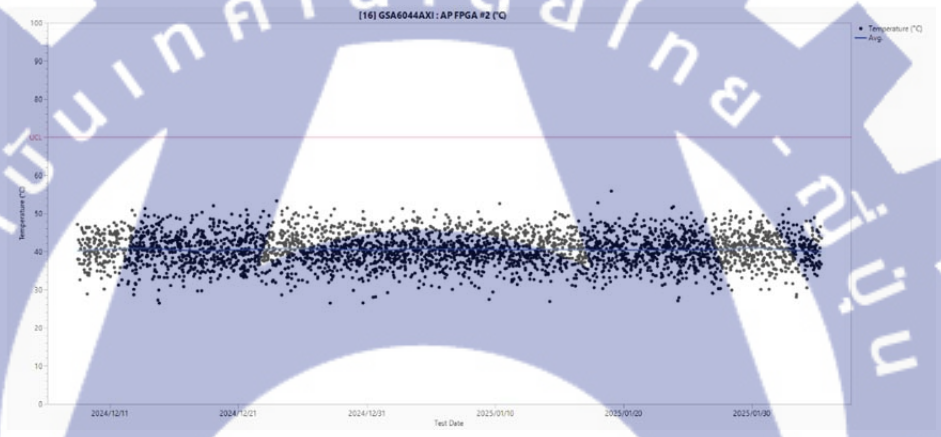
รูปที่ 3.17 การกระจายค่าความร้อนของโมดูล GSA6044AXI : PA #1



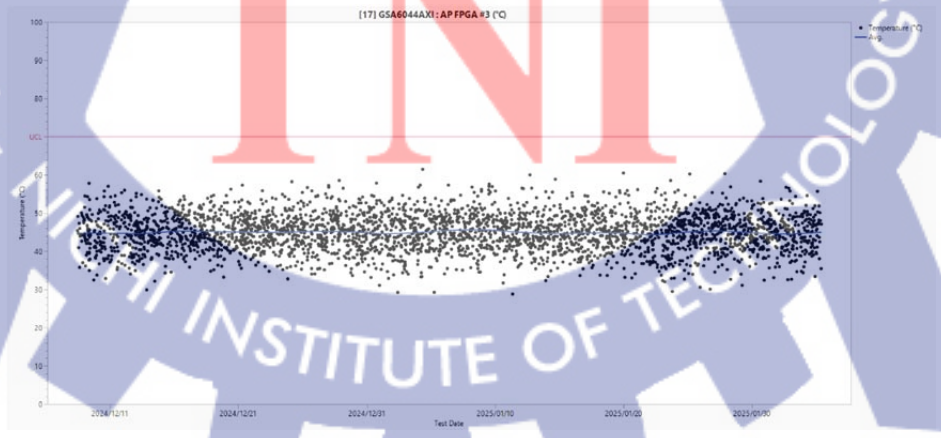
รูปที่ 3.18 การกระจายค่าความร้อนของโมดูล GSA6044AXI : AP FPGA #0



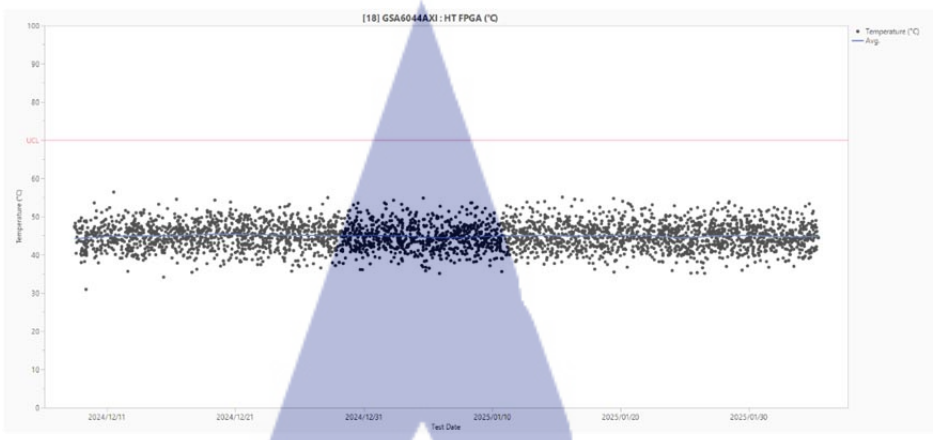
รูปที่ 3.19 การกระจายค่าความร้อนของโมดูล GSA6044AXI : AP FPGA #1



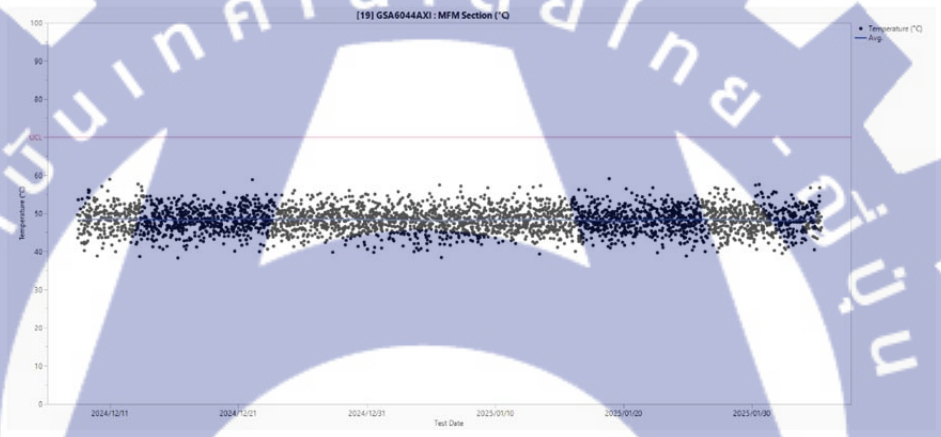
รูปที่ 3.20 การกระจายค่าความร้อนของโมดูล GSA6044AXI : AP FPGA #2



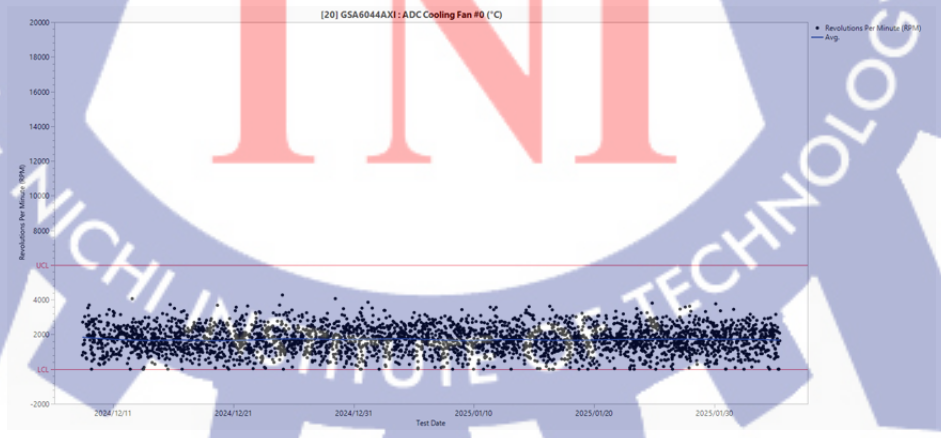
รูปที่ 3.21 การกระจายค่าความร้อนของโมดูล GSA6044AXI : AP FPGA #3



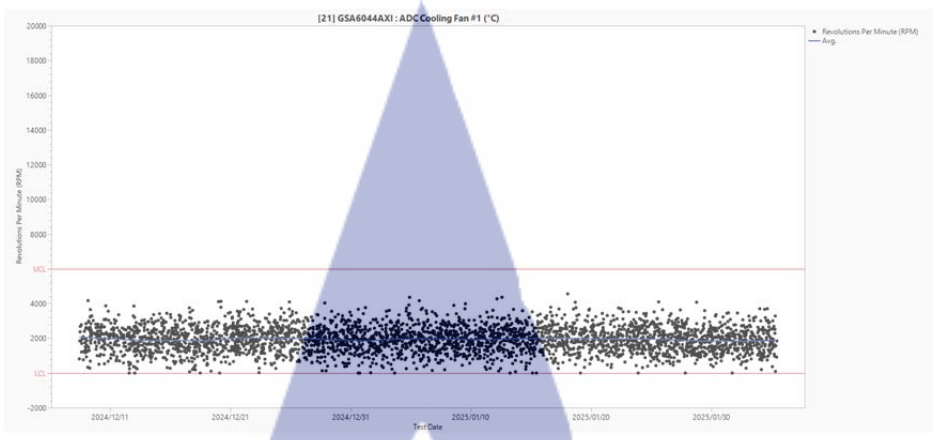
รูปที่ 3.22 การกระจายค่าความร้อนของโมดูล GSA6044AXI : HT FPGA



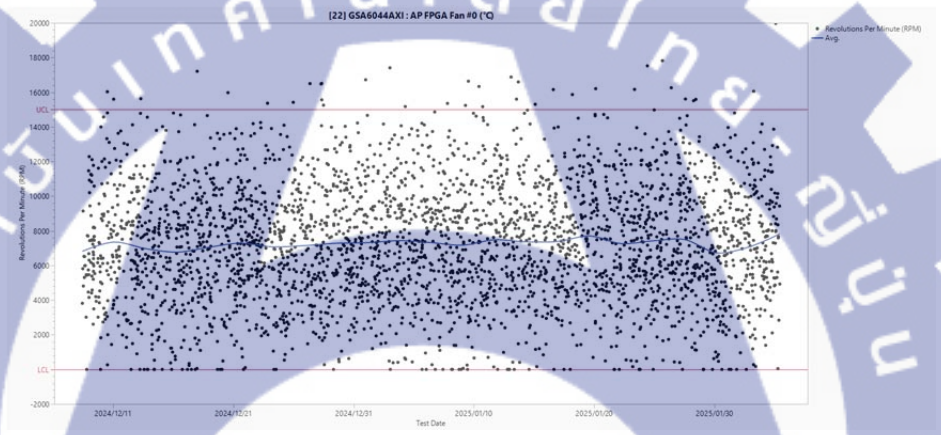
รูปที่ 3.23 การกระจายค่าความร้อนของโมดูล GSA6044AXI : MFM Section



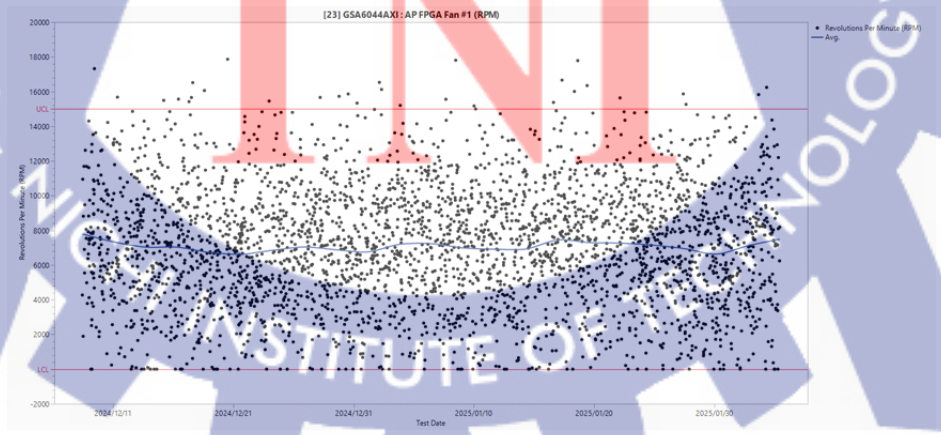
รูปที่ 3.24 การกระจายค่าความเร็วพัดลมของโมดูล GSA6044AXI : ADC Cooling Fan #0



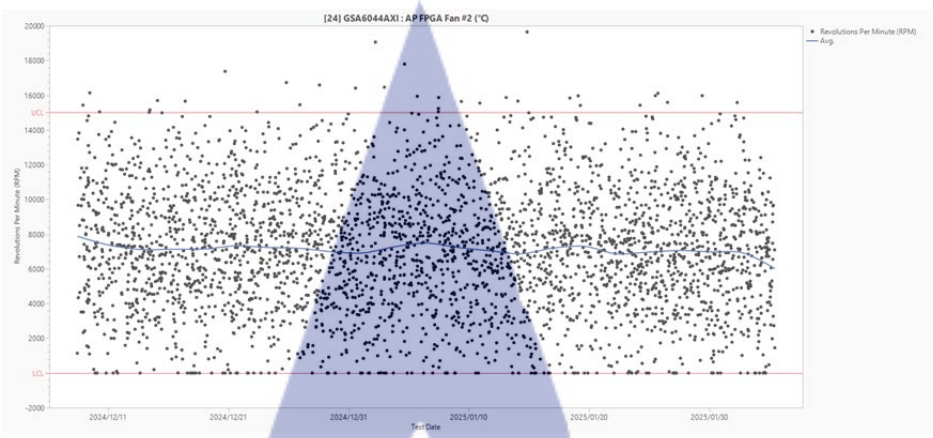
รูปที่ 3.25 การกระจายค่าความเร็วพัดลมของโมดูล GSA6044AXI : ADC Cooling Fan #1



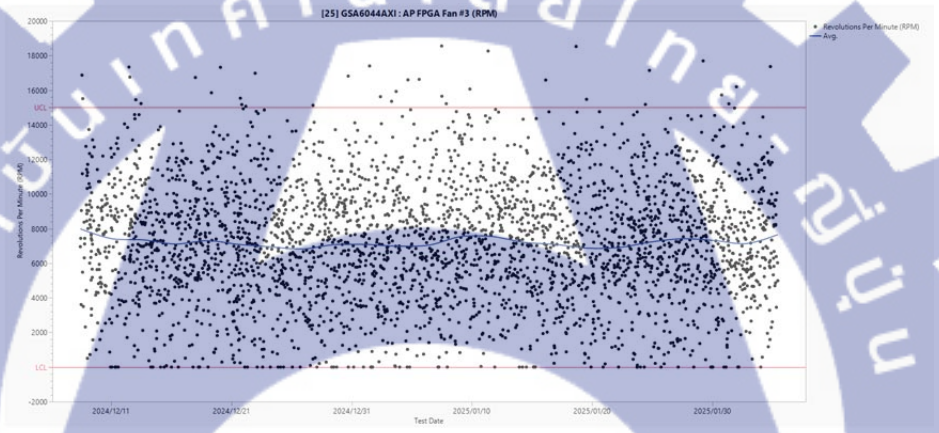
รูปที่ 3.26 การกระจายค่าความเร็วพัดลมของโมดูล GSA6044AXI : AP FPGA Fan #0



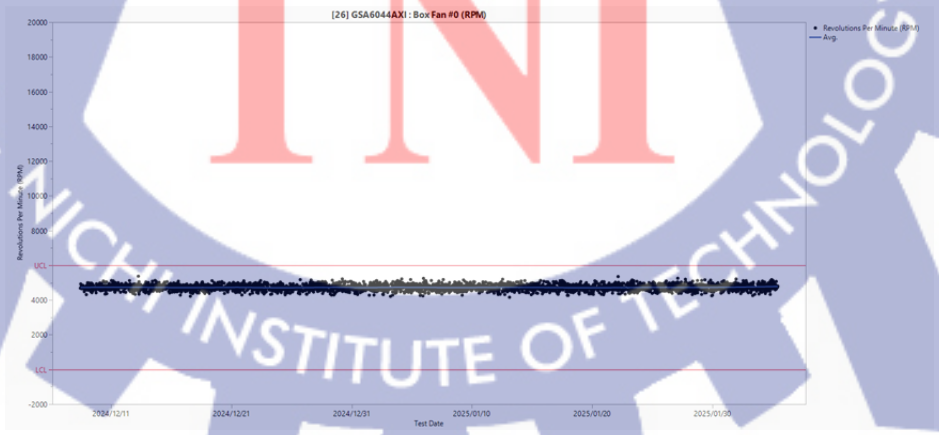
รูปที่ 3.27 การกระจายค่าความเร็วพัดลมของโมดูล GSA6044AXI : AP FPGA Fan #1



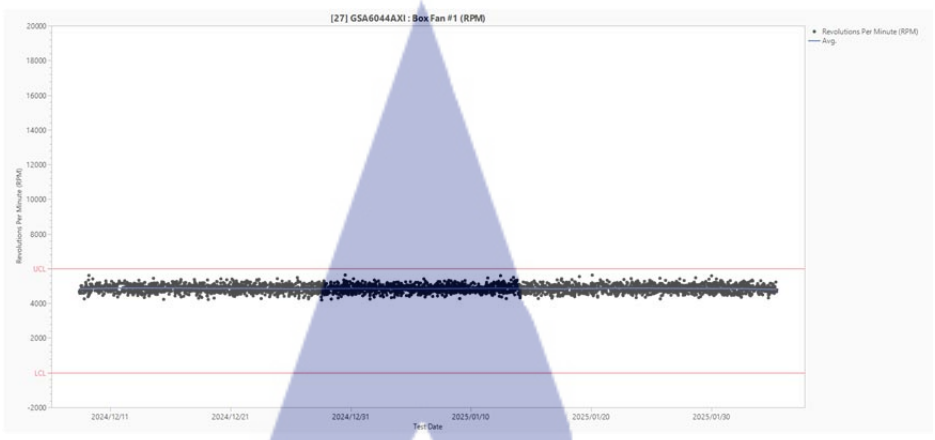
รูปที่ 3.28 การกระจายค่าความเร็วพัดลมของโมดูล GSA6044AXI : AP FPGA Fan #2



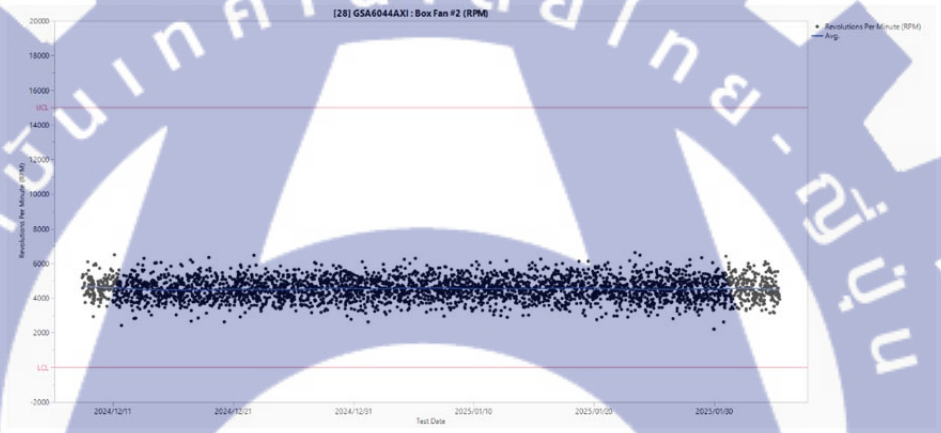
รูปที่ 3.29 การกระจายค่าความเร็วพัดลมของโมดูล GSA6044AXI : AP FPGA Fan #3



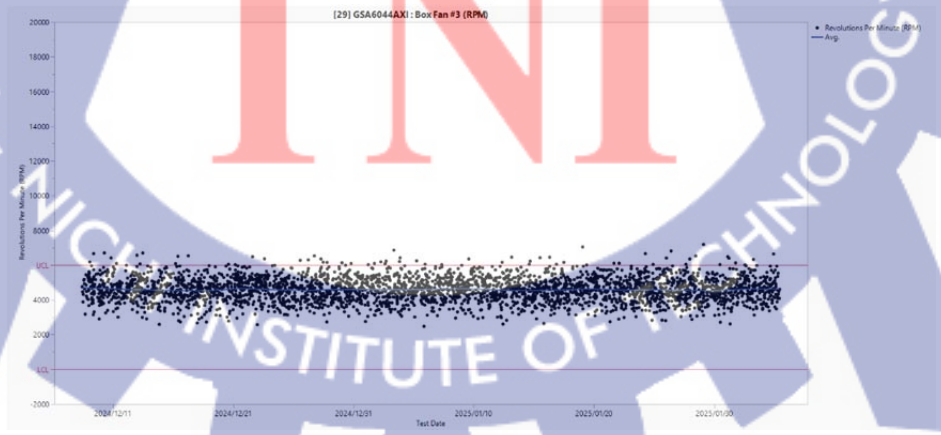
รูปที่ 3.30 การกระจายค่าความเร็วพัดลมของโมดูล GSA6044AXI : Box Fan #0



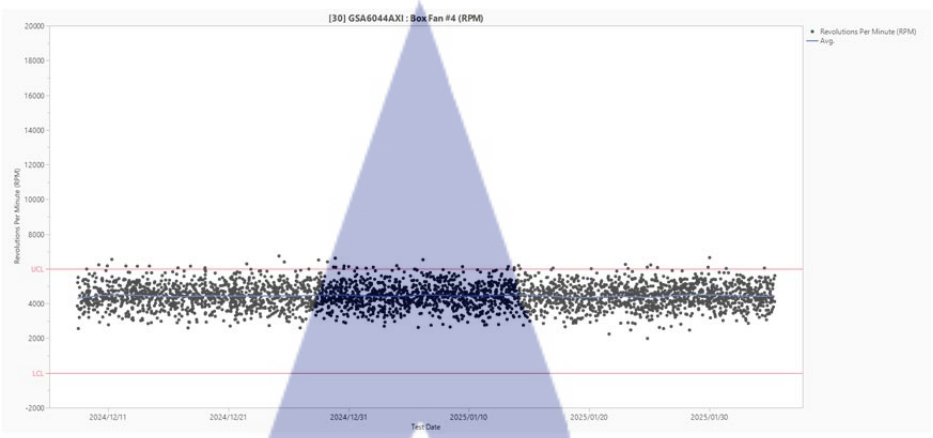
รูปที่ 3.31 การกระจายค่าความเร็วพัดลมของโมดูล GSA6044AXI : Box Fan #1



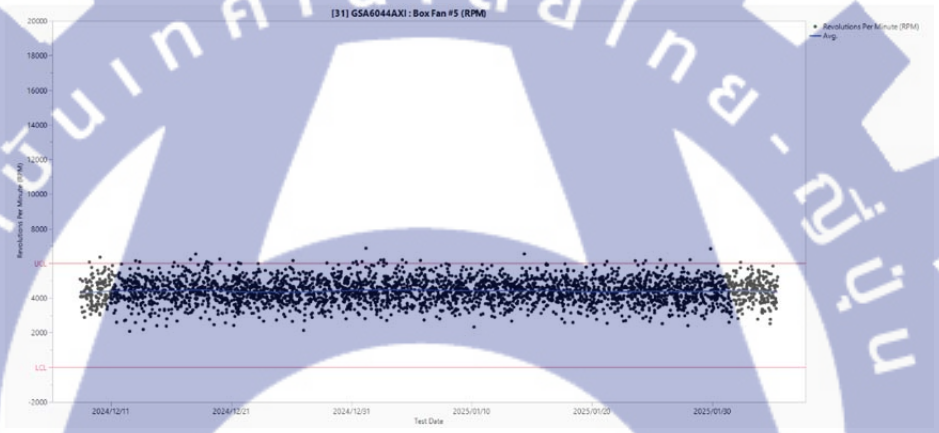
รูปที่ 3.32 การกระจายค่าความเร็วพัดลมของโมดูล GSA6044AXI : Box Fan #2



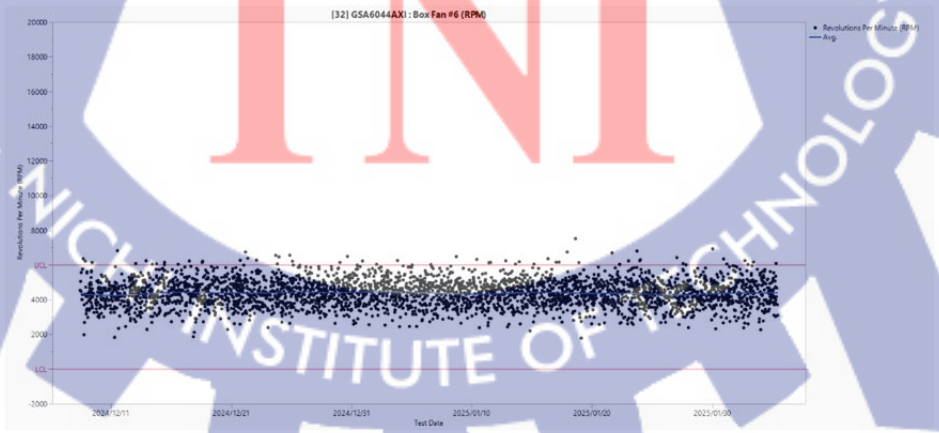
รูปที่ 3.33 การกระจายค่าความเร็วพัดลมของโมดูล GSA6044AXI : Box Fan #3



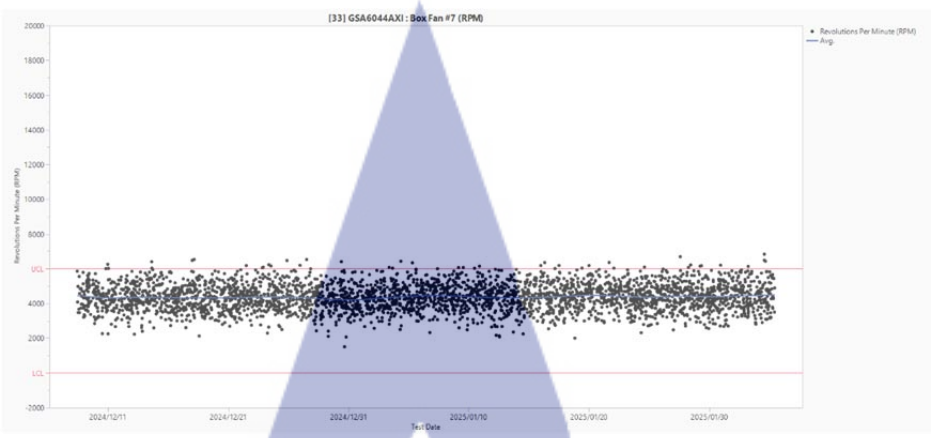
รูปที่ 3.34 การกระจายค่าความเร็วพัดลมของโมดูล GSA6044AXI : Box Fan #4



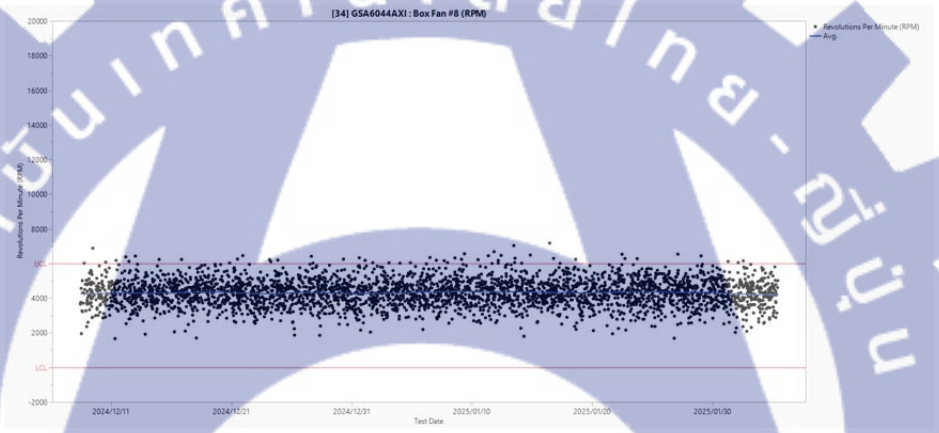
รูปที่ 3.35 การกระจายค่าความเร็วพัดลมของโมดูล GSA6044AXI : Box Fan #5



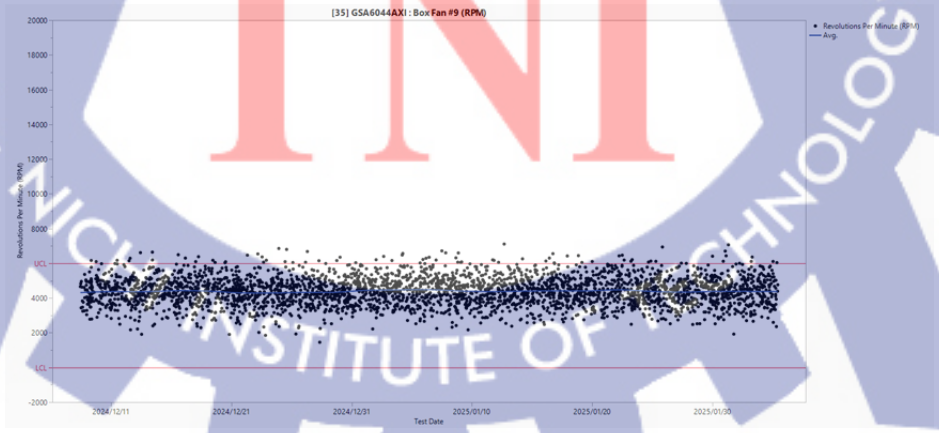
รูปที่ 3.36 การกระจายค่าความเร็วพัดลมของโมดูล GSA6044AXI : Box Fan #6



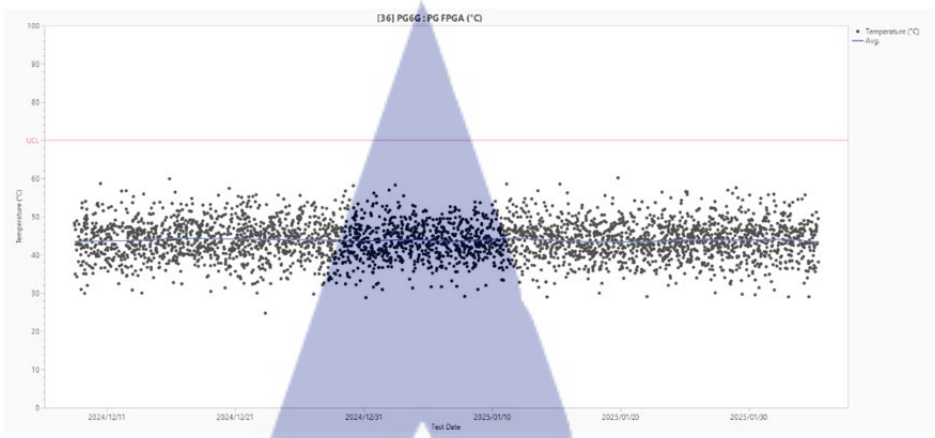
รูปที่ 3.37 การกระจายค่าความเร็วพัดลมของโมดูล GSA6044AXI : Box Fan #7



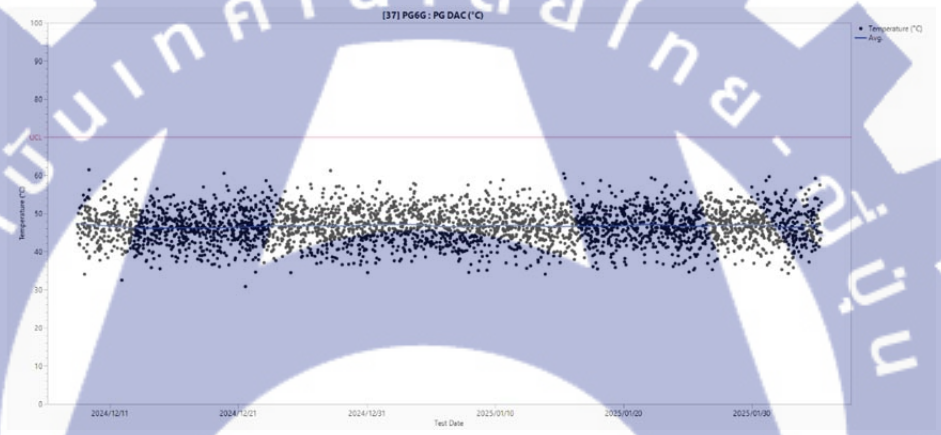
รูปที่ 3.38 การกระจายค่าความเร็วพัดลมของโมดูล GSA6044AXI : Box Fan #8



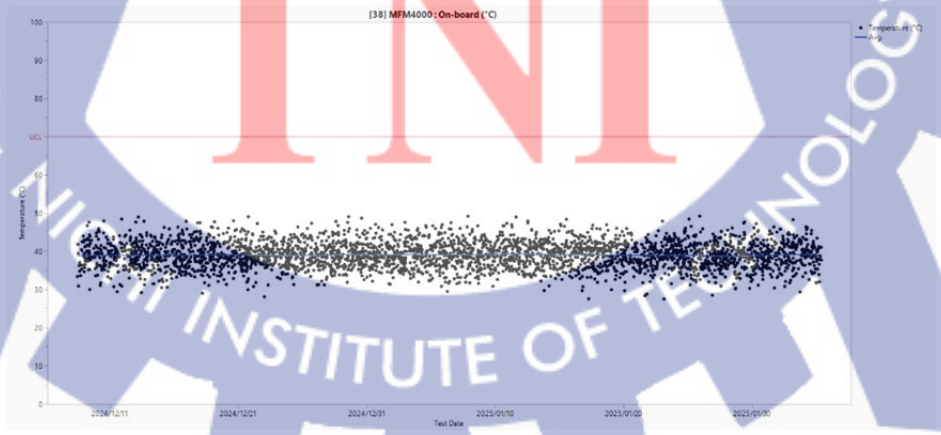
รูปที่ 3.39 การกระจายค่าความเร็วพัดลมของโมดูล GSA6044AXI : Box Fan #9



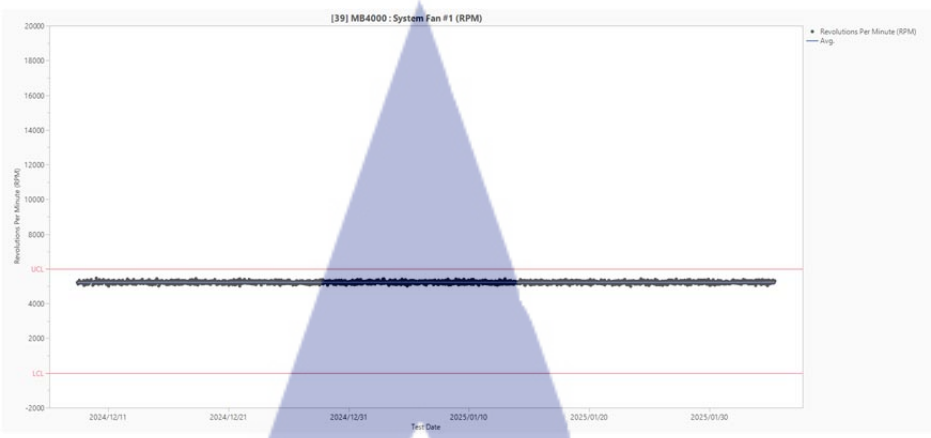
รูปที่ 3.40 การกระจายค่าความร้อนของโมดูล PG6G : PG FPGA



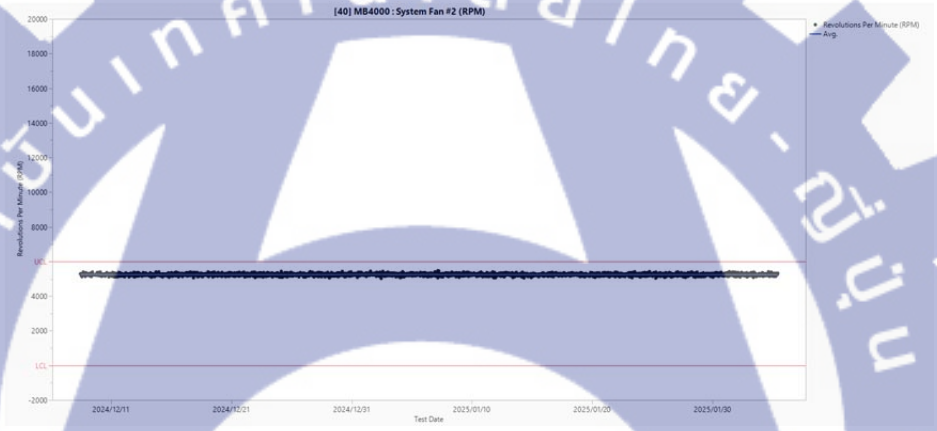
รูปที่ 3.41 การกระจายค่าความร้อนของโมดูล PG6G : DAC



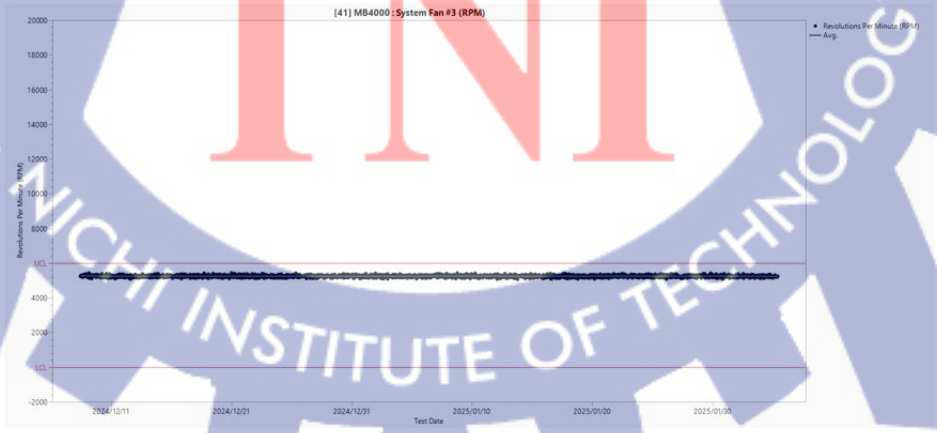
รูปที่ 3.42 การกระจายค่าความร้อนของโมดูล MFM4000 : On-board



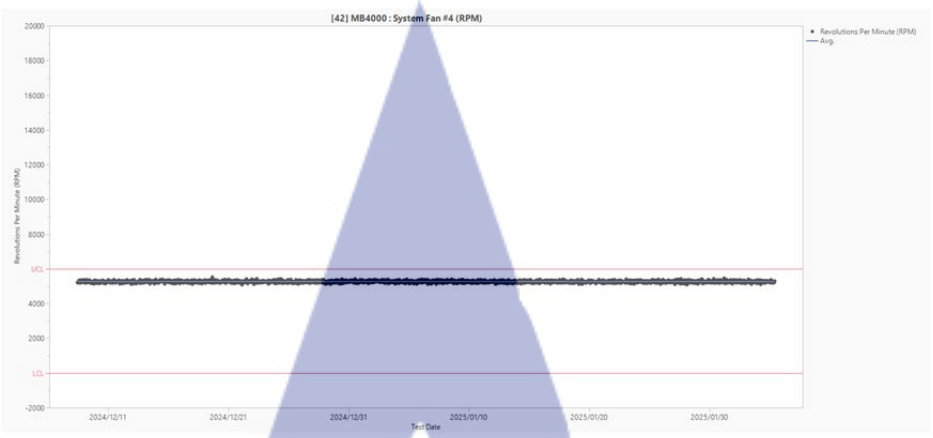
รูปที่ 3.43 การกระจายค่าความเร็วพัดลมของโมดูล MB4000 : System Fan #1



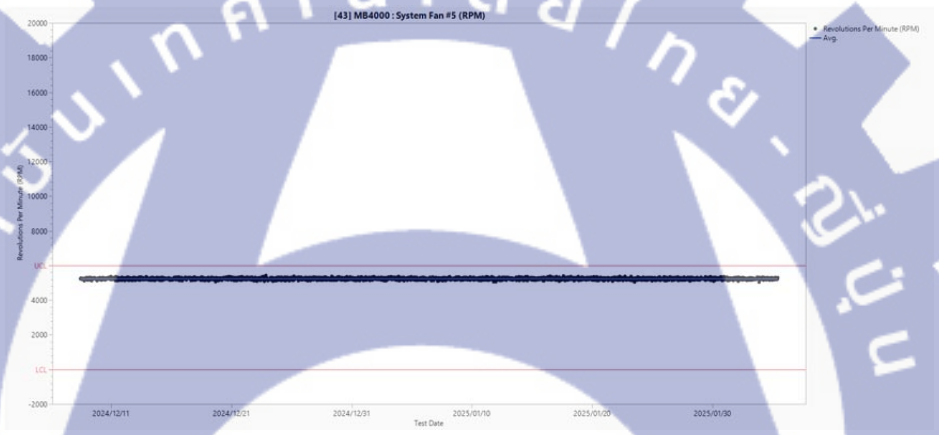
รูปที่ 3.44 การกระจายค่าความเร็วพัดลมของโมดูล MB4000 : System Fan #2



รูปที่ 3.45 การกระจายค่าความเร็วพัดลมของโมดูล MB4000 : System Fan #3



รูปที่ 3.46 การกระจายค่าความเร็วพัดลมของโมดูล MB4000 : System Fan #4



รูปที่ 3.47 การกระจายค่าความเร็วพัดลมของโมดูล MB4000 : System Fan #5

3.2.2 การจัดรูปแบบ Timestamp

ข้อมูลจากเครื่องตรวจสอบอุปกรณ์มักถูกบันทึกด้วย Timestamp ที่มีรูปแบบไม่สอดคล้องกัน เช่น ความละเอียดต่างระดับ (ระดับวินาที มิลลิวินาที) หรือ Time Zone ต่างกัน การจัดรูปแบบเวลาให้เป็นมาตรฐานเดียวกันช่วยให้สามารถเชื่อมโยงเหตุการณ์ วิเคราะห์ลำดับเวลา (Time-Series) และคำนวณค่าทางสถิติที่อ้างอิงเวลา เช่น Duration, Cycle Time หรือ Time Lag ระหว่างโมดูลของเครื่องจักรได้อย่างถูกต้อง

3.2.3 การจัดการ Missing Value

ค่าที่หายไป (Missing Value) เป็นปัญหาที่พบได้บ่อยในข้อมูลจากระบบโรงงานอุตสาหกรรม เช่น กรณีที่เซ็นเซอร์ไม่สามารถตรวจจับค่าได้ในบางช่วงเวลา การสูญหายของข้อมูลระหว่าง

การส่งผ่าน หรือความขัดข้องของระบบบันทึกข้อมูล วิธีการจัดการค่าที่หายไปต้องพิจารณาจากลักษณะของข้อมูล โครงสร้างเชิงเวลา และระดับความสำคัญของตัวแปรที่เกี่ยวข้อง แนวทางที่นิยมใช้ในการจัดการ Missing Value ได้แก่

1) การลบแถวข้อมูลที่มี Missing Value เป็นจำนวนมาก

2) การแทนค่าด้วยค่าเฉลี่ย (Mean), มัธยฐาน (Median) หรือค่าที่พบบ่อย (Mode)

3) การใช้วิธี Interpolation สำหรับข้อมูลแบบอนุกรมเวลา (Time-Series)

4) การใช้แบบจำลองเชิงสถิติหรือ Machine Learning เพื่อทำนายค่าที่หายไป (Model-Based Imputation)

สำหรับการศึกษานี้ เลือกใช้วิธี Interpolation เป็นแนวทางหลัก เนื่องจากข้อมูลเซ็นเซอร์จากระบบวิเคราะห์สัญญาณการอ่าน-เขียนของฮาร์ดดิสก์มีลักษณะเป็นข้อมูลแบบ Time-Series ที่เปลี่ยนแปลงอย่างต่อเนื่องตามเวลา เช่น ค่าอุณหภูมิและความเร็วรอบการทดสอบ ตัวแปรเหล่านี้มีความสัมพันธ์เชิงเวลาแบบค่อยเป็นค่อยไป (Smooth Progression) การประมาณค่าด้วยวิธีเชิงเส้นหรือการ Interpolation ตามเวลา (Time-Based Interpolation) จึงสามารถเติมค่าที่ขาดหายไปได้สอดคล้องกับพฤติกรรมจริงของเครื่องจักร และลดความเสี่ยงในการเกิดค่าผิดปกติแบบกระโดด ซึ่งอาจเกิดขึ้นจากการใช้ค่าคงที่ เช่น Mean หรือ Median

นอกจากนี้ การใช้ Interpolation ยังช่วยรักษาความต่อเนื่องของสัญญาณ (Signal Continuity) ซึ่งมีความสำคัญต่อประสิทธิภาพของแบบจำลอง Machine Learning โดยเฉพาะกลุ่ม Boosting และ Tree-Based Models ที่อาศัยข้อมูลเชิงลำดับเพื่อเรียนรู้แนวโน้มการเสื่อมสภาพของเครื่องจักรได้อย่างแม่นยำ

3.2.4 การจัดการ Outlier

Outlier หรือค่าผิดปกติ มักเกิดจากความผิดพลาดของเซ็นเซอร์ การปรับเทียบ (Calibration) ที่ไม่ถูกต้อง หรือเหตุการณ์ผิดปกติระหว่างการทำงานของเครื่องจักร การตัดสินใจว่าจะเก็บหรือกำจัด Outlier ออกจากชุดข้อมูลจำเป็นต้องพิจารณาว่าค่าดังกล่าวเป็น “ข้อมูลที่ผิดพลาด” หรือเป็น “สัญญาณที่สะท้อนพฤติกรรมผิดปกติที่มีความหมาย” ของระบบ วิธีการจัดการ Outlier ที่ใช้กันทั่วไป ได้แก่

1) การตรวจจับด้วยวิธีทางสถิติ เช่น Z-score และ Interquartile Range (IQR)

2) การตรวจจับด้วยแบบจำลอง เช่น Isolation Forest

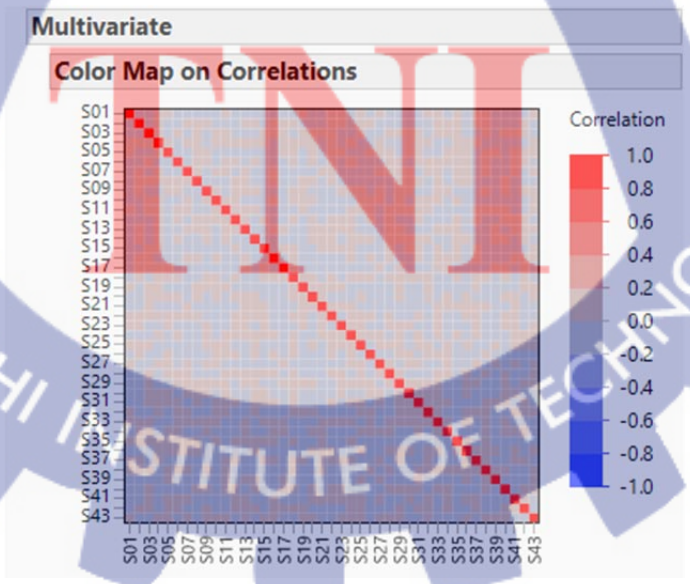
3) การปรับค่าโดยใช้ Capping/Flooring หรือการลบข้อมูลออก หากเป็นค่าที่ไม่สมเหตุสมผลทางกายภาพ

ในการดำเนินการวิเคราะห์ข้อมูลครั้งนี้ เลือกใช้แนวทางการลบค่า Outlier ที่ไม่สมเหตุสมผลออกจากชุดข้อมูล เนื่องจากค่าดังกล่าวเกิดจากความผิดพลาดในการบันทึกสัญญาณและไม่สามารถสะท้อนสภาพการทำงานจริงของเครื่องจักรได้อย่างถูกต้อง ตัวอย่างเช่น ค่าความเร็วรอบการหมุนของพัดลม (RPM) ที่ปรากฏเป็นค่าติดลบ ซึ่งไม่สามารถเกิดขึ้นได้ในทางกายภาพ ค่าลักษณะนี้จึงถูกตัดออกจากกระบวนการวิเคราะห์ข้อมูล เพื่อเพิ่มความถูกต้องและความน่าเชื่อถือของแบบจำลองที่พัฒนาขึ้นจากข้อมูลชุดนี้

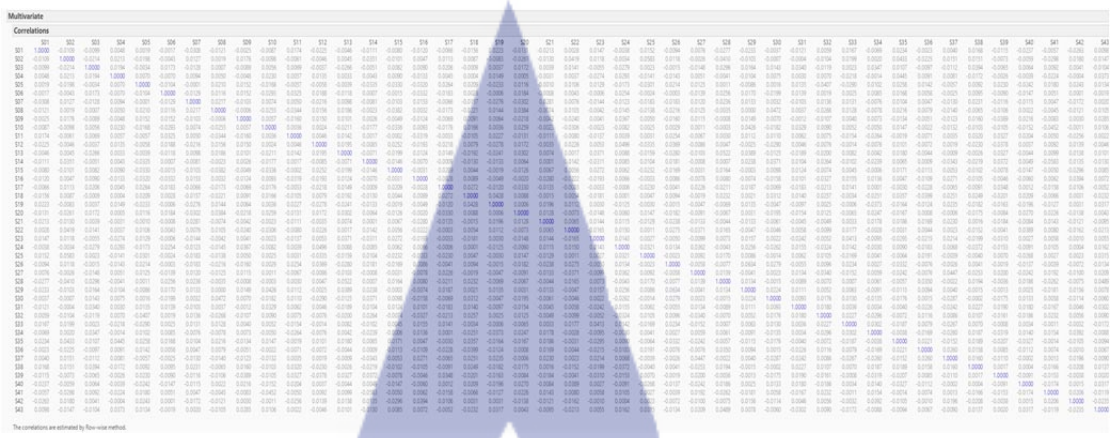
3.2.5 การวิเคราะห์ความสัมพันธ์ของข้อมูล

การวิเคราะห์ความสัมพันธ์ของข้อมูลถูกนำมาใช้เพื่อศึกษาความสัมพันธ์เชิงเส้นระหว่างตัวแปรเซ็นเซอร์ทั้งหมดในระบบ โดยมีเป้าหมายเพื่อตรวจสอบความซ้ำซ้อนของข้อมูล (Multicollinearity) และประเมินความเหมาะสมของตัวแปรก่อนนำไปสร้างแบบจำลองบำรุงรักษาเชิงคาดการณ์ การวิเคราะห์นี้ใช้ค่าสัมประสิทธิ์สหสัมพันธ์แบบเพียร์สัน (Pearson's Correlation Coefficient) ซึ่งสามารถสะท้อนระดับความสัมพันธ์ระหว่างตัวแปรเชิงตัวเลขได้อย่างชัดเจน

จากรูปที่ 3.48 และรูปที่ 3.49 พบว่าค่าความสัมพันธ์ระหว่างเซ็นเซอร์ส่วนใหญ่อยู่ในช่วงใกล้ศูนย์ โดยมีค่าประมาณระหว่าง –0.05 ถึง 0.05 แสดงให้เห็นว่าเซ็นเซอร์แต่ละตัวมีความเป็นอิสระต่อกันในระดับสูง และไม่มีความสัมพันธ์เชิงเส้นที่รุนแรงซึ่งอาจก่อให้เกิดปัญหา Multicollinearity ผลลัพธ์ดังกล่าวสะท้อนว่าแต่ละเซ็นเซอร์สามารถให้ข้อมูลเชิงพฤติกรรมของเครื่องจักรในมิติที่แตกต่างกัน ซึ่งเป็นประโยชน์ต่อการเรียนรู้ของโมเดล Machine Learning



รูปที่ 3.48 Color Map on Correlations แสดงความสัมพันธ์ระหว่างเซ็นเซอร์



รูปที่ 3.49 ตาราง Correlations แสดงความสัมพันธ์ระหว่างเซ็นเซอร์

เมื่อพิจารณาจาก Scatterplot Matrix ดังรูปที่ 3.50 จะเห็นว่าการกระจายของจุดข้อมูลระหว่างคู่ตัวแปรส่วนใหญ่มีลักษณะเป็นการกระจายแบบสุ่มและไม่ปรากฏแนวโน้มเชิงเส้นที่ชัดเจน สอดคล้องกับผลการวิเคราะห์เชิงสถิติข้างต้น ลักษณะดังกล่าวบ่งชี้ว่าความผิดปกติของเครื่องจักรมักเกิดจากความสัมพันธ์เชิงซับซ้อนและไม่เป็นเชิงเส้นของหลายเซ็นเซอร์ร่วมกัน ซึ่งเหมาะสมกับการนำโมเดลกลุ่ม Ensemble มาใช้ในการวิเคราะห์และพัฒนาระบบ Predictive Maintenance

รูปที่ 3.50 Scatterplot Matrix แสดงการกระจายของจุดข้อมูลระหว่างคู่ตัวแปร

3.2.6 การสร้างคุณลักษณะใหม่

การสร้างคุณลักษณะใหม่ (Feature Engineering) เป็นขั้นตอนสำคัญในการเพิ่มประสิทธิภาพการเรียนรู้ของแบบจำลอง Machine Learning โดยอาศัยการแปลงและสกัดสารสนเทศจากข้อมูลดิบให้สามารถสะท้อนพฤติกรรมการทำงานของเครื่องจักรได้อย่างชัดเจนยิ่งขึ้น ขั้นตอนนี้ช่วยให้แบบจำลองสามารถตรวจจับแนวโน้มการเสื่อมสภาพของอุปกรณ์ได้อย่างมีประสิทธิภาพ ในการดำเนินการมีการสร้างคุณลักษณะใหม่จากข้อมูลเซ็นเซอร์เดิม โดยสามารถแบ่งออกเป็นกลุ่มหลักดังนี้

1) ตัวชี้วัดสถิติจาก Time-Series ได้แก่ ค่าเฉลี่ย (Mean) ส่วนเบี่ยงเบนมาตรฐาน (Standard Deviation) ค่าต่ำสุด (Minimum) และค่าสูงสุด (Maximum) ซึ่งช่วยสรุปลักษณะการกระจายตัวและช่วงการเปลี่ยนแปลงของสัญญาณในแต่ละช่วงเวลา

2) ตัวชี้วัดที่สะท้อนพฤติกรรมการทำงานของเครื่องจักร เช่น Cycle Time, อัตราการเปลี่ยนแปลงของอุณหภูมิ (Temperature Gradient) และความผันผวนของความเร็วรอบการหมุน (RPM Variation) ซึ่งเป็นตัวแปรที่มีความสัมพันธ์โดยตรงกับสภาวะโหลด ความร้อนสะสม และประสิทธิภาพเชิงกลของระบบ

3) การผสานข้อมูลหลายแหล่ง (Feature Interaction) เป็นการสร้างคุณลักษณะใหม่จากความสัมพันธ์ระหว่างตัวแปรหลายชนิด เพื่อสะท้อนพฤติกรรมเชิงระบบของเครื่องจักรที่ไม่สามารถอธิบายได้ด้วยตัวแปรเดี่ยว

ผลลัพธ์ของการสกัดคุณลักษณะเชิงสถิติจากข้อมูลเซ็นเซอร์แสดงไว้ในตารางที่ 3.2 ซึ่งสรุปค่า Mean, Standard Deviation, Minimum และ Maximum ของอุณหภูมิและความเร็วรอบจากโมดูลและองค์ประกอบต่างๆ ของเครื่องทดสอบฮาร์ดดิสก์ พบว่าเซ็นเซอร์อุณหภูมิของโมดูลอิเล็กทรอนิกส์ เช่น FPGA, ADC และ Preamp มีค่าเฉลี่ยอยู่ในช่วงที่สอดคล้องกับการทำงานปกติของอุปกรณ์อิเล็กทรอนิกส์อุตสาหกรรม โดยมีส่วนเบี่ยงเบนมาตรฐานไม่สูงมาก แสดงถึงพฤติกรรมการเปลี่ยนแปลงที่ค่อนข้างเสถียร ขณะที่กลุ่มเซ็นเซอร์ความเร็วรอบพัดลม (Cooling Fan) มีค่าความแปรปรวนสูงกว่าอย่างชัดเจน ซึ่งสะท้อนถึงการปรับรอบการหมุนตามภาระความร้อนและสภาวะการทำงานของระบบในแต่ละช่วงเวลา

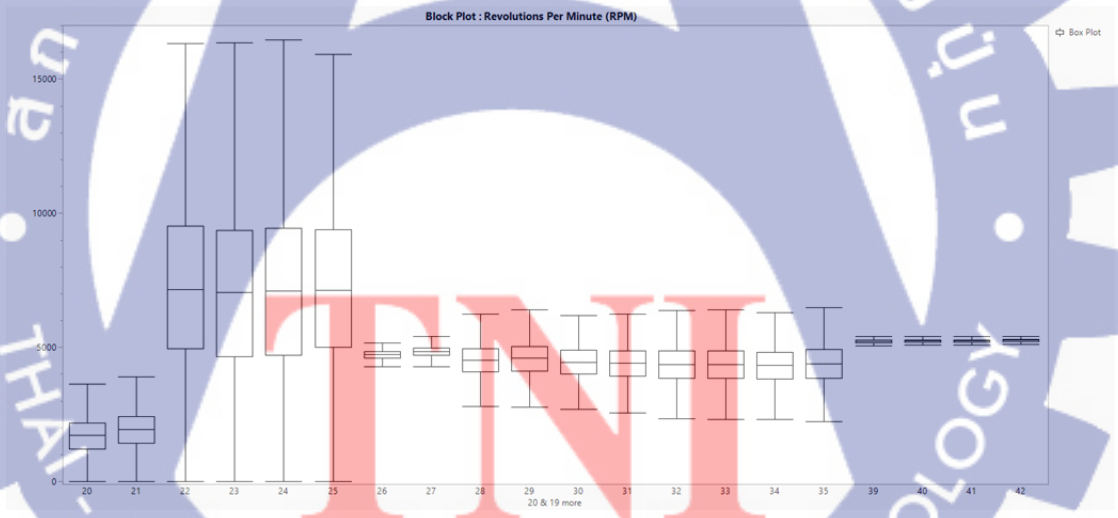
ลักษณะการกระจายของข้อมูลดังกล่าวสนับสนุนความเหมาะสมของการใช้คุณลักษณะเชิงสถิติและเชิงพฤติกรรมเป็นอินพุตของแบบจำลอง Machine Learning เนื่องจากสามารถถ่ายทอดทั้งระดับการทำงานโดยเฉลี่ยและความผันผวนของระบบ ซึ่งเป็นข้อมูลสำคัญต่อการตรวจจับความผิดปกติและการพยากรณ์ความเสื่อมสภาพของเครื่องจักรในระยะยาวดังรูปที่ 3.51 และ 3.52

ตารางที่ 3.2 คุณลักษณะเชิงสถิติที่สกัดจากข้อมูลเซ็นเซอร์แบบ Time-Series

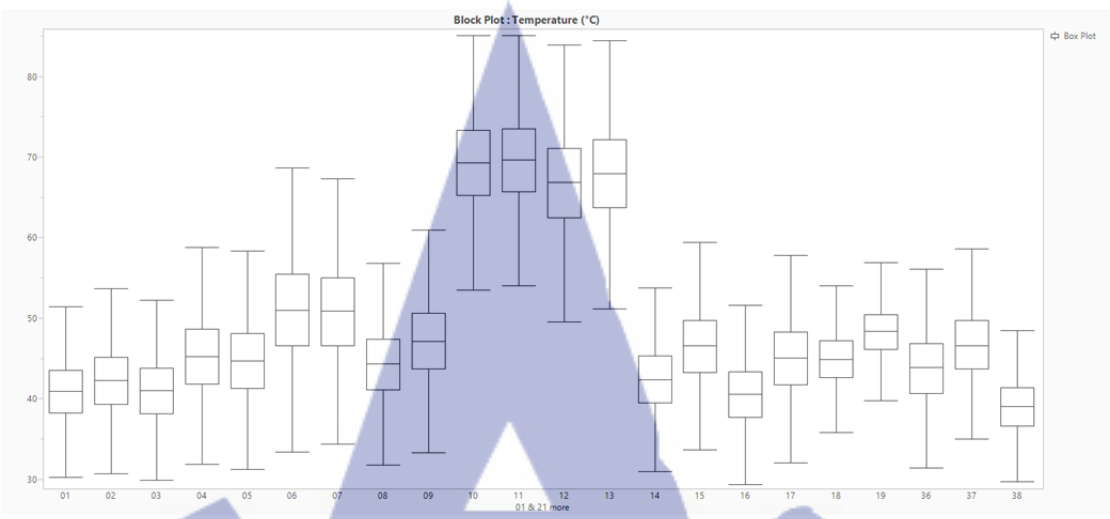
No.	Module	Component	Mean	Std Dev	Min	Max
1	CAI7000	On-board	40.9014	3.9887	27.7855	56.5584
2	CAI7000	FPGA	42.2334	4.30279	28.8157	56.2739
3	CAI7000	PRML IC	41.0083	4.2088	24.6311	55.7059
4	GSA6044AXI	Preamp #0	45.2234	5.1280	27.7624	62.9487
5	GSA6044AXI	Preamp #1	44.7347	5.0265	26.4722	66.9796
6	GSA6044AXI	Preamp #2	50.9604	6.4407	26.3147	76.0214
7	GSA6044AXI	Preamp #3	50.8790	6.07643	28.4495	73.1712
8	GSA6044AXI	ADC #0	44.2805	4.6520	27.6711	59.5611
9	GSA6044AXI	ADC #1	47.1133	5.0830	29.3242	65.8226
10	GSA6044AXI	ADC Section #0	69.3394	5.8466	43.9689	90.5660
11	GSA6044AXI	ADC Section #1	69.6226	5.7823	50.7882	88.9702
12	GSA6044AXI	PA #0	66.7658	6.3040	44.7038	87.6747
13	GSA6044AXI	PA #1	67.9209	6.1324	41.5798	89.4564
14	GSA6044AXI	AP FPGA #0	42.3782	4.2924	28.1667	57.0560
15	GSA6044AXI	AP FPGA #1	46.5071	4.8947	30.5892	63.0516
16	GSA6044AXI	AP FPGA #2	40.5068	4.1443	26.4460	55.8427
17	GSA6044AXI	AP FPGA #3	44.9570	4.8136	28.8027	61.4539
18	GSA6044AXI	HT FPGA	44.9362	3.3679	30.9403	56.4289
19	GSA6044AXI	MFM Section	48.3237	3.1417	38.3070	59.1077
20	GSA6044AXI	ADC Cooling Fan #0	1703.0093	721.2365	0	4274.4222
21	GSA6044AXI	ADC Cooling Fan #1	1935.9444	732.4346	0	4559.3464
22	GSA6044AXI	AP FPGA Fan #0	7245.8695	3342.8553	0	19977.9546
23	GSA6044AXI	AP FPGA Fan #1	7039.5154	3384.0880	0	17866.2497
24	GSA6044AXI	AP FPGA Fan #2	7113.1931	3477.3130	0	19644.1971
25	GSA6044AXI	AP FPGA Fan #3	7188.5561	3362.9076	0	18551.2953
26	GSA6044AXI	Box Fan #0	4726.6497	165.9964	4160.3081	5373.1181
27	GSA6044AXI	Box Fan #1	4851.9232	209.5678	4196.8489	5631.1460
28	GSA6044AXI	Box Fan #2	4529.5747	630.1346	2196.6188	6620.4454
29	GSA6044AXI	Box Fan #3	4590.7039	685.0368	2479.8630	7191.3301
30	GSA6044AXI	Box Fan #4	4441.7139	673.9969	1992.2485	6742.0606

ตารางที่ 3.2 คุณลักษณะเชิงสถิติที่สกัดจากข้อมูลเซ็นเซอร์แบบ Time-Series (ต่อ)

No.	Module	Component	Mean	Std Dev	Min	Max
31	GSA6044AXI	Box Fan #5	4398.2319	684.3454	2074.2899	6871.7462
32	GSA6044AXI	Box Fan #6	4369.9305	751.9510	1783.3574	7517.5523
33	GSA6044AXI	Box Fan #7	4366.5783	753.8522	1503.7781	6837.4363
34	GSA6044AXI	Box Fan #8	4313.5263	767.2768	1662.2412	7179.0764
35	GSA6044AXI	Box Fan #9	4387.9589	800.3514	1454.5352	7111.1915
36	PG6G	PG FPGA	43.8423	4.8098	24.7252	60.1545
37	PG6G	PG DAC	46.6586	4.3876	30.8029	61.4383
38	MFM4000	On-board	39.0395	3.4100	27.5291	49.2142
39	MB4000	System Fan #1	5225.8135	66.3349	4993.9819	5464.7292
40	MB4000	System Fan #2	5244.1298	63.3270	5020.8309	5473.1601
41	MB4000	System Fan #3	5244.6845	61.6962	5054.6232	5446.0716
42	MB4000	System Fan #4	5266.8912	58.8789	5071.1170	5538.6891
43	MB4000	System Fan #5	5233.4369	53.6268	5031.0822	5432.9511



รูปที่ 3.51 ลักษณะการกระจายของข้อมูลความเร็วรอบการหมุน (RPM)



รูปที่ 3.52 ลักษณะการกระจายของข้อมูลอุณหภูมิ (°C)

3.2.7 การจัดสมดุลข้อมูล

ในการตรวจจับความผิดปกติที่เกิดขึ้นจริง การแก้ไขทำได้โดยในงานตรวจจับความผิดปกติของเครื่องจักร (Pass/Fail) ข้อมูลมักมีลักษณะไม่สมดุล (Imbalanced Data) เนื่องจากเหตุการณ์ขัดข้องหรือความเสียหายของเครื่องจักรเกิดขึ้นไม่บ่อยเมื่อเทียบกับสภาวะการทำงานปกติ ส่งผลให้จำนวนตัวอย่างของกลุ่มความผิดปกติ (Minor Class) มีสัดส่วนต่ำกว่ากลุ่มปกติ (Major Class) อย่างมีนัยสำคัญ หากนำข้อมูลลักษณะนี้ไปฝึกแบบจำลองโดยไม่จัดการความไม่สมดุล โมเดลมีแนวโน้มเรียนรู้พฤติกรรมของกลุ่มปกติเป็นหลัก และอาจให้ค่าความแม่นยำโดยรวม (Accuracy) สูง แต่ไม่สามารถตรวจจับเหตุการณ์ความผิดปกติได้อย่างมีประสิทธิภาพ

การจัดสมดุลข้อมูลจึงเป็นขั้นตอนสำคัญในการลดอคติ (Bias) ของแบบจำลอง และเพิ่มความสามารถในการแยกแยะเหตุการณ์ความผิดปกติที่เกิดขึ้นจริง โดยแนวทางที่นิยมใช้สามารถแบ่งออกเป็น 3 กลุ่มหลัก ดังนี้

1) Oversampling เช่น SMOTE เพื่อเพิ่มตัวอย่างกลุ่ม Minor Class
2) Undersampling ลดจำนวนข้อมูลกลุ่ม Major Class
3) Class Weighting ปรับน้ำหนักความสำคัญในโมเดลให้กลุ่มที่พบได้น้อยมีผลต่อการเรียนรู้มากขึ้น

3.3 วิธีการแบ่งชุดข้อมูล

ชุดข้อมูลที่ใช้ในการศึกษานี้ประกอบด้วยข้อมูลทั้งหมดจำนวน 3,410 ตัวอย่าง แบ่งออกเป็นกลุ่มข้อมูลปกติ (Pass) จำนวน 2,865 ตัวอย่าง และกลุ่มข้อมูลผิดปกติ (Fail) จำนวน 545 ตัวอย่าง ซึ่งสะท้อนให้เห็นถึงปัญหาความไม่สมดุลของคลาสอย่างชัดเจน โดยข้อมูลกลุ่ม Fail มีสัดส่วนประมาณร้อยละ 16 ของข้อมูลทั้งหมด เพื่อให้การประเมินแบบจำลองมีความถูกต้องและหลีกเลี่ยงปัญหาการรั่วไหลของข้อมูล ชุดข้อมูลจึงถูกแบ่งออกเป็นชุดฝึกสอน ชุดตรวจสอบ และชุดทดสอบ ด้วยวิธี Stratified Sampling ในอัตราส่วน 80/10/10 โดยรักษาสัดส่วนของข้อมูลในแต่ละคลาสให้ใกล้เคียงกับโครงสร้างข้อมูลดั้งเดิม

จากการแบ่งชุดข้อมูล พบว่าชุดฝึกสอนมีจำนวน 2,728 ตัวอย่าง (Pass 2,292 ตัวอย่าง และ Fail 436 ตัวอย่าง) ชุดทดสอบมีจำนวน 342 ตัวอย่าง (Pass 287 ตัวอย่าง และ Fail 55 ตัวอย่าง) และชุดตรวจสอบมีจำนวน 340 ตัวอย่าง (Pass 286 ตัวอย่าง และ Fail 54 ตัวอย่าง) ซึ่งทั้งสามชุดยังคงสะท้อนลักษณะความไม่สมดุลของข้อมูลตามสภาพจริงของระบบ หลังจากนั้นจึงทำการแก้ไขปัญหาความไม่สมดุลของคลาสเฉพาะในชุดฝึกสอน โดยประยุกต์ใช้เทคนิค Synthetic Minority Over-Sampling Technique (SMOTE) เพื่อเพิ่มจำนวนข้อมูลในกลุ่ม Fail ให้มีความสมดุลกับกลุ่ม Pass และช่วยให้แบบจำลองสามารถเรียนรู้ลักษณะของข้อมูลกลุ่มส่วนน้อยได้อย่างมีประสิทธิภาพมากขึ้น

ก่อนนำข้อมูลไปใช้ในการสร้างแบบจำลอง ตัวแปรเชิงตัวเลขทั้งหมดจำนวน 43 ตัวแปรอิสระได้รับการปรับมาตรฐานด้วยเทคนิค Standardization เพื่อแปลงค่าให้อยู่ในรูปที่มีค่าเฉลี่ยเป็นศูนย์และส่วนเบี่ยงเบนมาตรฐานเป็นหนึ่ง ซึ่งช่วยลดผลกระทบจากความแตกต่างของช่วงค่าของตัวแปร และส่งเสริมให้กระบวนการเรียนรู้ของแบบจำลองมีความเสถียรและมีประสิทธิภาพมากยิ่งขึ้น โดยสรุปจำนวนข้อมูลในแต่ละคลาสหลังการแบ่งชุดข้อมูลแสดงไว้ในตารางที่ 3.3

ตารางที่ 3.3 จำนวนตัวอย่างข้อมูลในแต่ละคลาสของชุด Training, Validation และ Test

ชุดข้อมูล	จำนวนรวม	OK	Fail
Training Set (80%)	2,728	2,292	436
Validation Set (10%)	342	286	54
Test Set (10%)	340	287	55
รวม	3,410	2,865	545

3.4 อัลกอริทึมที่ใช้ในการวิจัย

1. XGBoost เป็นอัลกอริทึม Gradient Boosting ที่มีประสิทธิภาพสูงในการเรียนรู้ความสัมพันธ์ที่ไม่เชิงเส้นจากข้อมูลเซ็นเซอร์จำนวนมาก เหมาะสำหรับงาน PdM เนื่องจากสามารถจัดการข้อมูลที่มี Noise และความไม่สมดุลของคลาสได้ดี พร้อมทั้งให้ผลการทำนายที่มีความแม่นยำและเสถียร

2. LightGBM ใช้โครงสร้างต้นไม้แบบ Leaf-Wise ซึ่งช่วยเพิ่มความเร็วในการฝึกสอนและรองรับข้อมูลขนาดใหญ่ เหมาะสำหรับงาน PdM ที่ต้องประมวลผลข้อมูล Time-Series จากเซ็นเซอร์จำนวนมากภายใต้ข้อจำกัดด้านเวลาและทรัพยากรคำนวณ

3. AdaBoost ปรับน้ำหนักของข้อมูลโดยเน้นตัวอย่างที่ทำนายผิดในรอบก่อนหน้า จึงเหมาะสำหรับงาน PdM ที่เหตุการณ์ความล้มเหลวเกิดขึ้นไม่บ่อย และต้องการเพิ่มความสามารถในการตรวจจับสัญญาณผิดปกติของเครื่องจักร

4. CatBoost ถูกออกแบบมาเพื่อลด Bias จากการเข้ารหัสตัวแปรเชิงกลุ่ม และให้ผลลัพธ์ที่มีเสถียรภาพสูง เหมาะสำหรับงาน PdM ที่มีข้อมูลจากหลายโมดูลหรือหลายสภาวะการทำงานซึ่งประกอบด้วยตัวแปรเชิงกลุ่มร่วมกับข้อมูลเชิงตัวเลข

5. GBT เป็นอัลกอริทึม Boosting แบบดั้งเดิมที่ใช้เป็นแบบจำลองอ้างอิงเพื่อเปรียบเทียบกับวิธีการที่พัฒนาขึ้นใหม่ เหมาะสำหรับงาน PdM ในการประเมินประสิทธิภาพพื้นฐานของการพยากรณ์ความเสื่อมสภาพของเครื่องจักร

6. Bagging ช่วยลดความแปรปรวนของแบบจำลองผ่านการสุ่มตัวอย่างข้อมูลหลายชุดและรวมผลลัพธ์เข้าด้วยกัน เหมาะสำหรับงาน PdM ที่ต้องการความเสถียรในการทำนายภายใต้สภาวะการทำงานของเครื่องจักรที่มีความผันผวนสูง

3.5 การปรับค่า Hyperparameter

การปรับค่า Hyperparameter เป็นขั้นตอนสำคัญในการพัฒนาแบบจำลอง Machine Learning เนื่องจากค่าพารามิเตอร์ควบคุมเหล่านี้มีผลโดยตรงต่อความสามารถในการเรียนรู้ รูปแบบการสร้างแบบจำลองและประสิทธิภาพในการทำนายของอัลกอริทึมแต่ละชนิด Hyperparameter แตกต่างจากพารามิเตอร์ที่เรียนรู้จากข้อมูลโดยตรงตรงที่ต้องถูกกำหนดล่วงหน้าก่อนกระบวนการฝึกสอนแบบจำลอง การเลือกค่าที่เหมาะสมจึงมีบทบาทสำคัญต่อการลดปัญหา Overfitting และ Underfitting รวมถึงช่วยเพิ่มความสามารถในการสรุปผล (Generalization) ของแบบจำลองเมื่อนำไปใช้งานกับข้อมูลใหม่

ในการศึกษานี้ มีการกำหนดค่า Hyperparameter สำหรับแบบจำลองในกลุ่ม Tree-Based และ Ensemble Learning หลายอัลกอริทึม ได้แก่ LightGBM, XGBoost, CatBoost, AdaBoost, Gradient Boosting Tree (GBT) และ Bagging โดยค่าพารามิเตอร์ถูกเลือกจากแนวทางที่ได้รับการยอมรับในงานวิจัยที่เกี่ยวข้องและคำแนะนำจากเอกสารอ้างอิงของอัลกอริทึมแต่ละประเภท ทั้งนี้ ตาราง

ที่ 3.3 แสดงรายละเอียดของ Hyperparameter ที่ใช้ในการฝึกสอนแบบจำลอง ซึ่งครอบคลุมพารามิเตอร์ที่ควบคุมความลึกของต้นไม้ อัตราการเรียนรู้ จำนวนรอบการ Boosting และระดับของการสุ่มตัวอย่างข้อมูลและคุณลักษณะ

การกำหนดค่า Hyperparameter ดังกล่าวมีเป้าหมายเพื่อสร้างสมดุลระหว่างความซับซ้อนของแบบจำลองและประสิทธิภาพในการเรียนรู้จากข้อมูลจริง โดยอัลกอริทึมในกลุ่ม Boosting ถูกตั้งค่าให้มีจำนวนรอบการเรียนรู้และอัตราการเรียนรู้ที่เหมาะสมต่อการจับรูปแบบความผิดปกติของเครื่องจักร ขณะที่อัลกอริทึมในกลุ่ม Bagging ถูกใช้เพื่อเพิ่มความเสถียรและลดความแปรปรวนของผลลัพธ์ การตั้งค่าเหล่านี้จึงเป็นพื้นฐานสำคัญสำหรับการเปรียบเทียบประสิทธิภาพของแบบจำลองต่างๆ ซึ่งผลลัพธ์จะถูกนำเสนอและวิเคราะห์ในบทถัดไป

ตารางที่ 3.3 การกำหนดค่าพารามิเตอร์ควบคุม (Hyperparameter)

Algorithm	Hyperparameter	Value
LightGBM	Learning Rate	0.1
	Number of Leaves	50
	Minimum Child Samples	20
	Maximum Tree Depth	-1
	Number of Estimators	100
XGBoost	Number of Boosting Rounds	100
	Maximum Tree Depth	6
	Step Size Shrinkage (eta)	0.3
	Subsample Ratio of Training Data	1
	Subsample Ratio of Columns	1
CatBoost	Number of Boosting Iterations	1000
	Depth of Trees	6
	Learning Rate	0.03
	L2 Regularization Coefficient	3
AdaBoost	Number of Estimators	50
	Learning Rate	1
	Base Estimator	1

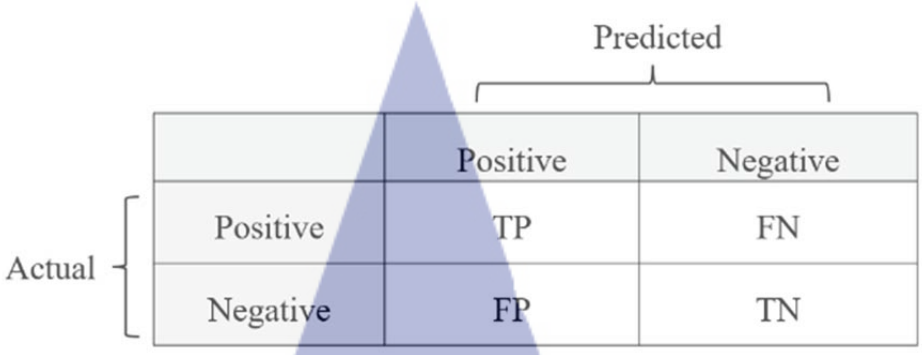
ตารางที่ 3.3 การกำหนดค่าพารามิเตอร์ควบคุม (Hyperparameter) (ต่อ)

Algorithm	Hyperparameter	Value
GBT	Number of Boosting Stages to Perform	100
	Maximum Depth of the Individual Regression Estimators	3
	Shrinks the Contribution of Each Tree by this Rate	0.1
	The Fraction of Samples to be Used for Fitting the Individual Base Learners	1.0
Bagging	Number of Base Estimators	10
	Maximum Samples	1.0
	Maximum Features	1.0

3.6 ตัวชี้วัดประสิทธิภาพ

Confusion Matrix เป็นเครื่องมือที่นิยมใช้กันอย่างแพร่หลายในการประเมินประสิทธิภาพของโมเดลจำแนกประเภท โดยแสดงการเปรียบเทียบระหว่างป้ายกำกับชั้นจริง (Actual Class Labels) กับผลลัพธ์ที่โมเดลทำนาย (Predicted Outputs) ซึ่งช่วยให้เห็นภาพรวมทั้งความแม่นยำและลักษณะของข้อผิดพลาดในการจำแนกประเภทได้อย่างชัดเจน รูปที่ 3.2 แสดงตัวอย่าง Confusion Matrix ที่ใช้ในการประเมินโมเดลจำแนกประเภทในงานวิจัยนี้ โดยประกอบด้วย 4 ส่วนหลัก ดังนี้

1. True Positive (TP): กรณีที่โมเดลทำนายว่า “Fail” ได้ถูกต้อง เช่น เซนเซอร์ที่ไม่ผ่านมาตรฐานถูกจัดประเภทอย่างถูกต้องเป็น Fail
2. True Negative (TN): กรณีที่โมเดลทำนายว่า “OK” ได้ถูกต้อง เช่น เซนเซอร์ที่ผ่านมาตรฐานถูกจัดประเภทอย่างถูกต้องเป็น OK
3. False Positive (FP): กรณีที่โมเดลทำนายว่า “OK” ผิดพลาด เช่น เซนเซอร์ที่ไม่ผ่านมาตรฐานถูกทำนายผิดเป็น Fail
4. False Negative (FN): กรณีที่โมเดลทำนายว่า “Fail” ผิดพลาด เช่น เซนเซอร์ที่ผ่านมาตรฐานถูกทำนายผิดเป็น OK



รูปที่ 3.53 หลักการทำงานของ Confusion Matrix

การประเมินประสิทธิภาพของโมเดลการจำแนกที่นำเสนออย่างรอบด้าน ได้มีการคำนวณตัวชี้วัดที่ได้รับการยอมรับอย่างแพร่หลายจากตารางสับสน (Confusion Matrix) ตามที่แสดงในรูปที่ 3.2 ตัวชี้วัดเหล่านี้ช่วยสะท้อนคุณภาพของการทำนายจากมุมมองที่หลากหลาย และมีความสำคัญเป็นพิเศษในกรณีที่ข้อมูลไม่สมดุล ซึ่งการพึ่งพาเพียงค่า Accuracy เพียงอย่างเดียวอาจนำไปสู่การตีความที่คลาดเคลื่อนได้ ดังนั้นตัวชี้วัดที่ใช้ในการประเมิน มีดังนี้

1. Accuracy วัดความถี่ที่โมเดลทำนายถูกต้อง โดยคำนวณจากจำนวนการทำนายที่ถูกต้องหารด้วยจำนวนตัวอย่างทั้งหมด

2. Precision วัดสัดส่วนของกรณี “Fail” ที่โมเดลทำนายได้ถูกต้อง เมื่อเทียบกับจำนวนกรณีที่โมเดลทำนายว่าเป็น “Fail” ทั้งหมด ซึ่งสะท้อนถึงความแม่นยำเฉพาะจุด (Exactness) ของตัวจำแนก

3. Recall (Sensitivity หรือ True Positive Rate) ประเมินความสามารถของโมเดลในการระบุกรณีที่เป็นบวก (Positive Cases) ได้อย่างถูกต้องทั้งหมด โดยให้ความสำคัญกับความครบถ้วน (Completeness) ของการตรวจจับ

4. F1-Score แสดงค่าเฉลี่ยฮาร์มอนิกระหว่างค่า Precision และ Recall โดยเป็นตัวชี้วัดเดียวที่รวมข้อได้เปรียบของทั้งสองตัวเข้าด้วยกัน เพื่อสะท้อนถึงความสมดุลระหว่างความผิดพลาดจากการจำแนกทั้งสองประเภท

5. Area Under the ROC Curve (AUC) วัดความสามารถของตัวจำแนกในการแยกคลาสบวกและคลาสลบได้ดีเพียงใดที่ค่า threshold การตัดสินใจต่างๆ ตัวชี้วัดนี้มีประโยชน์เป็นพิเศษสำหรับชุดข้อมูลที่ไม่สมดุล เนื่องจากสามารถสะท้อนความสัมพันธ์ระหว่าง ความไว (Sensitivity) และความจำเพาะ (Specificity)

โดยรวมแล้ว ตัวชี้วัดเหล่านี้ช่วยให้สามารถประเมินประสิทธิภาพและความมั่นคงของตัวจำแนกได้อย่างรอบด้าน สนับสนุนการตัดสินใจอย่างมีข้อมูลพื้นฐานเกี่ยวกับการนำโมเดลไปใช้งานจริง และการปรับปรุงประสิทธิภาพของโมเดลให้เหมาะสม

3.7 เครื่องมือและซอฟต์แวร์ที่ใช้ในงานวิจัย

รายละเอียดของฮาร์ดแวร์และซอฟต์แวร์ที่ใช้ในการทดลองแสดงไว้ในตารางที่ 3.4 ซึ่งถูกกำหนดให้เหมือนกันทุกการทดลองเพื่อให้การเปรียบเทียบผลลัพธ์ของแบบจำลองมีความสอดคล้องและเป็นธรรม

ตารางที่ 3.4 สรุปเครื่องมือและซอฟต์แวร์ที่ใช้ในการวิจัย

หมวดหมู่	รายการ	รายละเอียด
Hardware	หน่วยประมวลผลกลาง (CPU)	Intel Core i3-1115G4 ความเร็ว 3.00 GHz
	หน่วยความจำหลัก (RAM)	8.00 GB
	สถาปัตยกรรมระบบ	64-bit
Software	ระบบปฏิบัติการ	Windows 10 (64-bit)
	ภาษาโปรแกรม	Python Version 3.10
	โปรแกรมที่ใช้การพัฒนา	Jupyter Notebook
	ไลบรารีจัดการข้อมูล	NumPy, Pandas
	ไลบรารี	Machine Learning, Scikit-learn, Ensemble (Boosting, XGBoost, LightGBM, CatBoost)

บทที่ 4
ผลการวิจัย

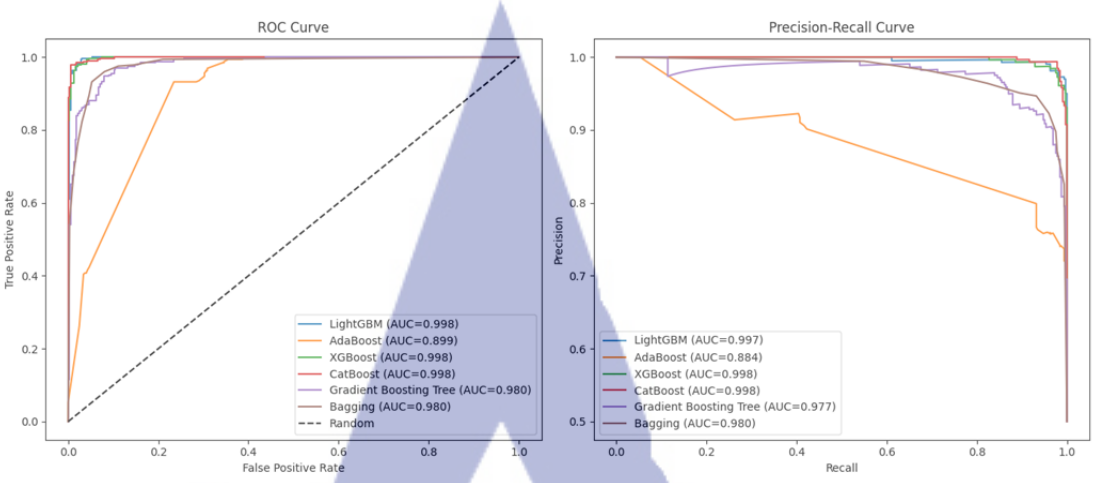
บทนี้นำเสนอผลการเปรียบเทียบประสิทธิภาพของแบบจำลอง Machine Learning ที่ใช้สำหรับการพยากรณ์ความผิดปกติของเครื่องจักรในงาน Predictive Maintenance (PdM) โดยผลลัพธ์เชิงตัวเลขแสดงในตารางที่ 4.1 และผลการวิเคราะห์เชิงกราฟแสดงผ่าน ROC Curve และ Precision–Recall Curve แสดงในรูปที่ 4.1

ตารางที่ **ผิดพลาด! ไม่มีข้อความของลักษณะที่ระบุในเอกสาร**.1 Confusion Matrix แต่ละเทคนิค

Model	True Negative	False Positive	False Negative	True Positive
CatBoost	320	4	7	317
XGBoost	319	5	9	315
LightGBM	318	6	6	318
GBT	296	28	17	307
Bagging	311	13	30	294
AdaBoost	226	98	17	307

ตารางที่ 4.2 ผลการประเมินแต่ละเทคนิค

Model	Accuracy	Precision	Recall	F1-Score	AUC
CatBoost	**0.9830**	**0.9875**	0.9784	**0.9829**	**0.9985**
XGBoost	0.9784	0.9844	0.9722	0.9783	0.9979
LightGBM	0.9815	0.9815	**0.9815**	0.9815	0.9976
GBT	0.9306	0.9164	0.9475	0.9317	0.9804
Bagging	0.9290	0.9603	0.8951	0.9265	0.9785
AdaBoost	0.8225	0.7580	**0.9475**	0.8422	0.8992



รูปที่ 4.1 ผลการวิเคราะห์เชิงกราฟแสดงผ่าน ROC Curve และ Precision–Recall Curve

4.1 ผลการเปรียบเทียบประสิทธิภาพของแบบจำลอง

จากผลในตารางที่ 4.2 พบว่าแบบจำลอง CatBoost ให้ประสิทธิภาพโดยรวมดีที่สุด โดยมีค่า Accuracy, Precision, F1-score และ AUC สูงที่สุดเมื่อเทียบกับแบบจำลองอื่น แสดงให้เห็นถึงความสามารถในการจำแนกข้อมูลปกติและข้อมูลผิดปกติได้อย่างถูกต้องและมีเสถียรภาพ รองลงมาคือ LightGBM และ XGBoost ซึ่งให้ผลลัพธ์ใกล้เคียงกัน โดย LightGBM มีค่า Recall สูง แสดงถึงความสามารถในการตรวจจับความผิดปกติได้อย่างครอบคลุม ขณะที่ XGBoost ให้สมดุลของตัวชี้วัดในระดับสูง

ในทางกลับกัน แบบจำลอง GBT และ Bagging มีประสิทธิภาพลดลงอย่างเห็นได้ชัด โดยเฉพาะค่า Accuracy และ F1-score ส่วน AdaBoost แม้จะสามารถตรวจจับความผิดปกติได้ดี (Recall สูง) แต่มีค่า Precision ต่ำ ซึ่งหมายถึงการแจ้งเตือนผิดพลาดจำนวนมาก ส่งผลให้ไม่เหมาะสมต่อการนำไปใช้งานจริงในระบบ PdM ที่ต้องการลดการแจ้งเตือนว่ามีความผิดปกติหรือความล้มเหลว ทั้งที่ความจริงเครื่องจักรยังอยู่ในสภาวะปกติ

4.2 การวิเคราะห์ผลลัพธ์เชิงกราฟสำหรับงาน Predictive Maintenance

จากกราฟ ROC Curve ในรูปที่ 4.1 พบว่า CatBoost, XGBoost และ LightGBM มีเส้นโค้งอยู่ใกล้มุมซ้ายบนของกราฟมากที่สุด และมีค่า AUC สูงกว่า 0.99 ซึ่งสอดคล้องกับผลการประเมินเชิงตัวเลขในตารางที่ 4.2 แสดงให้เห็นว่าโมเดลกลุ่มนี้สามารถแยกแยะสถานการณ์ทำงานปกติและความผิดปกติของเครื่องจักรได้อย่างมีประสิทธิภาพในทุกระดับของค่า threshold นอกจากนี้ เมื่อพิจารณากราฟ Precision-Recall Curve ซึ่งเหมาะสมกับข้อมูลที่มีความไม่สมดุลของคลาส พบว่า CatBoost และ XGBoost สามารถรักษาค่า Precision ในระดับสูงตลอดช่วง Recall ส่วนใหญ่ สะท้อนถึงความสามารถ

ในการตรวจจับความผิดปกติโดยไม่ก่อให้เกิดการแจ้งเตือนผิดพลาดในระดับสูง ซึ่งเป็นคุณสมบัติที่มีความสำคัญต่อการนำระบบ Predictive Maintenance ไปใช้งานจริงในภาคอุตสาหกรรม

4.3 สรุปผลการทดลองและการอภิปรายผล

จากผลการทดลองทั้งเชิงตัวเลขและเชิงกราฟ สามารถสรุปได้ว่า CatBoost เป็นแบบจำลองที่เหมาะสมที่สุดสำหรับงาน Predictive Maintenance ในงานวิจัยนี้ เนื่องจากสามารถตรวจจับความผิดปกติได้แม่นยำ มีอัตราการแจ้งเตือนผิดพลาดต่ำ และมีความเสถียรสูง รองลงมาคือ LightGBM และ XGBoost ซึ่งยังคงเป็นทางเลือกที่มีประสิทธิภาพสำหรับการนำไปประยุกต์ใช้งานจริงในระบบเฝ้าระวังสภาพเครื่องจักร

บทที่ 5
สรุปและอภิปรายผล

5.1 สรุปและอภิปรายผล

ผลการทดลองแสดงให้เห็นว่าแบบจำลองการเรียนรู้ของเครื่องสามารถนำมาใช้ในการทำนายความผิดปกติของเครื่องจักรในงาน Predictive Maintenance ได้อย่างมีประสิทธิภาพ โดยใช้ข้อมูลเซ็นเซอร์เชิงเวลาจากระบบทดสอบฮาร์ดดิสก์ซึ่งมีลักษณะข้อมูลไม่สมดุลระหว่างคลาส หลังจากดำเนินการเตรียมข้อมูล ได้แก่ การทำความสะอาดข้อมูล การจัดการค่าที่สูญหายและค่าผิดปกติ การสร้างคุณลักษณะใหม่ และการจัดสมดุลข้อมูลด้วยเทคนิค SMOTE แล้ว ข้อมูลถูกนำไปใช้ในการฝึกและประเมินแบบจำลอง

ผลการเปรียบเทียบแบบจำลองทั้ง 6 วิธี ได้แก่ CatBoost, XGBoost, LightGBM, Gradient Boosting Tree, Bagging และ AdaBoost พบว่าแบบจำลองในกลุ่ม Gradient Boosting ให้ประสิทธิภาพโดยรวมสูงกว่าแบบจำลองอื่น โดยเฉพาะ CatBoost ซึ่งให้ค่า Accuracy, F1-score และ AUC สูงที่สุด และมีความสมดุลระหว่างค่า Precision และ Recall ที่เหมาะสม ส่งผลให้สามารถตรวจจับความผิดปกติของเครื่องจักรได้อย่างแม่นยำ พร้อมลดอัตราการแจ้งเตือนผิดพลาด

ผลการวิเคราะห์ด้วย ROC Curve และ Precision–Recall Curve สนับสนุนผลเชิงตัวเลข โดยแสดงให้เห็นว่า CatBoost มีความสามารถในการแยกแยะสถานะการทำงานปกติและผิดปกติได้ดีในทุกช่วงของค่า Threshold และมีค่า False Negative ต่ำ ซึ่งเป็นปัจจัยสำคัญต่อระบบ Predictive Maintenance เนื่องจากการพลาดการตรวจจับความเสียหายอาจส่งผลให้เกิดการหยุดชะงักของกระบวนการผลิตและความเสียหายทางเศรษฐกิจ

5.2 ข้อจำกัดในการวิจัย

ข้อมูลที่ใช้ในการวิจัยนี้ได้มาจากระบบเครื่องทดสอบฮาร์ดดิสก์ที่ ครอบคลุมเฉพาะการใช้งานของระบบ Guzik Signal Analyzer (GSA) 6000 Series และ Read-Write Analyzer Systems 4000 TDMR Series เท่านั้น และเป็นข้อมูลที่เก็บรวบรวมจากช่วงเวลาการทำงานที่จำกัด ส่งผลให้รูปแบบข้อมูลและลักษณะความผิดปกติที่แบบจำลองได้เรียนรู้อาจสะท้อนพฤติกรรมการทำงานได้ดีเฉพาะในขอบเขตของระบบดังกล่าว แต่ยังไม่สามารถรับรองการประยุกต์ใช้งานกับเครื่องทดสอบหรือระบบที่มีสถาปัตยกรรมฮาร์ดแวร์ โครงสร้างสัญญาณ และกระบวนการทำงานที่แตกต่างออกไปได้อย่างครอบคลุม

นอกจากนี้ การจัดการปัญหาข้อมูลไม่สมดุลโดยใช้เทคนิค SMOTE แม้ว่าจะช่วยเพิ่มจำนวนตัวอย่างในกลุ่มความล้มเหลวและปรับปรุงประสิทธิภาพของแบบจำลองในเชิงสถิติ แต่ในบางกรณีอาจก่อให้เกิดข้อมูลสังเคราะห์ที่ไม่สอดคล้องกับพฤติกรรมการทำงานจริงของเครื่องทดสอบ โดยเฉพาะความ

ผิดปกติที่มีลักษณะเฉพาะเชิงกายภาพหรือเชิงกระบวนการ ซึ่งอาจส่งผลให้การนำแบบจำลองไปใช้งานกับข้อมูลจริงนอกขอบเขตของระบบ GSA 6000 Series และ 4000 TDMR Series มีข้อจำกัด

5.3 ข้อเสนอแนะในการวิจัย

งานวิจัยในอนาคตควรพิจารณาการจัดกลุ่มข้อมูลเซนเซอร์ (Sensor Grouping) ตามโครงสร้างและหน้าที่ของโมดูลภายในเครื่องทดสอบฮาร์ดดิสก์ เช่น กลุ่มเซนเซอร์ของระบบ Preamp, ADC, FPGA และระบบระบายความร้อน เพื่อสะท้อนความสัมพันธ์เชิงกายภาพของอุปกรณ์และลดความซับซ้อนของข้อมูลเชิงเวลา การจัดกลุ่มเซนเซอร์ดังกล่าวจะช่วยให้สามารถระบุแหล่งที่มาของความผิดปกติในระดับโมดูล (Module-Level Fault Identification) ได้อย่างแม่นยำมากยิ่งขึ้น

นอกจากนี้ งานวิจัยในอนาคตอาจศึกษาการประยุกต์ใช้เทคนิคการจัดการข้อมูลไม่สมดุลเพิ่มเติม เช่น Cost-Sensitive Learning หรือ Adaptive Thresholding ร่วมกับข้อมูลที่ถูกจัดกลุ่มตามโมดูล เพื่อเพิ่มความสามารถของแบบจำลองในการตรวจจับเหตุการณ์ความล้มเหลวที่เกิดขึ้นไม่บ่อย

อีกทั้งสามารถต่อยอดโดยการประยุกต์ใช้แบบจำลองเชิงลึก (Deep Learning) ที่สามารถเรียนรู้รูปแบบเชิงเวลาโดยตรงจากข้อมูลเซนเซอร์ที่จัดกลุ่มแล้ว เช่น Long Short-Term Memory (LSTM) หรือ Transformer Model เพื่อเพิ่มประสิทธิภาพในการพยากรณ์ความผิดปกติของเครื่องทดสอบในระยะยาว และสนับสนุนการบำรุงรักษาเชิงพยากรณ์ในระดับโมดูลได้อย่างมีประสิทธิภาพยิ่งขึ้น


สถาบันเทคโนโลยีไทย-ญี่ปุ่น
THAI - NICHI INSTITUTE OF TECHNOLOGY
TNI

บรรณานุกรม

บรรณานุกรม

[1] E. E. Kriezis, "Index IEEE transactions on magnetics," *IEEE Transactions on Magnetics,* vol. 33, no. 6, pp. 4,575-4,704, November 1997.

[2] Z. S. Shan et al., "Energy barrier and magnetic properties of exchange-coupled hard-soft bilayer," *IEEE Transactions on Magnetics,* vol. 38, no. 5, pp. 2,907-2,909, September 2002.

[3] G. Ababei, "Index IEEE Transactions on Magnetics," *IEEE Transactions on Magnetics,* vol. 45, no. 12, pp. 5,417-5,560, December 2009.

[4] F. Cannillo, "Index proceedings of the IEEE," *Proceedings of the IEEE,* vol. 97, no. 12, pp. 2,113-2,175, December 2009.

[5] อาชนัน เกาะไพบูลย์, "เครือข่ายการผลิตระหว่างประเทศในอุตสาหกรรมฮาร์ดดิสก์ของไทย : นัยต่อการพัฒนาอุตสาหกรรม," *การประชุมวิชาการข่ายงานวิศวกรรมอุตสาหการ 2553, IE Network 2010,* กรุงเทพมหานคร, 31 มกราคม 2553, หน้า 1–54.

[6] Q. Cheng et al., "Direct measurement of disk-to-head back-heating in HAMR using a non-flying test stage," *Applied Physics Letters,* vol. 120, no. 24, p. 241,602, June 2022.

[7] T. Coughlin and R. Hoyt, "IEEE roadmap outlines development of mass digital storage technology," *IEEE Transactions on Magnetics,* vol. 57, no. 9, pp. 111-116, September 2024.

[8] K. S. Chan and M. R. Elidrissi, "A system level study of two-dimensional magnetic recording (TDMR)," *IEEE Transactions on Magnetics,* vol. 49, no. 6, pp. 2,812-2,817, June 2013.

[9] R. H. Wang et al., "Head protrusion and its implications on head-disk interface reliability," *IEEE Transactions on Magnetics,* vol. 37, no. 4, pp. 1,842-1,844, July 2001.

[10] N. Normadhi et al., "Challenges of the head-disk interface for near contact and contact recording," *IEEE Transactions on Magnetics,* vol. 35, no. 5, pp. 2,466-2,468, September 1999.

[11] B. D. Strom et al., "Hard disk drive reliability modeling and failure prediction," *Proceedings of the 2004 Northeast Decision Science Institute Conference, NEDSI 2004 Proceedings,* New Jersey, USA, November 2006, pp. 1-2.

[12] T. Coughlin and R. F. Hoyt, *"IRDS Mass Data Storage,"* [Online]. Available : https://irds.ieee.org/images/files/pdf/2023/2023IRDS_MDS.pdf. [Accessed : August 16, 2025].

[13] Seagate Technology LLC, *"Optimizing Manufacturing Using AI,"* [Online]. Available : https://www.seagate.com/content/dam/seagate/migrated-assets/www-content/solutions/high-performance-computing/_shared/files/using-ai-to-boost-factory-efficiency.pdf. [Accessed : September 7, 2025].

[14] T. Wanglomklang et al., "Using fault detection and classification techniques for machine breakdown reduction of the HGA process caused by the slider loss defect," *Manufacturing Review,* vol. 9, no. 21, pp. 256-259, August 2022.

[15] K. Mori and Y. Matsui, "Manufacturing equipment technologies for hard disk's challenge of physical limits," *Fujitsu Scientific and Technical Journal,* vol. 42, no. 1, pp. 148-154, January 2006.

[16] Y. Yu et al., "Mechanism of u-HAST failure caused by flux residue containing bromide," *IEEE Electronics Packaging Technology Conference, EPTC 2020 Proceedings,* New Jersey, USA, December 8-12, 2020, pp. 44-48.

[17] C. Zorn and N. Kaminski, "Temperature–humidity–bias testing on insulated-gate bipolar transistor modules-failure modes and acceleration due to high voltage," *IET Power Electronics,* vol. 8, no. 12, pp. 2,329-2,335, December 2015.

[18] C. Zorn and N. Kaminski, "Acceleration of temperature humidity bias (THB) testing on IGBT modules by high bias levels," *2015 IEEE 27th International Symposium on Power Semiconductor Devices & ICs, ISPSD 2015 Proceedings,* Hong Kong, China, May 18-20, 2015, pp. 385-388.

[19] L. Decker et al., "Real-time anomaly detection in data centers for log-based predictive maintenance using an evolving fuzzy-rule-based approach," *International Conference on Powder Metallurgy and Particulate Materials,* PM2TEC 2020, Las Vegas, USA, July 28, 2020, pp. 1-8.

[20] Q. Kang et al., "Reliability research of a MEMS flow sensor with accelerated degradation testing," *International Conference on System Reliability and Safety, ICSRS 2023,* Bologna, Italy, November 6-9, 2023, pp. 88-93.

[21] V. Madonna et al., "Introducing physics of failure considerations in the electrical machines design," *IEEE International Electric Machines & Drives Conference, IEMDC,* San Diego, USA, May 10-13, 2019, pp. 2,233-2,238.

[22] A. Zineb et al., "Impact of improving machines availability using stochastic petri nets on overall equipment effectiveness," *Maintenance Management,* vol. 34, no. 3, pp. 267-274, June 2020.

[23] P. Alavian et al., "The precise estimates of MTBF and MTTR : Definition calculation and observation time," *IEEE Transactions on Automation Science and Engineering,* vol. 18, no. 3, pp. 1,469-1,477, July 2021.

[24] H. S. Solari et al., "Optimal estimation of weibull distribution parameters in order to provide preventive-corrective maintenance program for power transformers," *Journal of Quality in Maintenance Engineering,* vol. 27, no. 4, pp. 145-150, April 2021.

[25] K. Aggarwal et al., "Two birds with one network : Unifying failure event prediction and time-to-failure modeling," *IEEE International Conference on Big Data, BDCAT,* Seattle, USA, December 12-15, 2018, pp. 1,308-1,317.

[26] A. D. Santo et al., "Deep learning for HDD health assessment : An application based on LSTM," *IEEE Transactions on Computers,* vol. 71, no. 1, pp. 69-80, January 2022.

[27] J. Ircio et al., "A multivariate time series streaming classifier for predicting hard drive failures," *IEEE Computational Intelligence Magazine,* vol. 17, no. 1, pp. 102-114, February 2022.

[28] F. Amato et al., "A comparative assessment of explainable AI tools in predicting hard disk drive health," *Symposium on Advanced Database Systems,* vol. 9, no. 3, pp. 1-15, June 2021.

[29] J. L. Gomez et al., "Optimal feature selection for defect classification in semiconductor wafers," *IEEE Transactions on Semiconductor Manufacturing,* vol. 35, no. 2, pp. 324-331, May 2022.

[30] G. Makridis et al., "Predictive maintenance leveraging machine learning for time-series forecasting in the maritime industry," *International Conference on Intelligent Transportation Systems, ITSC 2020,* Rhodes, Greece, September 8-10, 2020, pp. 1-8.

[31] M. Saad et al., "Machine learning based approaches for imputation in time series data and their impact on forecasting," *International Conference on Systems, SMC 2020,* Toronto, Canada, October 5-9, 2020, pp. 2,621-2,627.

[32] A. A. Alfrhan et al., "Smote class imbalance problem in intrusion detection system," *International Conference on Computing and Information Technology, ICCIT 2020,* Tabuk, Saudi Arabia, September 28-30, 2020, pp. 1-5.

[33] AMES Group, *"Basic Manufacturing Process,"* [Online]. Available : https://futurenetworks.ieee.org/images/files/pdf/INGR-2022-Edition/IEEE_INGR_AIML_Chapter_2022-Edition-FINAL.pdf. [Accessed : August 17, 2025].

[34] F. Wilhelmi et al., "AI/ML-based Load Prediction in IEEE 802.11 enterprise networks," IEEE International Conference on Machine Learning for Communication and Networking, ICMLCN 2024, New Jersey, USA, May 6-9, 2024, pp. 50-55.

[35] T. N. Rincy and R. Gupta, "Ensemble learning techniques and its efficiency in Machine learning : A survey," *International Conference on Data, Engineering and Applications, IDEA 2020,* Bhopal, India, February 7-9, 2020, pp. 1-6.

[36] L. Ma et al., "Bagging likelihood-based belief decision trees," International Conference on Information Fusion, PM2TEC 2017, Las Vegas, USA, July 25, 2017, pp. 1-6.

[37] A. Konstantinov et al., "Gradient boosting machine with partially randomized decision trees," *28th Conference of Open Innovations Association, FRUCT 2021,* Moscow, Russia, January 15-17, 2021, pp. 167-173.

[38] R. Alhalaseh and K. Alhalaseh, "Stacked generalization concept for electrical load prediction," *4th International Conference on Smart and Sustainable Technologies, IACC 2019,* Bhimavaram, India, June 27-28, 2019, pp. 1-5.

[39] P. Senthan et al., "Development of churn prediction model using XGBoost- telecommunication industry in sri lanka," *IEEE International IOT Electronics and Mechatronics Conference, IEMTRONICS,* Toronto, Canada, April 10-13, 2021, pp. 1-7.

[40] M. R. Machado et al., "LightGBM : An effective decision tree gradient boosting Method to predict customer loyalty in the finance industry," *14th International Conference on Computer Science & Education, ICCSE 2019,* Toronto, Canada, August 18, 2019, pp. 1,111-1,116.

[41] Y. Zhang et al., "Research and application of adaboost algorithm based on SVM," *IEEE 8th Joint International Information Technology and Artificial Intelligence Conference, ITAIC 2019,* Chongqing, China, May 3-5, 2019, pp. 662-666.

[42] J. Ding et al., "Sales forecasting based on catboost," *2nd International Conference on Information Technology and Computer Application, ITCA 2020,* Guangzhou, China, December 7-10, 2020, pp. 636-639.

[43] O. E. Yurek and D. Birant, "Remaining useful life estimation for predictive maintenance using feature engineering," *Innovations in Intelligent Systems and Applications Conference,* ASYU 2019, Izmir, Turkey, October 15-20, 2019, pp. 1-5.

[44] C. Chen et al., "A risk-averse remaining useful life estimation for predictive maintenance," *Journal of Automatica Sinica,* vol. 8, no. 2, pp. 412-422, February 2021.

[45] O. Koca et al., "Advanced predictive maintenance with machine learning failure estimation in industrial packaging robots," *International Conference on Development and Application Systems, NIPS 2020,* Suceava, Romania, May 6-9, 2020, pp. 1-6.

[46] A. Binding et al., "Machine learning predictive maintenance on data in the wild," *IEEE 5th World Forum on Internet of Things, WF-IoT,* Limerick, Ireland, April 19-21, 2019, pp. 507-512.

[47] G. Makridis et al., "Predictive maintenance leveraging machine learning for time-series forecasting in the maritime industry," *IEEE 23rd International Conference on Intelligent Transportation Systems, ITSC 2020,* Rhodes, Greece, September 2-5, 2020, pp. 1-8.

[48] Guzik Technical Enterprises, *"Read-Write Analyzer Systems 4000 TDMR Series,"* [Online]. Available : https://www.guzik.com/product/read-write-analyzer-systems-4000-tdmr-series/. [Accessed : June 3, 2025].

ภาคผนวก

ภาคผนวก ก.

โปรแกรมการทำงาน

```
import pandas as pd
import matplotlib.pyplot as plt
import time

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

from sklearn.ensemble import (
    AdaBoostClassifier,
    GradientBoostingClassifier,
    BaggingClassifier
)

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,
    roc_auc_score,
    confusion_matrix,
    classification_report,
    roc_curve,
    precision_recall_curve,
    auc
)

# ============================================================================
# 1) Load data
# ============================================================================
df = pd.read_csv("20251213_2.csv")

# ============================================================================
# 2) Select Sensor columns and Label
```

```
# ================================================================================
# Clean column names (remove leading/trailing spaces)
df.columns = df.columns.str.strip()

sensor_cols = [f"Sensor {str(i).zfill(2)}" for i in range(1, 44)]

# Check if columns exist, if not try to auto-detect format
if not all(col in df.columns for col in sensor_cols):
    print("Warning: 'Sensor 01' format not found. Checking alternatives...")
    # Try "Sensor01" (no space)
    alt_cols = [f"Sensor{str(i).zfill(2)}" for i in range(1, 44)]
    if all(col in df.columns for col in alt_cols):
        print("-> Using 'Sensor01' format.")
        sensor_cols = alt_cols
    else:
        # Fallback: use all columns starting with "Sensor"
        sensor_cols = [c for c in df.columns if c.startswith("Sensor")]
        print(f "-> Found {len(sensor_cols)} columns starting with 'Sensor'.")

if not sensor_cols:
    raise KeyError(f "No sensor columns found. Available: {df.columns.tolist()}")

X = df[sensor_cols]
y = df["Status"]

# Map Pass / Fail -> 1 / 0
label_map = {"Pass": 1, "Fail": 0}
y = y.map(label_map)

# ================================================================================
# 3) Scaling & SMOTE (Before Split)
# ================================================================================
scaler = StandardScaler()
X_scaled = pd.DataFrame(scaler.fit_transform(X), columns=sensor_cols)

print("Before SMOTE:")
print(y.value_counts())

smote = SMOTE(random_state=42)
X_resampled, y_resampled = smote.fit_resample(X_scaled, y)
```

```
print("After SMOTE:")
print(pd.Series(y_resampled).value_counts())

# ============================================================================
# 4) Split 80/10/10
# ============================================================================
X_train, X_temp, y_train, y_temp = train_test_split(
    X_resampled,
    y_resampled,
    test_size=0.2,
    random_state=42,
    stratify=y_resampled
)

X_val, X_test, y_val, y_test = train_test_split(
    X_temp,
    y_temp,
    test_size=0.5,
    random_state=42,
    stratify=y_temp
)

print("Data Split")
print("Train:", X_train.shape)
print("Validation:", X_val.shape)
print("Test:", X_test.shape)

# ============================================================================
# 7) Models
# ============================================================================
models = {
    "LightGBM": LGBMClassifier(min_data_in_leaf=1, num_leaves=50),
    "AdaBoost": AdaBoostClassifier(),
    "XGBoost": XGBClassifier(
        use_label_encoder=False,
        eval_metric="logloss"
    ),
    "CatBoost": CatBoostClassifier(verbose=0),
    "Gradient Boosting Tree": GradientBoostingClassifier(),
    "Bagging": BaggingClassifier()
}
```

```
# ================================================================================
# 8) Train & Evaluate (TEST set only)
# ================================================================================
results = []  # <-- reset ให้สะอาด

# Prepare plots
fig, (ax_roc, ax_pr) = plt.subplots(1, 2, figsize=(14, 6))

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

    y_pred = model.predict(X_test)

    if hasattr(model, "predict_proba"):
        y_prob = model.predict_proba(X_test)[:, 1]
    else:
        y_prob = y_pred

    results.append({
        "Model": name,
        "Accuracy": accuracy_score(y_test, y_pred),
        "Precision": precision_score(y_test, y_pred, zero_division=0),
        "Recall": recall_score(y_test, y_pred, zero_division=0),
        "F1-Score": f1_score(y_test, y_pred, zero_division=0),
        "AUC": roc_auc_score(y_test, y_prob)
    })

    # Plot ROC
    fpr, tpr, _ = roc_curve(y_test, y_prob)
    ax_roc.plot(fpr, tpr, label=f"{name} (AUC={results[-1]['AUC']:.3f})")

    # Plot Precision-Recall
    prec, rec, _ = precision_recall_curve(y_test, y_prob)
    pr_auc = auc(rec, prec)
    ax_pr.plot(rec, prec, label=f"{name} (AUC={pr_auc:.3f})")

    print(f"\n=== {name} ===")
    print("Confusion Matrix:")
    print(confusion_matrix(y_test, y_pred))
    print("Classification Report:")
```

```
    print(classification_report(y_test, y_pred, zero_division=0))

# Finalize and show plots
ax_roc.plot([0, 1], [0, 1], “k—”, label=“Random”)
ax_roc.set_title(“ROC Curve”)
ax_roc.set_xlabel(“False Positive Rate”)
ax_roc.set_ylabel(“True Positive Rate”)
ax_roc.legend(loc=“lower right”)

ax_pr.set_title("Precision-Recall Curve")
ax_pr.set_xlabel("Recall")
ax_pr.set_ylabel("Precision")
ax_pr.legend(loc="lower left")

plt.tight_layout()
plt.show()

# ============================================================================
# 9) Summary (ONLY required columns)
# ============================================================================
results_df = pd.DataFrame(
    results,
    columns=[“Model”, “Accuracy”, “Precision”, “Recall”, “F1-Score”, “AUC”]
).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))
```