การทำนายสถานะการทำงานของเครื่องจักรอุตสาหกรรมด้วยข้อมูลจากอุปกรณ์อินเทอร์เน็ตของสรรพสิ่ง

INDUSTRIAL MACHINE STATUS PREDICTION USING INTERNET OF THINGS DEVICES DATA

สุชาภา ธรรมเอกสกุล

SUCHAPA THAMEKSAKUL

การค้นคว้าอิสระนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการข้อมูลและการวิเคราะห์

ศูนย์วิเคราะห์ข้อมูลดิจิทัลอัจฉริยะพระจอมเกล้าลาดกระบัง คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2565

KMITL-2022-SC-M-017-104

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

INDUSTRIAL MACHINE STATUS PREDICTION USING INTERNET OF THINGS DEVICES DATA

SUCHAPA THAMEKSAKUL

AN INDEPENDENT STUDY SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF MASTER OF SCIENCE IN DATA SCIENCE AND ANALYTICS

KMITL-DIGITAL ANALYTICS AND INTELLIGENCE CENTER SCHOOL OF SCIENCE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2022

KMITL-2022-SC-M-017-104

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COPYRIGHT 2022

SCHOOL OF SCIENCE

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อการค้นคว้าอิสระ	การทำนายสถานะการทำงานของเครื่องจักรอุตสาหกรรมด้วยข้อมูลจากอุปกรณ์อินเทอร์เน็ตของสรรพสิ่ง
ชื่อนักศึกษา	นางสาวสุชาภา ธรรมเอกสกุล
รหัสประจำตัว	63605092
ปริญญา	วิทยาศาสตรมหาบัณฑิต (วิทยาการข้อมูลและการวิเคราะห์) ศูนย์วิเคราะห์ข้อมูลดิจิทัลอัจฉริยะพระจอมเกล้าลาดกระบัง
พ.ศ.	2565
อาจารย์ที่ปรึกษาการค้นคว้าอิสระ	ผู้ช่วยศาสตราจารย์ ดร.วรางคณา กิ้มปาน

บทคัดย่อ

เทคโนโลยีอินเทอร์เน็ตของสรรพสิ่ง (Internet of things : IoT) เป็นเทคโนโลยีที่จำเป็นสำหรับการเปลี่ยนแปลงทางดิจิทัล (Digital Transformation) และโรงงานอัจฉริยะ (Smart Factory) โดยข้อมูลที่รวบรวมจากอุปกรณ์ต่างๆ จะช่วยให้เข้าใจสถานะการทำงานในปัจจุบัน ประสิทธิภาพโดยรวม และความผิดปกติของอุปกรณ์ต่างๆ ในช่วงที่ผ่านมามีงานวิจัยจำนวนมากที่นำเสนอการทำงานด้วยระบบอินเทอร์เน็ตของสรรพสิ่ง เช่น อุตสาหกรรมการผลิต พลังงาน เกษตรกรรม การขนส่งและสาธารณูปโภค เป็นต้น การศึกษาค้นคว้าอิสระนี้มุ่งเน้นการนำเทคโนโลยีอินเทอร์เน็ตของสรรพสิ่งมาใช้ในการบริหารจัดการเกี่ยวกับกระบวนการทำงานของเครื่องจักร และแก้ไขส่วนที่มีความผิดปกติได้อย่างทันท่วงที อีกทั้งยังสามารถยืดอายุการใช้งานของเครื่องจักรได้อีกด้วย จึงเป็นอีกหนึ่งเทคโนโลยีที่ทำให้ผู้ประกอบการสามารถเข้าถึงอุตสาหกรรมยุค 4.0 ได้ง่ายขึ้น

การค้นคว้าอิสระนี้มีวัตถุประสงค์เพื่อศึกษาความสัมพันธ์ระหว่างอุปกรณ์ (Sensors) และเครื่องจักร ผ่านการเก็บข้อมูลระบบอินเทอร์เน็ตของสรรพสิ่ง และเปรียบเทียบประสิทธิภาพของวิธีการจำแนกกลุ่ม โดยเลือกใช้วิธีต้นไม้ตัดสินใจ วิธีโครงข่ายประสาทเทียม และวิธีซัพพอร์ตเวกเตอร์แมชชีน จากการเปรียบเทียบประสิทธิภาพวิธีการจำแนกกลุ่มข้อมูลเซนเซอร์ โดยเปรียบเทียบจากค่าความถูกต้อง วิธีการจำแนกกลุ่มที่มีประสิทธิภาพการจำแนกดีที่สุดคือ วิธีต้นไม้ตัดสินใจ ซึ่งให้ค่าความถูกต้อง คือ 99.96% เมื่อเทียบกับวิธีซัพพอร์ตเวกเตอร์แมชชีน ค่าความถูกต้องเฉลี่ย 99.93% และวิธีโครงข่ายประสาทเทียม ค่าความถูกต้องเฉลี่ยที่ 99.80%

คำสำคัญ: อินเทอร์เน็ตของสรรพสิ่ง, วิธีต้นไม้ตัดสินใจ, วิธีโครงข่ายประสาทเทียม, วิธีซัพพอร์ตเวกเตอร์แมชชีน, การปรับปรุงประสิทธิภาพของเครื่องจักร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Independent Study Title	INDUSTRIAL MACHINE STATUS PREDICTION USING INTERNET OF THINGS DEVICES DATA
Student Name	Miss Suchapa Thameksakul
Student ID	63605092
Degree	Master of Science (Data science and Analytics) KMITL-Digital Analytics and Intelligence Center
Year	2022
Independent Study Advisor	Asst.Prof.Dr. Warangkhana Kimpan

ABSTRACT

Internet of things (IOT) technology is an essential technology for digital transformation and smart factory. Information collected from different devices will help to understand current working status, overall performance, and malfunction of the device. Recently, there has been more and more research on Internet of Things systems. such as Manufacturing, Energy, Agriculture, Transportation and Utilities, etc.

The purposes of this study were to study the relationship between devices (Sensors) and machines through the collection of information by the Internet of Things system and compare the efficiency of the classification method. Three methods were used: Decision tree, Neural network, and Support vector machine. The results of the comparison of the efficiency for Three methods method by comparing the accuracy. The most effective of classification method is the Decision Tree method, with an accuracy of 99.96% compared to the Support Vector Machine has 99.93% accuracy, and Neural Network has 99.80% accuracy.

Keywords: Internet of things, Decision Tree, Neural Network, Support Vector Machine, Synthetic Minority Over-Sampling Technique (SMOTE), K-Fold Cross-Validation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การศึกษาค้นคว้าอิสระฉบับนี้สำเร็จลุล่วงได้ด้วยดี เนื่องจากได้รับความกรุณาอย่างสูงจาก ผู้ช่วยศาสตราจารย์ ดร.วรางคณา กิ้มปาน อาจารย์ที่ปรึกษาการศึกษาค้นคว้าอิสระ ที่ให้คำแนะนำ ปรึกษาตลอดจนปรับปรุงแก้ไขข้อบกพร่องต่างๆ รวมทั้งบุคคลที่ผู้ศึกษาค้นคว้าอิสระนำมาอ้างอิงทางวิชาการตามที่ปรากฏในเอกสารอ้างอิง เพื่อให้การศึกษาค้นคว้าอิสระฉบับนี้มีความสมบูรณ์มากที่สุด

ขอขอบคุณคณาจารย์ทุกท่านที่เคยสั่งสมให้ความรู้มาตลอดการศึกษา ขอขอบคุณเจ้าหน้าที่ประจำหลักสูตรวิทยาการข้อมูลและการวิเคราะห์ทุกท่านที่คอยให้คำปรึกษา คอยช่วยเหลือในการศึกษาระดับปริญญาโท และขอขอบคุณขอบคุณเพื่อนๆทุกคนที่เคยช่วยเหลือไม่ว่าทางตรงหรือทางอ้อม

ท้ายสุดนี้ขอขอบคุณบิดา มารดา และครอบครัว ตลอดจนผู้มีพระคุณทุกท่าน ที่คอยสนับสนุนช่วยเหลือ และเป็นกำลังใจตลอดการศึกษา

นางสาวสุชาภา ธรรมเอกสกุล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญตาราง	จ
สารบัญรูป	ฉ
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์ของการค้นคว้าอิสระ	2
1.3 ขอบเขตของการค้นคว้าอิสระ	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ	2
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	3
2.1 แนวคิดเกี่ยวกับการเรียนรู้ของเครื่องจักร	3
2.2 แนวคิดเกี่ยวกับการเรียนรู้แบบมีผู้สอน	4
2.2.1 ซัพพอร์ต์เวกเตอร์แมชชีน (Support Vector Machines: SVM)	5
2.2.2 การสร้างต้นไม้ตัดสินใจ (Decision Tree)	6
2.2.3 โครงข่ายประสาทเทียม (Neural Network)	7
2.3 แนวคิดเกี่ยวกับเซนเซอร์อินเทอร์เน็ตของสรรพสิ่ง	9
2.4 เทคนิคการสุ่มเพิ่มตัวอย่างกลุ่มน้อย	10
2.5 การวัดประสิทธิภาพ	10
2.6 วิธีการวิเคราะห์ความแม่นตรงของตัวแบบ (K-Fold Cross-Validation)	13
2.7 งานวิจัยที่เกี่ยวข้อง	13
2.7.1 การบำรุงรักษาเชิงคาดการณ์ของเครื่องจักรอุตสาหกรรม	13
2.7.2 อินเทอร์เน็ตของสรรพสิ่งกับอุตสาหกรรมการผลิต	16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
บทที่ 3 วิธีการดำเนินการ	20
3.1 ขั้นตอนของการศึกษาค้นคว้าอิสระ	20
3.1.1 ขั้นตอนการเตรียมข้อมูล	20
3.1.2 อัลกอริทึมที่ใช้ในการหาค่าความผิดปกติของเครื่องจักรในโรงงานอุตสาหกรรม	27
3.1.3 วิธีการทำนายความผิดปกติของเครื่องจักรในโรงงานอุตสาหกรรม	27
3.1.4 การวัดผลการทดลอง (Evaluation)	27
3.2 การออกแบบการทดลอง	27
3.2.1 ขั้นตอนการเลือกชุดข้อมูล	27
3.2.2 วิธีการทดลอง	30
3.3 เครื่องมือที่ใช้ในการทดลอง	32
บทที่ 4 ผลการทดลองและอภิปรายผล	33
4.1 การทดลองชุดที่ 1 และชุดที่ 2	33
4.2 เปรียบเทียบผลการทดลองชุดที่ 1 และชุดที่ 2	37
บทที่ 5 บทสรุปและข้อเสนอแนะ	39
5.1 สรุป	39
5.2 วิเคราะห์ผลการทดลอง	39
5.3 ปัญหาและข้อเสนอแนะ	40
เอกสารอ้างอิง	42
ภาคผนวก	44
ภาคผนวก ก ขั้นตอนการจัดเตรียมข้อมูลเพื่อใช้ในการสอนและทดสอบโมเดล	45
ภาคผนวก ข ขั้นตอนการสอน ทดสอบ และปรับแต่งค่าพารามิเตอร์ให้เหมาะสมกับโมเดล	55
ประวัติผู้เขียน	64

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1 กราฟแสดงการแบ่งแยกกลุ่มของ Support Vector Machine	5
2.2 นิวรอน (Neurons) และจุดประสานประสาท (Synapses) แต่ละเซลล์ประสาท	8
2.3 โครงสร้างการเรียนรู้ของ Neural network	9
2.4 การทำนาย Confusion Matrix	10
2.5 ประสิทธิภาพการทำนาย ROC AUC score	12
2.6 5-Fold Cross-Validation	13
2.7 อุปกรณ์ต่างๆ ภายในเครื่องจักร	14
2.8 เครื่องตัดที่ใช้ในกระบวนการผลิต	15
2.9 ตำแหน่งที่ใช้เกตเวย์เพื่อเชื่อมต่อระบบควบคุมแบบแยกส่วนกับคลาวด์	17
2.10 การไหลของอินเทอร์เน็ตของสรรพสิ่ง	17
2.11 บล็อกไดอะแกรมสำหรับกระบวนการส่งข้อมูลอุปกรณ์ต่างๆไปยังยังคลาวด์	18
3.1 สถานะการทำงานของเครื่องจักร	23
3.2 จำนวนครั้งของสถานะการทำงานปกติ และผิดปกติของเครื่องจักร	23
3.3 ฮิสโตแกรม (Histogram) ของเซนเซอร์ 51 เซนเซอร์	26
4.1 การทำนายค่า Confusion Matrix โมเดล Support Vector Machine ของการทดลองชุดที่ 1	35
4.2 การทำนายค่า Confusion Matrix โมเดล Decision Tree ของการทดลองชุดที่ 1	35
4.3 การทำนายค่า Confusion Matrix โมเดล Neural Network ของการทดลองชุดที่ 1	35
4.4 การทำนายค่า Confusion Matrix โมเดล Support vector machine ของการทดลองชุดที่ 2	36
4.5 การทำนายค่า Confusion Matrix โมเดล Decision Tree ของการทดลองชุดที่ 2	36
4.6 การทำนายค่า Confusion Matrix โมเดล Neural Network ของการทดลองชุดที่ 2	37
ก.1 อ่านข้อมูลจาก .csv ไฟล์	45
ก.2 ตัวอย่างลักษณะข้อมูลการทดลองชุดที่ 1 จากไฟล์ .csv	46
ก.3 ตัวอย่างการทำความสะอาดข้อมูลการทดลองชุดที่ 1 (1)	46
ก.4 ตัวอย่างการทำความสะอาดข้อมูลการทดลองชุดที่ 1 (2)	47
ก.5 ตัวอย่างการทำความสะอาดข้อมูลการทดลองชุดที่ 1 (3)	47

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
ก.6 ตัวอย่างการทำความสะอาดข้อมูลการทดลองชุดที่ 1 (4)	48
ก.7 Heatmap ข้อมูลการทดลองชุดที่ 1 ก่อนทำความสะอาดข้อมูล	48
ก.8 ตัวอย่างการทำความสะอาดข้อมูลการทดลองชุดที่ 1 (5)	49
ก.9 ชุดข้อมูลสำหรับการสอน และทดสอบโมเดลของข้อมูลการทดลองชุดที่ 1	49
ก.10 เทคนิคการสุ่มเพิ่มตัวอย่างกลุ่มน้อยของข้อมูลการทดลองชุดที่ 1	50
ก.11 ตัวอย่างการทำความสะอาดข้อมูลการทดลองชุดที่ 2 (1)	51
ก.12 ตัวอย่างการทำความสะอาดข้อมูลการทดลองชุดที่ 2 (2)	51
ก.13 ตัวอย่างการทำความสะอาดข้อมูลการทดลองชุดที่ 2 (3)	52
ก.14 ตัวอย่างการทำความสะอาดข้อมูลการทดลองชุดที่ 2 (4)	52
ก.15 Heatmap ข้อมูลการทดลองชุดที่ 2 ก่อนทำความสะอาดข้อมูล	53
ก.16 ตัวอย่างการทำความสะอาดข้อมูลการทดลองชุดที่ 2 (5)	53
ก.17 ชุดข้อมูลสำหรับการสอน และทดสอบโมเดลของข้อมูลการทดลองชุดที่ 2	54
ก.18 เทคนิคการสุ่มเพิ่มตัวอย่างกลุ่มน้อยของข้อมูลการทดลองชุดที่ 2	54
ข.1 ตัวอย่างการสอน และทดสอบโมเดลเบื้องต้นของการทดลองชุดที่ 1	56
ข.2 ผลลัพธ์การทดสอบโมเดลของการทดลองชุดที่ 1	56
ข.3 ปรับแต่งค่าพารามิเตอร์ของแต่ละโมเดลในการทดลองชุดที่ 1	57
ข.4 ค่าพารามิเตอร์ที่ดีที่สุดของแต่ละโมเดลในการทดลองชุดที่ 1	57
ข.5 ค่าพารามิเตอร์ที่ดีที่สุดของ Neural Network ในการทดลองชุดที่ 1	57
ข.6 ผลการทดลองโมเดล Support Vector Machine ของการทดลองชุดที่ 1	58
ข.7 ผลการทดลองโมเดล Decision Tree ของการทดลองชุดที่ 1	58
ข.8 ผลการทดลองโมเดล Neural network ของการทดลองชุดที่ 1 (1)	58
ข.9 ผลการทดลองโมเดล Neural network ของการทดลองชุดที่ 1 (2)	59
ข.10 ตัวอย่างการสอน และทดสอบโมเดลเบื้องต้นของการทดลองชุดที่ 2	60
ข.11 ผลลัพธ์การทดสอบโมเดลของการทดลองชุดที่ 2	60
ข.12 ปรับแต่งค่าพารามิเตอร์ของแต่ละโมเดลในการทดลองชุดที่ 2	61
ข.13 ค่าพารามิเตอร์ที่ดีที่สุดของแต่ละโมเดลในการทดลองชุดที่ 2	61

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
ข.14 ค่าพารามิเตอร์ที่ดีที่สุดของ Neural network ในการทดลองชุดที่ 2	61
ข.15 ผลการทดลองโมเดล Support Vector Machine ของการทดลองชุดที่ 2	62
ข.16 ผลการทดลองโมเดล Decision Tree ของการทดลองชุดที่ 2	62
ข.17 ผลการทดลองโมเดล Neural network ของการทดลองชุดที่ 2 (1)	62
ข.18 ผลการทดลองโมเดล Neural network ของการทดลองชุดที่ 2 (2)	63

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
3.1 รายละเอียดคุณลักษณะที่นำมาสร้างตัวแบบ	21
3.2 สถิติชุดข้อมูลทั้งหมด	24
3.3 สถิติชุดข้อมูลช่วงที่1 (วันที่ 1 เมษายน – 30 มิถุนายน 2561)	28
3.4 สถิติชุดข้อมูลช่วงที่ 2 (วันที่ 1 กรกฎาคม – 30 สิงหาคม 2561)	29
3.5 ชุดข้อมูลสำหรับการสอน และข้อมูลสำหรับการทดสอบ ของการทดลองชุดที่ 1 และ2	30
3.6 ชุดข้อมูลสำหรับการสอน และข้อมูลสำหรับการทดสอบ หลังจากทำการสุ่มเพิ่มตัวอย่างกลุ่มน้อยของการทดลองชุดที่ 1 และ2	31
4.1 ค่าประสิทธิภาพการทำนายโมเดล Support Vector Machine, Decision Tree และ Neural Network ของการทดลองชุดที่ 1	33
4.2 ค่าประสิทธิภาพการทำนายโมเดล Support Vector Machine, Decision Tree และ Neural Network ของการทดลองชุดที่ 2	34
4.3 การเปรียบเทียบความถูกต้องของโมเดลต่างๆ ในการทำนายความผิดปกติของเครื่องจักร	38

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในยุคสังคมปัจจุบัน เทคโนโลยี และข้อมูลขนาดใหญ่ (Big Data) หรือ ข้อมูลจำนวนมากมหาศาล ที่เก็บรวบรวมข้อมูล ทุกเรื่อง ทุกแง่มุม ทุกรูปแบบ ซึ่งอาจเป็นข้อมูลที่มีโครงสร้าง (Structured Data) เช่น ข้อมูลที่เก็บอยู่ในตารางข้อมูลต่างๆ หรืออาจเป็นข้อมูลกึ่งโครงสร้าง (Semi-Structured Data) นับว่ามีบทบาทสำคัญในทางอุตสาหกรรม และได้รับการยอมรับว่าเป็นเครื่องมือสำคัญอย่างหนึ่งต่อการทำงานทั้งภาครัฐ และภาคเอกชน การนำข้อมูลจำนวนมาก จากหลายแหล่งข้อมูลมาจัดการด้วยกระบวนการต่างๆ และนำเอาข้อมูลขนาดใหญ่มาวิเคราะห์ด้วยเทคโนโลยีที่มีความก้าวหน้าที่สูงขึ้น เช่น ปัญญาประดิษฐ์ ทำการวิเคราะห์ข้อมูลเหล่านั้น เพื่อค้นหาคำตอบที่ช่วยในการ ลดต้นทุน ลดเวลา พัฒนาผลิตภัณฑ์ใหม่ หรือหาแนวทางที่ดีที่สุดในรูปแบบต่างๆของแต่ละองค์กร และช่วยในการตัดสินใจอย่างชาญฉลาด

อุตสาหกรรมการผลิตในยุค 4.0 เป็นยุคของการผลิตที่นำเครือข่ายอินเทอร์เน็ตเข้ามาเชื่อมโยงข้อมูลการผลิตระหว่างเครื่องจักร เป็นยุคของการรวมระหว่างเทคโนโลยีดิจิทัลควบคุมเครื่องจักร ให้เครื่องจักรสื่อสารข้อมูลกันเอง โดยที่ระบบอัตโนมัติ (Automation) และอินเทอร์เน็ตของสรรพสิ่ง (Internet of things : IOT) เริ่มมีการผสมผสานกัน สื่อสารและควบคุมได้ เป็นยุคที่จะเปลี่ยนผ่านจากโรงงานปกติ (Factory) ไปสู่ความเป็นระบบการผลิตอัจฉริยะ (Smart Manufacturing) จึงได้นำวิธีการวิเคราะห์ข้อมูล (Data Analytics) เพื่อขับเคลื่อน ควบคุม และตัดสินใจบนพื้นฐานของข้อมูลที่ดี ส่งผลให้อุตสาหกรรมการผลิตมีประสิทธิภาพมากยิ่งขึ้น

ในระบบการผลิตของโรงงานอุตสาหกรรมประกอบด้วยเครื่องจักรหลากหลายประเภท และมีหน้าที่หลักในการทำงานไม่เหมือนกัน ซึ่งจะแบ่งแยกไปในแต่ละขั้นตอนการผลิต ส่งผลทำให้ข้อมูลที่ได้จากการบันทึกระหว่างการผลิตมีความแตกต่างกัน และทำให้การวางแผนการผลิตเกิดข้อผิดพลาดขึ้นได้ ดังนั้นจึงมีวิธีการรวบรวมข้อมูลของเครื่องจักรที่ใช้ในการผลิตขนาดใหญ่นำมาวิเคราะห์และทำนายลักษณะการทำงานที่ผิดปกติของอุปกรณ์ต่างๆ ของเครื่องจักร สำหรับการวางแผนการผลิตรวมถึงการวางแผนระบบการซ่อมบำรุงเครื่องจักร ทำให้สามารถแก้ไขปัญหาเครื่องจักรหยุดทำงาน (Breakdown) เครื่องจักรสามารถทำงานได้อย่างต่อเนื่อง และยังสามารถช่วยเพิ่มประสิทธิภาพของระบบการผลิตให้ดียิ่งขึ้นอีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 วัตถุประสงค์ของการค้นคว้าอิสระ

1) เพื่อศึกษาสถานะของเครื่องจักร ปกติ หรือ ผิดปกติ จากอุปกรณ์ภายในเครื่องจักร โดยใช้การจำแนกแบบกลุ่ม (Classification)

2) เพื่อศึกษาโมเดล Neural network, Support Vector Machine และ Decision Tree ในการวัดประสิทธิภาพของกลุ่มตัวอย่าง

3) ศึกษาความเหมาะสมของโมเดลกับกลุ่มตัวอย่าง เพื่อการนำไปประยุกต์ใช้ในอนาคต

1.3 ขอบเขตของการค้นคว้าอิสระ

1) ขอบเขตด้านกลุ่มตัวอย่าง : กลุ่มตัวอย่างเครื่องจักรในโรงงานอุตสาหกรรม

2) ตัวแปรและทฤษฎีที่ใช้ในการศึกษา ได้แก่ การเรียนรู้แบบมีผู้สอน (Supervised Learning) ประเภท การจำแนกกลุ่ม และวัดประสิทธิภาพของแต่ละโมเดล รวมทั้งสิ้น 3 โมเดล คือ Neural network, Support Vector Machine และ Decision Tree

- ตัวแปรอิสระ ได้แก่ เซนเซอร์
- ตัวแปรตาม ได้แก่ สถานะของเครื่องจักร

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1) สามารถนำเทคโนโลยีอินเทอร์เน็ตของสรรพสิ่งมาใช้ในการบริหารและจัดการเกี่ยวกับกระบวนการทำงานของเครื่องจักร และแก้ไขส่วนที่มีความผิดปกติได้อย่างทันท่วงที

2) เพื่อเป็นแนวทางในการเพิ่มประสิทธิภาพ ปรับปรุง พัฒนา กระบวนการผลิต เพื่อตอบสนองความต้องการของลูกค้าอย่างต่อเนื่องทั้งในด้านสินค้า และบริการ

3) ผลการวิจัยนี้ทำให้ผู้ประกอบการสามารถเข้าถึงอุตสาหกรรมยุค 4.0 ได้ง่ายขึ้น และสามารถนำเทคโนโลยีนี้ปรับใช้ในกระบวนการผลิตสินค้า และบริการในโรงงานอุตสาหกรรมได้เป็นอย่างดี

4) สามารถบอกแนวโน้มความเป็นไปได้ในกระบวนการผลิต ทำให้แผนการทำงานสามารถดำเนินการไปได้อย่างราบรื่น

5) สามารถนำผลลัพธ์ที่ได้จากการค้นคว้าอิสระไปทำแผนการลงทุนเพื่อพัฒนาการผลิตในโรงงานอุตสาหกรรม ลดต้นทุนในการจัดซื้ออุปกรณ์ และติดตั้งเครื่องจักรในการผลิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

การศึกษาค้นคว้าอิสระเรื่อง การบำรุงรักษาเชิงคาดการณ์ของเครื่องจักรอุตสาหกรรมด้วยการเชื่อมโยงข้อมูลของเซนเซอร์อินเทอร์เน็ตของสรรพสิ่ง คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง พ.ศ.2563-2564 ผู้ค้นคว้าอิสระได้ศึกษางานวิจัยที่เกี่ยวข้องดังต่อไปนี้ แนวคิดเกี่ยวกับการเรียนรู้ของเครื่อง แนวคิดเกี่ยวกับการเรียนรู้แบบมีผู้สอน ได้แก่ Support Vector Machines (SVM) การสร้างต้นไม้ตัดสินใจ (Decision Tree) และโครงข่ายประสาทเทียม (Neural Network) แนวคิดเกี่ยวกับเซนเซอร์อินเทอร์เน็ตของสรรพสิ่ง และงานวิจัยที่เกี่ยวข้อง

2.1 แนวคิดเกี่ยวกับการเรียนรู้ของเครื่อง

การเรียนรู้ของเครื่อง (Machine Learning) [1] เป็นสาขาหนึ่งของปัญญาประดิษฐ์ (Artificial Intelligent) ที่พัฒนามาจากการศึกษาการรู้การจดจำรูปแบบ (Pattern Recognition) เกี่ยวข้องกับการศึกษา การสร้างอัลกอริทึมที่สามารถเรียนรู้ข้อมูล และทำนายข้อมูลได้ อัลกอริทึมนั้นจะทำงานโดยอาศัยโมเดล ที่สร้างมาจากชุดข้อมูลตัวอย่างขาเข้า (Training Data) เพื่อการทำนาย (Predict) หรือตัดสินใจในภายหลังแทนที่จะทำงานตามลำดับของคำสั่งโปรแกรมคอมพิวเตอร์

การเรียนรู้ของเครื่อง คือ การทำให้ระบบคอมพิวเตอร์เรียนรู้ได้ด้วยตนเอง โดยอาศัยข้อมูล (Data) เมื่อคอมพิวเตอร์ สมาร์ทโฟน หรือเครื่องจักรอุตสาหกรรมที่ควบคุมโดยคอมพิวเตอร์มีความฉลาดมากพอที่จะเรียนรู้ด้วยตัวเองจะรู้ว่าสิ่งใด กระบวนการใดที่เกิดซ้ำ ทำบ่อยๆ สภาพแวดล้อมแบบใดที่จะต้องเจอและไม่คุ้นเคย แล้วเกิดกระบวนการแยกแยะ จากนั้นจะประมวลผล และทำงานผ่านปัญญาประดิษฐ์ ออกมาเป็นผลลัพธ์ (Output) ส่งเสริมความสะดวกสบายของมนุษย์ ซึ่งองค์กรธุรกิจใดสามารถเข้าถึงเรื่องนี้และนำมาปรับใช้กับธุรกิจของตน ก็จะทำให้ได้เปรียบในการแข่งขันมาก เพราะสามารถลดเวลาการทำงาน และลดต้นทุนแรงงาน จึงเรียกได้ว่าสิ่งนี้เป็นศักยภาพแห่งอนาคตที่เข้ามาตอบโจทย์ธุรกิจอุตสาหกรรมได้ดีซึ่งแตกต่างกับโปรแกรมสั่งการทั่วไป เนื่องจากการเขียนโปรแกรมคอมพิวเตอร์ทั่วไปคอมพิวเตอร์จะทำงานตามคำสั่งที่ผู้เขียนโปรแกรมสั่งให้ทำแบบเฉพาะเจาะจง เป็นลำดับขั้นตอน เพื่อให้ได้ผลลัพธ์ แต่การเรียนรู้ของเครื่อง คอมพิวเตอร์มีความฉลาดมากยิ่งขึ้น ซึ่งสามารถโปรแกรมได้เองโดยไม่ต้องมีผู้สั่งให้ทำแบบเฉพาะเจาะจง แต่จำเป็นต้องมีการใส่ข้อมูลขาเข้า (Input) และผลลัพธ์เข้าไป เพื่อให้คอมพิวเตอร์หาโปรแกรมที่จะนำไปตอบโจทย์ในอนาคตได้ว่าข้อมูลลักษณะนี้ ได้ผลลัพธ์เป็นอย่างไร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเรียนรู้ของเครื่อง จำแนกออกได้เป็นสามรูปแบบคือ การเรียนรู้แบบมีผู้สอน (Supervised Learning) การเรียนรู้แบบไม่มีผู้สอน (Unsupervised Learning) และการเรียนรู้แบบเสริมกำลัง (Reinforcement Learning) มีรายละเอียดดังนี้

1) การเรียนรู้แบบมีผู้สอน (Supervised Learning) เป็นการนำข้อมูลตัวอย่างและผลลัพธ์ที่ “ผู้สอน” ต้องการป้อนเข้าสู่คอมพิวเตอร์ เป้าหมาย คือ การสร้างกฎทั่วไปที่สามารถเชื่อมโยงข้อมูลขาเข้ากับขาออกได้

2) การเรียนรู้แบบไม่มีผู้สอน (Unsupervised learning) เป็นการนำข้อมูลตัวอย่างและผลลัพธ์ที่ “ผู้สอน” ต้องการป้อนเข้าสู่คอมพิวเตอร์ โดยไม่มีการกำหนดผลลัพธ์ใดๆ และให้คอมพิวเตอร์หาโครงสร้างของข้อมูลขาเข้าเอง

3) การเรียนรู้แบบเสริมกำลัง (Reinforcement learning) คือ คอมพิวเตอร์มีปฏิสัมพันธ์และเรียนรู้กับสิ่งแวดล้อมที่เปลี่ยนไปตลอดเวลา โดยคอมพิวเตอร์จะต้องทำงานบางอย่าง(Action) เช่น ขับรถ โดยที่ไม่มี “ผู้สอน” คอยบอกอย่างจริงจังว่าวิธีการที่ทำอยู่นั้นเข้าใกล้เป้าหมายแล้วหรือไม่ ตัวอย่างเช่น การเรียนรู้เพื่อเล่นเกม เป็นต้น

2.2 แนวคิดเกี่ยวกับการเรียนรู้แบบมีผู้สอน

การเรียนรู้แบบมีผู้สอน (Supervised Learning) [2] เป็นการเรียนรู้ที่เครื่องจะต้องอาศัยข้อมูลในการฝึกฝน (Training Set) กล่าวคือ การที่ต้องคอยแยกประเภท หรือบอกผลลัพธ์ที่ควรจะเป็นไว้ล่วงหน้า จากนั้นใช้ข้อมูลตัวอย่างชุดฝึกฝนข้อมูลนำไปผ่านอัลกอริทึมสำหรับสร้างโมเดลในการคาดการณ์ผลลัพธ์ที่จะเกิดขึ้นในอนาคต เมื่อเกิดการเรียนรู้แล้วจึงนำข้อมูลไปทดสอบ ผ่านข้อมูลชุดใหม่ที่คอมพิวเตอร์ยังไม่เคยได้ประมวลผลมาก่อน (Test Set) และสังเกตผลลัพธ์ที่ถูกคาดการณ์โดยการเรียนรู้ของเครื่อง ตัวอย่างที่ถูกนำมาประยุกต์ใช้งานในเชิงธุรกิจ คือ การคำนวณราคาบ้าน หรือการวิเคราะห์ผลฟุตบอล เป็นต้น

การเรียนรูปแบบมีผู้สอน แบ่งเป็น 2 ลักษณะ คือ

1) การจำแนกประเภทข้อมูล (Classification) คือ โมเดลจะต้องมีผลลัพธ์ หรือตัวแปรที่ต้องการศึกษา เป็นตัวต้นแบบให้เรียนรู้ โดยข้อมูลเป็นลักษณะกลุ่ม หรือข้อมูลที่ไม่มีความต่อเนื่อง เช่น ผลลัพธ์แสดงเป็น ใช่/ไม่ใช่ หรือ อนุมัติ/ไม่อนุมัติ เป็นต้น

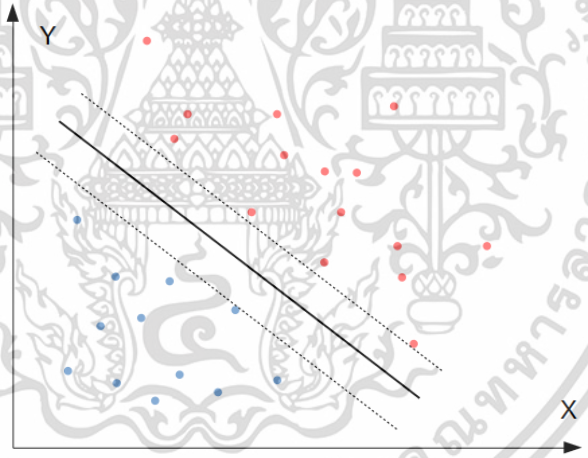
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) การถดถอย (Regression) คือ มีความคล้ายคลึงกับการแยกประเภท แต่ผลลัพธ์ หรือตัวแปรที่ต้องการศึกษา เป็นข้อมูลที่มีความต่อเนื่อง ไม่ได้แบ่งเป็นกลุ่ม เช่น การทำนายยอดขาย การทำนายราคาสินค้าหรือราคาบ้าน เป็นต้น

สำหรับอัลกอริทึมการเรียนรู้แบบมีผู้สอน ยกตัวอย่าง 3 อัลกอริทึม ได้แก่ ซัพพอร์ต์เวกเตอร์แมชชีน (SVM) การสร้างต้นไม้ตัดสินใจ (Decision Tree) และโครงข่ายประสาทเทียม (Neural Network) ดังต่อไปนี้

2.2.1 ซัพพอร์ต์เวกเตอร์แมชชีน (Support Vector Machines: SVM)

Support Vector Machines [3] เป็นอัลกอริทึมที่สามารถนำมาช่วยแก้ปัญหาการจำแนกข้อมูลและวิเคราะห์ข้อมูล ซึ่งทั้งยืดหยุ่นและทำงานได้ดีโดยเฉพาะอย่างยิ่งเมื่อข้อมูลมีความซับซ้อน (หลาย Feature) แต่จำนวนตัวอย่างไม่มาก โดยอาศัยหลักการ การหาสัมประสิทธิ์ของสมการเพื่อสร้างเส้นแบ่งแยกกลุ่มข้อมูลที่ถูกป้อนเข้าสู่กระบวนการการเรียนรู้แบบมีผู้สอน โดยเน้นไปยังเส้นแบ่งแยกแยะกลุ่มข้อมูลได้ดีที่สุด ดังรูปที่ 2.1



รูปที่ 2.1 กราฟแสดงการแบ่งแยกกลุ่มของ Support Vector Machine

แนวความคิดของ Support Vector Machine จะเป็นการจำแนกประเภทข้อมูล โดยการแบ่งข้อมูลออกเป็นกลุ่ม (Class) ซึ่งสามารถใช้การแบ่งด้วยสมการเชิงเส้นได้ทั้งเส้นตรง (Linear) และไม่เป็นเส้นตรง (Non-Linear) สำหรับสมการเชิงเส้น การแบ่งกลุ่มเกิดจากการที่นำค่าของกลุ่มข้อมูลมาวางลงในปริภูมิคุณลักษณะ (Feature Space) จากนั้นจึงหาเส้นที่ใช้แบ่งข้อมูลทั้งสองออกจากกันโดยจะสร้างเส้นแบ่ง (Hyperplane) ที่เป็นเส้นตรงขึ้นมา เพื่อให้ทราบว่าเส้นตรงที่แบ่งกลุ่มออกจากกันนั้นเส้นตรงใดเป็นเส้นที่แบ่งแยกได้ดีที่สุด โดยมีเงื่อนไขว่าจะต้องหาคู่ของเส้นประที่กว้างที่สุดเท่าที่จะเป็นไปได้ และข้อมูลที่มีลักษณะไม่เป็นเชิงเส้น จะมีฟังก์ชันเคอร์เนล (Kernel Function) ที่สามารถแปลงข้อมูลไปยังมิติที่สูงขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในปริภูมิคุณลักษณะ (Feature Space) เพื่อให้สามารถแบ่งข้อมูลแบบเชิงเส้นได้ และจำแนกข้อมูลที่มีความคลุมเครือได้อย่างมีประสิทธิภาพ ประสิทธิภาพของการจำแนกประเภท หรือการประมาณค่านั้นจะขึ้นอยู่กับฟังก์ชันเคอร์เนลที่เลือกใช้กับลักษณะของชุดข้อมูล และพารามิเตอร์ต่าง ๆ

การวิเคราะห์จำแนกกลุ่มถูกออกแบบมาสำหรับกลุ่มเป้าหมายประเภท การจำแนกแบบไบนารี (Binary Classification) แต่สามารถนำไปประยุกต์ใช้กับการจำแนกกลุ่มหลายประเภท (Multiclass classification) และการวิเคราะห์การถดถอยเชิงเส้น (Linear Regression) ได้โดยง่าย การวิเคราะห์ด้วยวิธีนี้นอกจากจะสามารถจำแนกความแตกต่างระหว่างกลุ่มได้ และยังสามารถบอกได้ว่าตัวแปรใดจำแนกได้ดีมากน้อยกว่ากัน นั่นคือสามารถบอกประสิทธิภาพ หรือน้ำหนักในการจำแนกของการจัดกลุ่ม และยังสามารถพยากรณ์การเข้าสู่กลุ่มของข้อมูลใหม่ด้วย

2.2.2 การสร้างต้นไม้ตัดสินใจ (Decision Tree)

เทคนิคการสร้างต้นไม้ตัดสินใจ (Decision Tree) [4] คือ แบบจำลองทางคณิตศาสตร์ เพื่อหาทางเลือกที่ดีที่สุด ซึ่งเป็นเทคนิคหนึ่งในการสร้างแบบจำลองพยากรณ์รูปแบบโครงสร้างต้นไม้ การเรียนรู้โดยการจำแนกประเภทข้อมูล (Classification) โดยแบ่งข้อมูลออกเป็นกลุ่มต่างๆ โดยใช้คุณลักษณะข้อมูล (Attribute) ในการจำแนกประเภทต้นไม้ตัดสินใจที่ได้จากการเรียนรู้ ทำให้ทราบว่าคุณลักษณะใดเป็นตัวกำหนดการจำแนกประเภทของข้อมูล และคุณลักษณะแต่ละตัวมีความสำคัญมากน้อยต่างกันอย่างไร โดยอัลกอริทึมที่ใช้ในการสร้างต้นไม้ตัดสินใจ เช่น C4.5 Algorithm, Iterative Dichotomiser 3 (ID3), Logistic Model Trees (LMT), Classification and Regression Trees (CART) และ Random Forest (RF) เป็นต้น

หลักการพื้นฐานของการสร้างต้นไม้ตัดสินใจ เป็นการสร้างลักษณะจากบนลงล่าง (Top-Down) คือ เริ่มจากการสร้างรากของต้นไม้ ก่อนแล้วจึงแตกกิ่งไปจนถึงใบ โครงสร้างของต้นไม้ตัดสินใจประกอบด้วย

- โหนด (Node) คือ คุณลักษณะต่างๆ ของข้อมูล ซึ่งเมื่อข้อมูลใดๆ ตกลงมาที่โหนด จะใช้คุณลักษณะนี้เป็นตัวตัดสินใจว่าข้อมูลจะไปในทิศทางใด โดยโหนดที่เป็นจุดเริ่มต้นของต้นไม้ เรียกว่าโหนดราก (Root node)
- กิ่ง (Branch, Link) เป็นค่าของคุณลักษณะต่างๆในโหนดที่แตกออกมา ซึ่งโหนดจะแตกกิ่งเป็นจำนวนเท่ากับจำนวนของคุณลักษณะในโหนดนั้น
- โหนดใบ (Leaf node) คือ กลุ่มของผลลัพธ์ในการจำแนกประเภทข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต้นไม้ตัดสินใจมีค่าวัดความสามารถของการจัดกลุ่ม 2 กลุ่ม ของแต่ละคุณลักษณะหรือปัจจัย ดังต่อไปนี้

1) Gini Impurity [13] เป็นการวัดความไม่บริสุทธิ์ของกลุ่มข้อมูล หากกลุ่มใดแบ่งแยกออกมาได้ชัดเจน ค่า Gini impurity ก็จะยิ่งต่ำ และเป็นค่าที่บ่งบอกว่าคุณลักษณะหรือปัจจัยใดควรนำมาใช้เป็นคุณลักษณะในการแบ่งกลุ่ม ส่วนมากจะใช้กับอัลกอรึทึม J48 และ CART ดังสมการที่ (2.1)

$$\text{Gini}(t_i) = 1 - \sum_{i=1}^{N} [p(t_i)]^2 \tag{2.1}$$

2) Entropy เป็นการวัดความไม่แน่นอน (Randomness) ของข้อมูล เช่น การโยนเหรียญ ดังนั้นจึงจำเป็นต้องลดความไม่แน่นอนให้น้อยที่สุด เพื่อเพิ่มความแม่นยำในการจัดกลุ่มให้มากที่สุด ค่าคาดคะเนของข้อมูลเป็นค่าที่แยกโดยใช้ ลักษณะประจำของอัลกอริทึม ID3 ดังสมการที่ (2.2)

$$\text{Entropy}(t_i) = 1 - \sum_{i=1}^{N} [p(t_i)]\log_2 p(t_i) \tag{2.2}$$

โดยที่ t_i คือ คุณลักษณะที่นำมาวัดค่า และ Entropy $P(t_i)$ คือ สัดส่วนของจำนวนสมาชิกของกลุ่ม i กับจำนวนสมาชิกทั้งหมดของกลุ่มตัวอย่าง

ซึ่งแต่ละอัลกอริทึมจะให้ผลของโครงสร้างต้นไม้ตัดสินใจที่แตกต่างกันไป ดังนั้น Gini impurity และ Entropy มีจุดประสงค์เหมือนกัน คือ พยายามทำให้การแยกโหนด (Split Node) แต่ละครั้งได้กลุ่มออกมาให้มีความบริสุทธิ์ของกลุ่มใดกลุ่มหนึ่งในตัวแปรเป้าหมาย (Target Variable) มากที่สุด

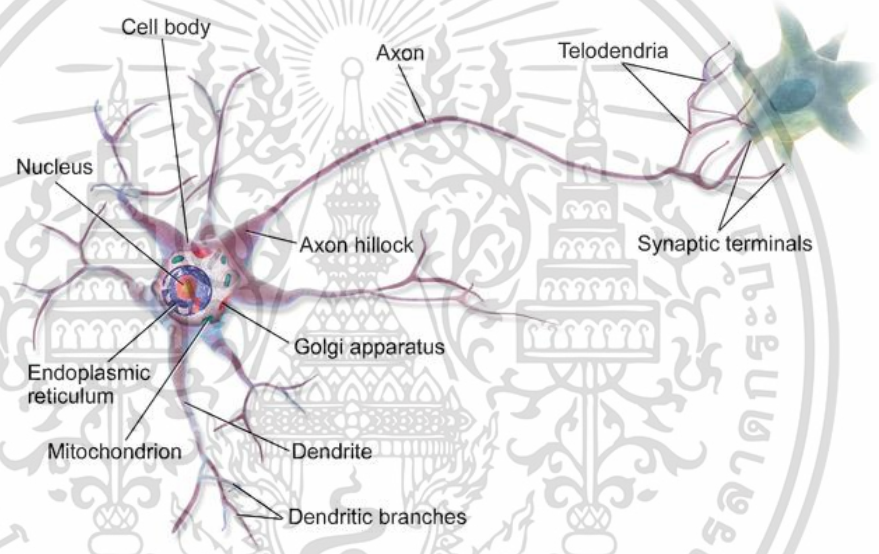
2.2.3 โครงข่ายประสาทเทียม (Neural Network)

โครงข่ายประสาทเทียม (Neural Networks) [5] คือ การสร้างคอมพิวเตอร์ที่จำลองเอาวิธีการทำงานของสมองมนุษย์ เพื่อช่วยให้คอมพิวเตอร์ฟังภาษามนุษย์ได้เข้าใจ อ่านออก และรู้จำได้ ซึ่งอาจเรียกได้ว่าเป็น “สมองกล”

โครงข่ายประสาทเทียมเป็นหนึ่งในเทคนิคของการทำเหมืองข้อมูล (Data Mining) คือ โมเดลทางคณิตศาสตร์ สำหรับประมวลผลสารสนเทศด้วยการคำนวณแบบคอนเนคชันนิสต์ (Connectionist) เพื่อจำลองการทำงานของเครือข่ายประสาทในสมองมนุษย์ ด้วยวัตถุประสงค์ที่จะสร้างเครื่องมือที่มีความสามารถในการเรียนรู้การจดจำรูปแบบ (Pattern Recognition) และการสร้างความรู้ใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

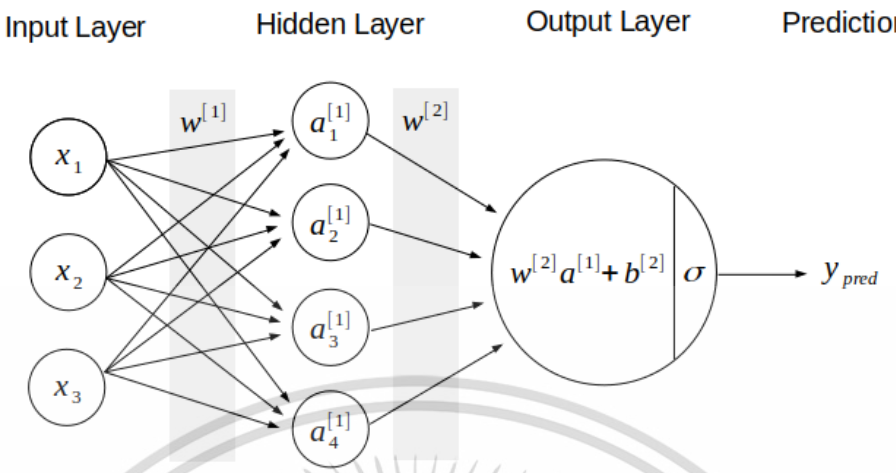
(Knowledge Extraction) เช่นเดียวกับความสามารถที่มีในสมองมนุษย์ แนวคิดเริ่มต้นของเทคนิคนี้ได้มาจากการศึกษาโครงข่ายไฟฟ้าชีวภาพ (Bioelectric Network) ในสมอง แสดงดังรูปที่ 2.2 ซึ่งประกอบด้วยเซลล์ประสาท หรือ นิวรอน (Neurons) และจุดประสานประสาท (Synapses) แต่ละเซลล์ประสาทประกอบด้วยปลายในการรับกระแสประสาท เรียกว่า เดนไดรท์ (Dendrite) เป็นข้อมูลที่ป้อนเข้า และส่วนปลายในการส่งกระแสประสาท เรียกว่า แอกซอน (Axon) เป็นเหมือนหน่วยส่งข้อมูลออกของเซลล์ ซึ่งเซลล์เหล่านี้ทำงานด้วยปฏิกิริยาไฟฟ้าเคมี เมื่อมีการกระตุ้นด้วยสิ่งเร้าภายนอกหรือกระตุ้นด้วยเซลล์ด้วยกัน กระแสประสาทจะวิ่งผ่านเดนไดรท์เข้าสู่นิวเคลียสซึ่งจะเป็นตัวตัดสินว่าต้องกระตุ้นเซลล์อื่น ๆ ต่อหรือไม่ ถ้ากระแสประสาทแรงพอ นิวเคลียสก็จะกระตุ้นเซลล์อื่น ๆ ต่อไปผ่านทางแอคซอน



รูปที่ 2.2 นิวรอน (Neurons) และจุดประสานประสาท (Synapses) แต่ละเซลล์ประสาท

สำหรับในคอมพิวเตอร์เซลล์ประสาท (Neuron) ประกอบด้วยข้อมูลที่ป้อนเข้า และผลลัพธ์ โดยจำลองให้ ข้อมูลที่ป้อนเข้าแต่ละตัวมีน้ำหนัก (Weight) เป็นตัวกำหนดน้ำหนักของข้อมูล โดยเซลล์ประสาทแต่ละหน่วยจะมีค่าเกณฑ์ (Threshold) เป็นตัวกำหนดว่าน้ำหนักรวมของข้อมูลต้องมากเพียงใดจึงจะสามารถส่งผลลัพธ์ไปยังเซลล์ประสาทตัวอื่นได้ กล่าวคือ ถ้าผลรวมมีค่ามากกว่าค่าเกณฑ์ที่กำหนด เซลล์ประสาทจะส่งผลลัพธ์ออกไป ซึ่งผลลัพธ์นี้จะถูกส่งไปเป็นข้อมูลขาเข้าของเซลล์ประสาทอื่น ๆ ที่เชื่อมกันในเครือข่าย (Network) แต่ถ้าค่าน้อยกว่าค่าเกณฑ์ที่กำหนดก็จะไม่เกิดผลลัพธ์ แสดงดังรูปที่ 2.3 เมื่อนำเซลล์ประสาทแต่ละหน่วยมาต่อกัน และให้ทำงานร่วมกันการทำงานนี้จะเหมือนกับปฏิกิริยาเคมีที่เกิดในสมองของมนุษย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 โครงสร้างการเรียนรู้ของ Neural Network

2.3 แนวคิดเกี่ยวกับเซนเซอร์อินเทอร์เน็ตของสรรพสิ่ง

อินเทอร์เน็ตของสรรพสิ่ง (Internet of things) [6] คือ เทคโนโลยีที่ทำให้อุปกรณ์อิเล็กทรอนิกส์ต่างๆ สามารถเชื่อมโยง และรับส่งข้อมูลระหว่างกันได้อย่างง่ายดาย และสามารถสั่งการเพื่อควบคุมอุปกรณ์ต่างๆได้ผ่านทางระบบเครือข่ายอินเทอร์เน็ต ไม่ว่าจะเป็นเมืองอัจฉริยะ (Smart City) บ้านอัจฉริยะ (Smart Home) ระบบโครงข่ายสำหรับส่งไฟฟ้าอัจฉริยะ (Smart Grid) หรืออุปกรณ์สวมใส่อัจฉริยะ (Smart Wearable) เป็นต้น ซึ่งการเชื่อมโยงนั้นจะสามารถเก็บและรวบรวมข้อมูลได้อย่างเป็นระบบ นอกจากนี้แล้วยังมีระบบคลาวด์ที่จัดเก็บและประมวลผลข้อมูลผ่านออนไลน์ หลักการใช้งานสามารถควบคุม กำหนดความเป็นส่วนตัว และสามารถเข้าถึงข้อมูลได้ตลอดเวลา

ประโยชน์ในการใช้งานอินเทอร์เน็ตของสรรพสิ่ง คือการที่สามารถนำเทคโนโลยีมาใช้ได้อย่างมีประสิทธิภาพ ไม่ว่าจะเป็นการรับส่งข้อมูลในรูปแบบดิจิทัล ตลอดเวลา และสามารถทำงานได้ทันที อีกทั้งยังช่วยในการลดภาระงานของบุคลากร รวมไปถึงการเข้าไปตรวจสอบในจุดที่อาจตกหล่น ทั้งนี้เพื่อช่วยลดความเสี่ยงที่อาจจะเกิดขึ้นได้อีกด้วย นอกจากนี้แล้วสิ่งสำคัญในการทำงานร่วมกับอินเทอร์เน็ตของสรรพสิ่ง คือการจัดการข้อมูลและวิเคราะห์สิ่งต่างๆ ได้อย่างทันท่วงทีในรูปแบบเรียลไทม์ไม่ว่าจะเป็นการวิเคราะห์ข้อมูลขนาดใหญ่หรือจัดการข้อมูลระดับย่อย การเชื่อมต่อระบบอุปกรณ์ต่างๆ ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 เทคนิคการสุ่มเพิ่มตัวอย่างกลุ่มน้อย

Synthetic Minority Over-sampling Technique: SMOTE [13] เป็นเทคนิคที่ใช้ในการแก้ปัญหาที่ต้องการจำแนกข้อมูลไม่สมดุล ซึ่งข้อมูลมีจำนวนตัวอย่างแตกต่างกันมากในแต่ละกลุ่ม เมื่อทำการจำแนกประเภทจะทำให้มีการเรียนรู้แต่ข้อมูลกลุ่มที่มาก ผลที่ได้ก็จะจำแนกไปในข้อมูลกลุ่มมาก ดังนั้นวิธี SMOTE เป็นวิธีการเพิ่มจำนวนข้อมูลกลุ่มน้อยให้เพิ่มปริมาณข้อมูลใกล้เคียงกับประเภทที่มีมากที่สุด โดยสุ่มค่าขึ้นมาหนึ่งค่าและหาค่าระยะห่างระหว่างค่าที่เลือกกับทุกๆ ค่า เลือกค่าที่ใกล้เคียงที่สุด เช่น กำหนดไว้ 5 ค่า สุ่มค่าจากที่เลือก 1 ใน 5 หาค่าอยู่ระหว่างค่าที่เลือกตอนแรก และค่าที่สุ่มมาตอนหลัง เพื่อนำค่าที่ได้มาเพิ่มจำนวนข้อมูล ดังสมการที่ (2.3)

$$X_{new} = x_i + (\hat{x}_i - x_i) \times \delta \tag{2.3}$$

โดยที่ x_{new} คือ ข้อมูลใหม่

x_i คือ ข้อมูลที่สุ่มในตอนแรก

$\hat{x}_i$ คือ ข้อมูลที่สุ่มมาอีก เช่น สุ่มมาอีก 5 จุด

δ คือ ค่าสุ่มตั้งแต่ 0-1

2.5 การวัดประสิทธิภาพ

ในการวัดผลการทดลองการทำนายความผิดปกติของเครื่องจักร ในกรณีโมเดลมีลักษณะการจำแนกแบบกลุ่ม หรือตัวแปรตามเป็นลักษณะกลุ่ม สามารถวัดประสิทธิภาพของโมเดลจาก Confusion Matrix [14] โดยมีเกณฑ์การทำนายดังรูปที่ 2.4

	Predicted: Yes	Predicted: No
Actual: Yes	TP (true positive)	FN (false negatives)
Actual: No	FP (false positive)	TN (true negative)

รูปที่ 2.4 การทำนาย Confusion Matrix

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปแสดงการทำนาย Confusion Matrix หากตัวแปรเป้าหมายในการทำนายมี 2 กลุ่ม เช่น Yes และ No จะพบว่ามี 4 สถานการณ์ที่เกิดขึ้นได้ ดังนี้ TP (True Positive) แสดงจำนวนที่ตัวแบบทำนาย Yes ถูกต้อง TN (True negative) แสดงจำนวนที่ตัวแบบทำนาย No ถูกต้อง FP (False Positive) แสดงจำนวนที่ตัวแบบทำนาย Yes ผิด หรือ ทำนาย TP ผิดพลาด และ FN (False Negative) แสดงจำนวนที่ตัวแบบทำนาย No ผิด หรือ ทำนาย TN ผิดพลาด และสามารถคำนวณประสิทธิภาพของโมเดลได้ ดังสมการที่ (2.4) ถึงสมการที่ (2.8) [14]

$$Accuracy = \frac{(TP+TN)}{(TP+FP+TN+FN)} \quad (2.4)$$

$$True\ positive\ rate = \frac{TP}{(TP+FN)} \quad (2.5)$$

$$False\ positive\ rate = \frac{FP}{(TN+FP)} \quad (2.6)$$

$$Precision = \frac{TP}{(TP+FP)} \quad (2.7)$$

$$F-measure = \frac{2\ Recall*Precision}{Recall+Precision} \quad (2.8)$$

จากสมการที่ (2.4) ถึงสมการที่ (2.8) ค่าเริ่มต้นจาก TP, TN, FP และ FN จะถูกใช้ในการคำนวณเพื่อสร้างเป็นเกณฑ์ในการวัดค่าความถูกต้องในการจำแนกประเภทแบบกลุ่ม และคำนวณเพื่อวัดประสิทธิภาพของโมเดล มีทั้งหมดทั้งหมด 5 สมการ ดังนี้

1) ค่าความถูกต้องโดยรวม (Accuracy Score) คือ เกณฑ์การวัดความถูกต้องโดยรวมในการจำแนกประเภทแบบกลุ่ม เพื่อพิจารณาการทำนายของตัวแบบว่าถูกต้องมากน้อยเพียงใด โดยนับจำนวนที่ทำนายถูกทั้งหมดในทุกประเภทเทียบกับจำนวนตัวอย่างทั้งหมด พิจารณาค่าจากเมทริกซ์คอนฟิวชั่น (Confusion Matrix) หรือ สัดส่วนเปอร์เซ็นต์ความถูกต้อง คำนวณได้จากสูตร สมการที่ (2.4)

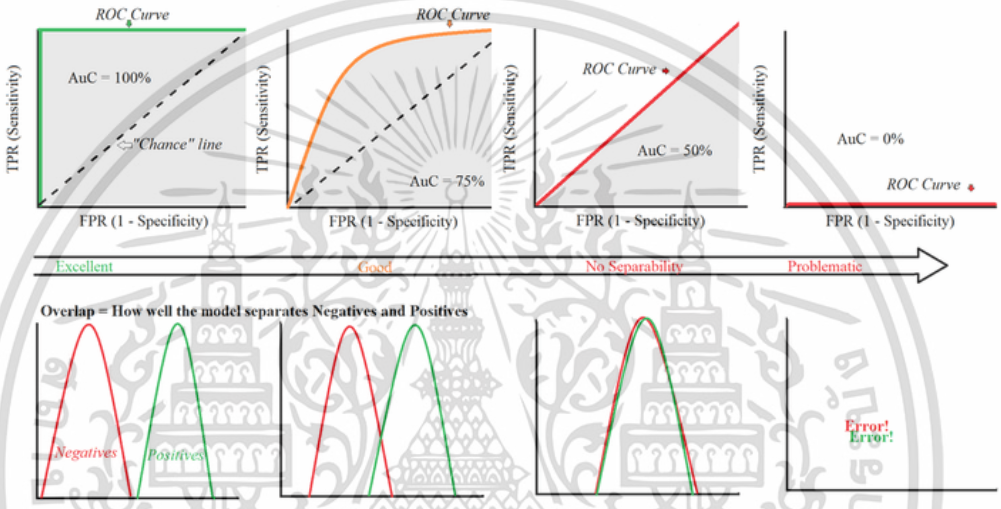
2) Precision คือ ค่าความถูกต้องในการทำนายกลุ่ม Yes ถูกต้อง หารด้วยจำนวนครั้งที่ทายว่า Yes ทั้งหมด หรือเรียกว่า Positive Predictive Value (PPV) คำนวณได้จากสูตร สมการที่ (2.7)

3) Recall คือ ค่าความถูกต้องในการทำนายกลุ่ม Yes ถูกต้อง หารด้วยจำนวน Yes ทั้งหมดของข้อมูล หรือเรียกว่า TPR (True Positive Rate) คำนวณได้จากสูตร สมการที่ (2.5)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) ค่าการวัดเอฟ (F-measure) หรือ F1-Score เป็นเกณฑ์การวัดความถูกต้องในการจำแนกประเภทแบบกลุ่ม ใช้ค่าเฉลี่ยฮาร์โมนิค (Harmonic Mean) ระหว่าง True Positive Rate หรือที่เรียกว่า Recall และ Precision คำนวณได้จากสูตร สมการที่ (2.8)

5) ROC AUC score [14] คือเกิดจากการหาพื้นที่ใต้กราฟที่มีการพลอตระหว่าง TPR (True Positive Rate) และ FPR (False Positive Rate) ที่หาจากค่าความน่าจะเป็น (Probability Score) ดังรูปที่ 2.5



รูปที่ 2.5 ประสิทธิภาพการทำนาย ROC AUC score

จากรูปที่ 2.5 พื้นที่ใต้กราฟ คือ ค่า AUC (Area Under The Curve) ซึ่งการเลือกจุดตัดบนกราฟที่เหมาะสมที่สุด (Optimal Cut-off Point) ที่จะได้ค่า AUC มากที่สุด และให้ผลลัพธ์ดีที่สุด คือ จุดที่อยู่ใกล้มุมซ้ายบนมากที่สุด ซึ่งจะเป็นจุดที่ TRP (Sensitivity) สูงและ FPR (Specificity) สูงด้วยเช่นกัน โดยมาเกณฑ์ของ AUC Score ดังนี้

1) AUC Score = 0.50 ไม่ต่างอะไรกับการเดาสุ่ม
2) AUC Score > 0.70 คือเกณฑ์มาตรฐานสำหรับโมเดลส่วนใหญ่
3) AUC Score > 0.80 โมเดลทำงานได้ดี
4) AUC Score > 0.90 โมเดลทำงานได้ดีมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 วิธีการวิเคราะห์ความแม่นตรงของตัวแบบ (K-fold cross-validation)

การตรวจสอบไขว้กัน (Cross Validation) [13] เป็นวิธีการตรวจสอบค่าความผิดพลาดในการคาดการณ์ของตัวแบบ โดยพื้นฐานของวิธีการตรวจสอบไขว้กันคือ การสุ่มตัวอย่าง โดยเริ่มจากแบ่งชุดข้อมูลออกเป็นส่วนๆ และนำบางส่วนจากชุดข้อมูลนั้นมาตรวจสอบผลลัพธ์จากการทำการตรวจสอบไขว้กัน มักถูกใช้เป็นตัวเลือกในการกำหนดตัวแบบ ในกรณีการทำ K-fold cross-validation จะแบ่งข้อมูลออกเป็น K ชุด เท่าๆ กัน เช่น 5 ชุด จะทำการคำนวณค่าความผิดพลาด 5 รอบ โดยแต่ละรอบการคำนวณ ข้อมูลหนึ่งชุดจากข้อมูลทั้งหมด 5 ชุด จะถูกเลือกออกมาเพื่อเป็นข้อมูลทดสอบ และข้อมูลอีก 4 ชุดจะถูกใช้เป็นข้อมูลสำหรับการเรียนรู้ ดังรูปที่ 2.6

lteration 1: train on	2	3	4	5	test on	1
lteration 1: train on	1	3	4	5	test on	2
lteration 1: train on	1	2	4	5	test on	3
lteration 1: train on	1	2	3	5	test on	4
lteration 1: train on	1	2	3	4	test on	5

รูปที่ 2.6 5-fold cross-validation

2.7 งานวิจัยที่เกี่ยวข้อง

การศึกษาเรื่อง การบำรุงรักษาเชิงคาดการณ์ของเครื่องจักรในอุตสาหกรรมด้วยการเชื่อมโยงข้อมูลของเซนเซอร์อินเทอร์เน็ตของสรรพสิ่ง ซึ่งยังเป็นประเด็นหนึ่งที่อุตสาหกรรมการผลิตให้ความสนใจเป็นอย่างมาก จากการได้ศึกษางานวิจัยที่เกี่ยวข้องสามารถสรุปได้ดังนี้

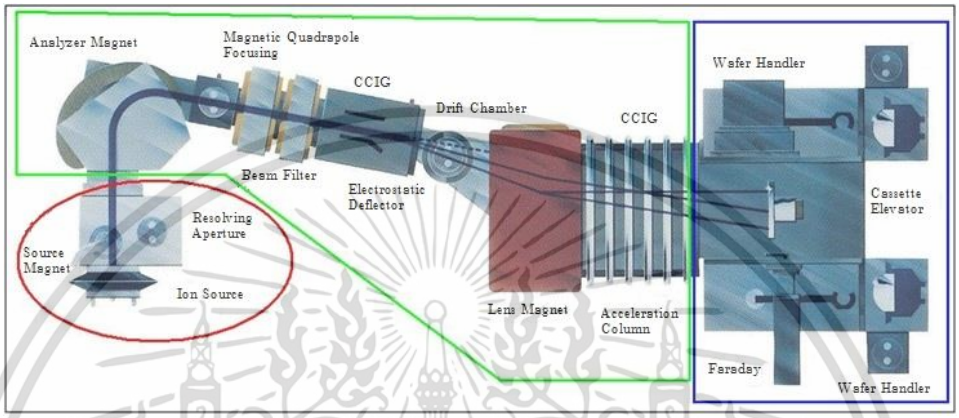
2.7.1 การบำรุงรักษาเชิงคาดการณ์ของเครื่องจักรอุตสาหกรรม

Susto, G. A., Schirru, A., Pampuri, S., McLoone, S., & Beghi, A. (2015) [7] ในงานวิจัยนี้ผู้วิจัยได้นำเสนอวิธีการเรียนรู้ของเครื่อง จากการจำลองปัญหาการบำรุงรักษาเครื่องจักรในกระบวนการผลิตเซมิคอนดักเตอร์ เช่น อายุการใช้งานของอุปกรณ์ต่างๆ และสิ่งแวดล้อม เป็นต้น โดยใช้โมดูลการจำแนกกลุ่มหลายประเภท (Multiple classification modules)

จากการสำรวจเครื่องจักร ผู้วิจัยพบว่าปัจจัยหลายประการที่อาจส่งผลต่อระยะเวลาการทำงานของเครื่องจักรในกระบวนการผลิต เช่น ค่าความดัน แรงดันไฟ และกระแสไฟฟ้า เป็นต้น นำไปสู่การ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บำรุงรักษาเชิงคาดการณ์ (Predictive Maintenance) ซึ่งเป็นกลยุทธ์ที่โดดเด่นในการจัดการกับปัญหาการบำรุงรักษา อีกทั้งสามารถใช้กับปัญหาข้อมูลที่มีมิติสูง และปัญหาของเซนเซอร์ได้อีกด้วย เพื่อป้องกันการหยุดการทำงานของเครื่องจักรโดยไม่คาดคิด อีกทั้งยังลดระยะเวลาในการทำงานของผู้ผลิตสินค้าอุปกรณ์ต่างๆภายในเครื่องจักรสามารถแสดงได้ดังรูปที่ 2.7



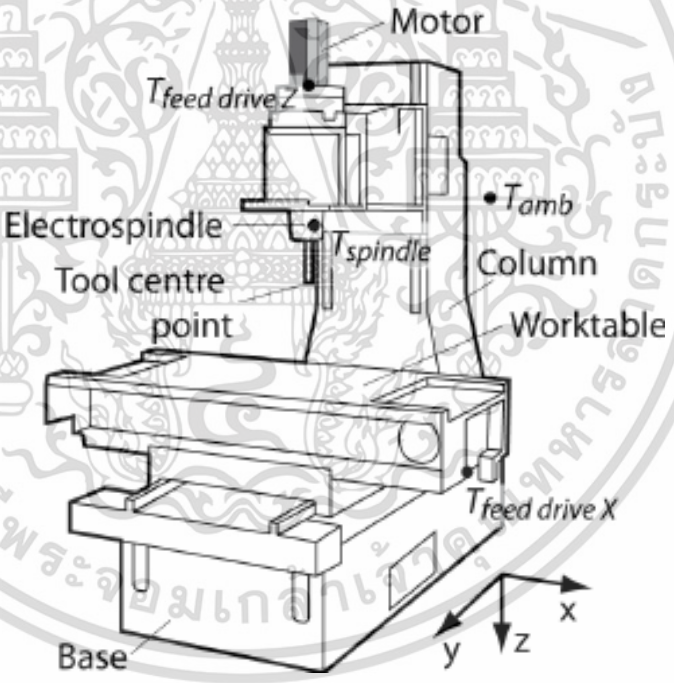
รูปที่ 2.7 อุปกรณ์ต่างๆ ภายในเครื่องจักร

การแก้ปัญหาการบำรุงรักษาเชิงคาดการณ์ ผู้วิจัยนำเสนอเทคโนโลยีการเรียนรู้ของเครื่องแบบการเรียนรู้แบบมีผู้สอน โดยใช้อัลกอริทึมการจำแนกกลุ่มหลายประเภท ที่รู้จักและใช้กันอย่างแพร่หลาย 2 โมเดล ได้แก่ Support Vector Machines (SVMs) และ K-Nearest Neighbors (K-NN) นำมาเปรียบเทียบประสิทธิภาพของแต่ละโมเดล เพื่อเป็นแนวทางในการตัดสินใจในอุตสาหกรรม ในด้านที่สำคัญ เช่น การจัดกำหนดการต่างๆ การจัดการการบำรุงรักษา และการปรับปรุงคุณภาพ ซึ่งผลลัพธ์กรณีศึกษานี้แสดงให้เห็นว่าเมื่อนำไปทดสอบกับเครื่องจักรในกระบวนการผลิตเซมิคอนดักเตอร์ เทคนิค SVMs มีประสิทธิภาพที่เหนือกว่าโมเดลประเภท K-NN ผลลัพธ์ที่ได้ทำให้เกิดความเข้าใจในกระบวนการผลิตมากขึ้น เพื่อนำไปปรับปรุงการตัดสินใจในการจัดการการบำรุงรักษาในแง่ของการลดต้นทุนการดำเนินงาน และสามารถนำไปใช้กับปัญหาการบำรุงรักษาใดๆ ได้

Marina Paolanti, Luca Romeo, Andrea Felicetti, Adriano Mancini, Emanuele Frontoni (July 2018) [8] งานวิจัยนี้ ผู้วิจัยศึกษาเรื่อง แนวทางการเรียนรู้ของเครื่องจักรสำหรับการบำรุงรักษาเชิงคาดการณ์ในอุตสาหกรรมยุค 4.0 การตรวจสอบสภาพความพร้อมในการใช้งาน กับการบำรุงรักษาเชิง-คาดการณ์ของมอเตอร์ไฟฟ้าและอุปกรณ์อื่นๆภายในอุตสาหกรรม ซึ่งช่วยหลีกเลี่ยงความสูญเสียที่เกิดขึ้นในการผลิตอย่างรุนแรงอันเป็นผลมาจากความล้มเหลวของมอเตอร์ที่ไม่คาดคิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผู้วิจัยได้ทำการทดสอบจากตัวอย่างเครื่องตัดภายในอุตสาหกรรม โดยรวบรวมข้อมูลจากอุปกรณ์ควบคุมการทำงานของเครื่องจักร (PLC-Programmable Logic Controller) เช่น เซนเซอร์ แกนหมุน เวลา มอเตอร์ เป็นต้น โดยรายละเอียดเครื่องตัดภายในอุตสาหกรรมสามารถแสดงได้ดังรูปที่ 2.8 การนำแนวทางการเรียนรู้ของเครื่องมาใช้ในการวิเคราะห์ระบบของข้อมูล และเปรียบเทียบกับการวิเคราะห์เครื่องมือ โดยจำลองด้วยโมเดล Random Forest แบบการจำแนกประเภทข้อมูล ซึ่งเป็นการสร้างต้นไม้ติดสินใจหลายๆโมเดลเข้าด้วยกัน หลักการ คือ การฝึกฝนโมเดลที่เหมือนกันหลายๆครั้งบนข้อมูลชุดเดียวกัน โดยแต่ละครั้งของการฝึกฝนจะเลือกส่วนของข้อมูลที่ไม่เหมือนกัน แล้วนำผลการทำนายของโมเดลเหล่านั้นมาโหวต (Voting) เพื่อดูว่ากลุ่มไหนถูกเลือกมากที่สุด ภายใต้เงื่อนไขที่ว่า โมเดลผู้ทำนายแต่ละตัวจะต้องเรียนรู้อย่างเป็นอิสระต่อกันให้มากที่สุด ข้อดี คือ เมื่อนำต้นไม้ติดสินใจหลายๆ โมเดลมาทำนายร่วมกันจะทำให้ประสิทธิภาพในการทำงานสูงขึ้น และแม่นยำมากกว่าการสร้างต้นไม้ตัดสินใจแบบเดี่ยว



รูปที่ 2.8 เครื่องตัดที่ใช้ในกระบวนการผลิต

จากการทดสอบวิธีบำรุงรักษาเชิงคาดการณ์ ด้วยโมเดล Random Forest แบบการจำแนกประเภทข้อมูล บน Azure Machine Learning Studio ผลลัพธ์เบื้องต้นแสดงถึงแนวทางในการคาดการณ์สถานะของเครื่องจักรต่างๆ ด้วยความแม่นยำสูงถึง 95% ในชุดข้อมูล 530,731 ข้อมูล และการอ่านข้อมูลบนคุณสมบัติต่างๆของเครื่องด้วยตัวแปร 15 แบบ ที่รวบรวมแบบเรียลไทม์จากเครื่องตัดที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

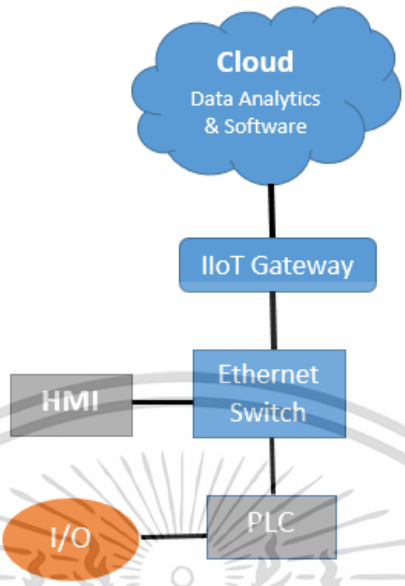
ทดสอบ ดังนั้นการบำรุงรักษาเชิงคาดการณ์จึงเป็นกลยุทธ์ที่นำไปใช้ได้จริงเมื่อต้องรับมือกับปัญหาการบำรุงรักษา เนื่องจากความต้องการที่เพิ่มขึ้นในกระบวนการผลิต นำไปสู่การป้องกันการหยุดทำงานโดยไม่คาดคิด และลดต้นทุนที่เกี่ยวข้อง

2.7.2 อินเทอร์เน็ตของสรรพสิ่งกับอุตสาหกรรมการผลิต

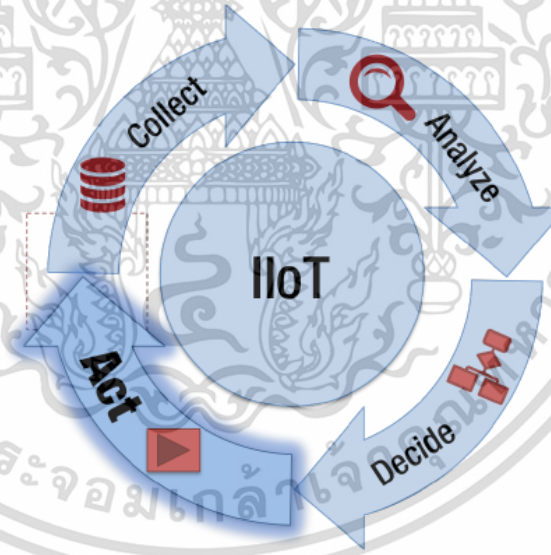
Ásgeir Halldórsson (2017) [9] ในงานวิจัยผู้วิจัยศึกษาเรื่อง อินเทอร์เน็ตของสรรพสิ่งสำหรับการวัดประสิทธิผลโดยรวมของอุปกรณ์ (Overall Equipment Effectiveness : OEE) โดยจุดมุ่งหมายของงานวิจัยนี้ คือ การรวบรวมรายการข้อกำหนดต่างๆสำหรับการเชื่อมต่อระหว่างคอมพิวเตอร์กับอุปกรณ์ภายในอุตสาหกรรม โดยรวบรวมข้อมูลและส่งไปยังระบบคลาวด์ เพื่อตอบสนองทุกความต้องการของผู้ใช้งาน อีกทั้งสามารถวิเคราะห์ข้อมูลเพื่อปรับปรุงประสิทธิภาพโดยรวมของอุปกรณ์ (OEE) ช่วยให้บริษัทใช้วิธีการผลิตอย่างมีประสิทธิภาพและประสิทธิผลมากขึ้น

ทุกวันนี้โรงงานอุตสาหกรรมหลายแห่งดำเนินการผ่านคอมพิวเตอร์ โดยใช้ข้อมูลที่มีอยู่เพื่อเพิ่มการผลิตให้สูงสุด โดยใช้การจดจำรูปแบบและการเรียนรู้ของเครื่อง หรือที่เรียกว่าปัญญาประดิษฐ์ อย่างไรก็ตามอุปกรณ์มากถึง 85% ภายในระบบแยกจากกัน หรือไม่เชื่อมต่อกัน การรวบรวมข้อมูลทั้งหมดต้องใช้เงินเป็นจำนวนมาก และใช้เวลานาน การรวบรวมข้อมูลสำหับผู้วิจัยถือเป็นสิ่งที่ยากที่สุด เนื่องจากมีโปรโตคอลการสื่อสารอินเทอร์เน็ตของสรรพสิ่ง และข้อจำกัดของเครือข่ายหลายประเภทที่ต้องแก้ไข เช่น OPC UA, HTTP, REST, JSON, MQTT, CoAP, DDS และ AMQP เป็นต้น นอกจากนี้ ปัญหาของคอมพิวเตอร์อุตสาหกรรมที่พบบ่อย คือ คอมพิวเตอร์มักไม่สามารถเข้าถึงการสื่อสารภายนอกได้ จึงจำเป็นต้องมีเกตเวย์เป็นตัวกลางในการอ่านข้อมูล และส่งข้อมูลไปยังคลาวด์ เนื่องจากการวัดประสิทธิผลโดยรวมของอุปกรณ์ไม่ได้เป็นสมการทางคณิตศาสตร์ที่ซับซ้อน แต่เป็นการรวบรวมข้อมูลง่ายๆระหว่างการทำงาน และการวิเคราะห์แนวโน้มอย่างง่าย เช่น การเปรียบเทียบผลผลิตระหว่างวัน เวลา หรือช่วงเวลาของวัน ตำแหน่งที่ใช้เกตเวย์เพื่อเชื่อมต่อระบบควบคุมแบบแยกส่วนกับคลาวด์ และการไหลของอินเทอร์เน็ตของสรรพสิ่งสามารถแสดงได้ดังรูปที่ 2.9 และรูปที่ 2.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 ตำแหน่งที่ใช้เกตเวย์เพื่อเชื่อมต่อระบบควบคุมแบบแยกส่วนกับคลาวด์



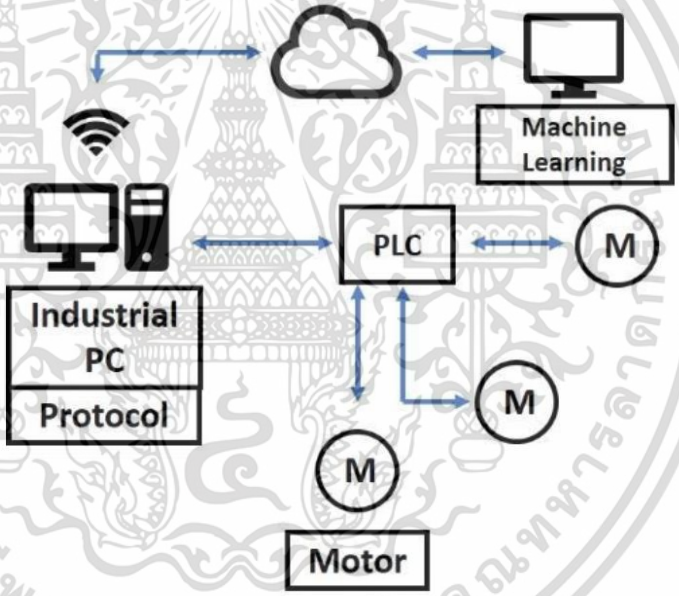
รูปที่ 2.10 การไหลของอินเทอร์เน็ตของสรรพสิ่ง

ผู้วิจัยชี้ให้เห็นถึงพื้นฐานของการพัฒนาผลิตภัณฑ์อนาคตในเรื่องของการเชื่อมต่ออุปกรณ์ต่างๆในอุตสาหกรรมการผลิต และวิธีรวบรวมข้อมูลด้วยวิธีการต่างๆ โดยมีวัตถุประสงค์เพื่อให้การวัดประสิทธิผลโดยรวมของอุปกรณ์แบบเรียลไทม์ ซึ่งเป็นเครื่องมือที่มีประสิทธิภาพเหมาะแก่การขับเคลื่อน ปรับปรุงและพัฒนาสำหรับผู้ใช้ในอนาคต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Ameeth Kanawaday, Aditya Sane (November 2017) [10] ผู้วิจัยได้ศึกษาเรื่อง อินเทอร์เน็ตของสรรพสิ่งในเชิงอุตสาหกรรม คือการใช้เทคโนโลยีอินเทอร์เน็ตของสรรพสิ่งในอุตสาหกรรมการผลิตโดยควบคุม และเก็บข้อมูลเครื่องจักรจากเซนเซอร์ต่างๆ นำไปวิเคราะห์เพื่อให้ได้ข้อมูลที่เป็นประโยชน์กับผู้ใช้งานมากที่สุดในด้านประสิทธิภาพการทำงานของเครื่องจักร และค่าใช้จ่ายในการบำรุงรักษาที่เกี่ยวข้อง บล็อกไดอะแกรมจำลองกระบวนการส่งข้อมูลอุปกรณ์ต่างๆไปยังคลาวด์ แสดงได้ดังรูปที่ 2.11

ผู้วิจัยได้ทดสอบด้วยแบบจำลองการคาดการณ์แบบค่าเฉลี่ยเคลื่อนที่แบบบูรณาการการถดถอยอัตโนมัติ (AutoRegressive Integrated Moving Average : ARIMA) กับข้อมูลอนุกรมเวลาที่รวบรวมจากเซนเซอร์ต่างๆ จากเครื่องตัดในอุตสาหกรรม ภายใต้โมเดลการเรียนรู้แบบมีผู้สอน และคาดการณ์สถานะของเครื่องจักรในอนาคตที่อาจนำไปสู่ความผิดปกติ หรือเพื่อคาดการณ์ความล้มเหลวที่อาจเกิดขึ้นและข้อบกพร่องด้านคุณภาพ ซึ่งจะช่วยปรับปรุงกระบวนการผลิตได้ดียิ่งขึ้น



รูปที่ 2.11 บล็อกไดอะแกรมสำหรับกระบวนการส่งข้อมูลอุปกรณ์ต่างๆไปยังคลาวด์

วัตถุประสงค์การดำเนินการทดสอบ คือ เปรียบเทียบการเรียนรู้แบบมีผู้สอน แบบการจำแนกประเภทข้อมูล (Classification) ทั้งหมด 4 โมเดล ได้แก่ Naive Bayes, Support Vector Machine, CART และ Deep Neural Network โดยที่แต่ละโมเดลจะได้รับการฝึกฝนข้อมูลจากข้อมูลในอดีตเช่นเดียวกับโมเดล ARIMA เพื่อทำนายค่าพารามิเตอร์ในรอบการผลิตในอนาคต และค่าเหล่านั้นจะถูกป้อนไปยังแบบจำลองการเรียนรู้แบบมีผู้สอนเพื่อจัดประเภทต่อไป โดยจะแบ่งชุดข้อมูลสำหรับการฝึกฝน 70% การตรวจสอบความถูกต้อง (Cross Validation) 10% และการทดสอบ 20% ซึ่งผลลัพธ์ที่ได้ คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โมเดล Naive Bayes มีค่าความแม่นยำ 96.61% โมเดล Support Vector Machine มีค่าความแม่นยำในการทำนาย 95.52% โมเดล CART มีค่าความแม่นยำในการทำนาย 94.46% และโมเดล Deep Neural Network มีค่าความแม่นยำในการทำนายสูงสุดถึง 98.69% ดังนั้นโมเดลการเรียนรู้แบบมีผู้สอนสามารถใช้เพื่อค้นหาข้อมูลเชิงลึก และคาดการณ์เหตุการณ์ต่างๆในเครื่องจักรได้ ทำให้แน่ใจว่ากระบวนการผลิตดำเนินไปอย่างมีประสิทธิภาพ โดยลดค่าใช้จ่ายในการบำรุงรักษา และลดการเสื่อมคุณภาพของผลิตภัณฑ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
วิธีการดำเนินการ

บทนี้อธิบายถึงขั้นตอนการค้นคว้าอิสระที่เริ่มตั้งแต่การเลือกชุดข้อมูลที่ใช้ในการทดลอง อัลกอริทึมที่ใช้ในการหาความผิดปกติของเครื่องจักร วิธีทำนายความผิดปกติของเครื่องจักร วิธีวัดความถูกต้องในการทำนายความผิดปกติของเครื่องจักร เครื่องมือที่ใช้ในการทดลอง และการออกแบบกระบวนการทดลอง

3.1 ขั้นตอนของการศึกษาค้นคว้าอิสระ

ขั้นตอนของการศึกษาค้นคว้าอิสระ ประกอบด้วย 4 ส่วน คือ

1) ขั้นตอนการเตรียมข้อมูล (Data Preparation) เป็นขั้นตอนที่เลือกข้อมูลที่ใช้ในการทดลอง ซึ่งมาจากเก็บรวบรวมข้อมูล (Data Collection) เซนเซอร์ของเครื่องจักรแต่ละตัวภายในโรงงานอุตสาหกรรม

2) อัลกอริทึมที่ใช้ในการหาค่าความผิดปกติของเครื่องจักรในโรงงานอุตสาหกรรม

3) วิธีการทำนายความผิดปกติของเครื่องจักรในโรงงานอุตสาหกรรม

4) การวัดผลการทดลอง (Evaluation) เป็นส่วนที่ตรวจสอบว่าการทำนายความผิดปกติของเครื่องจักรในโรงงานอุตสาหกรรมไปในทิศทางเดียวกับความผิดปกติที่เกิดขึ้นจากสถานการณ์จริงหรือไม่

3.1.1 ขั้นตอนการเตรียมข้อมูล

เนื่องจากในการศึกษาค้นคว้าอิสระนี้ต้องการข้อมูลเซนเซอร์ของเครื่องจักรในโรงงานอุตสาหกรรมหลายประเภท เช่น อุณหภูมิ อัตราการไหล ความชื้น โดยชุดข้อมูลที่นำมาทดลองต้องมีองค์ประกอบของข้อมูลดังกล่าวครบถ้วน และจากการศึกษางานวิจัยที่เกี่ยวข้อง พบว่างานวิจัยที่ทำในลักษณะคล้ายกันใช้ข้อมูลจากข้อมูลที่เกิดขึ้นจริง หรือจำลองชุดข้อมูลขึ้นมาเอง (Simulation Data) ดังนั้นผู้ค้นคว้าอิสระจึงได้ศึกษา และวิเคราะห์หาชุดข้อมูลที่สามารถนำไปประยุกต์ใช้กับการทดลอง โดยชุดข้อมูลที่นำมาทดลองเป็นระบบปั๊มน้ำ (Water Pump System) ในโรงงานอุตสาหกรรมแห่งหนึ่ง มีทั้งหมด 51 เซนเซอร์ แสดงดังตารางที่ 3.1 จำนวนข้อมูลทั้งหมด 220,320 ข้อมูล และในชุดข้อมูลเซนเซอร์ภายในระบบปั๊มน้ำ เช่น อุณหภูมิ อัตราการไหล แรงสั่นสะเทือน แรงดัน เป็นต้น ช่วงเวลาในการเก็บข้อมูลอยู่ในช่วงวันที่ 1 เมษายน พ.ศ. 2561 ถึง 30 สิงหาคม พ.ศ. 2561 ข้อมูลที่นำมาทดลองได้จาก Kaggle [11] เพื่อนำมาวิเคราะห์ และทำนายสถานะการทำงานของเครื่องจักร แสดงดังรูปที่ 3.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.1 รายละเอียดคุณลักษณะที่นำมาสร้างตัวแบบ

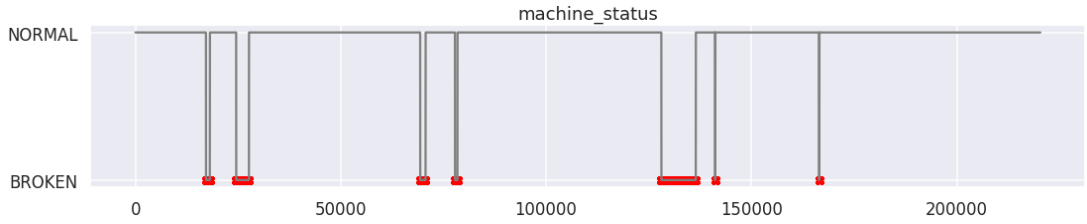
ลำดับ	ชื่อคุณลักษณะ	คำอธิบาย
1	SENSOR_00	การสั่นสะเทือนของมอเตอร์
2	SENSOR_01	ความถี่มอเตอร์ A
3	SENSOR_02	ความถี่มอเตอร์ B
4	SENSOR_03	ความถี่มอเตอร์ C
5	SENSOR_04	ความเร็วมอเตอร์
6	SENSOR_05	กระแสไฟของมอเตอร์
7	SENSOR_06	กำลังไฟฟ้าสุทธิของมอเตอร์ (Motor Active Power)
8	SENSOR_07	กำลังงานที่ปรากฏของมอเตอร์ (Motor Apparent Power)
9	SENSOR_08	กำลังรีแอคทีฟของมอเตอร์ (Motor Reactive Power)
10	SENSOR_09	กำลังขับของมอเตอร์ (Motor Shaft Power)
11	SENSOR_10	กระแสไฟมอเตอร์เฟส A
12	SENSOR_11	กระแสไฟมอเตอร์เฟส B
13	SENSOR_12	กระแสไฟมอเตอร์เฟส C
14	SENSOR_13	การสั่นสะเทือนของมอเตอร์ตัวขับ (Motor Coupling Vibration)
15	SENSOR_14	แรงดันไฟฟ้าของมอเตอร์เฟส AB
16	SENSOR_15	แรงดันไฟฟ้าของมอเตอร์เฟส BC
17	SENSOR_16	แรงดันไฟฟ้าของมอเตอร์เฟส CA
18	SENSOR_17	การสั่นสะเทือนโดยรอบของปั๊ม (Pump Casing Vibration)
19	SENSOR_18	ความเร็วรอบใบพัดของปั๊ม 1 (Pump Impeller Speed No.1)
20	SENSOR_19	ความเร็วรอบใบพัดของปั๊ม 2 (Pump Impeller Speed No.2)
21	SENSOR_20	ความเร็วรอบใบพัดของปั๊ม 3 (Pump Impeller Speed No.3)
22	SENSOR_21	ความเร็วรอบใบพัดของปั๊ม 4 (Pump Impeller Speed No.4)
23	SENSOR_22	ความเร็วรอบใบพัดของปั๊ม 5 (Pump Impeller Speed No.5)
24	SENSOR_23	ความเร็วรอบใบพัดของปั๊ม 6 (Pump Impeller Speed No.6)
25	SENSOR_24	ความเร็วรอบใบพัดของปั๊ม 7 (Pump Impeller Speed No.7)
26	SENSOR_25	ความเร็วรอบใบพัดของปั๊ม 8 (Pump Impeller Speed No.8)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.1(ต่อ) รายละเอียดคุณลักษณะที่นำมาสร้างตัวแบบ

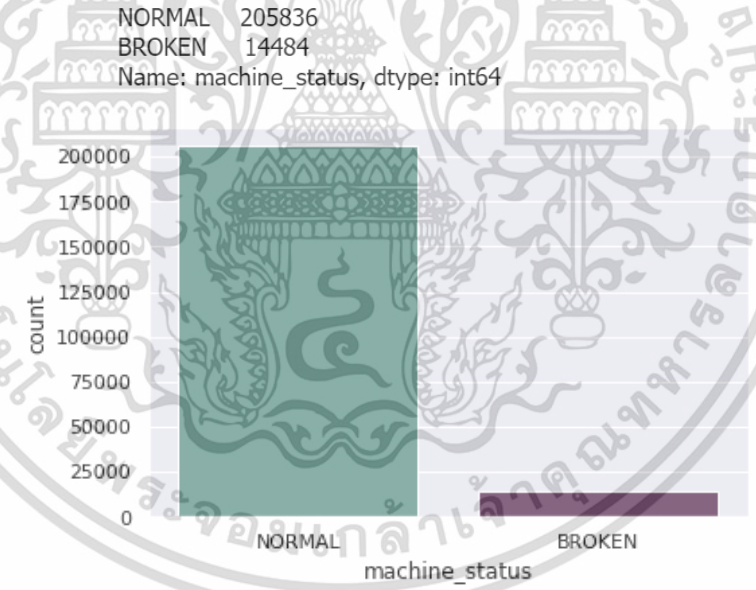
ลำดับ	ชื่อคุณลักษณะ	คำอธิบาย
27	SENSOR_26	ความเร็วรอบใบพัดของปั๊ม 9 (Pump Impeller Speed No.9)
28	SENSOR_27	ความเร็วรอบใบพัดของปั๊ม 10 (Pump Impeller Speed No.10)
29	SENSOR_28	ความเร็วรอบใบพัดของปั๊ม 11 (Pump Impeller Speed No.11)
30	SENSOR_29	ความเร็วรอบใบพัดของปั๊ม 12 (Pump Impeller Speed No.12)
31	SENSOR_30	ความเร็วรอบใบพัดของปั๊ม 13 (Pump Impeller Speed No.13)
32	SENSOR_31	ความเร็วรอบใบพัดของปั๊ม 14 (Pump Impeller Speed No.14)
33	SENSOR_32	ความเร็วรอบใบพัดของปั๊ม 15 (Pump Impeller Speed No.15)
34	SENSOR_33	รอบอัตราการไหลขาเข้าของปั๊ม (Pump Inlet Flow)
35	SENSOR_34	รอบอัตราการไหลขาออกของปั๊ม (Pump Discharge Flow)
36	SENSOR_35	รอบอัตราการไหลขาเข้าของปั๊มสำรอง (Pump Inlet Flow)
37	SENSOR_36	ระดับน้ำมันหล่อลื่นปั๊มน้ำมันหล่อลื่นเหนือศีรษะ
38	SENSOR_37	อุณหภูมิปั๊มน้ำมันหล่อลื่น (ขากลับ)
39	SENSOR_38	อุณหภูมิปั๊มน้ำมันหล่อลื่น (ขาเข้า)
40	SENSOR_39	อุณหภูมิแรงขับแบริ่งของปั๊ม ขณะที่ไม่ใช้งาน 1
41	SENSOR_40	อุณหภูมิแรงขับปลายแบริ่งของปั๊ม 1
42	SENSOR_41	อุณหภูมิแรงขับปลายแบริ่งของปั๊ม 2
43	SENSOR_42	อุณหภูมิแรงขับแบริ่งของปั๊ม ขณะที่ไม่ใช้งาน 2
44	SENSOR_43	อุณหภูมิแรงขับปลายแบริ่งของปั๊ม 3
45	SENSOR_44	อุณหภูมิแรงขับปลายแบริ่งของปั๊ม 5
46	SENSOR_45	อุณหภูมิแรงขับปลายแบริ่งของปั๊ม 6
47	SENSOR_46	อุณหภูมิแรงขับปลายแบริ่งของปั๊ม 7
48	SENSOR_47	ความดันขาเข้าของปั๊ม (Pump Inlet Pressure)
49	SENSOR_48	อุณหภูมิขาเข้าของปั๊ม
50	SENSOR_49	ความดันขาออกของปั๊ม 1 (Pump Discharge Pressure 1)
51	SENSOR_50	ความดันขาออกของปั๊ม 2 (Pump Discharge Pressure 2)
52	Machine_Status	BROKEN = เครื่องจักรเสีย NORMAL = เครื่องจักรทำงานปกติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1 สถานะการทำงานของเครื่องจักร

จากตารางที่ 3.1 และรูปที่ 3.1 แสดงตัวแปรเป้าหมาย ข้อมูลสถานะการทำงานของเครื่องจักร ที่เกิดจากการทำงานของเซนเซอร์ทั้งหมด 51 เซนเซอร์ ซึ่งมีข้อมูลจำนวนทั้งหมด 220,320 ข้อมูล ข้อมูลวันที่ 1 เมษายน พ.ศ. 2561 – 30 สิงหาคม พ.ศ. 2561 ข้อมูลที่นำมาทดลองได้จาก Kaggle [11] ซึ่งจากการศึกษาความสัมพันธ์ระหว่างการทำงานของเครื่องจักร และเซนเซอร์ พบว่ามีข้อจำกัดในด้านการคำนวณหาความผิดปกติของเครื่องจักร เนื่องจากข้อมูลไม่ซับซ้อนเท่าที่ควร คือ ตัวแปรเป้าหมาย (ปกติ/ผิดปกติ) ที่มีจำนวนข้อมูลไม่เท่ากัน และอาจนำไปสู่การเป็นข้อมูลที่ไม่สมดุล (Imbalanced data) ดังรูปที่ 3.2



รูปที่ 3.2 จำนวนครั้งของสถานะการทำงานปกติ และผิดปกติของเครื่องจักร

จากรูปที่ 3.2 สถานะเครื่องจักรจำนวน 220,320 ข้อมูล ประเภทของข้อมูลเป็นตัวเลขที่มีขนาด 64 bit พบว่าตัวแปรเป้าหมายเป็นข้อมูลที่ไม่สมดุล เนื่องจากสถานะเครื่องจักรปกติ มีจำนวน 205,836 ข้อมูล คิดเป็น 93.43% ของข้อมูลทั้งหมด และสถานะเครื่องจักรผิดปกติ มี 14,484 ข้อมูล คิดเป็น 6.57% ของข้อมูลทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.2 สถิติชุดข้อมูลทั้งหมด

Sensor	count	mean	std	min	25%	50%	75%	max
sensor_00	210112	2.372220816	0.412227352	0	2.438831	2.456539	2.499826	2.549016
sensor_01	219951	47.5916107	3.296665578	0	46.31076	48.13367844	49.47916	56.72743
sensor_02	220301	50.86739215	3.666820373	33.15972	50.39062	51.6493	52.77777	56.03299
sensor_03	220301	43.75248103	2.418886574	31.64062	42.83853912	44.22742844	45.3125	48.22049
sensor_04	220301	590.6739359	144.0239119	2.798032	626.6204	632.638916	637.6157227	800
sensor_05	220301	73.39641389	17.29824702	0	69.97626	75.57679	80.91215	99.99988
sensor_06	215522	13.50153744	2.163736194	0.01446759	13.34635	13.64294	14.53993	22.25116
sensor_07	214869	15.84315187	2.201155272	0	15.90712	16.16753	16.42795	23.59664
sensor_08	215213	15.20072127	2.037390474	0.02893518	15.18374	15.49479	15.69734	24.34896
sensor_09	215725	14.79920961	2.091963463	0	15.05353	15.08247	15.11863	25
sensor_10	220301	41.47033902	12.09351871	0	40.70526	44.29134	47.46376	76.10686
sensor_11	220301	41.91831906	13.05642524	0	38.85642	45.36314	49.65654	60
sensor_12	220301	29.13697472	10.11393506	0	28.68681	32.51583	34.93973	45
sensor_13	220301	7.078857518	6.90175467	0	1.538516	2.929809	12.85952	31.18755
sensor_14	220299	376.8600409	113.2063816	32.40955	418.10325	420.1062	420.9971	500
sensor_15	220289	416.4728921	126.0726417	0	459.4534	462.8561	464.3027	739.7415
sensor_16	220274	421.1275169	129.1561746	0	454.138825	462.02025	466.857075	599.999939
sensor_17	220274	2.303784946	0.765883056	0	2.447542	2.5337035	2.587682	4.87325
sensor_18	220304	590.8297752	199.34582	0	662.768975	665.6724	667.1467	878.9179
sensor_19	220304	360.8051652	101.9741177	0	398.0215	399.367	400.0884	448.9079
sensor_20	220304	796.2259422	226.6793166	95.52766	875.4644	879.6976	882.1299	1107.526
sensor_21	220279	459.7928149	154.5283375	0	478.9626	531.8559	534.25485	594.0611
sensor_22	220304	922.609264	291.83528	0	950.9224	981.925	1090.808	1227.564
sensor_23	220304	556.2353971	182.2979786	0	601.15105	625.8735	628.607725	1000
sensor_24	220284	649.1447987	220.865166	0	693.9578	740.2035	750.357125	839.575
sensor_25	220300	786.4117809	246.6636076	43.15479	790.489575	861.8696	919.104775	1214.42
sensor_26	220304	501.506589	169.8231727	0	448.29795	494.46845	536.27455	2000
sensor_27	220304	851.6903389	313.0740321	4.319347	782.682625	967.27985	1043.9765	1841.146
sensor_28	220248	576.1953046	225.7640911	0.6365742	518.947225	564.8725	744.021475	1466.281
sensor_29	220059	614.5964416	195.726872	0	627.7778	668.9814	697.2222	1600
sensor_30	220304	863.3231004	283.5447601	23.95833	839.0624	917.7083	981.2499	1800
sensor_31	220252	804.2839148	260.6023608	0.2407161	760.607475	878.85075	943.877625	1839.211
sensor_32	220304	486.4059795	150.7518358	6.460602	489.761075	512.27175	555.163225	1578.6
sensor_33	220304	234.9717765	88.37606459	54.88237	172.4863	226.35605	316.84495	425.5498
sensor_34	220304	427.1298166	141.772519	0	353.176625	473.34935	528.891025	694.479126
sensor_35	220304	593.0338759	289.3855113	2.26097	288.547575	709.66805	837.333025	984.0607

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

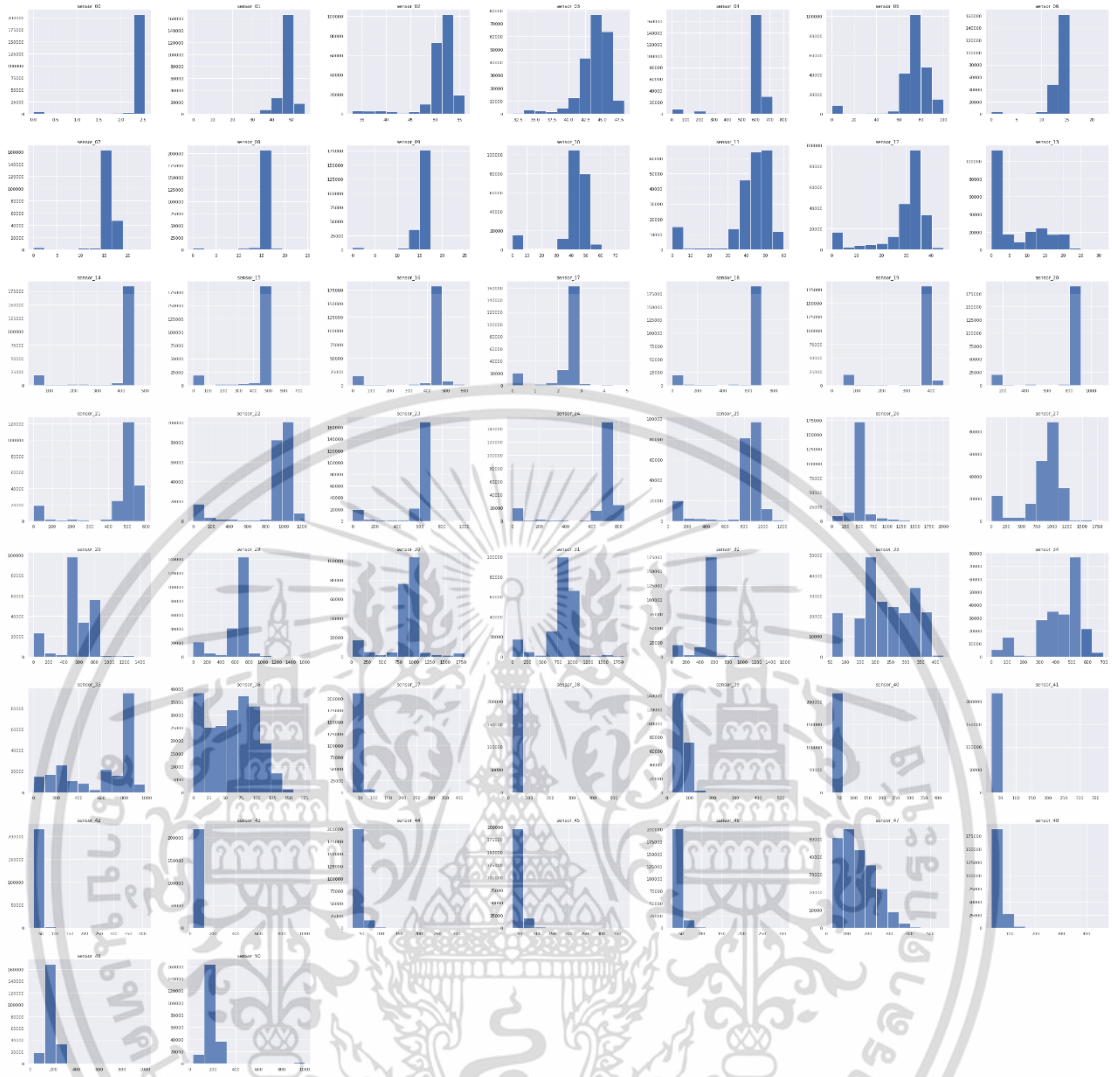
ตารางที่ 3.2(ต่อ) สถิติชุดข้อมูลทั้งหมด

Sensor	count	mean	std	min	25%	50%	75%	max
sensor_36	220304	60.78736021	37.60488301	0	28.79922	64.295485	90.8219275	174.9012
sensor_37	220293	49.65594644	10.54039745	24.47916603	45.57291	49.47916	53.64583	417.7083
sensor_38	220293	36.61044406	15.61372284	19.27083	32.55208	35.41666	39.0625	547.9166
sensor_39	220293	68.84452963	21.3711386	23.4375	57.8125	66.40625	77.86458	512.7604
sensor_40	220293	35.36512621	7.898665188	20.83333	32.55208	34.89583206	37.76041	420.3125
sensor_41	220293	35.45345524	10.25952072	22.13541603	32.8125	35.15625	36.97916412	374.2188
sensor_42	220293	43.87959103	11.04440379	24.47916603	39.58333	42.96875	46.61458	408.5937
sensor_43	220293	42.65689303	11.57634606	25.75231552	36.74768448	40.50926	45.13889	1000
sensor_44	220293	43.09498437	12.83751987	26.33101845	36.74768448	40.21991	44.84954	320.3125
sensor_45	220293	48.01858528	15.64128354	26.33101845	40.50925827	44.84954	51.21528	370.3704
sensor_46	220293	44.34090313	10.44243709	27.19907	39.0625	42.53472	46.58565	303.5301
sensor_47	220293	150.8890438	82.2449568	26.33101845	83.91203	138.0208	208.3333	561.632
sensor_48	220293	57.11996849	19.14359772	26.62037	47.74306	52.66204	60.76389	464.4097
sensor_49	143303	183.0492605	65.25865033	27.48842621	167.5347	193.8657	219.9074	1000
sensor_50	204937	202.6996668	109.5886072	27.77777863	179.1088	197.338	216.7245	1000

จากตารางที่ 3.2 จะแสดงข้อมูลสถิติเชิงบรรยาย รวมถึงสถิติที่สรุปแนวโน้มศูนย์กลางการกระจายตัว และรูปร่างของการแจกแจงของชุดข้อมูลทั้งหมด จากการสำรวจชุดข้อมูล (Explore Data) นำไปสู่การเตรียมชุดข้อมูล ในการทดลองมีขั้นตอนดังต่อไปนี้

1) ใส่ค่าที่ขาดหายไป (Imputing Missing Value) ด้วยค่ามัธยฐาน (Median) เนื่องจากประเภทของข้อมูล และการกระจายตัวของข้อมูลส่วนมากมีลักษณะเบ้ซ้าย หรือลักษณะเบ้ขวา อีกทั้งข้อมูลมีค่าผิดปกติ (Outlier) ดังรูปที่ 3.2 ดังนั้นหากใส่ค่าที่ขาดหายไปด้วยค่าเฉลี่ย (Mean) อาจทำให้ผลลัพธ์ที่ได้มีค่าสูงหรือต่ำกว่าข้อมูลส่วนใหญ่มาก สำหรับค่าฐานนิยม (Mode) เหมาะกับชุดข้อมูลประเภทหมวดหมู่ (Category)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 ฮิสโตแกรม (Histogram) ของเซนเซอร์ 51 เซนเซอร์

2) ปรับขนาดของข้อมูล (Feature Scaling) ด้วยการ Rescaling หรือ Min-Max Normalization คือ การปรับข้อมูลให้อยู่ระหว่าง 0 ถึง 1 เหมาะกับข้อมูลที่มีส่วนเบี่ยงเบนมาตรฐาน (Standard Deviation) น้อยมากๆ มีสูตรการคำนวณดังสมการที่ (3.1)

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{3.1}$$

โดยที่ x' คือ ค่าที่ปรับ (Rescaled Value) อยู่ระหว่าง 0 ถึง 1 และ x คือ ค่าดั้งเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) ตรวจสอบความสัมพันธ์เชิงเส้นระหว่างกันของตัวแปรอิสระ (Multicollinearity) โดยในการทดลองนี้จะพิจารณาสถิติสหสัมพันธ์ (Correlation statistics) ซึ่งค่าสัมประสิทธิ์ของความสัมพันธ์ระหว่างตัวแปรอิสระต้องไม่เกิน 0.8 หากเกินจะนำข้อมูลออกจากชุดข้อมูล

3.1.2 อัลกอริทึมที่ใช้ในการหาค่าความผิดปกติของเครื่องจักรในโรงงานอุตสาหกรรม

ในการค้นคว้าอิสระนี้ต้องการศึกษาความสัมพันธ์ระหว่างอุปกรณ์และเครื่องจักรโดยใช้การจำแนกแบบกลุ่ม (Classification) ซึ่งจะทำการสร้างโมเดลหรือตัวจำแนกข้อมูล (Classifier) เพื่อทำนายหมวดหมู่ของข้อมูล (Categories/Class) คือ สถานะปกติ หรือ ไม่ปกติ

3.1.3 วิธีการทำนายความผิดปกติของเครื่องจักรในโรงงานอุตสาหกรรม

การทำนายความผิดปกติของเครื่องจักร ใช้ 3 โมเดล ได้แก่ Neural Network, Support Vector Machine และ Decision Tree ในการวัดประสิทธิภาพของกลุ่มตัวอย่าง เพื่อเปรียบเทียบการทำงานของโมเดลที่มีผลต่อประสิทธิภาพในการทำงานของเครื่องจักร ทำให้ผลผลิตเพิ่มมากขึ้น

3.1.4 การวัดผลการทดลอง (Evaluation)

ในการวัดผลการทดลองการทำนายความผิดปกติของเครื่องจักร ในกรณีโมเดลมีลักษณะการจำแนกแบบกลุ่ม หรือตัวแปรตามเป็นลักษณะกลุ่ม สามารถวัดประสิทธิภาพของโมเดลจากค่า Accuracy, Precision, Recall , F1-score , ROC AUC score และ Confusion Matrix

3.2 การออกแบบการทดลอง

3.2.1 ขั้นตอนการเลือกชุดข้อมูล

ชุดข้อมูลสถานะเครื่องจักรจำนวน 220,320 ข้อมูล ซึ่งเป็นข้อมูลจำนวนมาก พบว่ามีข้อจำกัดด้านทรัพยากรคอมพิวเตอร์ ซึ่งความเร็ว และข้อมูลจำเพาะของเครื่องคอมพิวเตอร์ที่ใช้ในการทดลอง คือ อุปกรณ์ DESKTOP-VDLN527 หน่วยประมวลผล Intel(R) Core(TM) i5-1035G7 CPU @ 1.20GHz 1.50 GHz, RAM ที่ติดตั้ง 8.00 GB ระบบปฏิบัติการ 64 บิต ส่งผลให้แบ่งชุดข้อมูลออกเป็น 2 ช่วง คือ ชุดข้อมูลช่วงที่ 1 (วันที่ 1 เมษายน – 30 มิถุนายน พ.ศ. 2561) และ ชุดข้อมูลช่วงที่ 2 (วันที่ 1 กรกฎาคม – 30 สิงหาคม พ.ศ. 2561) ดังตารางที่ 3.3 และ 3.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.3 สถิติชุดข้อมูลช่วงที่ 1 (วันที่ 1 เมษายน – 30 มิถุนายน พ.ศ. 2561)

Sensor	count	mean	std	min	25%	50%	75%	max
0	131040	0.926665643	0.160864349	0	0.957545578	0.963334871	0.971053928	1
1	131040	0.7607645	0.078576542	0	0.731770873	0.770833277	0.80468738	1
2	131040	0.75237099	0.164789961	0	0.712694567	0.786191281	0.846324902	1
4	131040	0.73531911	0.176022322	0	0.780482228	0.788612413	0.795000632	1
5	131040	0.741600175	0.191277585	0	0.726735947	0.774499129	0.820259159	1
6	131040	0.58344318	0.096136755	0	0.594664987	0.60312308	0.611580722	1
8	131040	0.643064019	0.086667448	0	0.648228577	0.654443793	0.667495402	1
9	131040	0.656895389	0.096962294	0	0.66645047	0.676508978	0.678130898	1
12	131040	0.616469848	0.223170981	0	0.601555444	0.695598889	0.748082167	1
13	131040	0.112203887	0.145056303	0	0.039182639	0.062075764	0.104802918	1
14	131040	0.676444855	0.296905384	0	0.798956491	0.827880146	0.830684309	1
26	131040	0.232755661	0.085166054	0	0.217592075	0.2295669	0.255231663	1
28	131040	0.409139317	0.196558536	0	0.349258536	0.493488266	0.534843283	1
29	131040	0.346469339	0.143995004	0	0.3778935	0.3984375	0.418113375	1
33	131040	0.388494698	0.232702448	0	0.286975187	0.329599852	0.612328766	1
35	131040	0.47500244	0.315029654	0	0.200524403	0.338361399	0.842906251	1
36	131040	0.475898331	0.155773167	0	0.38344138	0.482638312	0.579727155	1
37	131040	0.065784913	0.029128277	0	0.053595673	0.064450474	0.077340579	1
38	131040	0.034121183	0.033445995	0	0.025628379	0.032035483	0.037949727	1
39	131040	0.082823952	0.041774271	0	0.064997331	0.078849236	0.09536494	1
40	131040	0.077617526	0.04413902	0	0.060446801	0.074901451	0.090670177	1
43	131040	0.07000091	0.036380342	0	0.049107141	0.065476188	0.084821458	1
44	131040	0.078890402	0.052605386	0	0.053731298	0.068656676	0.089552215	1
45	131040	0.060960681	0.045503092	0	0.041379318	0.05418722	0.068965518	1
46	131040	0.08998751	0.048227902	0	0.065068484	0.083904094	0.104452021	1
47	131040	0.181508065	0.120209987	0	0.092570057	0.153471303	0.251522571	1
48	131040	0.086189176	0.045618804	0	0.064642509	0.078354558	0.096963761	1
49	131040	0.351046497	0.125395509	0	0.312539724	0.365372311	0.421387675	1
50	131040	0.169076906	0.078811163	0	0.155177555	0.17039695	0.185616242	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.4 สถิติชุดข้อมูลช่วงที่ 2 (วันที่ 1 กรกฎาคม – 30 สิงหาคม 2561)

Sensor	count	mean	std	min	25%	50%	75%	max
0	89280	0.945952575	0.154483143	0	0.966615093	0.975155262	0.988353992	1
1	89280	0.833650413	0.071722336	0	0.800305954	0.848508005	0.876817125	1
2	89280	0.772896947	0.18853803	0	0.769980459	0.820662609	0.861598347	1
5	89280	0.722761353	0.141155384	0	0.67510261	0.722223617	0.780657387	1
7	89280	0.686948824	0.083281665	0	0.684775838	0.695825705	0.705340844	1
13	89280	0.457476392	0.237801034	0	0.388847384	0.478390103	0.66155138	1
14	89280	0.821907141	0.053053951	0	0.825934096	0.827527584	0.829185359	1
16	89280	0.718773942	0.046163142	0	0.713594276	0.723892488	0.733525998	1
25	89280	0.800906862	0.096827795	0	0.703112098	0.849950601	0.859149559	1
26	89280	0.277167101	0.077279868	0	0.248644625	0.26033315	0.282850375	1
27	89280	0.653022987	0.164673617	0	0.579046091	0.691063614	0.779443946	1
28	89280	0.564166699	0.052897419	0	0.541913523	0.574542655	0.591943894	1
29	89280	0.649228542	0.059563306	0	0.625594964	0.642721552	0.671265729	1
30	89280	0.578206303	0.068547943	0	0.551180856	0.585073348	0.617938794	1
31	89280	0.67657066	0.104919778	0	0.673572422	0.708056147	0.737885099	1
32	89280	0.510218806	0.040124591	0	0.491472842	0.508263456	0.534165583	1
33	89280	0.604787919	0.173187214	0	0.445635054	0.574555096	0.780764068	1
34	89280	0.703860898	0.064573971	0	0.663811934	0.697302322	0.736749155	1
35	89280	0.796215108	0.120971209	0	0.702119153	0.84766317	0.874791034	1
36	89280	0.205706311	0.177177606	0	0.048859989	0.178581736	0.326819564	1
37	89280	0.062773535	0.024157114	0	0.053642373	0.063576149	0.073509925	1
38	89280	0.037221252	0.027607748	0	0.029179334	0.03465046	0.042553204	1
39	89280	0.111143466	0.045194247	0	0.086495522	0.108816957	0.13616069	1
41	89280	0.040204875	0.019889009	0	0.034364254	0.040378011	0.046391735	1
43	89280	0.021004535	0.015765792	0	0.013365015	0.017820021	0.024651025	1
44	89280	0.064359131	0.053369727	0	0.038385824	0.050196841	0.073818893	1
45	89280	0.075475187	0.051238499	0	0.048780496	0.065601332	0.089150554	1
46	89280	0.069238877	0.046472917	0	0.045026195	0.058638764	0.078534032	1
47	89280	0.335373432	0.154311466	0	0.24054053	0.342162051	0.438378388	1
48	89280	0.082670982	0.055161749	0	0.054196958	0.069398562	0.098479856	1
49	89280	0.093308799	0.040650362	0	0.092234454	0.092234454	0.092234454	1
50	89280	0.191309931	0.140954613	0	0.156547668	0.177380971	0.199404742	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตารางที่ 3.3 และตารางที่ 3.4 เป็นข้อมูลหลังจากการเตรียมข้อมูล และทำความสะอาดข้อมูล (Cleansing Data) เรียบร้อยแล้ว พบว่าข้อมูลช่วงที่ 1 (วันที่ 1 เมษายน – 30 มิถุนายน พ.ศ. 2561) มีข้อมูลจำนวนทั้งหมด 131,040 ข้อมูล จำนวนเซนเซอร์ 29 เซนเซอร์ โดยที่สถานะเครื่องจักรปกติ จำนวน 122,065 ข้อมูล คิดเป็น 93.15% ของข้อมูลทั้งหมด และสถานะเครื่องจักรผิดปกติจำนวน 8,975 ข้อมูล คิดเป็น 6.85% ของข้อมูลทั้งหมด ชุดข้อมูลช่วงที่ 2 (วันที่ 1 กรกฎาคม – 30 สิงหาคม พ.ศ. 2561) มีข้อมูลจำนวนทั้งหมด 89,280 ข้อมูล จำนวนเซนเซอร์ 32 เซนเซอร์ โดยที่สถานะเครื่องจักรปกติจำนวน 83,771 ข้อมูล คิดเป็น 93.83% ของข้อมูลทั้งหมด และสถานะเครื่องจักรผิดปกติจำนวน 5,509 ข้อมูล คิดเป็น 6.17% ของข้อมูลทั้งหมด

3.2.2 วิธีการทดลอง

เลือกชุดข้อมูลเซนเซอร์แต่ละช่วง มาแบ่งข้อมูลออกเป็น 2 ส่วน คือ ชุดข้อมูลสำหรับการสอน (Training Dataset) เพื่อสอนให้เครื่องเรียนรู้ หรือรู้จักข้อมูล และชุดข้อมูลสำหรับการทดสอบ (Test Dataset) เพื่อนำมาทดสอบความถูกต้องของโมเดลหลังจากที่เรียนรู้ข้อมูลเรียบร้อยแล้ว โดยแบ่งชุดข้อมูลสำหรับการสอนออกเป็น 70 เปอร์เซ็นต์ และแบ่งชุดข้อมูลสำหรับการทดสอบความถูกต้องของโมเดลออกเป็น 30 เปอร์เซ็นต์ ของข้อมูลทั้งหมด การทดลองชุดที่ 1 มีชุดข้อมูลสำหรับการทดสอบจำนวน 91,728 ข้อมูล ชุดข้อมูลสำหรับการทดสอบจำนวน 39,312 ข้อมูล และการทดลองชุดที่ 2 มีชุดข้อมูลสำหรับการทดสอบ จำนวน 62,496 ข้อมูล ชุดข้อมูลสำหรับการทดสอบจำนวน 26,784 ข้อมูล ดังตารางที่ 3.5

ตารางที่ 3.5 ชุดข้อมูลสำหรับการสอน และข้อมูลสำหรับการทดสอบ ของการทดลองชุดที่ 1 และชุดที่ 2

	ข้อมูลการทดลองชุดที่ 1	**ข้อมูลการทดลองชุดที่ 2**
ชุดข้อมูลสำหรับการสอน (Training Dataset)	91,728	62,496
ชุดข้อมูลสำหรับการทดสอบ (Test Dataset)	39,312	26,784

ในชุดข้อมูลสำหรับการสอน และข้อมูลสำหรับการทดสอบ พบว่าข้อมูลเกิดความไม่สมดุลกันในชุดข้อมูลเป้าหมาย (Target Dataset) ดังรูปที่ 3.2 ดังนั้นจึงทำการสุ่มเพิ่มตัวอย่างกลุ่มน้อย (Synthetic Minority Oversampling Technique : SMOTE) ซึ่งเป็นวิธีที่นิยมใช้มากในงานวิจัย [13] เนื่องจากเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเพิ่มจำนวนข้อมูลกลุ่มน้อยให้มีจำนวนเพิ่มขึ้น เพื่อให้การกระจายของกลุ่มข้อมูลมีความสมดุลมากขึ้น โดยทำการสุ่มค่าข้อมูลที่อยู่ในกลุ่มข้อมูลน้อยขึ้นมา 1 ค่า หลังจากนั้นพิจารณาค่าข้อมูลใกล้เคียงอีกจำนวน K ค่า (K-Nearest Neighbor) แล้วคำนวณค่าระยะทาง (Euclidean Distance) ระหว่างค่าที่สุ่มกับค่าข้อมูลใกล้เคียงแต่ละค่า เพื่อหาค่าระยะทางที่น้อยที่สุดระหว่างค่าที่สุ่มกับค่าข้อมูลใกล้เคียง จากนั้นจึงสร้างข้อมูลเทียมระหว่างค่าข้อมูลที่สุ่มกับค่าข้อมูลใกล้เคียงตัวที่ให้ค่าระยะทางที่น้อยที่สุด ดังนั้นข้อมูลชุดที่ 1 มีชุดข้อมูลสำหรับการทดสอบจำนวน 91,728 ข้อมูล ประกอบด้วยข้อมูลเป้าหมายประเภท ปกติ จำนวน 85,425 ข้อมูล และผิดปกติ จำนวน 6,303 ข้อมูล หลังจากการสุ่มเพิ่มตัวอย่างกลุ่มน้อย พบว่าชุดข้อมูลสำหรับการทดสอบเพิ่มจำนวนเป็น 170,850 ข้อมูล ประกอบด้วยข้อมูลเป้าหมายประเภท ปกติ และผิดปกติ จำนวน 85,425 ข้อมูล ชุดที่ 2 มีชุดข้อมูลสำหรับการทดสอบจำนวน 91,728 ข้อมูล ประกอบด้วยข้อมูลเป้าหมายประเภท ปกติ จำนวน 58,669 ข้อมูล และ ผิดปกติ จำนวน 3,827 ข้อมูล หลังจากการสุ่มเพิ่มตัวอย่างกลุ่มน้อย พบว่าชุดข้อมูลสำหรับการทดสอบเพิ่มจำนวนเป็น 117,338 ข้อมูล ประกอบด้วยข้อมูลเป้าหมายประเภท ปกติ และผิดปกติ จำนวน 58,669 ข้อมูล ดังตารางที่ 3.6

ตารางที่ 3.6 ชุดข้อมูลสำหรับการสอน และข้อมูลสำหรับการทดสอบ หลังจากทำการสุ่มเพิ่มตัวอย่างกลุ่มน้อย ของการทดลองชุดที่ 1 และ 2

SMOTE Technique	**ข้อมูลการทดลองชุดที่ 1**	**ข้อมูลการทดลองชุดที่ 2**
ชุดข้อมูลสำหรับการสอน (Training Dataset)	170,850 (Normal= 85,425, Broken= 85,425)	117,338 (Normal= 58,669, Broken= 58,669)
ชุดข้อมูลสำหรับการทดสอบ (Test Dataset)	39,312	26,784

หลังจากทำการสุ่มเพิ่มตัวอย่างกลุ่มน้อยเรียบร้อยแล้ว นำข้อมูลชุดการทดลองไปทำการสอนและทดสอบโมเดล พร้อมทั้งปรับแต่งค่าพารามิเตอร์ (Turning Parameter) ให้มีประสิทธิภาพในการทำนายที่ดี และลดค่าความผิดพลาดในการทำนาย ได้แก่ การวิเคราะห์ความแม่นตรงของตัวแบบ (K-Fold Validation) , การสุ่มถอดบางโหนดในระหว่างการสอน (Dropout Regularization) เพื่อช่วยลดการเกิดปัญหาโมเดลชำนาญเกินไปในการเรียนรู้ชุดข้อมูลสำหรับการสอน (Overfitting) ซึ่งทำการทดลองทั้ง 2 ชุดการทดลอง คือ ทดลองข้อมูลช่วงที่ 1 (วันที่ 1 เมษายน – 30 มิถุนายน พ.ศ. 2561) และข้อมูลช่วงที่ 2 (วันที่ 1 กรกฎาคม – 30 สิงหาคม พ.ศ. 2561) ด้วยเงื่อนไขเดียวกัน เพื่อเปรียบเทียบประสิทธิภาพของโมเดล 3 โมเดลกับชุดข้อมูล ได้แก่ Neural network, Support vector machine และ Decision tree

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ด้วยค่า Accuracy, Precision, Recall, F1-score, ROC AUC score และ Confusion matrix พร้อมทั้งดูค่าความผิดพลาดในการทำนายร่วมด้วย

3.3 เครื่องมือที่ใช้ในการทดลอง

การทดลองแบ่งออกเป็น 3 ส่วน คือ ส่วนที่นำชุดข้อมูลอุปกรณ์ของเครื่องจักรในโรงงานมาทำความสะอาด (Data Cleaning) และปรับให้อยู่ในรูปแบบที่ต้องการนำไปใช้ ส่วนแบบจำลองอัลกอริทึมเพื่อการทำนาย และส่วนที่ใช้ในการคำนวณความถูกต้องและความแม่นยำของอัลกอริทึมที่ใช้ในแบบจำลอง

ชุดข้อมูลที่ผ่านการทำความสะอาดแล้วจะถูกเก็บในรูปแบบไฟล์ .csv ในส่วนการทำงานของอัลกอริทึม และการสร้างโมเดลทั้งหมดจะถูกคำนวณผ่าน Google Colab ที่เขียนโดยภาษา Python

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ผลการทดลองและการอภิปรายผล

บทนี้นำเสนอผลการทดลองการทำนายความผิดปกติของเครื่องจักร และเปรียบเทียบประสิทธิภาพของโมเดล แบ่งออกเป็น 2 ชุดการทดลอง การทดลองชุดที่ 1 เป็นการทดลองช่วงวันที่ 1 เมษายน – 30 มิถุนายน พ.ศ. 2561 และการทดลองชุดที่ 2 เป็นการทดลองช่วง 1 กรกฎาคม – 30 สิงหาคม พ.ศ. 2561 โดยทั้งสองชุดการทดลองใช้วิธีการวัดประสิทธิภาพของโมเดล การวิเคราะห์ผลการทดลอง และการวัดความถูกต้องแบบเดียวกัน

4.1 การทดลองชุดที่ 1 และชุดที่ 2

การทดลองชุดที่ 1 และชุดที่ 2 ทำเปรียบเทียบประสิทธิภาพโมเดล Neural network, Support Vector Machine และ Decision Tree ในทำนายความผิดปกติของเครื่องจักร ในกรณีโมเดลมีลักษณะการจำแนกแบบกลุ่ม (ปกติ/ผิดปกติ) ด้วยค่า Accuracy, Precision, Recall, F1-score, ROC AUC score และ Confusion Matrix ดังตารางที่ 4.1 และตารางที่ 4.2

ตารางที่ 4.1 ตารางแสดงค่าประสิทธิภาพการทำนายโมเดล Support Vector Machine, Decision Tree และ Neural Network ของการทดลองชุดที่ 1

Algorithm Model	SVM	Decision Tree	Neural Network
Accuracy (%)	99.93	99.95	99.69
Precision	0.99	1	0.98
Recall	1	1	1
F1-Score	1	1	0.99
ROC AUC Score	0.99	0.99	0.99

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 ตารางแสดงค่าประสิทธิภาพการทำนายโมเดล Support Vector Machine, Decision Tree และ Neural Network ของการทดลองชุดที่ 2

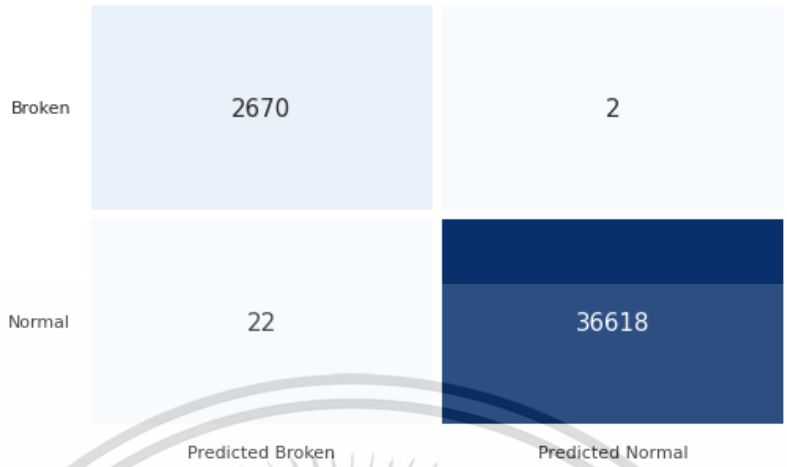
Algorithm Model	SVM	Decision Tree	Neural Network
Accuracy (%)	99.93	99.97	99.91
Precision	0.99	1	1
Recall	1	1	0.99
F1-Score	0.99	1	0.99
ROC AUC Score	0.99	0.99	0.99

จากตารางที่ 4.1 และตารางที่ 4.2 แสดงประสิทธิภาพการทำนายโมเดล Support Vector Machine, Decision Tree และ Neural Network หลังจากการทำความสะอาดข้อมูล พร้อมทั้งปรับค่าพารามิเตอร์ (Turning parameter) ด้วยการค้นหาแบบกริด (Grid Search) และทำการวิเคราะห์ความแม่นตรงของตัวแบบ (K-Fold Validation) โดยการแบ่งชุดข้อมูลสำหรับการสอนออกเป็นจำนวน K ส่วน แบบสุ่มจำนวนเท่าๆ กัน เพื่อสร้างและสอนโมเดลก่อนนำไปทำนายประสิทธิภาพด้วยชุดข้อมูลสำหรับการทดสอบ (Test Dataset) ในการทดลองนี้ทำการแบ่งชุดข้อมูลออกเป็นจำนวน 10 ส่วน พร้อมทั้งสุ่มถอดบางโหนดในระหว่างการสอน (Dropout Regularization) สำหรับโมเดล Neural Network เพื่อลดการเกิดปัญหาโมเดลชำนาญเกินไปในการเรียนรู้ชุดข้อมูลสำหรับการสอน (Overfitting)

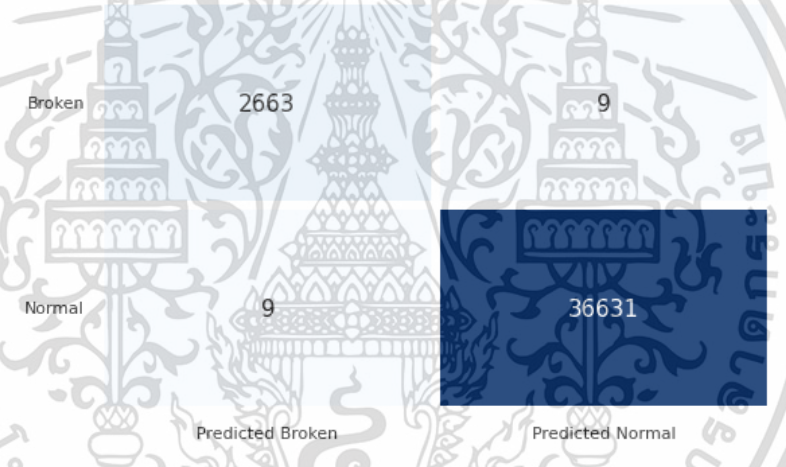
ประสิทธิภาพการทำนายความผิดพลาดของเครื่องจักรในการทดลองชุดที่ 1 พบว่าโมเดล Decision Tree ให้ผลลัพธ์ดีที่สุดด้วยค่าความถูกต้อง 99.95% พร้อมด้วยค่า Confusion Matrix ที่ให้ผลลัพธ์ดีที่สุด ดังรูปที่ 4.1 ถึงรูปที่ 4.3 โมเดล Support Vector Machine ทำนายค่าความถูกต้อง 99.93% และโมเดล Neural Network ทำนายค่าความถูกต้อง 99.69%

ประสิทธิภาพการทำนายความผิดพลาดของเครื่องจักรในการทดลองชุดที่ 2 พบว่าโมเดล Decision Tree ให้ผลลัพธ์ดีที่สุดด้วยค่าความถูกต้อง 99.97% พร้อมด้วยค่า Confusion matrix ที่ให้ผลลัพธ์ดีที่สุด ดังรูปที่ 4.4 ถึงรูปที่ 4.6 โมเดล Support vector machine ทำนายค่าความถูกต้อง 99.93% และโมเดล Neural Network ทำนายค่าความถูกต้อง 99.91%

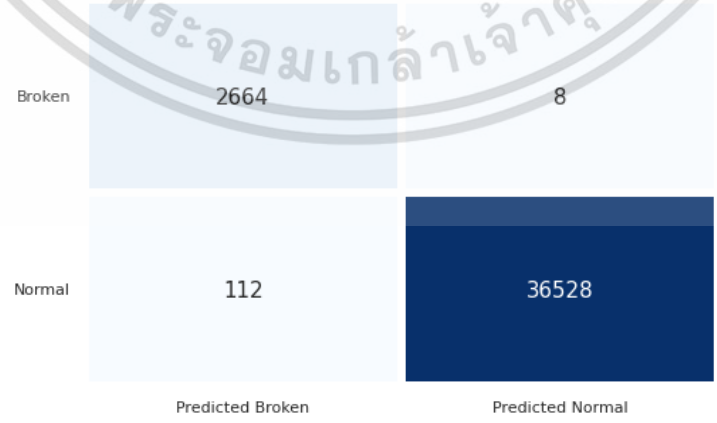
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.1 การทำนายค่า Confusion Matrix โมเดล Support Vector Machine ของการทดลองชุดที่ 1



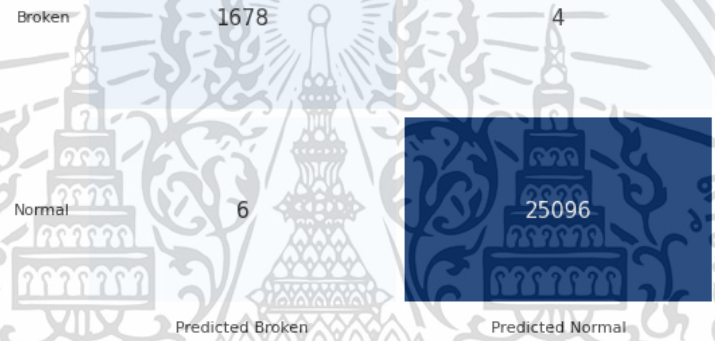
รูปที่ 4.2 การทำนายค่า Confusion Matrix โมเดล Decision Tree ของการทดลองชุดที่ 1



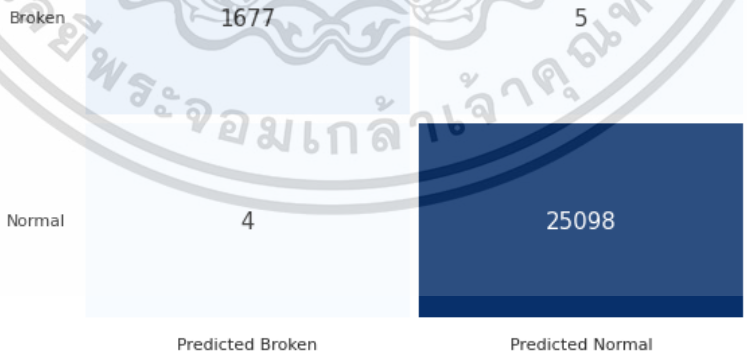
รูปที่ 4.3 การทำนายค่า Confusion Matrix โมเดล Neural Network ของการทดลองชุดที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.1 ถึงรูปที่ 4.3 แสดงการทำนาย Confusion Matrix ของการทดลองชุดที่ 1 พบว่าโมเดล Decision Tree ให้ผลลัพธ์ดีที่สุด ด้วยการทำนาย ‘ผิดปกติ (Broken)’ ถูกต้อง จำนวน 2,663 ครั้ง ทำนาย ‘ปกติ (Normal)’ ถูกต้อง จำนวน 36,631 ครั้ง ทำนาย ‘ผิดปกติ’ ผิดพลาด จำนวน 9 ครั้ง และทำนาย ‘ปกติ’ ผิดพลาด จำนวน 9 ครั้ง โมเดล Support Vector Machine ทำนาย ‘ผิดปกติ’ ถูกต้อง จำนวน 2,670 ครั้ง ทำนาย ‘ปกติ’ ถูกต้อง จำนวน 36,618 ครั้ง ทำนาย ‘ผิดปกติ’ ผิดพลาด จำนวน 22 ครั้ง และทำนาย ‘ปกติ’ ผิดพลาด จำนวน 2 ครั้ง โมเดล Neural Network ทำนาย ‘ผิดปกติ’ ถูกต้อง จำนวน 2,664 ครั้ง ทำนาย ‘ปกติ’ ถูกต้อง จำนวน 36,528 ครั้ง ทำนาย ‘ผิดปกติ’ ผิดพลาด จำนวน 112 ครั้ง และทำนาย ‘ปกติ’ ผิดพลาด จำนวน 8 ครั้ง ตามลำดับ

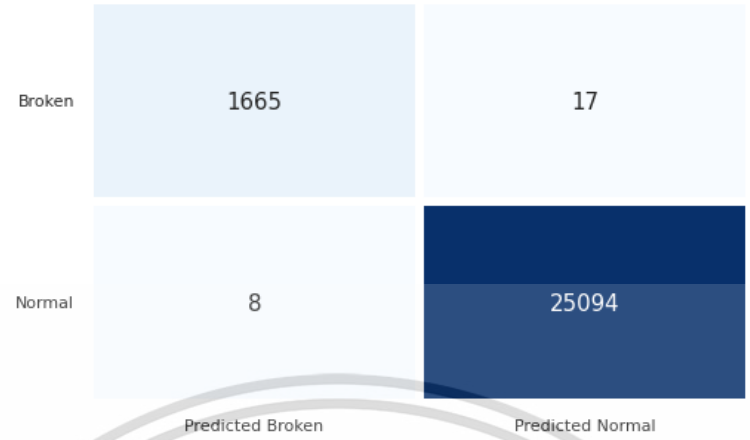


รูปที่ 4.4 การทำนายค่า Confusion Matrix โมเดล Support vector machine ของการทดลองชุดที่ 2



รูปที่ 4.5 การทำนายค่า Confusion Matrix โมเดล Decision Tree ของการทดลองชุดที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 การทำนายค่า Confusion Matrix โมเดล Neural Network ของการทดลองชุดที่ 2

จากรูปที่ 4.4 ถึงรูปที่ 4.6 แสดงการทำนาย Confusion Matrix ของการทดลองชุดที่ 2 พบว่า โมเดล Decision Tree ให้ผลลัพธ์ดีที่สุด ด้วยการทำนาย ‘ผิดปกติ (Broken)’ ถูกต้อง จำนวน 1,677 ครั้ง ทำนาย ‘ปกติ (Normal)’ ถูกต้อง จำนวน 25,098 ครั้ง ทำนาย ‘ผิดปกติ’ ผิดพลาด จำนวน 4 ครั้ง และทำนาย ‘ปกติ’ ผิดพลาด จำนวน 5 ครั้ง โมเดล Support Vector Machine ทำนาย ‘ผิดปกติ’ ถูกต้อง จำนวน 1,678 ครั้ง ทำนาย ‘ปกติ’ ถูกต้อง จำนวน 25,096 ครั้ง ทำนาย ‘ผิดปกติ’ ผิดพลาด จำนวน 6 ครั้ง และทำนาย ‘ปกติ’ ผิดพลาด จำนวน 4 ครั้ง โมเดล Neural Network ทำนาย ‘ผิดปกติ’ ถูกต้อง จำนวน 1,665 ครั้ง ทำนาย ‘ปกติ’ ถูกต้อง จำนวน 25,094 ครั้ง ทำนาย ‘ผิดปกติ’ ผิดพลาด จำนวน 8 ครั้ง และทำนาย ‘ปกติ’ ผิดพลาด จำนวน 17 ครั้ง ตามลำดับ

4.2 เปรียบเทียบผลการทดลองชุดที่ 1 และชุดที่ 2

การทดลองชุดที่ 1 และชุดที่ 2 เป็นการทำนายความผิดปกติของเครื่องจักร และเปรียบเทียบประสิทธิภาพของโมเดล โดยการใช้วิธีการทำความสะอาดข้อมูล และปรับค่าพารามิเตอร์เหมือนกัน จากการทดลองได้ผลดังตารางที่ 4.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.3 การเปรียบเทียบประสิทธิภาพของโมเดลต่างๆ ในการทำนายความผิดปกติของเครื่องจักร

Algorithm Model	SVM	Decision Tree	Neural Network
<u>การทดลองชุดที่ 1</u>			
Accuracy (%)	99.93	**99.95**	99.69
Precision	0.99	**1**	0.98
Recall	1	**1**	1
F1-Score	1	**1**	0.99
ROC AUC Score	0.99	**0.99**	0.99
<u>การทดลองชุดที่ 2</u>			
Accuracy (%)	99.93	**99.97**	99.91
Precision	0.99	**1**	1
Recall	1	**1**	0.99
F1-Score	0.99	**1**	0.99
ROC AUC Score	0.99	**0.99**	0.99

จากตารางที่ 4.3 พบว่าโมเดล Decision Tree จะให้ผลความถูกต้องของการทำนายมากที่สุดทั้งการทดลองชุดที่ 1 และชุดที่ 2 ด้วยค่าความถูกต้องเฉลี่ย 99.96% และค่า Confusion Matrix ให้ผลลัพธ์ดีที่สุด เมื่อเทียบกับ Support vector machine ที่มีค่าความถูกต้องเฉลี่ย 99.93 % ตามด้วย Neural Network ค่าความถูกต้องเฉลี่ย 99.80% ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทสรุปและข้อเสนอแนะ

5.1 สรุป

การจำแนกประเภทข้อมูลเป็นกระบวนการสร้างโมเดลจัดการข้อมูลให้อยู่ในกลุ่มที่กำหนดมาให้ โดยการสร้างกฎเพื่อช่วยในการตัดสินใจจากข้อมูลที่มีอยู่ เพื่อใช้ทำนายแนวโน้มของข้อมูลที่ยังไม่เกิดขึ้น ในการทำนายหมวดหมู่ของข้อมูล การศึกษาค้นคว้าอิสระครั้งนี้มีวัตถุประสงค์เพื่อศึกษาสถานะของเครื่องจักร ปกติ หรือ ผิดปกติ จากอุปกรณ์ภายในเครื่องจักรโดยใช้การจำแนกแบบกลุ่ม (Classification) ด้วยวิธี Support Vector Machine, Decision Tree และ Neural Network เนื่องจากเป็นอัลกอริทึมที่งานวิจัย [7],[10] ประเภทอุตสาหกรรมส่วนมากนิยมนำมาใช้ในการทดสอบ เพราะลักษณะข้อมูลประเภทอุปกรณ์เครื่องจักรไม่ซับซ้อนมากนัก โดยใช้หลักการเหมืองข้อมูลมาใช้ในการจำแนกประเภทข้อมูล จากการทดลองแบ่งข้อมูลออกเป็น 2 ชุดการทดลอง โดยใช้เงื่อนไขการทดลองที่เหมือนกัน

จากการทดลอง พบว่าโมเดล Decision Tree เป็นวิธีที่มีประสิทธิภาพสูงสุดทั้งการทดลองชุดที่ 1 และชุดที่ 2 ด้วยค่าความถูกต้องเฉลี่ย 99.96% และค่า Confusion Matrix ให้ผลลัพธ์ดีที่สุด เมื่อเทียบกับ Support vector machine ที่มีค่าความถูกต้องเฉลี่ย 99.93% และตามด้วย Neural Network ค่า ค่าความถูกต้องเฉลี่ยที่ 99.80%

การสรุปผลการศึกษาค้นคว้าอิสระครั้งนี้ วิธี Decision Tree มีประสิทธิภาพในการจำแนกความผิดปกติของเครื่องจักรมากที่สุดทั้งสองชุดการทดลอง เพราะว่ามีค่าความถูกต้องอยู่ในระดับสูงที่สุด และค่า Confusion Matrix ให้ผลลัพธ์ดีที่สุด ดังนั้นสามารถนำผลลัพธ์ที่ได้จากแบบจำลองของ Decision Tree ไปใช้ในการทำนายสถานะการทำงานของเครื่องจักรที่ลักษณะข้อมูลประเภทเดียวกับกลุ่มตัวอย่าง เพื่อเป็นแนวทางในการสนับสนุนการตัดสินใจเกี่ยวกับกระบวนการผลิต และแก้ไขส่วนที่มีความผิดปกติได้อย่างทันท่วงที อีกทั้งยังสามารถบอกแนวโน้มความเป็นไปได้ในกระบวนการผลิต ส่งผลให้แผนการทำงานสามารถดำเนินการไปได้อย่างราบรื่น

5.2 วิเคราะห์ผลการทดลอง

โดยจากผลการทดลองทั้ง 2 ชุดข้อมูล พบว่า โมเดล Decision Tree ทำนายประสิทธิภาพได้แม่นยำที่สุด ความถูกต้องเฉลี่ย 99.96% เนื่องจากเป็นอัลกอริทึมที่ถูกออกแบบมาเพื่อให้รองรับข้อมูลที่ไม่เป็นเส้นตรง (Non-Linear) สามารถตีความ และอธิบายที่มาของผลการทำนายได้ง่าย รวดเร็ว ส่วนมากจะใช้กับลักษณะข้อมูลที่ไม่ซับซ้อนมาก สำหรับโมเดล Support Vector Machine ความถูกต้องเฉลี่ย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

99.93 เปอเซ็นต์ เป็นอัลกอริทึมที่รองรับข้อมูลที่เป็นเส้นตรง และไม่เป็นเส้นตรงเช่นกัน ส่วนมากนิยมใช้กับข้อมูลที่มีจำนวนไม่มาก ซึ่งยากต่อการเข้าใจผลการทำนาย ตีความ และใช้งาน เนื่องจากชุดข้อมูลที่นำมาทดลองมีลักษณะไม่เป็นเส้นตรง ดังนั้นจึงจำเป็นต้องกำหนดค่าเคอร์เนลฟังก์ชัน (Kernel Function) ให้เหมาะสมในแต่ละชุดข้อมูล และโมเดล Neural Network ค่าเฉลี่ยความถูกต้องที่ 99.80% เป็นอัลกอริทึมที่รองรับข้อมูลที่ซับซ้อนมาก แต่ข้อมูลที่นำมาทดลองไม่ซับซ้อนเท่าที่ควร และข้อมูลมีจำนวนมาก ส่งผลให้โมเดล Neural Network อาจใช้เวลาในการทดสอบนานกว่าตัวแบบอื่น

โมเดลที่นำมาเปรียบเทียบประสิทธิภาพในข้อมูล 2 ชุดการทดลองนี้ พบว่าค่าความถูกต้อง และค่าความคลาดเคลื่อนของข้อมูลงตัวแบบทั้ง 3 โมเดลอยู่ในระดับที่ดีมาก ดังนั้นการนำไปประยุกต์ใช้ขึ้นอยู่กับความเหมาะสมของลักษณะงาน ทั้งในด้านการบริหารเวลาและต้นทุนในการผลิต เป็นต้น

5.3 ปัญหาและข้อเสนอแนะ

ปัญหา และข้อจำกัดที่พบในการศึกษาค้นคว้าอิสระ มีดังต่อไปนี้

1) ข้อมูลที่ใช้เป็นข้อมูลจริง และมีจำนวนมาก แต่ลักษณะของข้อมูลไม่ซับซ้อนเท่าที่ควร เนื่องจากเซนเซอร์ภายในเครื่องจักรไม่เกิดความเสียหายบ่อยนัก

2) จากข้อมูลชุดการทดลองพบว่า ผลจากเซนเซอร์ภายในเครื่องจักรไม่เกิดความเสียหายบ่อยนัก ดังนั้นชุดข้อมูลเป้าหมายมีข้อมูลลักษณะไม่สมดุล คือ สถานะเครื่องจักร ปกติ จำนวน 205,836 ข้อมูล คิดเป็น 93.43% และผิดปกติ จำนวน 14,484 ข้อมูล คิดเป็น 6.57% ซึ่งอาจมีผลต่อการเปรียบเทียบประสิทธิภาพของโมเดล

3) ทรัพยากรคอมพิวเตอร์ ความเร็ว และข้อมูลจำเพาะของเครื่องคอมพิวเตอร์ที่ใช้ในการทดลอง ส่งผลให้ต้องแบ่งการทดลองออกเป็น 2 ชุดการทดลอง

อย่างไรก็ตามตัวแบบในการทำนายที่ได้นำเสนอมานั้น เป็นเพียงการใช้เทคนิคหนึ่งของการทำเหมืองข้อมูลจากหลากหลายเทคนิคที่มีอยู่ ซึ่งในอนาคตหากมีแนวทางที่จะพัฒนาต่อไป หรือมีการพัฒนาการค้นคว้าอิสระให้มีประสิทธิภาพมากขึ้น มีข้อเสนอแนะดังต่อไปนี้

1) ควรทดสอบด้วยชุดข้อมูลที่เหมาะสม เช่น เพิ่มปริมาณข้อมูลที่นำมาใช้ในการสร้างตัวแบบ เพิ่มประเภท และลักษณะของข้อมูลให้มีความหลากหลายมากยิ่งขึ้น เช่น Time series เป็นต้น

2) อาจมีวิธีการจำแนกกลุ่มอื่นๆ เพื่อประสิทธิภาพในการวิเคราะห์ข้อมูลเครื่องจักรมากกว่าวิธีดังกล่าว เช่น K-Nearest Neighbors (K-NN), Naive Bayes, Random Forest เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปัญหา ข้อจำกัด และข้อเสนอแนะเหล่านี้อาจทำให้ประสิทธิภาพของตัวแบบเปลี่ยนไป และสุดท้ายสามารถนำผลการทดลองในการค้นคว้าอิสระครั้งนี้ไปต่อยอด หรือเป็นแนวทางในการพัฒนาการบำรุงรักษาเครื่องจักรให้มีประสิทธิภาพดียิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[1]. SAS Institute Inc. 2021 [Online]. การเรียนรู้ของเครื่อง (machine learning). แหล่งข้อมูล : https://www.sas.com/th_th/insights/analytics/machine-learning.html

[2] Natdanai James — July 14, 2019 [Online]. Supervised Learning (การเรียนรู้แบบมีผู้สอน). แหล่งข้อมูล : https://www.glurgeek.com/education/ie311supervisedlearning/

[3] ชิตพงษ์ กิตตินราดร. มกราคม 2020 [Online]. Support Vector Machines. แหล่งข้อมูล : https://guopai.github.io/ml-blog08.html

[4] นายมธุพจน์ คำพา (2562). การวิเคราะห์ข้อมูลการใช้แอพสโตร์ด้วยเทคนิคเหมืองข้อมูล. สาขาวิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏกาญจนบุรี แหล่งข้อมูล: http://coms.kru.ac.th/tee/Projects/ShowPdf?name=637234216504001551.pdf&chk=False

[5] ชิตพงษ์ กิตตินราดร.มกราคม 2020 [Online]. Neural Network Algorithm. แหล่งข้อมูล : https://guopai.github.io/ml-blog14.html

[6] Mr. Digital Hits: 135163. 2021 [Online]. IOT กำลังจะเปลี่ยนโลก. แหล่งข้อมูล : https://www.ops.go.th/main/index.php/knowledge-base/article-pr/655-iot-กำลังจะเปลี่ยนโลก

[7] Susto, G. A., Schirru, A., Pampuri, S., McLoone, S., & Beghi, A. (2015). Machine Learning for Predictive Maintenance: A Multiple Classifiers Approach. IEEE Transactions on Industrial Informatics, 11(3), 812-820. แหล่งข้อมูล : https://doi.org/10.1109/TII.2014.2349359

[8] Marina Paolanti, Luca Romeo, Andrea Felicetti, Adriano Mancini, Emanuele Frontoni (July 2018). Machine Learning approach for Predictive Maintenance in Industry 4.0. DOI:10.1109/MESA.2018.8449150.
แหล่งข้อมูล : https://www.researchgate.net/publication/327334242

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[9] Ásgeir Halldórsson (2017). IIoT data collection for OEE measurements. แหล่งข้อมูล : https://skemman.is/handle/1946/26708

[10] Ameeth Kanawaday, Aditya Sane (Nov. 2017). Machine learning for predictive maintenance of industrial machines using IoT sensor data. 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS). แหล่งข้อมูล : https://www.researchgate.net/publication/324725326

[11] UNKNOWNCLASS. 2018 [Online]. pump_sensor_data. แหล่งข้อมูล : https://www.kaggle.com/datasets/nphantawee/pump-sensor-data

[12] C&G DataCommunity. กันยายน 2020 [Online]. MAE, MSE, RMSE — เลือกใช้ยังไงดีมาลองดูที่ความหมาย. แหล่งข้อมูล : https://medium.com/c-g-datacommunity/mse-rmse-mae- %A7%E0%B8%B2%E0%B8%A1%E0%B8%AB%E0%B8%A1%E0%B8%B2%E0%B8%A2-17b37b0b14b3

[13] Paranya Palwisut (January - June 2016). Improving Decision Tree Technique in Imbalanced Data Sets Using SMOTE for Internet Addiction Disorder Data. Available: https://ph01.tci-thaijo.org/index.php/IT_Journal/article/download/72840/58586/

[14] Amornthep Thongchiw. สิงหาคม 2018 [Online]. ทำความเข้าใจ accuracy, precision, recall, f1-score. แหล่งข้อมูล : https://thaidatascience.wordpress.com/2018/04/

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก
ขั้นตอนการจัดเตรียมข้อมูลเพื่อใช้ในการสอนและทดสอบโมเดล

ในส่วนนี้เป็นขั้นตอนในการศึกษาข้อมูล ทำความสะอาดข้อมูล พร้อมทั้งจัดเตรียมชุดข้อมูลเพื่อสามารถนำข้อมูลใช้ในการสอน และทดสอบโมเดล การศึกษาค้นคว้าอิสระได้จากแหล่งข้อมูล Kaggle มีดังต่อไปนี้

	timestamp	sensor_00	sensor_01	sensor_02	sensor_03	sensor_04	sensor_05	sensor_06	sensor_07	sensor_08	sensor_09	sensor_10	sensor_11	sensor_12	sensor_13	sensor_14	sensor_15	machine_status
0	04-01-18 00:00	2.465394	47.09201	53.2118	46.31076	634.375	76.45975	13.41146	16.13136	15.56713	15.05353	37.2274	47.52422	31.11716	1.681353	419.5747	461.8781	NORMAL
1	04-01-18 00:01	2.465394	47.09201	53.2118	46.31076	634.375	76.45975	13.41146	16.13136	15.56713	15.05353	37.2274	47.52422	31.11716	1.681353	419.5747	461.8781	NORMAL
2	04-01-18 00:02	2.444734	47.35243	53.2118	46.39757	638.8889	73.54598	13.32465	16.03733	15.61777	15.01013	37.86777	48.17723	32.08894	1.708474	420.848	462.7798	NORMAL
3	04-01-18 00:03	2.460474	47.09201	53.1684	46.39757	628.125	76.98898	13.31742	16.24711	15.69734	15.08247	38.57977	48.65607	31.67221	1.579427	420.7494	462.898	NORMAL
4	04-01-18 00:04	2.445718	47.13541	53.2118	46.39757	636.4583	76.58897	13.35359	16.21094	15.69734	15.08247	39.48939	49.06298	31.95202	1.683831	419.8926	461.4906	NORMAL
5	04-01-18 00:05	2.453588	47.09201	53.1684	46.39757	637.6157	78.18568	13.41146	16.16753	15.89265	15.16204	39.29406	49.37051	32.23816	1.673484	418.9049	461.8948	NORMAL
6	04-01-18 00:06	2.455556	47.04861	53.1684	46.39757	633.3333	75.81614	13.43316	16.13136	15.65393	15.08247	38.29974	49.57146	32.00982	1.684984	420.3324	464.2402	NORMAL
7	04-01-18 00:07	2.449653	47.13541	53.1684	46.39757	630.6713	75.77331	13.25231	16.12413	16.19647	15.08247	37.3396	49.32732	31.8832	1.646842	417.552	462.4563	NORMAL
8	04-01-18 00:08	2.463426	47.09201	53.1684	46.39757	631.9444	74.58916	13.28848	16.13136	15.47309	15.11863	38.45401	50.28795	32.09234	1.686156	422.0777	463.4988	NORMAL
9	04-01-18 00:09	2.445718	47.17882	53.1684	46.39757	641.7823	74.57428	13.38252	16.24711	15.61777	15.11863	39.52119	50.44635	32.25679	1.637774	421.4344	463.4123	NORMAL
10	04-01-18 00:10	2.46441	47.48264	53.125	46.39757	637.7314	76.05148	13.41146	16.16753	15.65393	15.11863	39.90199	50.48941	32.80076	1.678884	419.334	462.5085	NORMAL
11	04-01-18 00:11	2.444734	47.91666	53.1684	46.39757	635.6482	74.58654	13.41146	16.16753	15.84925	15.11863	39.78521	50.7882	33.14559	1.62582	420.2469	464.127	NORMAL
12	04-01-18 00:12	2.460474	48.26389	53.125	46.39757	630.0926	76.95988	13.34635	16.16753	15.73351	15.01013	40.04284	50.50479	33.46326	1.641763	420.9848	462.6014	NORMAL
13	04-01-18 00:13	2.448669	48.4375	53.1684	46.39757	638.6574	75.6731	13.31742	16.16753	15.84925	15.16927	40.90296	50.96519	33.63691	1.67578	417.9775	463.6844	NORMAL
14	04-01-18 00:14	2.453588	48.56771	53.1684	46.39757	632.4074	80.65949	13.38976	16.13136	15.53096	15.0897	41.82584	51.72565	34.46501	1.620947	419.8773	460.1368	NORMAL
15	04-01-18 00:15	2.455556	48.3941	53.125	46.39757	642.3611	78.13193	13.35359	16.21094	15.45139	15.12587	43.12774	51.89335	35.90224	1.675951	420.5849	462.8748	NORMAL
16	04-01-18 00:16	2.449653	48.3941	53.1684	46.31076	630.2084	77.89381	13.30295	16.16753	15.89265	15.08247	43.59909	51.45698	36.59018	1.701383	420.1925	463.1302	NORMAL
17	04-01-18 00:17	2.463426	48.4809	53.68924	46.31076	643.6343	77.30572	13.34635	16.16753	15.61777	15.01013	43.86082	51.56866	36.85331	1.743944	421.5702	465.1814	NORMAL
18	04-01-18 00:18	2.445718	48.61111	53.125	46.31076	632.9861	76.66199	13.34635	16.21094	15.81308	15.05353	43.36131	51.7037	36.81083	1.703058	417.1931	460.6873	NORMAL
19	04-01-18 00:19	2.46441	48.61111	53.1684	46.31076	644.3287	78.49116	13.34635	16.13136	15.70457	15.08247	42.28162	51.2277	36.4342	1.732553	420.9559	461.9449	NORMAL
20	04-01-18 00:20	2.445718	49.08854	53.03819	46.31076	633.4491	76.95741	13.34635	16.16753	15.76968	15.11863	42.12564	51.53938	36.60661	1.784164	419.6276	463.2421	NORMAL
21	04-01-18 00:21	2.460474	49.21875	53.125	46.31076	626.2731	78.76208	13.34635	16.16753	15.45139	15.11863	41.94852	51.88231	37.28117	1.866817	421.0786	463.0934	NORMAL
22	04-01-18 00:22	2.448669	48.78472	53.125	46.26736	635.4166	76.26164	13.41146	16.21094	15.65393	15.08247	42.94177	51.06642	37.80445	1.892598	419.7959	463.1438	NORMAL
23	04-01-18 00:23	2.453588	49.08854	53.1684	46.26736	635.4166	79.25443	13.34635	16.21094	15.81308	15.08247	44.50991	51.11165	37.12904	1.794096	420.4793	462.9269	NORMAL

รูปที่ ก.1 อ่านข้อมูลจาก .csv ไฟล์

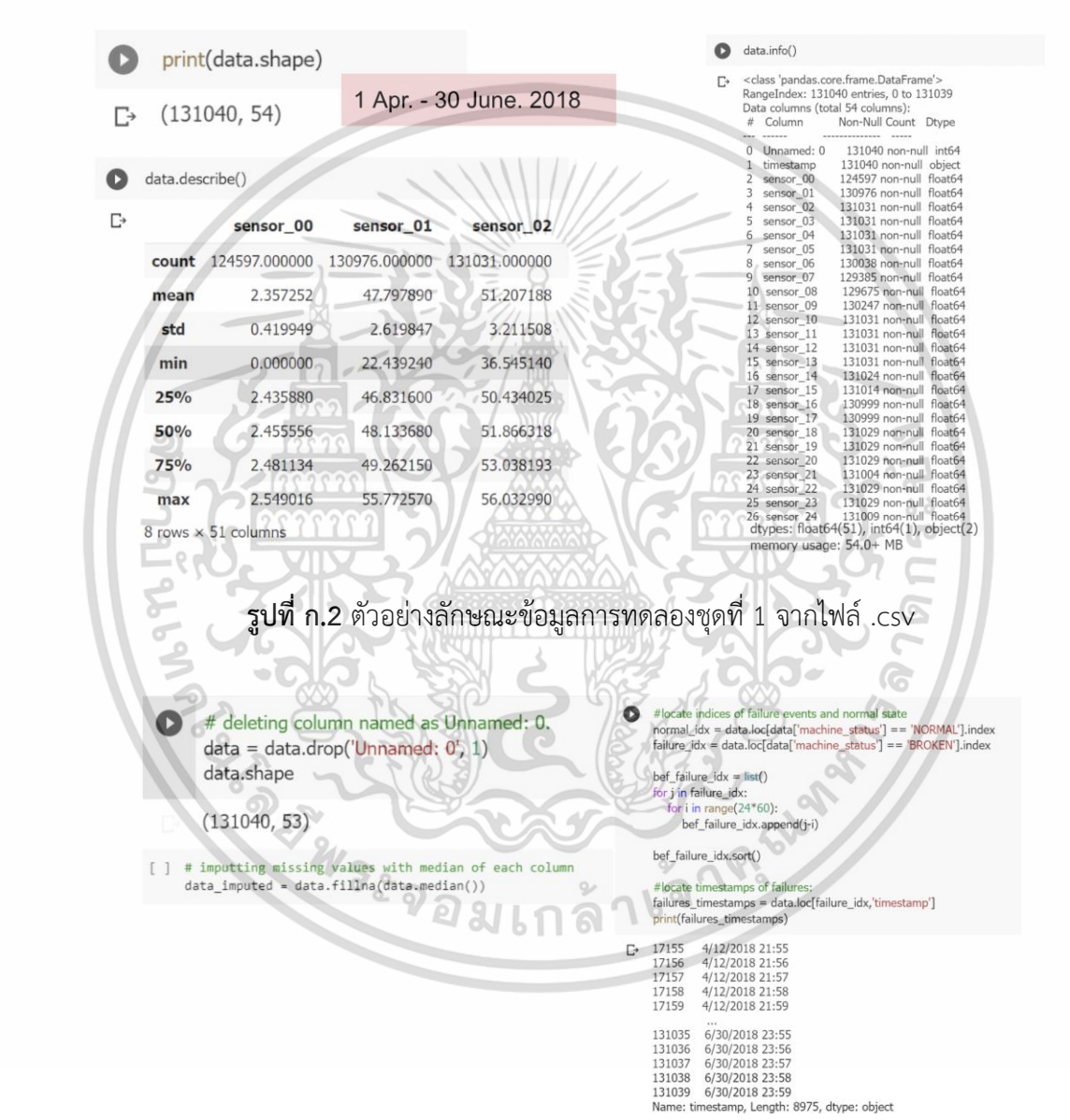
จากรูปที่ ก.1 แสดงตัวอย่างประเภทของของมูลแบบคร่าวๆ พบว่ามีการเก็บข้อมูลข้อมูลเป็นวินาที ลักษณะข้อมูลที่นำมาใช้เป็นทศนิยม 5 ตำแหน่ง และตัวแปรเป้าหมายมี 2 ประเภท คือ ‘NORMAL’ และ ‘BROKEN’ ซึ่งพิจารณาจากชุดข้อมูลที่นำมาทดลองเป็นระบบปั๊มน้ำ (Water Pump System) จำนวนอุปกรณ์ที่ใช้งานทั้งหมด 51 เซนเซอร์ จำนวนข้อมูลที่ถูกบันทึกทั้งหมด 220,320 ข้อมูล

ก.1 ชุดข้อมูลการทดลองชุดที่ 1

การทดลองชุดที่ 1 เป็นข้อมูลวันที่ 1 เมษายน – 30 มิถุนายน พ.ศ. 2561 ขั้นตอนการทำความสะอาดข้อมูล ขั้นตอนแรกศึกษาลักษณะของข้อมูลดังรูปที่ ก.2 จากนั้นทำความสะอาดข้อมูลเริ่มจากการลบตารางที่ไม่เกี่ยวข้องออก และเพิ่มค่ามัธยฐาน (Median) ลงไปในข้อมูลที่ว่าง หรือ N/A ดังรูปที่ ก.3 จากนั้นทำการปรับขนาดของข้อมูลให้อยู่ระหว่าง 0 ถึง 1 ด้วยการ Rescaling หรือ Min-Max Normalization ดังรูปที่ ก.6 พร้อมทั้งศึกษา Heatmap เพื่อดูความสัมพันธ์ของตัวแปรอิสระ หากความสัมพันธ์มีมากถึง 0.8 ให้ทำการนำคอลัมน์นั้นออก ดังรูปที่ ก.7 และรูปที่ ก.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการเตรียมข้อมูลเพื่อใช้ในการสอน และทดสอบโมเดล เริ่มจากนำชุดข้อมูลที่ทำความสะอาดเรียบร้อยแล้วมาแบ่งข้อมูลเป็นชุดข้อมูลสำหรับการสอนโมเดล และชุดข้อมูลสำหรับทดสอบโมเดลในอัตราส่วน 70 และ 30 ตามลำดับ ดังรูปที่ ก.59 พบว่าข้อมูลตัวแปรเป้าหมายมีความไม่สมดุลกัน จึงใช้เทคนิคการสุ่มเพิ่มตัวอย่างกลุ่มน้อย (SMOTE) เพื่อแก้ปัญหา ดังรูปที่ ก.60 และนำข้อมูลเหล่านี้ไปสร้างตัวแบบในขั้นตอนต่อไป



รูปที่ ก.2 ตัวอย่างลักษณะข้อมูลการทดลองชุดที่ 1 จากไฟล์ .csv

รูปที่ ก.3 ตัวอย่างการทำความสะอาดข้อมูลการทดลองชุดที่ 1 (1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
[ ]  #removing outliers using zscore
     z_scores = zscore(data_imputed.iloc[:,1:52])
     abs_z_scores = np.abs(z_scores)
     filtered_entries = (abs_z_scores < 3).all(axis=1)
     data_outlir = data_imputed[filtered_entries]
     data_outlir
```

	timestamp	sensor_00	sensor_01	sensor_02	sensor_03	sensor_04	sensor_05	sensor_06	sensor_07	sensor_08	...
0	4/1/2018 0:00	2.465394	47.09201	53.21180	46.310760	634.375000	76.45975	13.41146	16.13136	15.56713	...
1	4/1/2018 0:01	2.465394	47.09201	53.21180	46.310760	634.375000	76.45975	13.41146	16.13136	15.56713	...
2	4/1/2018 0:02	2.444734	47.35243	53.21180	46.397570	638.888900	73.54598	13.32465	16.03733	15.61777	...
3	4/1/2018 0:03	2.460474	47.09201	53.16840	46.397568	628.125000	76.98898	13.31742	16.24711	15.69734	...
4	4/1/2018 0:04	2.445718	47.13541	53.21180	46.397568	636.458300	76.58897	13.35359	16.21094	15.69734	...
...	...	...	...	...	...	...	...	...	...	...	...
127669	6/28/2018 15:49	2.005961	47.78646	50.13021	44.314236	562.500000	99.80158	12.59404	16.21094	15.69734	...
127672	6/28/2018 15:52	2.003009	47.09201	50.13021	44.314236	452.555000	99.55959	12.59404	16.06626	15.77691	...
127680	6/28/2018 16:00	1.663600	47.91666	50.13021	44.357636	203.683441	99.10869	12.59404	16.16753	15.90712	...
127681	6/28/2018 16:01	1.663600	47.91666	50.13021	44.357636	203.683441	99.10869	12.59404	16.16753	15.90712	...
127682	6/28/2018 16:02	1.656713	47.74305	50.08680	44.357636	200.558441	99.60864	12.62297	16.16753	15.61777	...

103326 rows × 53 columns

รูปที่ ก.4 ตัวอย่างการทำความสะอาดข้อมูลการทดลองชุดที่ 1 (2)

```
# how many data has been lost after removing outliers
((131040 - 103326)/131040) * 100
```

21.149267399267398

```
[ ]  #After remove outlier
     data_outlir['machine_status'].value_counts()
```

NORMAL 103296
BROKEN 30
Name: machine_status, dtype: int64

```
[ ]  #Before remove outlier
     data_imputed['machine_status'].value_counts()
```

NORMAL 122065
BROKEN 8975
Name: machine_status, dtype: int64

```
[ ]  # how many data has been lost after removing outliers(Target : Broken)
     ((8975 - 30)/8975) * 100
```

99.66573816155989

```
[ ]  # how many data has been lost after removing outliers(Target : Normal)
     ((122065 - 103296)/122065) * 100
```

15.376233973702536

รูปที่ ก.5 ตัวอย่างการทำความสะอาดข้อมูลการทดลองชุดที่ 1 (3)

จากรูปที่ ก.4 ทำการลบค่าที่ผิดปกติ (Outlier) ของข้อมูลออก สังเกตว่าหลังจากตัดข้อมูลที่ผิดปกติออกแล้ว ตัวแปรเป้าหมาย สถานะ ‘NORMAL’ จำนวน 103,296 ข้อมูล คิดเป็น 15.38% และสถานะ ‘BROKEN’ จำนวน 30 ข้อมูล คิดเป็น 99.67% ดังรูปที่ ก.5 ดังนั้นเพื่อไม่ให้ส่งผลกับประสิทธิภาพของตัวแบบในภายหลัง จึงเลือกวิธีไม่ลบค่าผิดปกติออกจากข้อมูล

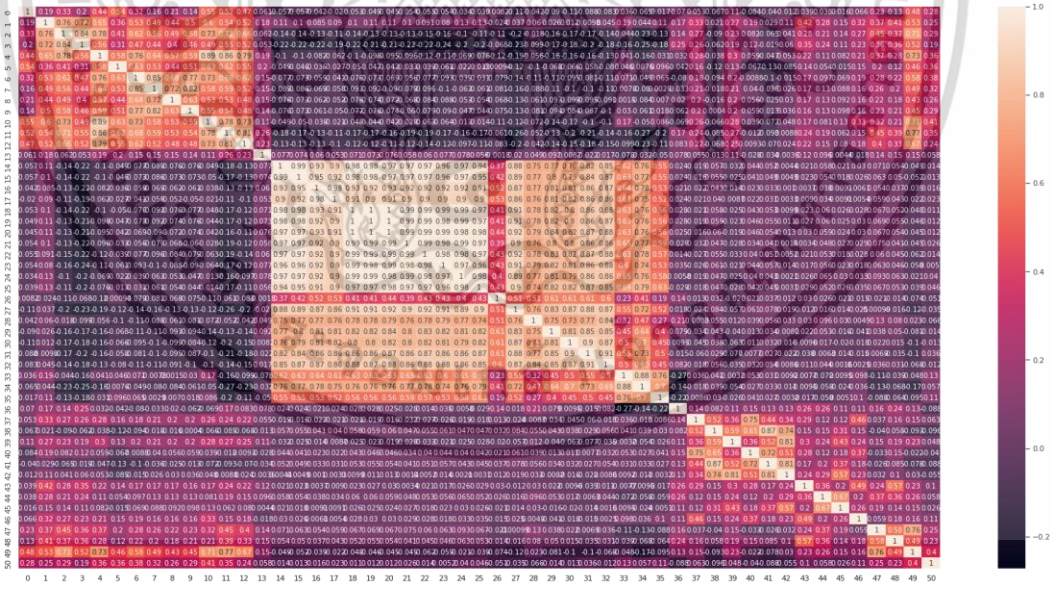
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
# standarizing the dataset using minmax scalar
scaler = MinMaxScaler()
data_std = scaler.fit_transform(data_imputed.iloc[:,1:52])
data_std = pd.DataFrame(data_std)
data_std
```

	0	1	2	3	4	5	6	7	8	9	...
0	0.967194	0.739583	0.855233	0.872093	0.792242	0.764598	0.602472	0.686154	0.667495	0.675211	...
1	0.967194	0.739583	0.855233	0.872093	0.792242	0.764598	0.602472	0.686154	0.667495	0.675211	...
2	0.959089	0.747396	0.855233	0.877907	0.797904	0.735461	0.598568	0.682154	0.669671	0.673264	...
3	0.965264	0.739583	0.853006	0.877907	0.784402	0.769891	0.598243	0.691077	0.673089	0.676509	...
4	0.959475	0.740885	0.855233	0.877907	0.794855	0.765891	0.599870	0.689539	0.673089	0.676509	...
...	...	...	...	...	...	...	...	...	...	...	...
131035	0.963335	0.407552	0.120267	0.145349	0.000675	1.000000	0.603123	0.686154	0.654444	0.676509	...
131036	0.963335	0.407552	0.120267	0.145349	0.000436	1.000000	0.603123	0.686154	0.654444	0.676509	...
131037	0.963335	0.407552	0.118040	0.145349	0.000820	1.000000	0.603123	0.686154	0.654444	0.676509	...
131038	0.963335	0.407552	0.120267	0.142441	0.000290	1.000000	0.603123	0.686154	0.654444	0.676509	...
131039	0.963335	0.407552	0.118040	0.142442	0.000675	1.000000	0.603123	0.686154	0.654444	0.676509	...

131040 rows × 51 columns

รูปที่ ก.6 ตัวอย่างการทำความสะอาดข้อมูลการทดลองชุดที่ 1 (4)



รูปที่ ก.7 Heatmap ข้อมูลการทดลองชุดที่ 1 ก่อนทำความสะอาดข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
le = LabelEncoder()
Y = le.fit_transform(data_imputed['machine_status'])
Y
```

```
array([1, 1, 1, ..., 0, 0, 0])
```

```
X = correlation(data_std, 0.8)
X
```

	0	1	2	4	5	6	8	9	12	13	...
0	0.967194	0.739583	0.855233	0.792242	0.764598	0.602472	0.667495	0.675211	0.691492	0.053911	...
1	0.967194	0.739583	0.855233	0.792242	0.764598	0.602472	0.667495	0.675211	0.691492	0.053911	...
2	0.959089	0.747396	0.855233	0.797904	0.735461	0.598568	0.669671	0.673264	0.713088	0.054781	...
3	0.965264	0.739583	0.853006	0.784402	0.769891	0.598243	0.673089	0.676509	0.703827	0.050643	...
4	0.959475	0.740885	0.855233	0.794855	0.765891	0.599870	0.673089	0.676509	0.710045	0.053990	...
...	...	...	...	...	...	...	...	...	...	...	...
131035	0.963335	0.407552	0.120267	0.000675	1.000000	0.603123	0.654444	0.676509	0.000000	0.000000	...
131036	0.963335	0.407552	0.120267	0.000436	1.000000	0.603123	0.654444	0.676509	0.000000	0.000000	...
131037	0.963335	0.407552	0.118040	0.000820	1.000000	0.603123	0.654444	0.676509	0.000000	0.000000	...
131038	0.963335	0.407552	0.120267	0.000290	1.000000	0.603123	0.654444	0.676509	0.000000	0.000000	...
131039	0.963335	0.407552	0.118040	0.000675	1.000000	0.603123	0.654444	0.676509	0.000000	0.000000	...

131040 rows × 29 columns

รูปที่ ก.8 ตัวอย่างการทำความสะอาดข้อมูลการทดลองชุดที่ 1 (5)

```
[19] X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.3, random_state=1234)

[20] print("Number transactions x_train dataset: ", X_train.shape)
     print("Number transactions y_train dataset: ", y_train.shape)
     print("Number transactions x_test dataset: ", X_test.shape)
     print("Number transactions y_test dataset: ", y_test.shape)

     Number transactions x_train dataset:  (91728, 29)
     Number transactions y_train dataset:  (91728,)
     Number transactions x_test dataset:  (39312, 29)
     Number transactions y_test dataset:  (39312,)
```

รูปที่ ก.9 ชุดข้อมูลสำหรับการสอน และทดสอบโมเดลของข้อมูลการทดลองชุดที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
[33] from imblearn.over_sampling import SMOTE
     from imblearn.over_sampling import RandomOverSampler

[ ]  print("Before OverSampling, counts of label '1': {}".format(sum(y_train==1)))
     print("Before OverSampling, counts of label '0': {} \n".format(sum(y_train==0)))

     sm = SMOTE(random_state=1234, sampling_strategy='minority')
     X_train_res, y_train_res = sm.fit_resample(X_train, y_train)

     print('After OverSampling, the shape of train_X: {}'.format(X_train_res.shape))
     print('After OverSampling, the shape of train_y: {} \n'.format(y_train_res.shape))

     print("After OverSampling, counts of label '1': {}".format(sum(y_train_res==1)))
     print("After OverSampling, counts of label '0': {}".format(sum(y_train_res==0)))

     Before OverSampling, counts of label '1': 85425
     Before OverSampling, counts of label '0': 6303

     After OverSampling, the shape of train_X: (170850, 29)
     After OverSampling, the shape of train_y: (170850,)

     After OverSampling, counts of label '1': 85425
     After OverSampling, counts of label '0': 85425
```

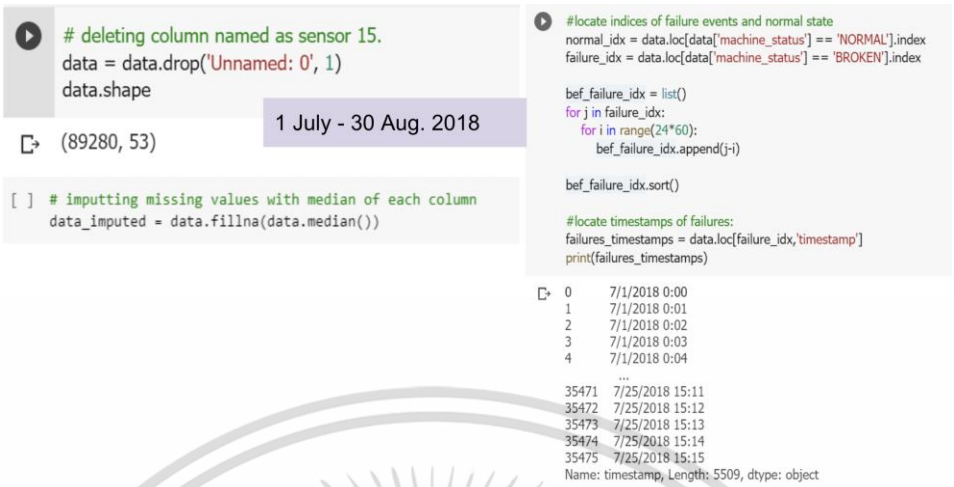
รูปที่ ก.10 เทคนิคการสุ่มเพิ่มตัวอย่างกลุ่มน้อยของข้อมูลการทดลองชุดที่ 1

ก.2 ชุดข้อมูลการทดลองชุดที่ 2

การทดลองชุดที่ 1 เป็นข้อมูลวันที่ 1 กรกฎาคม – 30 สิงหาคม พ.ศ. 2561 ขั้นตอนการทำความสะอาดข้อมูล เงื่อนไขเหมือนกับการทดลองชุดที่ 1 โดยขั้นตอนแรกศึกษาลักษณะของข้อมูล จากนั้นทำความสะอาดข้อมูลเริ่มจากการลบตารางที่ไม่เกี่ยวข้องออก และเพิ่มค่ามัธยฐาน (Median) ลงไปในข้อมูลที่ว่าง หรือ N/A ดังรูปที่ ก.11 จากนั้นทำการปรับขนาดของข้อมูลให้อยู่ระหว่าง 0 ถึง 1 ด้วยการ Rescaling หรือ Min-Max Normalization ดังรูปที่ ก.14 พร้อมทั้งศึกษา Heatmap เพื่อดูความสัมพันธ์ของแปรอิสระ หากความสัมพันธ์มีมากถึง 0.8 ให้ทำการนำคอลัมน์นั้นออก ดังรูปที่ ก.15 และรูปที่ ก.16

ขั้นตอนการเตรียมข้อมูลเพื่อใช้ในการสอน และทดสอบโมเดล เริ่มจากนำชุดข้อมูลที่ทำความสะอาดเรียบร้อยแล้วมาแบ่งข้อมูลเป็นชุดข้อมูลสำหรับการสอนโมเดล และชุดข้อมูลสำหรับทดสอบโมเดลในอัตราส่วน 70 และ 30 ตามลำดับ ดังรูปที่ ก.17 พบว่าข้อมูลตัวแปรเป้าหมายมีความไม่สมดุลกัน จึงใช้เทคนิคการสุ่มเพิ่มตัวอย่างกลุ่มน้อย (SMOTE) เพื่อแก้ปัญหา ดังรูปที่ ก.18 และนำข้อมูลเหล่านี้ไปสร้างตัวแบบในขั้นตอนต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```
# deleting column named as sensor 15.
data = data.drop('Unnamed: 0', 1)
data.shape
```

1 July - 30 Aug. 2018

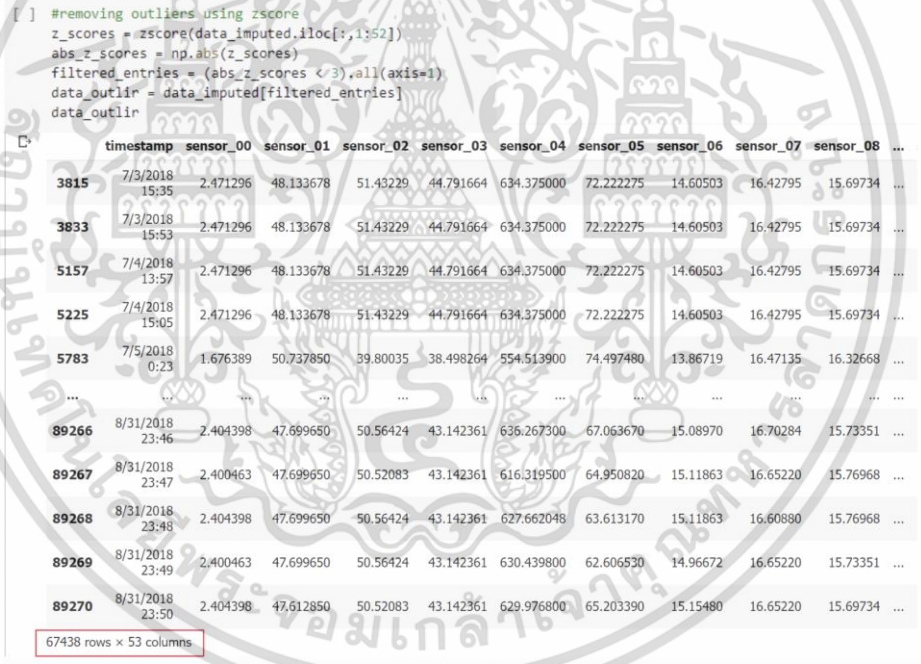
```
(89280, 53)
```

```
[ ] # imputting missing values with median of each column
    data_imputed = data.fillna(data.median())
```

```
#locate indices of failure events and normal state
normal_idx = data.loc[data['machine_status'] == 'NORMAL'].index
failure_idx = data.loc[data['machine_status'] == 'BROKEN'].index

bef_failure_idx = list()
for j in failure_idx:
    for i in range(24*60):
        bef_failure_idx.append(j-i)

bef_failure_idx.sort()

#locate timestamps of failures:
failures_timestamps = data.loc[failure_idx,'timestamp']
print(failures_timestamps)
```

```
0        7/1/2018 0:00
1        7/1/2018 0:01
2        7/1/2018 0:02
3        7/1/2018 0:03
4        7/1/2018 0:04
            ...
35471    7/25/2018 15:11
35472    7/25/2018 15:12
35473    7/25/2018 15:13
35474    7/25/2018 15:14
35475    7/25/2018 15:15
Name: timestamp, Length: 5509, dtype: object
```

รูปที่ ก.11 ตัวอย่างการทำความสะอาดข้อมูลการทดลองชุดที่ 2 (1)



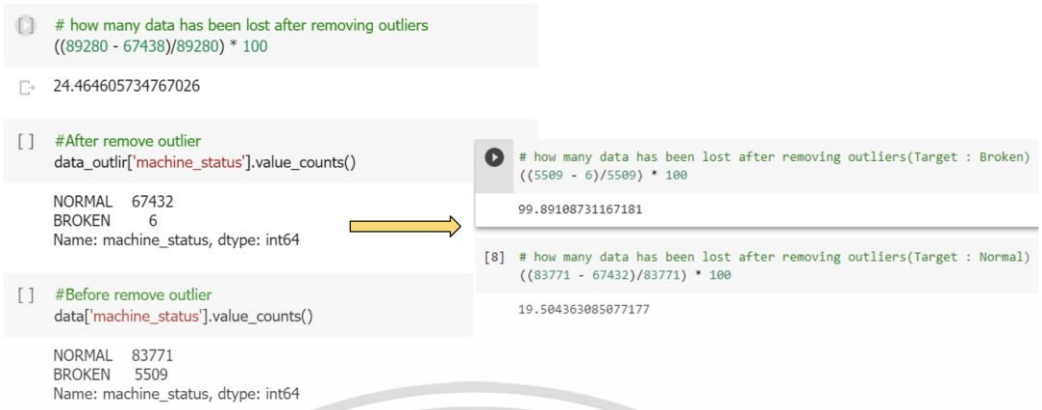
```
[ ] #removing outliers using zscore
    z_scores = zscore(data_imputed.iloc[:,1:52])
    abs_z_scores = np.abs(z_scores)
    filtered_entries = (abs_z_scores < 3).all(axis=1)
    data_outlir = data_imputed[filtered_entries]
    data_outlir
```

	timestamp	sensor_00	sensor_01	sensor_02	sensor_03	sensor_04	sensor_05	sensor_06	sensor_07	sensor_08	...
3815	7/3/2018 15:35	2.471296	48.133678	51.43229	44.791664	634.375000	72.222275	14.60503	16.42795	15.69734	...
3833	7/3/2018 15:53	2.471296	48.133678	51.43229	44.791664	634.375000	72.222275	14.60503	16.42795	15.69734	...
5157	7/4/2018 13:57	2.471296	48.133678	51.43229	44.791664	634.375000	72.222275	14.60503	16.42795	15.69734	...
5225	7/4/2018 15:05	2.471296	48.133678	51.43229	44.791664	634.375000	72.222275	14.60503	16.42795	15.69734	...
5783	7/5/2018 0:23	1.676389	50.737850	39.80035	38.498264	554.513900	74.497480	13.86719	16.47135	16.32668	...
...	...	...	...	...	...	...	...	...	...	...	...
89266	8/31/2018 23:46	2.404398	47.699650	50.56424	43.142361	636.267300	67.063670	15.08970	16.70284	15.73351	...
89267	8/31/2018 23:47	2.400463	47.699650	50.52083	43.142361	616.319500	64.950820	15.11863	16.65220	15.76968	...
89268	8/31/2018 23:48	2.404398	47.699650	50.56424	43.142361	627.662048	63.613170	15.11863	16.60880	15.76968	...
89269	8/31/2018 23:49	2.400463	47.699650	50.56424	43.142361	630.439800	62.606530	14.96672	16.65220	15.73351	...
89270	8/31/2018 23:50	2.404398	47.612850	50.52083	43.142361	629.976800	65.203390	15.15480	16.65220	15.69734	...

67438 rows × 53 columns

รูปที่ ก.12 ตัวอย่างการทำความสะอาดข้อมูลการทดลองชุดที่ 2 (2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.13 ตัวอย่างการทำความสะอาดข้อมูลการทดลองชุดที่ 2 (3)

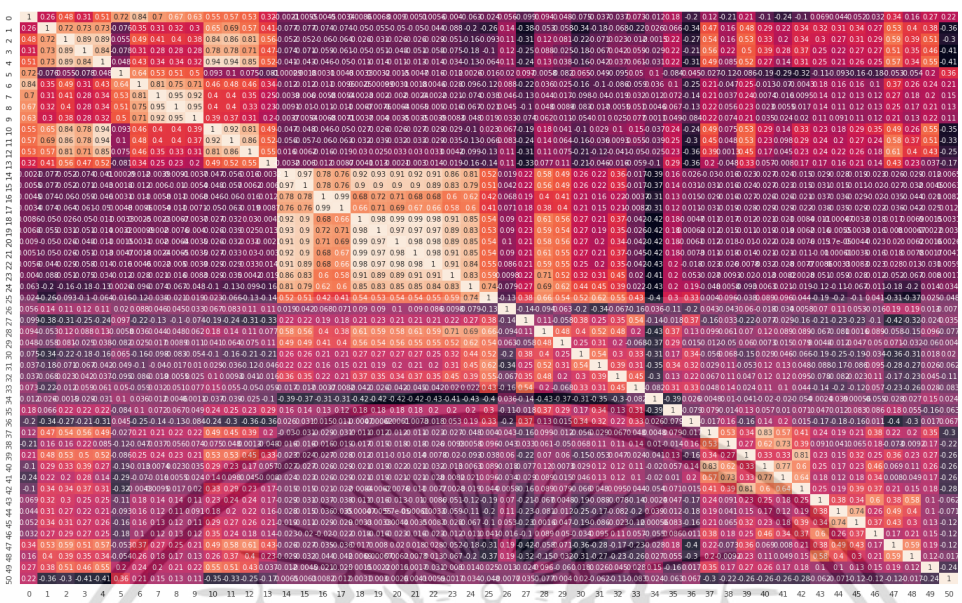
จากรูปที่ ก.12 ทำการลบค่าที่ผิดปกติ (Outlier) ของข้อมูลออก สังเกตว่าหลังจากตัดข้อมูลที่ผิดปกติออกแล้ว มีตัวแปรเป้าหมาย สถานะ ‘NORMAL’ จำนวน 67,432 ข้อมูล คิดเป็น 19.50% และสถานะ ‘BROKEN’ จำนวน 6 ข้อมูล คิดเป็น 99.89% ดังรูปที่ ก.13 ดังนั้นเพื่อไม่ให้ส่งผลกับประสิทธิภาพของตัวแบบในภายหลัง จึงเลือกวิธีไม่ลบค่าผิดปกติออกจากข้อมูล

	0	1	2	3	4	5	6	7	8	9	...
0	0.975155	0.634277	0.255361	0.234501	0.000517	1.000000	0.849621	0.695826	0.644259	0.604631	...
1	0.975155	0.634277	0.255361	0.234501	0.000517	1.000000	0.849621	0.695826	0.644259	0.604631	...
2	0.975155	0.634277	0.255361	0.234501	0.000973	1.000000	0.849621	0.695826	0.644259	0.604631	...
3	0.975155	0.634277	0.255361	0.234501	0.000344	1.000000	0.849621	0.695826	0.644259	0.604631	...
4	0.975155	0.634277	0.255361	0.234501	0.000801	1.000000	0.849621	0.695826	0.644259	0.604631	...
...	...	...	...	...	...	...	...	...	...	...	...
89275	0.949923	0.840857	0.779727	0.714286	0.940405	0.645910	0.879528	0.705341	0.642474	0.606368	...
89276	0.947205	0.840857	0.781677	0.714286	0.934721	0.658337	0.881634	0.707490	0.642474	0.604631	...
89277	0.945652	0.840857	0.779727	0.714286	0.927314	0.672945	0.877843	0.707490	0.644259	0.604631	...
89278	0.949534	0.840857	0.779727	0.714286	0.941783	0.650918	0.879528	0.701657	0.646044	0.604631	...
89279	0.945652	0.840857	0.779727	0.714286	0.947983	0.654564	0.879528	0.705341	0.642474	0.600290	...

89280 rows × 51 columns

รูปที่ ก.14 ตัวอย่างการทำความสะอาดข้อมูลการทดลองชุดที่ 2 (4)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.15 Heatmap ข้อมูลการทดลองชุดที่ 2 ก่อนทำความสะอาดข้อมูล

```
[ ]  le = LabelEncoder()
     Y = le.fit_transform(data_imputed['machine_status'])
     Y

     array([0, 0, 0, ..., 1, 1, 1])

     X = correlation(data_std, 0.8)
     X
```

	0	1	2	5	7	13	14	16	25	26	...
0	0.975155	0.634277	0.255361	1.000000	0.695826	0.000000	0.829055	0.718860	0.824919	0.257532	...
1	0.975155	0.634277	0.255361	1.000000	0.695826	0.000000	0.829055	0.718860	0.824919	0.257532	...
2	0.975155	0.634277	0.255361	1.000000	0.695826	0.000000	0.824791	0.726040	0.822962	0.239550	...
3	0.975155	0.634277	0.255361	1.000000	0.695826	0.000000	0.824791	0.726040	0.822962	0.239550	...
4	0.975155	0.634277	0.255361	1.000000	0.695826	0.000000	0.827386	0.723018	0.824937	0.228788	...
...	...	...	...	...	...	...	...	...	...	...	...
89275	0.949923	0.840857	0.779727	0.645910	0.705341	0.492069	0.828343	0.710697	0.695560	0.346057	...
89276	0.947205	0.840857	0.781677	0.658337	0.707490	0.491214	0.831689	0.735893	0.699270	0.348900	...
89277	0.945652	0.840857	0.779727	0.672945	0.707490	0.489225	0.826906	0.736370	0.699462	0.352330	...
89278	0.949534	0.840857	0.779727	0.650918	0.701657	0.488662	0.827777	0.714279	0.692578	0.353485	...
89279	0.945652	0.840857	0.779727	0.654564	0.705341	0.486898	0.829350	0.706224	0.699113	0.351813	...

89280 rows × 32 columns

รูปที่ ก.16 ตัวอย่างการทำความสะอาดข้อมูลการทดลองชุดที่ 2 (5)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.3, random_state=1234)

print("Number transactions x_train dataset: ", X_train.shape)
print("Number transactions y_train dataset: ", y_train.shape)
print("Number transactions x_test dataset: ", X_test.shape)
print("Number transactions y_test dataset: ", y_test.shape)

Number transactions x_train dataset:  (62496, 32)
Number transactions y_train dataset:  (62496,)
Number transactions x_test dataset:  (26784, 32)
Number transactions y_test dataset:  (26784,)
```

รูปที่ ก.17 ชุดข้อมูลสำหรับการสอน และทดสอบโมเดลของข้อมูลการทดลองชุดที่ 2

```
[33] from imblearn.over_sampling import SMOTE
     from imblearn.over_sampling import RandomOverSampler

print("Before OverSampling, counts of label '1': {}".format(sum(y_train==1)))
print("Before OverSampling, counts of label '0': {} \n".format(sum(y_train==0)))

sm = SMOTE(random_state=1234, sampling_strategy='minority')
X_train_res, y_train_res = sm.fit_resample(X_train, y_train)

print('After OverSampling, the shape of train_X: {}'.format(X_train_res.shape))
print('After OverSampling, the shape of train_y: {} \n'.format(y_train_res.shape))

print("After OverSampling, counts of label '1': {}".format(sum(y_train_res==1)))
print("After OverSampling, counts of label '0': {}".format(sum(y_train_res==0)))

Before OverSampling, counts of label '1': 58669
Before OverSampling, counts of label '0': 3827

After OverSampling, the shape of train_X: (117338, 32)
After OverSampling, the shape of train_y: (117338,)

After OverSampling, counts of label '1': 58669
After OverSampling, counts of label '0': 58669
```

รูปที่ ก.18 เทคนิคการสุ่มเพิ่มตัวอย่างกลุ่มน้อยของข้อมูลการทดลองชุดที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

ขั้นตอนการสอน ทดสอบ และปรับแต่งค่าพารามิเตอร์ให้เหมาะสมกับโมเดล

ในขั้นตอนนี้เป็นส่วนการสอน ทดสอบ และปรับแต่งค่าพารามิเตอร์ให้เหมาะสมกับโมเดล แต่ละชุดการทดลองแบ่งออกเป็น 2 ส่วน คือส่วนที่ทำการสอน-ทดสอบโมเดล และส่วนที่ปรับแต่งค่าพารามิเตอร์มีดังต่อไปนี้

1) ศึกษาภาพรวมของข้อมูลที่ทำความสะอาดเรียบร้อยแล้วจากชุดข้อมูลการสอน และชุดข้อมูลการทดสอบเพื่อนำไปใช้กับโมเดล

2) ศึกษาผลการทดสอบ ค่า Accuracy, Precision, Recall, F1-Scores, ROC AUC score ที่ได้จากโมเดลแต่ละตัว พร้อมทั้งปรับแต่งค่าพารามิเตอร์ให้เหมาะสม

ข.1 ชุดข้อมูลการทดลองชุดที่ 1

จากการศึกษาภาพรวมของข้อมูลที่ทำความสะอาดเรียบร้อยแล้วจากชุดข้อมูลการสอน และชุดข้อมูลการทดสอบ นำไปสู่การสร้างโมเดล ดังรูปที่ ข.1 โดยทำการสร้างโมเดลทั้งหมด 7 โมเดล และเลือกมา 3 โมเดล คือ Support Vector Machine, Decision Tree และ Neural Network เพื่อเปรียบเทียบประสิทธิภาพวิธีการจำแนกกลุ่มข้อมูล โดยมีการกำหนดวิธีการวิเคราะห์ความแม่นตรงของตัวแบบ K=10

ผลการทดสอบก่อนปรับค่าพารามิเตอร์พบว่า โมเดล Decision Tree มีค่าความถูกต้อง 99.95% และ Support Vector Machine มีค่าความถูกต้องที่ 99.86% ดังรูปที่ ข.2 จากนั้นทำการปรับแต่งค่าพารามิเตอร์ โดยใช้ GridSarchCV ดังรูปที่ ข.3 พบว่าค่าพารามิเตอร์ที่ดีที่สุดของโมเดล Decision Tree มีเกณฑ์การแบ่งข้อมูลแบบ ‘Entropy’, Support Vector Machine ใช้วิธีเรเดียลเบสิสฟังก์ชัน (Radial Basis Function : RBF) เป็นฟังก์ชันเคอร์เนลยอดนิยมที่ใช้ในอัลกอริทึมการเรียนรู้เคอร์เนลต่างๆ โดยกำหนดค่า ‘c =10’ ซึ่งเป็นตัวกำหนดขนาดของเรกูลาไรซ์ เพื่อเลือกระดับของการอนุญาตให้มีการละเมิดขอบเขตเส้นประ ดังรูปที่ 2.1 และ โมเดล Neural Network กำหนดขนาด Batch size =32, Epochs = 50 รายละเอียดดังรูปที่ ข.4 และรูปที่ ข.5

หลังจากปรับแต่งค่าพารามิเตอร์ให้เหมาะสมกับแต่ละโมเดลเรียบร้อยแล้ว จากนั้นทำการสอนและทดสอบแต่ละโมเดลอีกรอบ ได้ผลลัพธ์ดังรูปที่ ข.6 ถึงรูปที่ ข.9 พบว่าโมเดล Decision Tree ให้ค่าความถูกต้องสูงที่สุด และค่าความคลาดเคลื่อนต่ำที่สุด ตามด้วย Support Vector Machine และ Neural Network ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
models = []
models.append(['Logistic Regreesion', LogisticRegression(random_state=0)])
models.append(['SVM', SVC(random_state=0)])
models.append(['KNeighbors', KNeighborsClassifier()])
models.append(['GaussianNB', GaussianNB()])
models.append(['Decision Tree', DecisionTreeClassifier(random_state=0)])
models.append(['Random Forest', RandomForestClassifier(random_state=0)])
models.append(['XGBoost', XGBClassifier(eval_metric= 'error')])

lst_1= []

for m in range(len(models)):
    lst_2= []
    model = models[m][1]
    model.fit(X_train_res, y_train_res)
    y_pred = model.predict(X_test)
    cm = confusion_matrix(y_test, y_pred)  #Confusion Matrix
    accuracies = cross_val_score(estimator = model, X = X_train_res, y = y_train_res, cv = 10)   #K-Fold Validation
    roc = roc_auc_score(y_test, y_pred)  #ROC AUC Score
    precision = precision_score(y_test, y_pred)  #Precision Score
    recall = recall_score(y_test, y_pred)  #Recall Score
    f1 = f1_score(y_test, y_pred)  #F1 Score
    print(models[m][0],':')
    print(cm)
    print('Accuracy Score: ',accuracy_score(y_test, y_pred))
    print('')
    print("K-Fold Validation Mean Accuracy: {:.2f} %".format(accuracies.mean()*100))
    print('')
    print("Standard Deviation: {:.2f} %".format(accuracies.std()*100))
    print('')
    print('ROC AUC Score: {:.2f}'.format(roc))
    print('')
    print('Precision: {:.2f}'.format(precision))
    print('')
    print('Recall: {:.2f}'.format(recall))
    print('')
    print('F1: {:.2f}'.format(f1))
    print('----------------------------------')
    print('')
    lst_2.append(models[m][0])
    lst_2.append((accuracy_score(y_test, y_pred))*100)
    lst_2.append(accuracies.mean()*100)
    lst_2.append(accuracies.std()*100)
    lst_2.append(roc)
    lst_2.append(precision)
    lst_2.append(recall)
    lst_2.append(f1)
```

รูปที่ ข.1 ตัวอย่างการสอน และทดสอบโมเดลเบื้องต้นของการทดลองชุดที่ 1

SVM :
[[2671 1]
[94 36546]]
Accuracy Score: 0.997583435083435

K-Fold Validation Mean Accuracy: 99.86 %

Standard Deviation: 0.03 %

ROC AUC Score: 1.00

Precision: 1.00

Recall: 1.00

F1: 1.00

Decision Tree :
[[2663 9]
[10 36630]]
Accuracy Score: 0.999516687016687

K-Fold Validation Mean Accuracy: 99.98 %

Standard Deviation: 0.01 %

ROC AUC Score: 1.00

Precision: 1.00

Recall: 1.00

F1: 1.00

รูปที่ ข.2 ผลลัพธ์การทดสอบโมเดลของการทดลองชุดที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
from sklearn.model_selection import GridSearchCV

[38] grid_models = [(LogisticRegression(),[{'C':[7,8,9,10],'random_state':[0]}]),
                    (KNeighborsClassifier(),[{'n_neighbors':[3,5,7,9], 'metric': ['euclidean','manhattan', 'chebyshev', 'minkowski']}]),
                    (SVC(),[{'C':[7,8,9,10],'kernel':['linear', 'rbf'],'random_state':[0]}]),
                    (GaussianNB(),[{'var_smoothing': [1e-09]}]),
                    (DecisionTreeClassifier(),[{'criterion':['gini','entropy'],'random_state':[0]}]),
                    (RandomForestClassifier(),[{'n_estimators':[100,150,200,300],'criterion':['gini','entropy'],'random_state':[0]}]),
                   (XGBClassifier(), [{'learning_rate': [0.01, 0.05, 0.1], 'eval_metric': ['error']}])]

for i,j in grid_models:
    grid = GridSearchCV(estimator=i,param_grid = j, scoring = 'accuracy',cv = 10)
    grid.fit(X_train_res, y_train_res)
    best_accuracy = grid.best_score_
    best_param = grid.best_params_
    print('{}:\nBest Accuracy : {:.2f}%'.format(i,best_accuracy*100))
    print('Best Parameters : ',best_param)
    print('')
    print('----------------')
    print('')
```

รูปที่ ข.3 ปรับแต่งค่าพารามิเตอร์ของแต่ละโมเดลในการทดลองชุดที่ 1

```
SVC():
Best Accuracy : 99.97%
Best Parameters :  {'C': 10, 'kernel': 'rbf', 'random_state': 0}


DecisionTreeClassifier():
Best Accuracy : 99.98%
Best Parameters :  {'criterion': 'entropy', 'random_state': 0}
```

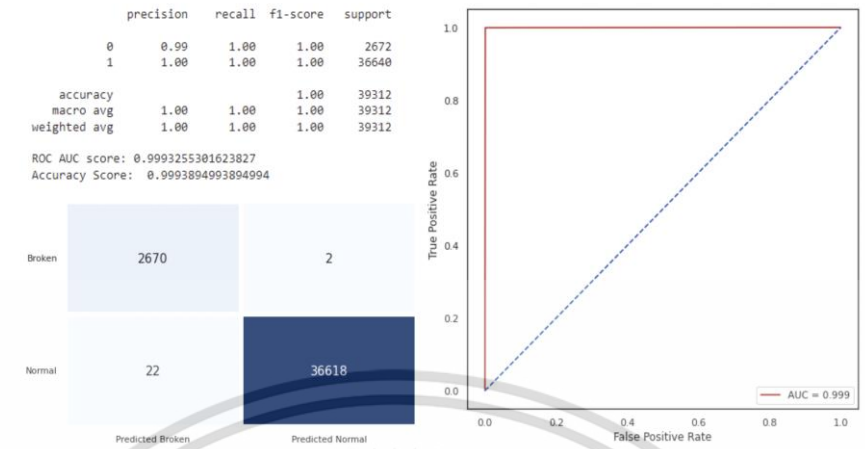
รูปที่ ข.4 ค่าพารามิเตอร์ที่ดีที่สุดของแต่ละโมเดลในการทดลองชุดที่ 1

Turning ANN	Batch size = 16	Batch size = 32	Batch size = 64
Epochs = 50	Acc = 99.42 Loss = 0.024	Acc = 99.69 Loss = 0.017	Acc = 99.42 Loss = 0.024
Epochs = 100	Acc = 99.56 Loss = 0.019	Acc = 98.56 Loss = 0.064	Acc = 99.30 Loss = 0.029

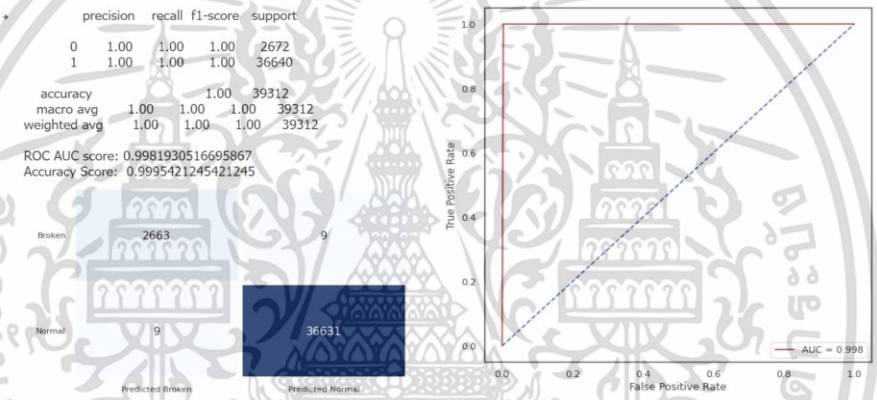
```
[134] ann = tf.keras.models.Sequential()
      ann.add(tf.keras.layers.Dense(units= 32, kernel_regularizer=l2(0.01), bias_regularizer=l2(0.01), activation='relu'))
      tf.keras.layers.Dropout(0.25)
      ann.add(tf.keras.layers.Dense(units= 32, kernel_regularizer=l2(0.01), bias_regularizer=l2(0.01), activation='relu'))
      tf.keras.layers.Dropout(0.5)
      ann.add(tf.keras.layers.Dense(units= 1, activation='sigmoid'))
      ann.compile(optimizer= 'adam', loss= 'binary_crossentropy', metrics= ['accuracy'])

[135] ann_history = ann.fit(X_train_res, y_train_res, batch_size= 32, epochs= 50, validation_split= 0.2)

      Epoch 1/50
      4272/4272 [==============================] - 7s 2ms/step - loss: 0.1168 - accuracy: 0.9856 - val_loss: 0.0809 - val_accuracy: 0.9939
      Epoch 2/50
      4272/4272 [==============================] - 6s 1ms/step - loss: 0.0550 - accuracy: 0.9921 - val_loss: 0.0330 - val_accuracy: 0.9996
      Epoch 3/50
      4272/4272 [==============================] - 7s 2ms/step - loss: 0.0460 - accuracy: 0.9931 - val_loss: 0.0367 - val_accuracy: 0.9987
      Epoch 4/50
      4272/4272 [==============================] - 6s 1ms/step - loss: 0.0414 - accuracy: 0.9936 - val_loss: 0.0393 - val_accuracy: 0.9985
      Epoch 5/50
      4272/4272 [==============================] - 6s 1ms/step - loss: 0.0387 - accuracy: 0.9940 - val_loss: 0.0313 - val_accuracy: 0.9989
      Epoch 6/50
      4272/4272 [==============================] - 7s 2ms/step - loss: 0.0371 - accuracy: 0.9942 - val_loss: 0.0272 - val_accuracy: 0.9994
      Epoch 7/50
      4272/4272 [==============================] - 6s 1ms/step - loss: 0.0353 - accuracy: 0.9944 - val_loss: 0.0338 - val_accuracy: 0.9985
      Epoch 8/50
      4272/4272 [==============================] - 7s 2ms/step - loss: 0.0330 - accuracy: 0.9948 - val_loss: 0.0302 - val_accuracy: 0.9988
      Epoch 9/50
      4272/4272 [==============================] - 7s 2ms/step - loss: 0.0309 - accuracy: 0.9952 - val_loss: 0.0416 - val_accuracy: 0.9972
```

รูปที่ ข.5 ค่าพารามิเตอร์ที่ดีที่สุดของ Neural Network ในการทดลองชุดที่ 1

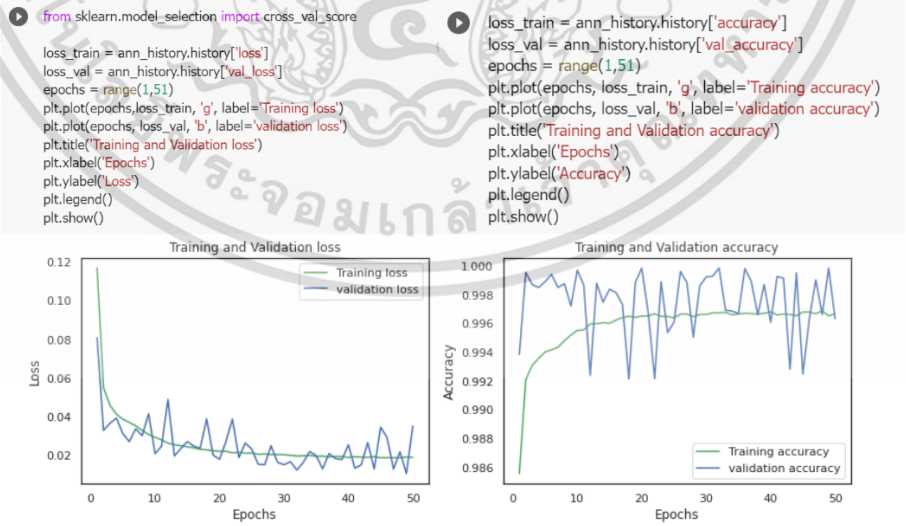
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.6 ผลการทดลองโมเดล Support Vector Machine ของการทดลองชุดที่ 1

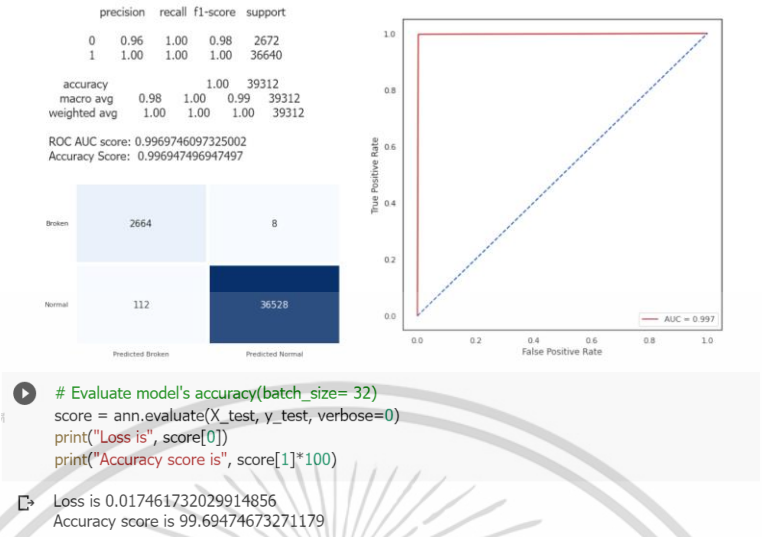


รูปที่ ข.7 ผลการทดลองโมเดล Decision Tree ของการทดลองชุดที่ 1



รูปที่ ข.8 ผลการทดลองโมเดล Neural Network ของการทดลองชุดที่ 1 (1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.9 ผลการทดลองโมเดล Neural Network ของการทดลองชุดที่ 1 (2)

ข.2 ชุดข้อมูลการทดลองชุดที่ 2

จากการศึกษาภาพรวมของข้อมูลที่ทำความสะอาดเรียบร้อยแล้วจากชุดข้อมูลการสอน และชุดข้อมูลการทดสอบ นำไปสู่การสร้างโมเดล ดังรูปที่ ข.10 โดยทำการสร้างโมเดลทั้งหมด 7 โมเดล และเลือกมา 3 โมเดล คือ Support Vector Machine, Decision Tree และ Neural Network เพื่อเปรียบเทียบประสิทธิภาพวิธีการจำแนกกลุ่มข้อมูล โดยมีการกำหนดวิธีการวิเคราะห์ความแม่นตรงของตัวแบบ K=10

ผลการทดสอบก่อนปรับค่าพารามิเตอร์พบว่า โมเดล Decision Tree มีค่าความถูกต้อง 99.98% และ Support Vector Machine มีค่าความถูกต้องที่ 99.85% ดังรูปที่ ข.11 จากนั้นทำการปรับแต่งค่าพารามิเตอร์ โดยใช้ GridSarchCV ดังรูปที่ ข.12 พบว่าค่าพารามิเตอร์ที่ดีที่สุดของโมเดล Decision Tree มีเกณฑ์การแบ่งข้อมูลแบบ ‘Entropy’, Support Vector Machine ใช้วิธีเรเดียลเบสิสฟังก์ชัน (Radial Basis Function : RBF) เป็นฟังก์ชันเคอร์เนลยอดนิยมที่ใช้ในอัลกอริทึมการเรียนรู้เคอร์เนลต่างๆ โดยกำหนดค่า ‘c =10’ และ โมเดล Neural Network กำหนดขนาด Batch size =32, Epochs = 50 รายละเอียดดังรูปที่ ข.13 และรูปที่ ข.14

หลังจากปรับแต่งค่าพารามิเตอร์ให้เหมาะสมกับแต่ละโมเดลเรียบร้อยแล้ว จากนั้นทำการสอนและทดสอบแต่ละโมเดลอีกรอบ ได้ผลลัพธ์ดังรูปที่ ข.15 ถึงรูปที่ ข.18 พบว่าโมเดล Decision Tree ให้ค่าความถูกต้องสูงที่สุด และค่าความคลาดเคลื่อนต่ำที่สุด ตามด้วย Support Vector Machine และ Neural Network ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
models = []
models.append(['Logistic Regreesion', LogisticRegression(random_state=0)])
models.append(['SVM', SVC(random_state=0)])
models.append(['KNeighbors', KNeighborsClassifier()])
models.append(['GaussianNB', GaussianNB()])
models.append(['Decision Tree', DecisionTreeClassifier(random_state=0)])
models.append(['Random Forest', RandomForestClassifier(random_state=0)])
models.append(['XGBoost', XGBClassifier(eval_metric= 'error')])

lst_1= []

for m in range(len(models)):
    lst_2= []
    model = models[m][1]
    model.fit(X_train_res, y_train_res)
    y_pred = model.predict(X_test)
    cm = confusion_matrix(y_test, y_pred)  #Confusion Matrix
    accuracies = cross_val_score(estimator = model, X = X_train_res, y = y_train_res, cv = 10)   #K-Fold Validation
    roc = roc_auc_score(y_test, y_pred)  #ROC AUC Score
    precision = precision_score(y_test, y_pred)  #Precision Score
    recall = recall_score(y_test, y_pred)  #Recall Score
    f1 = f1_score(y_test, y_pred)  #F1 Score
    print(models[m][0],':')
    print(cm)
    print('Accuracy Score: ',accuracy_score(y_test, y_pred))
    print('')
    print("K-Fold Validation Mean Accuracy: {:.2f} %".format(accuracies.mean()*100))
    print('')
    print("Standard Deviation: {:.2f} %".format(accuracies.std()*100))
    print('')
    print('ROC AUC Score: {:.2f}'.format(roc))
    print('')
    print('Precision: {:.2f}'.format(precision))
    print('')
    print('Recall: {:.2f}'.format(recall))
    print('')
    print('F1: {:.2f}'.format(f1))
    print('----------------------------------')
    print('')
    lst_2.append(models[m][0])
    lst_2.append((accuracy_score(y_test, y_pred))*100)
    lst_2.append(accuracies.mean()*100)
    lst_2.append(accuracies.std()*100)
    lst_2.append(roc)
    lst_2.append(precision)
    lst_2.append(recall)
    lst_2.append(f1)
```

รูปที่ ข.10 ตัวอย่างการสอน และทดสอบโมเดลเบื้องต้นของการทดลองชุดที่ 2

SVM :
[[1675 7]
[23 25079]]
Accuracy Score: 0.9988799283154122

K-Fold Validation Mean Accuracy: 99.86 %

Standard Deviation: 0.03 %

ROC AUC Score: 1.00

Precision: 1.00

Recall: 1.00

F1: 1.00

Decision Tree :
[[1677 5]
[9 25093]]
Accuracy Score: 0.9994772998805257

K-Fold Validation Mean Accuracy: 99.98 %

Standard Deviation: 0.01 %

ROC AUC Score: 1.00

Precision: 1.00

Recall: 1.00

F1: 1.00

รูปที่ ข.11 ผลลัพธ์การทดสอบโมเดลของการทดลองชุดที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
from sklearn.model_selection import GridSearchCV

grid_models = [(LogisticRegression(),[{'C':[7,8,9,10],'random_state':[0]}]),
               (KNeighborsClassifier(),[{'n_neighbors':[3,5,7,9], 'metric': ['euclidean','manhattan', 'chebyshev', 'minkowski']}]),
               (SVC(),[{'C':[7,8,9,10],'kernel':['linear', 'rbf'],'random_state':[0]}]),
               (GaussianNB(),[{'var_smoothing': [1e-09]}]),
               (DecisionTreeClassifier(),[{'criterion':['gini','entropy'],'random_state':[0]}]),
               (RandomForestClassifier(),[{'n_estimators':[100,150,200,300],'criterion':['gini','entropy'],'random_state':[0]}]),
              (XGBClassifier(), [{'learning_rate': [0.01, 0.05, 0.1], 'eval_metric': ['error']}])]

for i,j in grid_models:
    grid = GridSearchCV(estimator=i,param_grid = j, scoring = 'accuracy',cv = 10)
    grid.fit(X_train_res, y_train_res)
    best_accuracy = grid.best_score_
    best_param = grid.best_params_
    print('{}:\nBest Accuracy : {:.2f}%'.format(i,best_accuracy*100))
    print('Best Parameters : ',best_param)
    print('')
    print('----------------')
    print('')
```

รูปที่ ข.12 ปรับแต่งค่าพารามิเตอร์ของแต่ละโมเดลในการทดลองชุดที่ 2

```
DecisionTreeClassifier():
Best Accuracy : 99.98%
Best Parameters :  {'criterion': 'entropy', 'random_state': 0}


SVC():
Best Accuracy : 99.97%
Best Parameters :  {'C': 10, 'kernel': 'rbf', 'random_state': 0}
```

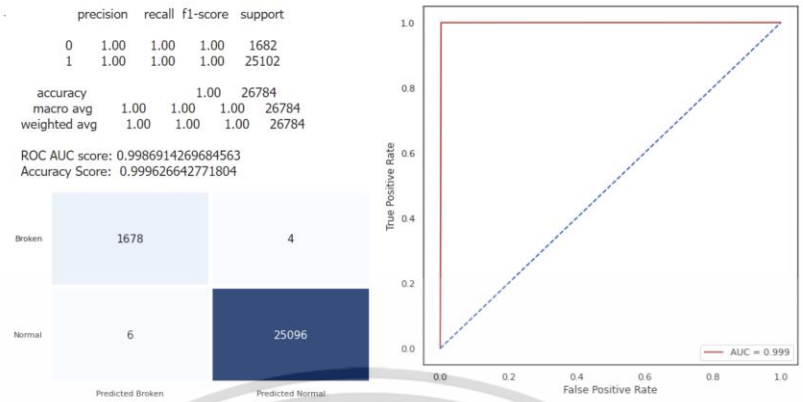
รูปที่ ข.13 ค่าพารามิเตอร์ที่ดีที่สุดของแต่ละโมเดลในการทดลองชุดที่ 2

Turning ANN	Batch size = 16	Batch size = 32	Batch size = 64
Epochs = 50	Acc = 99.68 Loss = 0.017	Acc = 99.91 Loss = 0.014	Acc = 99.61 Loss = 0.022
Epochs = 100	Acc = 99.88 Loss = 0.014	Acc = 99.79 Loss = 0.014	Acc = 99.37 Loss = 0.029

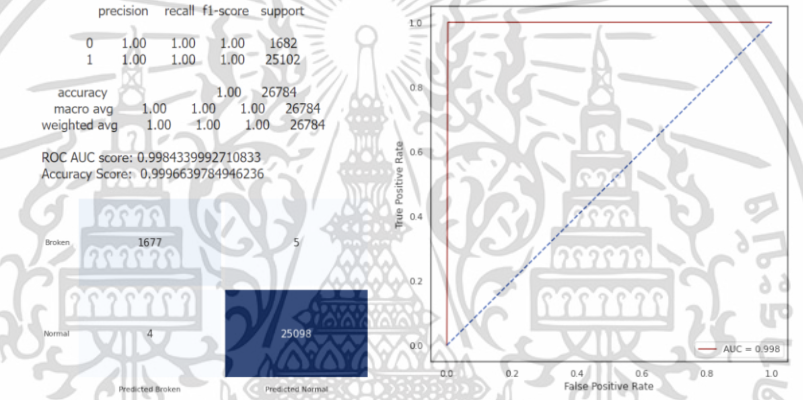
```
ann = tf.keras.models.Sequential()
ann.add(tf.keras.layers.Dense(units= 32, kernel_regularizer=l2(0.01), bias_regularizer=l2(0.01), activation='relu')),
tf.keras.layers.Dropout(0.25)
ann.add(tf.keras.layers.Dense(units= 32, kernel_regularizer=l2(0.01), bias_regularizer=l2(0.01), activation='relu')),
tf.keras.layers.Dropout(0.5)
ann.add(tf.keras.layers.Dense(units= 1, activation='sigmoid'))
ann.compile(optimizer= 'adam', loss= 'binary_crossentropy', metrics= ['accuracy'])

ann_history = ann.fit(X_train_res, y_train_res, batch_size= 32, epochs= 50, validation_split= 0.2)

Epoch 1/50
2934/2934 [==============================] - 8s 2ms/step - loss: 0.1473 - accuracy: 0.9865 - val_loss: 0.0717 - val_accuracy: 0.9879
Epoch 2/50
2934/2934 [==============================] - 6s 2ms/step - loss: 0.0649 - accuracy: 0.9917 - val_loss: 0.0644 - val_accuracy: 0.9881
Epoch 3/50
2934/2934 [==============================] - 7s 2ms/step - loss: 0.0525 - accuracy: 0.9926 - val_loss: 0.0560 - val_accuracy: 0.9893
Epoch 4/50
2934/2934 [==============================] - 6s 2ms/step - loss: 0.0457 - accuracy: 0.9930 - val_loss: 0.0553 - val_accuracy: 0.9882
Epoch 5/50
2934/2934 [==============================] - 6s 2ms/step - loss: 0.0412 - accuracy: 0.9935 - val_loss: 0.0468 - val_accuracy: 0.9895
Epoch 6/50
2934/2934 [==============================] - 6s 2ms/step - loss: 0.0382 - accuracy: 0.9938 - val_loss: 0.0718 - val_accuracy: 0.9867
```

รูปที่ ข.14 ค่าพารามิเตอร์ที่ดีที่สุดของ Neural network ในการทดลองชุดที่ 2

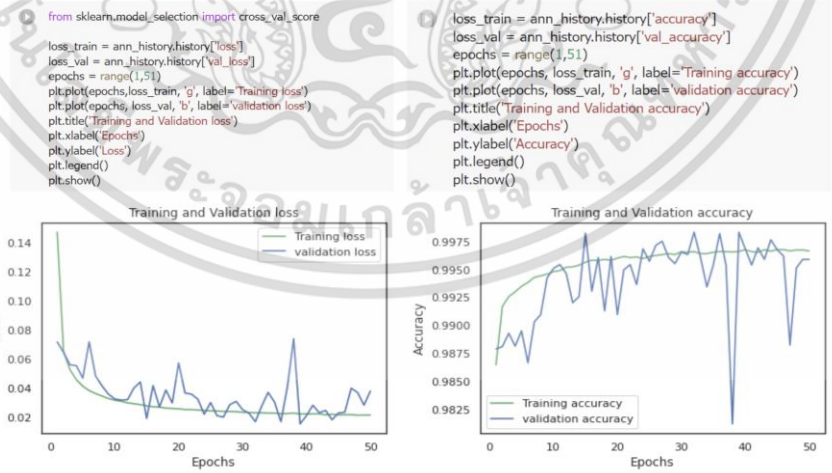
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.15 ผลการทดลองโมเดล Support Vector Machine ของการทดลองชุดที่ 2

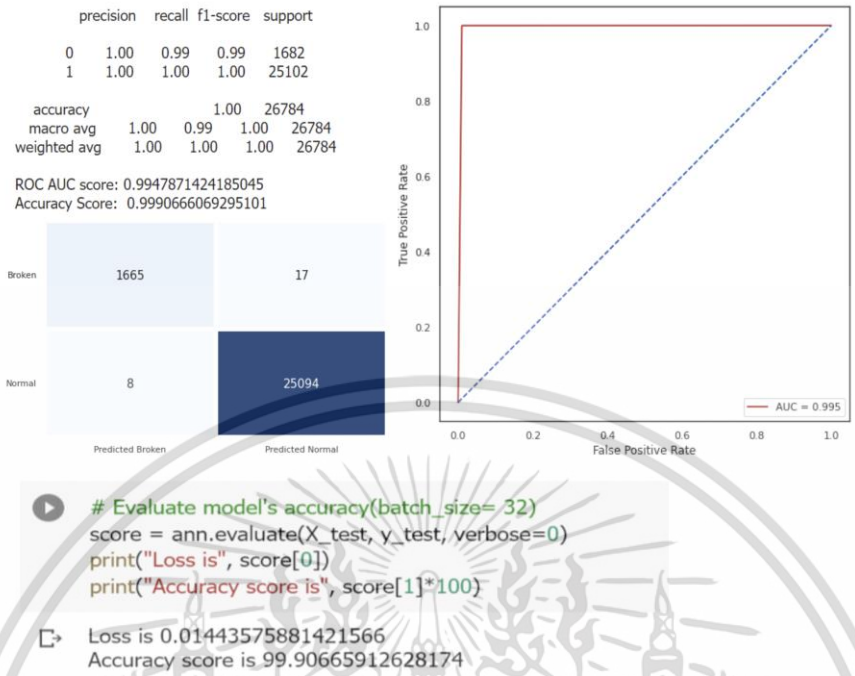


รูปที่ ข.16 ผลการทดลองโมเดล Decision Tree ของการทดลองชุดที่ 2



รูปที่ ข.17 ผลการทดลองโมเดล Neural network ของการทดลองชุดที่ 2 (1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.18 ผลการทดลองโมเดล Neural network ของการทดลองชุดที่ 2 (2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อ	นางสาวสุชาภา ธรรมเอกสกุล
วัน เดือน ปีเกิด	21 ตุลาคม พ.ศ. 2537
ที่อยู่ปัจจุบัน	88/41 ซ.กาญจนาภิเษก 46 ถ.กาญจนาภิเษก แขวงดอกไม้ เขตประเวศ กทม. 10250
ประวัติการศึกษา	ปีการศึกษา 2558 : ปริญญาตรี วิศวกรรมศาสตรบัณฑิต สาขา วิศวกรรมอัตโนมัติ เกรดเฉลี่ย 2.68 สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ทุนการศึกษา	-
ผลงานทางวิชาการ	-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้