

## การพยากรณ์ปริมาณความต้องการใช้ไฟฟ้าบนข้อมูลมิเตอร์อัจฉริยะด้วยการเรียนรู้ของเครื่องจักร

ณัฐนันท์ ยอดมา<sup>\*1</sup>, กนิษฐา อาศัวยราช<sup>\*2</sup>, ปณณธร รอดกำเนิด<sup>\*3</sup>, มกรา เกียรติเมธา<sup>\*4</sup>

<sup>\*</sup>ชมรมวิศวกร กฟผ.๑ (เขต.กฟผ.๑) การไฟฟ้าส่วนภูมิภาค เขต ๑ (ภาคกลาง) จ.พระนครศรีอยุธยา

<sup>1</sup>nattanon.yodm@pea.co.th, <sup>2</sup>kanittha.asa@pea.co.th,

<sup>3</sup>pannathon.r@outlook.com, <sup>4</sup>makara.kia@pea.co.th

### บทคัดย่อ

บทความนี้นำเสนอการพยากรณ์ความต้องการใช้ไฟฟ้าบนชุดข้อมูลจากระบบมิเตอร์อัจฉริยะ (Advanced Metering Infrastructure, AMI) ด้วยการเรียนรู้ของเครื่องจักร (Machine Learning, ML) และการเรียนรู้เชิงลึก (Deep Learning, DL) เพื่อค้นหาระเบียบวิธี (Algorithm) ที่มีความเหมาะสมกับชุดข้อมูลดังกล่าว จากผลการศึกษาพบว่า การพยากรณ์ด้วยชุดข้อมูลแบบอนุกรมเวลาตัวแปรต้นเดียว-ตัวแปรตามหนึ่งขั้น (One-Step Univariate Forecasting) นั้น ระเบียบวิธีการเรียนรู้แบบ Gradient Boosting (GB) มีความเหมาะสมมากที่สุด เมื่อเปรียบเทียบกับระเบียบวิธีอื่นที่นำมาศึกษา โดยมีค่าเฉลี่ยข้อผิดพลาดกำลังสอง (Mean Squared Error, MSE) เท่ากับ 0.0907 ต่อผู้ใช้ไฟฟ้าหนึ่งราย และใช้เวลาในการประมวลผลเท่ากับ 0.4499 วินาที ต่อผู้ใช้ไฟฟ้าหนึ่งราย ซึ่งน้อยกว่าระเบียบวิธีอื่นอย่างมีนัยสำคัญ นอกเหนือจากนั้น บทความนี้ยังได้นำเสนอ ผลของการนำแบบจำลองที่ถูกฝึกฝนแล้ว ไปประยุกต์ใช้ในสถานการณ์จริง พบว่า แบบจำลองดังกล่าวสามารถคาดการณ์พฤติกรรมความต้องการใช้ไฟฟ้าในอนาคตได้ประมาณ 15 วัน และมีประสิทธิภาพสูงสุดในการพยากรณ์แบบ 1 วัน ล่วงหน้า (Day Ahead)

**คำสำคัญ:** Machine Learning, Time-Series Forecasting, Advanced Metering Infrastructure

### 1. บทนำ

การไฟฟ้าส่วนภูมิภาค (กฟผ.) เป็นองค์กรที่บริการภาคจำหน่ายไฟฟ้าแก่ผู้ใช้ไฟฟ้า ที่ใช้ไฟฟ้าตั้งอยู่ในส่วนภูมิภาคครอบคลุมพื้นที่ 74 จังหวัด ดังนั้น การพัฒนาระบบโครงข่ายไฟฟ้าให้มีความมั่นคง มีประสิทธิภาพ และต้องประยุกต์ใช้เทคโนโลยีที่ทันสมัยอยู่เสมอ ถือเป็นพันธกิจที่สำคัญอย่างหนึ่งในอดีตการประเมินปริมาณความต้องการใช้ไฟฟ้า (Demand) ทำได้โดยใช้ข้อมูลการอ่านหน่วยการใช้พลังงานไฟฟ้า

(Electricity Unit) ผ่านเครื่องวัดพลังงานไฟฟ้าแบบจานหมุน (Kilowatt-Hour Meter) ที่ไม่สามารถสื่อสารได้

จนปัจจุบันเป็นระบบมิเตอร์อัจฉริยะ (AMI) ที่สามารถสื่อสารได้แบบสองทางระหว่างศูนย์สั่งการและเครื่องวัดที่ถูกติดตั้งหน้างาน โดยมีเตอร์ระบบ AMI นั้น มีความสำคัญยิ่งต่อระบบโครงข่ายอัจฉริยะ (Smart Grid, SG) ซึ่งผู้ให้บริการพลังงานไฟฟ้า เช่น กฟผ. สามารถนำข้อมูลมาใช้งานได้แบบทันทีทันใด (Real-Time) ไม่ว่าจะเป็น งานด้านการวิเคราะห์ การพยากรณ์ และบริหารจัดการทรัพยากร เป็นต้น

การพยากรณ์ ความต้องการใช้ไฟฟ้า (Demand Forecasting) เป็นการคาดคะเนการเจริญเติบโตหรือปริมาณความต้องการที่จะเกิดขึ้นในอนาคต ซึ่งมีความสำคัญต่อการวางแผนการปรับปรุงและบำรุงรักษาระบบไฟฟ้า หรือวิเคราะห์พฤติกรรมของผู้ใช้ไฟฟ้า โดยสิ่งที่ใช้เป็นข้อมูลขาเข้า (Input) ได้รับมาจากมิเตอร์ระบบ AMI ถูกป้อนเข้าสู่ระบบการเรียนรู้ของเครื่องจักร (ML) และหรือการเรียนรู้เชิงลึก (DL) เพื่อให้ได้ผลข้อมูลเป้าหมาย (Target) เป็นค่าที่ระบบสามารถพยากรณ์ได้ ซึ่งระบบ ML และ DL จำเป็นต้องมีการพัฒนาปรับปรุงให้มีประสิทธิภาพสูงขึ้น เพื่อให้มีความคลาดเคลื่อน (Error) น้อยที่สุดเท่าที่จะสามารถทำได้

บทความนี้จึงมีวัตถุประสงค์เพื่อพัฒนาแบบจำลอง (Model) ทั้งในส่วนของ ML และ DL เพื่อคัดเลือกระเบียบวิธีที่เหมาะสมกับชุดข้อมูลจากมิเตอร์ระบบ AMI มากที่สุด โดยมีเนื้อหาและรายละเอียดต่าง ๆ ดังต่อไปนี้

### 2. ทฤษฎีที่เกี่ยวข้อง

ในส่วนนี้จะขออธิบายถึงข้อมูลโดยทั่วไปของระเบียบวิธีของ ML และ DL ที่บทความนี้เลือกนำมาใช้ในการศึกษารวมถึงการแปลงข้อมูลให้เหมาะสมต่อการถูกวิเคราะห์โดยมีเนื้อหา ดังนี้

#### 2.1 การวิเคราะห์การถดถอยเชิงเส้นแบบ Elastic Net

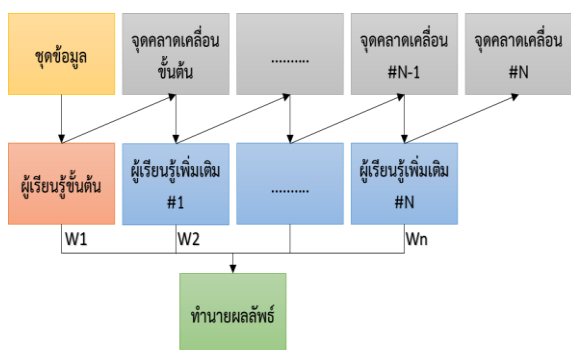
การวิเคราะห์การถดถอยเชิงเส้นแบบ Elastic Net (EN) เป็นหนึ่งในระเบียบวิธีมาตรฐานที่มักถูกนำมาใช้ในการ

วิเคราะห์ข้อมูลที่มีข้อมูลขาเข้าสัมพันธ์กับข้อมูลเป้าหมายเป็นลักษณะเชิงเส้น (Linear) โดยมีข้อดีคือ ผู้ใช้งานสามารถปรับตั้งค่าพารามิเตอร์  $\alpha$  ได้ ซึ่งค่านี้จะเป็นตัวบ่งบอกว่าต้องการให้ความสำคัญต่อการลดทอนน้ำหนักแบบใดระหว่าง L1 Penalty (Lasso) และ L2 Penalty (Ridge) ดังแสดงในสมการที่ 1 เพื่อให้มีความเหมาะสมกับข้อมูลขาเข้าที่นำมาวิเคราะห์ และยังเป็นการทำให้แบบจำลอง มีความเป็นมาตรฐานมากขึ้น (Regularization) เพื่อเตรียมให้แบบจำลองสามารถถูกนำไปใช้กับชุดข้อมูลในอนาคตได้ [1]

$$EN_{penalty} = (\alpha \times L1_{penalty}) + ((1 - \alpha) \times L2_{penalty}) \quad (1)$$

## 2.2 การเรียนรู้แบบ Gradient Boosting

การเรียนรู้แบบ Gradient Boosting (GB) เป็นการเรียนรู้แบบลำดับขั้น ที่ถูกพัฒนาต่อจากระเบียบวิธีแบบ AdaBoost บนแนวคิดการเรียนรู้จากความคลาดเคลื่อนของแบบจำลองก่อนหน้า ประกอบด้วย 3 องค์ประกอบหลัก ได้แก่ 1) ฟังก์ชันการสูญเสีย (Loss Function) ซึ่งปกติแล้วจะใช้เป็นค่าเฉลี่ยข้อผิดพลาดกำลังสอง (Mean Squared Error, MSE) สำหรับการวิเคราะห์เชิงถดถอย (Regression), 2) ผู้เรียนรู้ขั้นต้น (Weak Learner) เป็นระเบียบวิธีตั้งต้นที่จะถูกนำไปใช้เพื่อปรับปรุงต่อ ซึ่งปกติแล้วจะนิยมใช้เป็นระเบียบวิธีการตัดสินใจแบบต้นไม้ (Decision Tree) และ 3) ผู้เรียนรู้เพิ่มเติม (Additive Learners) เป็นแบบจำลองการเรียนรู้ที่จะถูกเพิ่มเข้ามาเพื่อปรับปรุงข้อผิดพลาดของผู้เรียนรู้ก่อนหน้า โดยระเบียบวิธีการเคลื่อนลงตามความชัน (Gradient Descent) จะถูกนำมาใช้ในการปรับปรุงค่าความคลาดเคลื่อนนี้ รูปที่ 1 แสดงผังการไหลของกระบวนการเรียนรู้แบบ GB [2]



รูปที่ 1 ผังการไหลของการเรียนรู้แบบ Gradient Boosting

## 2.3 การเรียนรู้เชิงลึกด้วยสถาปัตยกรรมแบบ CLDNN

การเรียนรู้เชิงลึกถูกพัฒนาด้วยแนวคิดการลอกเลียนแบบการทำงานของระบบประสาทในสมองของมนุษย์ โดยการสร้างโครงข่ายประสาทเทียม (Neuron Networks, NN) ทำให้มีความสามารถในการแก้ไขปัญหาที่ซับซ้อนขึ้นได้อย่างมีประสิทธิภาพ แต่ต้องใช้ทรัพยากรในการคำนวณที่มากขึ้นเช่นเดียวกัน ปัจจุบันสถาปัตยกรรม (Architecture) ของ NN นั้นมีหลากหลายรูปแบบ

และสามารถนำมาผสมผสานกันได้ เพื่อดึงเอาข้อดีของแต่ละแบบมาบูรณาการ (Integration) ร่วมกัน

โดยโครงสร้างแบบ Convolutional, Long Short-Term Memory, Fully Connected Deep Neural Networks (CLDNN) ที่นำเสนอ [3] เป็นการผสมผสานข้อดีของแต่ละระเบียบวิธี ได้แก่ 1) Convolutional Neural Networks (CNN) ที่มีความสามารถในการลดความแปรปรวนเชิงความถี่ของข้อมูล (Frequency Variations), 2) Long Short-Term Memory (LSTM) ที่มีความสามารถในการวิเคราะห์ข้อมูลเชิงเวลา ที่มีความต่อเนื่องกันในตัวเอง (Temporal Data) และ 3) Deep Neuron Networks (DNN) ซึ่งมีความสามารถในการวิเคราะห์ข้อมูลที่มีความซับซ้อน จากผลการศึกษาพบว่า CLDNN สามารถเพิ่มประสิทธิภาพของการพยากรณ์ได้ประมาณ 4 – 6 % เมื่อเปรียบเทียบกับการใช้แบบจำลองโครงสร้างใดโครงสร้างหนึ่งเพียงอย่างเดียว รูปที่ 2 แสดงผังการไหล เบื้องต้นของ CLDNN [3]



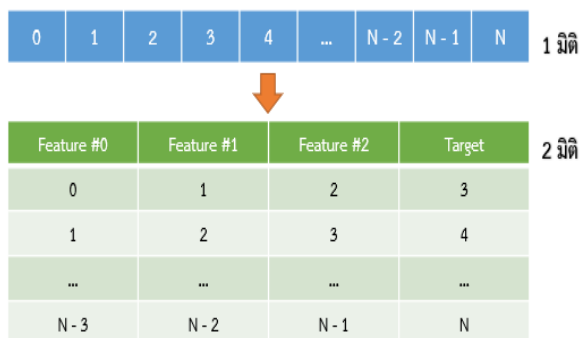
รูปที่ 2 ผังการไหลเบื้องต้นของการเรียนรู้แบบ CLDNN

## 2.4 การแปลงข้อมูลให้เหมาะสมต่อการถูกวิเคราะห์

การที่จะฝึกฝนแบบจำลองทั้ง ML และ DL ได้นั้น จำเป็นต้องทำการแปลงข้อมูล (Transformation) ที่เป็นลำดับต่อเนื่อง (Sequence) แบบ 1 มิติ ให้อยู่ในรูปแบบ (Multiple Input/Output Patterns) แบบ 2 มิติ [4] หรือทำให้มีทั้งข้อมูลขาเข้า (Input) และข้อมูลเป้าหมาย (Output/Target) โดยสามารถแปลงข้อมูลเบื้องต้น 3 รูปแบบ ได้ดังนี้ 1) One-Step Univariate Dataset หรือ ชุดข้อมูลอนุกรมเวลาแบบตัวแปรต้นเดียว-ตัวแปรตามหนึ่งขั้น 2) Multi-Step Univariate Dataset ชุดข้อมูลอนุกรมเวลาแบบตัวแปรต้นเดียว-ตัวแปรตามหลายขั้น และ 3) Multivariate Dataset ชุดข้อมูลอนุกรมเวลาแบบหลายตัวแปร [5] ซึ่งในบทความนี้ เลือกใช้ รูปแบบที่ 1 สำหรับขั้นตอนการแปลงข้อมูล

รายละเอียดการแปลงข้อมูลรูปแบบที่ 1 สามารถถูกอธิบายได้ดังต่อไปนี้ ชุดข้อมูลที่มีตัวแปรเพียงหนึ่งประเภทและตัวแปรนั้นจำเป็นต้องขึ้นอยู่กับความต่อเนื่องเชิงเวลา ตัวอย่างเช่น ข้อมูลที่นำมาศึกษาในบทความนี้คือ ค่าความต้องการพลังงานไฟฟ้าเป็นหน่วยของกิโลวัตต์ (kW) ของผู้ใช้ไฟฟ้าที่ถูกบันทึกทุก ๆ 15 นาที เป็นระยะเวลา 1 ปี โดยชุดข้อมูลประเภทนี้ มีความแปรปรวน (Variance) ตลอดช่วงเวลาที่เราสนใจ ดังนั้นจำนวนหลักของข้อมูลขาเข้า (Columns, Features) จึงบ่งบอกถึงช่วงเวลาย้อนหลังที่ต้องการให้แบบจำลองทำการเรียนรู้พฤติกรรมของข้อมูล (Previous Time Steps) เพื่อพยากรณ์ค่าเป้าหมาย ณ เวลาใดเวลาหนึ่ง เช่น หากเป็นข้อมูลที่ถูกรับบันทึกทุก ๆ 15 นาที และมีการ

กำหนดจำนวนหลักของข้อมูลขาเข้าเป็น 96 หลัก หมายความว่า มีความต้องการให้แบบจำลองเรียนรู้ข้อมูลย้อนหลัง 1 วัน ( $96 \times 15 = 1,440$  นาที = 1 วัน) เพื่อทำการพยากรณ์ข้อมูลเป้าหมายในช่วงเวลาถัดไปอีก 15 นาที เป็นต้น รูปที่ 3 แสดงตัวอย่างการแปลงข้อมูลจาก 1 มิติ เป็น 2 มิติ โดยมีจำนวนหลักของข้อมูลขาเข้า 3 หลัก และมีค่าเป้าหมาย 1 หลัก



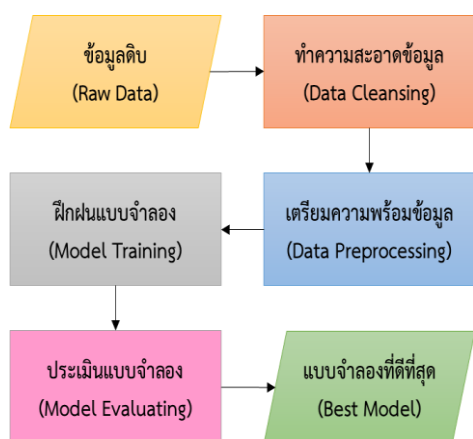
รูปที่ 3 การแปลงข้อมูลจาก 1 มิติ เป็น 2 มิติ

### 3. วิธีการและขั้นตอนการศึกษา

ในบทความนี้ได้พัฒนาระบบ ML และ DL ผ่านการเขียนโปรแกรมคอมพิวเตอร์ด้วยภาษาไพทอน (Python) [6] เพื่อใช้ในการพยากรณ์ปริมาณความต้องการใช้ไฟฟ้าของผู้ใช้ไฟฟ้าที่ได้รับการติดตั้งมิเตอร์ระบบ AMI ในพื้นที่เมืองพัทยา โดยมีการนำคลังโปรแกรม (Library) ตามตารางที่ 1 มาประยุกต์ใช้งาน ซึ่งผู้พัฒนาอนุญาตให้นำมาใช้ได้อย่างเสรี (Open Source) ทำให้ลดขั้นตอนการเขียนชุดคำสั่งใหม่ทั้งหมด รูปที่ 4 แสดงผังการไหลของกระบวนการศึกษาของบทความนี้

ตารางที่ 1 คลังโปรแกรมที่นำมาใช้ในการศึกษา

ลำดับ	Library	ความสามารถ
1	Pandas	จัดการกับชุดข้อมูล [7]
2	Numpy	เตรียมข้อมูลก่อนนำเข้าแบบจำลอง [8]
3	Matplotlib	แสดงผลในรูปแบบของกราฟ [9]
4	Scikit-learn	สร้างแบบจำลอง EN [10]
5	LightGBM	สร้างแบบจำลอง GB [11]
6	TensorFlow	สร้างแบบจำลอง CLDNN [12]



รูปที่ 4 ผังการไหลของกระบวนการศึกษา

### 3.1 การคัดเลือกข้อมูล (Data Selection)

ข้อมูลดิบ (Raw Data) ที่ใช้ในการศึกษาค้างนี้ เป็นชุดข้อมูลที่ได้จากมิเตอร์ระบบ AMI ที่ถูกติดตั้งนาร่องในเขตพื้นที่เมืองพัทยา โดยใช้ข้อมูลของปี 2563 และคัดเลือกผู้ใช้ไฟฟ้าในเขตพื้นที่ธุรกิจท้องถิ่น (Local Business Area, LBA) จำนวนทั้งสิ้น 53 ราย ภายใต้หม้อแปลงไฟฟ้าจำหน่าย ขนาด 250 kVA เครื่องเดียวกัน โดยมีข้อมูลขาเข้า 3 ลักษณะ (Feature) ได้แก่ 1) หมายเลขผู้ใช้ไฟฟ้าที่ผ่านการเข้ารหัสแล้ว (ID), 2) วันที่และเวลาในการอ่านค่า (Date & Time) และ 3) ปริมาณการใช้ไฟฟ้า มีหน่วยเป็นกิโลวัตต์ (kW)

### 3.2 การทำความสะอาดข้อมูล (Data Cleansing)

ข้อมูลดิบ (Raw Data) ที่นำมาใช้งานอาจมีหลายปัจจัยที่ส่งผลทำให้การอ่านค่าของมิเตอร์ระบบ AMI ไม่ครบถ้วนหรือมีความผิดพลาดเกิดขึ้น และในบางครั้งอาจไม่สามารถอ่านค่าได้ มีสาเหตุ เช่น ระบบเครือข่ายสื่อสารขัดข้อง และเกิดเหตุการณ์กระแสไฟฟ้าขัดข้อง เป็นต้น จึงมีความจำเป็นอย่างยิ่งที่จะต้องทำความสะอาดข้อมูลก่อนที่จะนำไปใช้งานเพื่อให้แบบจำลองที่สร้างขึ้น มีความแม่นยำและเกิดประสิทธิภาพสูงสุด

ขั้นตอนการทำความสะอาดข้อมูลดิบเริ่มจากการนำข้อมูลบางส่วนออก (Drop Data) เนื่องจากชุดข้อมูลที่อ่านได้จากมิเตอร์ระบบ AMI ของผู้ใช้ไฟฟ้าบางรายมีจำนวนไม่เพียงพอต่อการนำมาฝึกฝนแบบจำลอง โดยการคัดเลือกพิจารณาจากรายที่มีแถวของข้อมูล (Row) มากกว่าหรือเท่ากับ 90% ของจำนวนแถวข้อมูลทั้งหมด หลังจากนั้นทำการคัดเลือกเฉพาะรายที่มีจำนวนแถวของข้อมูลความต้องการใช้ไฟฟ้าเท่ากับ 0 กิโลวัตต์ น้อยกว่าหรือเท่ากับ 50% ของจำนวนแถวข้อมูลทั้งหมด ดังแสดงในรูปที่ 3 เมื่อเสร็จตามขั้นตอนชุดข้อมูลที่ทำความสะอาดสมบูรณ์ (Cleaned Data) ก็พร้อมใช้งาน

จากนั้นทำการสร้างชุดข้อมูล (Dataset) โดยกำหนดให้จำนวนแถวของข้อมูลเท่ากับ 35,136 แถว ซึ่งได้มาจากการคำนวณค่าที่มิเตอร์ระบบ AMI จำเป็นต้องอ่านค่าได้ทั้งปี 2563 (366 วัน, 24 ชั่วโมง, ทุก 15 นาที) จากนั้นนำข้อมูลที่ทำความสะอาดแล้วมาประสานกัน (Merging) และแทนค่าที่หายไป (Missing Values) ด้วยค่าศูนย์ ("0") ชุดข้อมูลขาเข้าที่ได้ก็พร้อมส่งเข้าสู่กระบวนการถัดไป

ตารางที่ 2 ขั้นตอนวิธีที่นำมาใช้ในการศึกษา

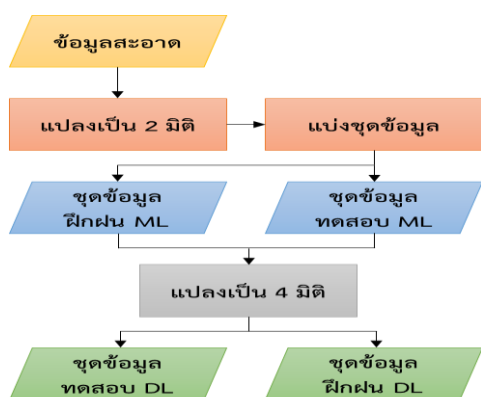
ลำดับ	ขั้นตอนวิธี	รูปแบบการเรียนรู้
1	Elastic Net Regularized Regression (EN)	ML
2	Gradient Boosted Decision Trees (GB)	ML
3	Convolutional, Long-Short Term, Deep Neural Network (CLDNN)	DL

### 3.3 การเตรียมความพร้อมข้อมูล (Data Preprocessing)

ก่อนการฝึกฝนแบบจำลองที่ได้เลือกไว้ตามตารางที่ 2 จำเป็นต้องนำชุดข้อมูลตามข้อ 3.2 เข้าสู่กระบวนการเตรียมความพร้อมของชุดข้อมูลเสียก่อน เพื่อให้ชุดข้อมูลอยู่ในรูปแบบที่เหมาะสมต่อการเรียนรู้ของ ML และ DL มากที่สุด

ในส่วนของบทความนี้ได้เลือก ML ขั้นตอนวิธี EN และ GB ดังนั้น จึงมีความจำเป็นต้องปรับเปลี่ยนข้อมูลจากเดิมที่เป็น 1 มิติ ให้เป็น 2 มิติ (ดังที่ได้กล่าวไว้ในหัวข้อ 2.4) โดยทำการกำหนดจำนวนหลักของข้อมูลขาเข้าเป็น 96 หลัก ทำให้ได้ชุดข้อมูล 2 ส่วนคือ ชุดข้อมูลขาเข้า (X) และชุดข้อมูลเป้าหมาย (y) จากนั้น ทำการแบ่งชุดข้อมูลเพื่อทำการฝึกฝนและการทดสอบ (Train-Test Splitting) ซึ่งทำให้สามารถประเมินประสิทธิภาพของแบบจำลองได้ ส่วนของชุดข้อมูลฝึกฝน (Training) ใช้ข้อมูล 11 เดือนแรกของปี 2563 (มกราคมถึงพฤศจิกายน) และชุดข้อมูลทดสอบ (Testing) ใช้ข้อมูลที่เหลือถัดไป 1 เดือน (ธันวาคม) ทำให้ได้ชุดข้อมูล X-train, X-test, y-train และ y-test สุดท้ายของชุดข้อมูลขาเข้า ทั้ง X-train และ X-test จะต้องถูกปรับช่วงของข้อมูล (Data Scaling) เพื่อทำให้ชุดข้อมูลไม่กระจัดกระจาย โดยบทความนี้เลือกใช้หลักการปรับช่วงแบบมาตรฐาน (Standard Scaling)

ในส่วน ของ DL เนื่องจากบทความนี้เลือกใช้สถาปัตยกรรมแบบ CLDNN ชุดข้อมูลขาเข้าทั้ง X-train และ X-test จะต้องถูกเปลี่ยนจากเดิมที่เป็น 2 มิติ ให้เป็น 4 มิติ ก่อนที่จะทำการป้อนเข้าสู่แบบจำลองด้วยเช่นกัน รูปที่ 5 แสดงผังการไหลของกระบวนการเตรียมความพร้อมข้อมูล



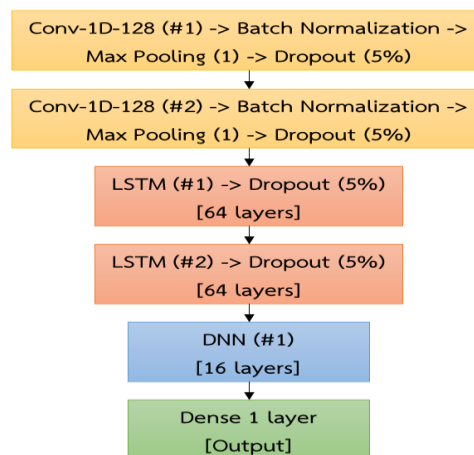
รูปที่ 5 ผังการไหลของกระบวนการเตรียมความพร้อมข้อมูล

### 3.4 การฝึกฝนแบบจำลอง (Model Training)

หลังจากได้ชุดข้อมูลตามข้อ 3.3 แล้ว ชุดข้อมูลดังกล่าว จะถูกนำเข้าสู่แบบจำลองทั้ง 3 รูปแบบ เพื่อทำการฝึกฝน โดยเป็นการฝึกฝนแบบเฉพาะเจาะจง บนข้อมูลของผู้ใช้ไฟฟ้าแต่ละราย ตารางที่ 3 แสดงการปรับตั้งค่าพารามิเตอร์ต่างๆ (Parameter Tuning) ของแบบจำลอง EN และ GB รูปที่ 6 แสดงโครงสร้างสถาปัตยกรรมของ CLDNN ที่ถูกสร้างเพื่อใช้ทำการศึกษาในบทความนี้

ตารางที่ 3 การปรับแต่งค่าพารามิเตอร์ของแบบจำลอง

ชื่อแพ็คเกจ	พารามิเตอร์ที่ปรับแต่งแล้ว
ElasticNet (Scikit-learn)	alpha=0.001, l1_ratio=0.23
LGBMRegressor (LightGBM)	learning_rate = 0.06873684210526317, reg_alpha = 0.75, reg_lambda = 0.5, min_split_gain = 0.255



รูปที่ 6 โครงสร้างสถาปัตยกรรมของ CLDNN

### 3.5 การประเมินแบบจำลอง (Model Evaluating)

หลังจากที่ได้ทำการสร้างและฝึกฝนแบบจำลองเสร็จเรียบร้อยแล้ว ขั้นตอนต่อไปคือการให้แบบจำลองเหล่านั้น ทำการพยากรณ์โดยการป้อนข้อมูล X-test เข้าไป ผลที่ได้ออกมา มักถูกเรียกว่า y-pred และเพื่อเป็นการทดสอบประสิทธิภาพของแบบจำลองทั้ง 3 แบบ ที่ได้ทำการเลือกไว้ในบทความนี้ เลือกใช้ฟังก์ชันการสูญเสีย (Loss Function) ค่าเฉลี่ยข้อผิดพลาดกำลังสอง (Mean Squared Error, MSE) เป็นตัววัดค่าประสิทธิภาพ ดังสมการที่ (2)

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 \quad (2)$$

เมื่อ n คือ จำนวนตัวอย่าง,  $Y_i$  คือ ค่าจริงของตัวอย่างที่ i และ  $\hat{Y}_i$  คือ ค่าพยากรณ์ของตัวอย่างที่ i สำหรับการคำนวณหาค่าความผิดพลาด (Error) โดยแบบจำลองที่มีค่า MSE น้อยที่สุด จะแสดงถึงความสามารถสูงที่สุด ซึ่งก็ จะถูกเลือกให้เป็นตัวแทนแบบจำลองของผู้ใช้ไฟฟ้าแต่ละราย

## 4. ผลการศึกษา

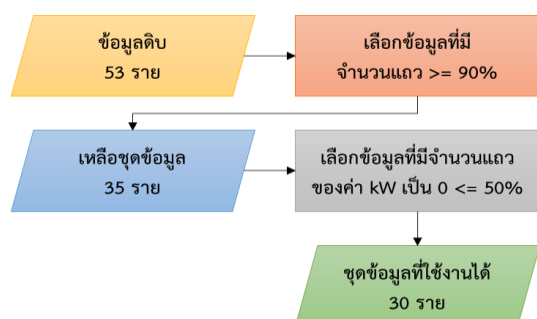
จากที่ได้ดำเนินการตามวิธีการและขั้นตอนการศึกษาในข้อที่ 3 เสร็จเรียบร้อยแล้ว ต่อไปจะเป็นการแสดงผลการศึกษา โดยมีรายละเอียดดังต่อไปนี้

### 4.1 ผลการคัดเลือกชุดข้อมูลขาเข้า

หลังจากได้นำชุดข้อมูลการใช้ไฟฟ้าของลูกค้าที่อ่านได้จากมิเตอร์ระบบ AMI ดังที่ได้กล่าวไว้ในหัวข้อ 3.1



และได้ดำเนินการทำความสะอาดข้อมูลตามหัวข้อ 3.2 เป็นที่เรียบร้อยแล้วนั้น ผังการไหลของผลการคัดเลือกชุดข้อมูลขาเข้า จะสามารถถูกแสดงได้ดังรูปที่ 7 ซึ่งแสดงให้เห็นว่า จากจำนวนผู้ใช้ไฟฟ้าทั้งสิ้น 53 ราย จะเหลือเพียง 30 ราย เท่านั้น ที่เข้าเงื่อนไขที่ตั้งไว้ และสามารถนำไปใช้ในการดำเนินการขั้นต่อไปได้



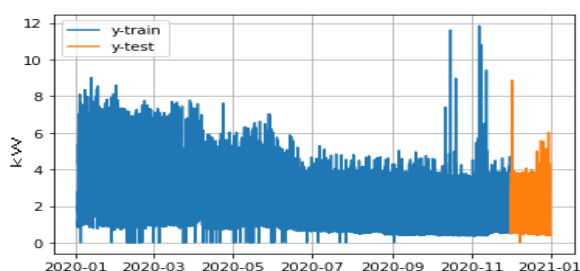
รูปที่ 7 ผังการไหลของผลการคัดเลือกชุดข้อมูลขาเข้า

## 4.2 ผลการเตรียมความพร้อมข้อมูล

จากชุดข้อมูลที่ได้ตามหัวข้อ 4.1 จะถูกนำเข้าสู่ขั้นตอนการเตรียมความพร้อมข้อมูลตามหัวข้อ 3.3 ตารางที่ 4 แสดงขนาดของชุดข้อมูลที่ถูกเตรียมความพร้อมแล้ว และรูปที่ 8 แสดงตัวอย่างของ y-train และ y-test ของผู้ใช้ไฟหนึ่งราย

ตารางที่ 4 ขนาดของชุดข้อมูลที่ถูกเตรียมความพร้อมแล้ว

แบบจำลอง	ชุดข้อมูล 2 มิติ	ชุดข้อมูล 4 มิติ
X-train	(32064, 96)	(32064, 1, 1, 96)
X-test	(2976, 96)	(2976, 1, 1, 96)
y-train	(32064, 1)	
y-test	(2976, 1)	



รูปที่ 8 ผลการเตรียมชุดข้อมูล y-train และ y-test

## 4.3 ผลการฝึกฝนและประเมินแบบจำลอง

หลังจากได้ชุดข้อมูลตามหัวข้อ 4.2 แล้วนั้น ต่อไปจะเป็นผลจากการฝึกฝนและประเมินแบบจำลอง ตามขั้นตอนที่ได้กล่าวไว้ในหัวข้อ 3.4 และ 3.5 การศึกษาครั้งนี้ได้เลือกใช้ระบบปฏิบัติการ Windows 11 Pro (64 Bits), CPU 6 cores 12 threads ความเร็ว 2.3GHz และ Ram ขนาด 16 GB ตารางที่ 5 แสดงผลการประเมินแบบจำลองบนข้อมูลผู้ใช้ไฟฟ้าทั้ง 30 ราย

จากตารางที่ 5 พบว่า แบบจำลอง GB เป็นแบบจำลองที่ถูกเลือกมากที่สุดคือ 19 ราย จากทั้งหมด 30 ราย ซึ่งหมายความว่า แบบจำลอง GB ให้ค่า MSE น้อยที่สุดสำหรับ

ผู้ใช้ไฟเหล่านั้น เมื่อเปรียบเทียบกับแบบจำลองอื่น โดยมีค่าเฉลี่ย MSE อยู่ที่ 0.0907 (ลำดับที่ 2) ถัดไปคือแบบจำลอง EN (9 จาก 30 ราย) โดยมีค่าเฉลี่ย MSE อยู่ที่ 0.0905 (ลำดับที่ 1) และ ลำดับสุดท้ายคือแบบจำลอง CLDNN (2 จาก 30 ราย) โดยมีค่าเฉลี่ย MSE อยู่ที่ 0.1001 (ลำดับที่ 3)

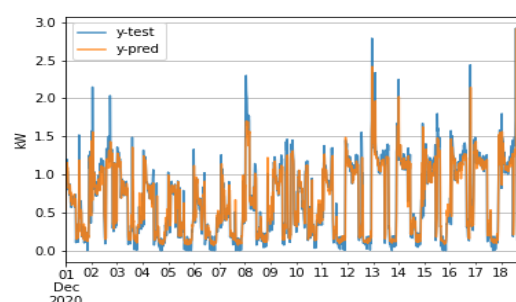
ตารางที่ 5 ผลการประเมินแบบจำลอง

แบบจำลอง	จำนวนผู้ใช้ไฟฟ้าที่เลือก (ราย)	ค่าเฉลี่ย MSE (ต่อราย)	ค่าเฉลี่ยเวลาในการประมวลผล (วินาทีต่อราย)
EN	9	0.0905	0.7736
GB	19	0.0907	0.4499
CLDNN	2	0.1001	66.4953

นอกจากนั้น จากตารางที่ 5 ยังมีข้อสังเกตที่น่าสนใจในส่วนของค่าเฉลี่ยเวลาที่ใช้ในการประมวลผล โดยพบว่าแบบจำลอง GB ใช้เวลาน้อยที่สุดที่ 0.4499 วินาทีต่อราย ถัดไปคือแบบจำลอง EN ใช้เวลา 0.7736 วินาทีต่อราย และสุดท้ายคือ CLDNN ใช้เวลามากถึง 66.4953 วินาทีต่อราย หรือประมาณ 1 นาที ตามลำดับ

## 4.4 ผลการพยากรณ์ด้วยชุดข้อมูลแบบมีผู้สอน

จากผลการฝึกฝนและประเมินแบบจำลอง พบว่าระเบียบวิธี GB เหมาะสมที่สุด ดังนั้นบทความนี้จึงเลือกแบบจำลองดังกล่าว มาทำการพยากรณ์ความต้องการใช้ไฟฟ้า รูปที่ 9 แสดงผลการพยากรณ์ด้วยชุดข้อมูลแบบมีผู้สอน (X-test) ทำให้ได้ค่าพยากรณ์ (y-pred) เปรียบเทียบกับค่าจริง (y-test) กับผู้ใช้ไฟฟ้ารายหนึ่ง ด้วยระยะเวลา 18 วันแรกของเดือนธันวาคม 2563

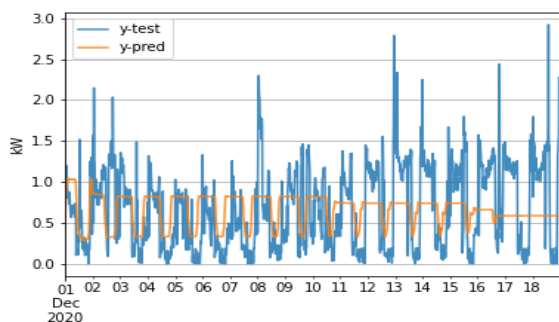


รูปที่ 9 ผลการพยากรณ์ความต้องการใช้ไฟฟ้า

จากรูปที่ 9 พบว่าแบบจำลองที่สร้างขึ้นด้วยขั้นตอนวิธีนี้มีค่า MSE เท่ากับ 0.0468 และใช้เวลาในการประมวลผลเท่ากับ 0.8787 วินาที ซึ่งเป็นผลลัพธ์ที่มีประสิทธิภาพสูง

## 5. การนำแบบจำลองที่ฝึกฝนแล้วไปประยุกต์ใช้

การใช้งานจริง ข้อมูลแบบมีผู้สอนตามหัวข้อ 4.4 ไม่ได้ถูกจัดเตรียมไว้ให้ ดังนั้นจึงต้องทำการพยากรณ์แบบ Recursive Multi-step Forecasting [13] ซึ่งเป็นการนำค่าความต้องการใช้ไฟฟ้าที่พยากรณ์ได้ มาใช้เป็นข้อมูลขาเข้าในการพยากรณ์ครั้งถัด ๆ ไป ผลการดำเนินการแสดงในรูปที่ 10



รูปที่ 10 ผลการประยุกต์ใช้การพยากรณ์ความต้องการใช้ไฟฟ้า

จากการพยากรณ์แบบ Recursive Multi-step Forecasting พบว่าวิธีดังกล่าว บนระยะเวลาที่เท่ากับในหัวข้อ 4.4 นั้น มีค่า MSE เท่ากับ 0.2259 ซึ่งไม่สามารถทำนายได้อย่างมีประสิทธิภาพ แต่สามารถคาดการณ์พฤติกรรมการใช้ไฟฟ้าของผู้ใช้ได้ประมาณ 15 วัน และหลังจากนั้นจะให้ผลเป็นค่าคงที่ค่าหนึ่ง หากนำไปใช้งานจริง อาจจะพยากรณ์ได้เพียง 1 วันล่วงหน้าเท่านั้น

## 6. สรุปและอภิปรายผล

บทความนี้แนะนำเสนอการพยากรณ์ความต้องการใช้ไฟฟ้าบนชุดข้อมูลจากมิเตอร์ระบบ AMI ด้วยหลักการ ML และ DL เพื่อค้นหาระเบียบวิธี ที่มีความเหมาะสมกับชุดข้อมูลแบบอนุกรมเวลาตัวแปรต้นเดียว-ตัวแปรตามหนึ่งชั้น จากผลการศึกษาที่ได้กล่าวไปในข้างต้น สรุปได้ว่า แบบจำลอง GB เป็นแบบจำลองที่เหมาะสมกับชุดข้อมูลดังกล่าวมากที่สุด เนื่องจากเป็นแบบจำลองที่ได้รับเลือกมากที่สุดคิดเป็น 63% จากจำนวนผู้ใช้ไฟฟ้าทั้งสิ้น 30 ราย รวมถึงมีค่าเฉลี่ย MSE ต่อผู้ใช้ไฟฟ้าที่ต่ำ และใช้เวลาการประมวลผลเฉลี่ยต่อผู้ใช้ไฟฟ้าน้อยกว่าแบบจำลองบนระเบียบวิธี EN และ CDLNN

ในส่วนของการใช้งานจริง บทความนี้ได้นำเสนอผลจากการประยุกต์ใช้วิธีการพยากรณ์แบบ Recursive Multi-step Forecasting ซึ่งเป็นการนำค่าความต้องการใช้ไฟฟ้าที่พยากรณ์ได้ มาใช้เป็นข้อมูลขาเข้าในการพยากรณ์ครั้งถัดไป จากผลการศึกษาพบว่า แบบจำลองที่ถูกฝึกฝนขึ้นมานั้น เหมาะกับการนำไปใช้ในงานพยากรณ์พฤติกรรมการใช้ไฟฟ้า โดยไม่ควรคาดการณ์ล่วงหน้าเกิน 1 วัน (Day Ahead) และหากต้องการเพิ่มประสิทธิภาพในการพยากรณ์ล่วงหน้านั้น อาจจะทำได้โดยการแปลงข้อมูลรูปแบบอื่น ที่ได้กล่าวไว้ในข้อ 2.4 มาใช้เป็นข้อมูลขาเข้า แทนการใช้ข้อมูลแบบอนุกรมเวลาตัวแปรต้นเดียว-ตัวแปรตามหนึ่งชั้น

นอกจากการฝึกฝนแบบจำลองเพื่อพยากรณ์ความต้องการใช้ไฟฟ้าในผู้ใช้ไฟฟ้าแต่ละรายแล้ว กฟภ. ยังสามารถทำการฝึกฝนแบบจำลอง เพื่อนำไปประยุกต์ใช้กับงานด้านอื่นได้ เช่น การวางแผนปรับปรุงระบบจำหน่ายไฟฟ้า รวมถึงการเตรียมความพร้อมเข้าสู่ตลาดการซื้อขายพลังงานไฟฟ้าที่จะมีการเปลี่ยนแปลงในอนาคตต่อไป

## 7. กิตติกรรมประกาศ

บทความฉบับนี้เขียนสำเร็จลงด้วยดี เนื่องจากได้รับความอนุเคราะห์ข้อมูลจากมิเตอร์ระบบ AMI จากกองแผนงานระบบไฟฟ้าอัจฉริยะ กฟภ. รวมถึงคำแนะนำและข้อเสนอแนะจากคุณอนุรักษ์ เขยชุม ซึ่งทางคณะผู้เขียนบทความหวังเป็นอย่างยิ่งว่า จะเป็นประโยชน์ต่อองค์กรและผู้ที่เกี่ยวข้องต่อไป

## เอกสารอ้างอิง

- [1] “How to Develop Elastic Net Regression Models in Python”. [Online] Available: <https://machinelearningmastery.com/elastic-net-regression-in-python/> [Accessed October, 2021]
- [2] “A Gentle Introduction to the Gradient Boosting Algorithm for Machine Learning”. [Online] Available: <https://machinelearningmastery.com/gentle-introduction-gradient-boosting-algorithm-machine-learning/> [Accessed October, 2021]
- [3] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, “Convolutional, Long Short-Term Memory, fully connected Deep Neural Networks”, in 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4580-4584, 2015.
- [4] “How to Develop LSTM Models for Time Series Forecasting”. [Online] Available: <https://machinelearningmastery.com/how-to-develop-lstm-models-for-time-series-forecasting/> [Accessed October, 2021]
- [5] “How to Convert a Time Series to a Supervised Learning Problem in Python”. [Online] Available: <https://machinelearningmastery.com/convert-time-series-supervised-learning-problem-python/> [Accessed November, 2021]
- [6] G. Van Rossum, and F. L. Drake, “Python 3 Reference Manual”, Scotts Valley, CA: CreateSpace, 2015.
- [7] W. McKinney, “Data Structures for Statistical Computing in Python”, Proceedings of the 9th Python in Science Conference, pp. 56-61, 2010.
- [8] Harris et al., “Array programming with NumPy”, in Nature, vol. 585, pp. 357-362, 2020.
- [9] J.D. Hunter, “Matplotlib: A 2D Graphics Environment. Computing in Science & Engineering”, in IEEE COMPUTER SOC, Vol 9, No.3, pp. 90-95, 2007.
- [10] Pedregosa et al., “Scikit-learn: Machine Learning in Python”, in Journal of Machine Learning Research, Vol 12, pp. 2825-2830, 2011.
- [11] Ke et al., “Lightgbm: A highly efficient gradient boosting decision tree”, in Advances in Neural Information Processing Systems, Vol 30, pp. 3146-3154, 2017.
- [12] Abadi et al., “TensorFlow: Large-scale machine learning on heterogeneous systems”, White Paper from Google Research, Available at TensorFlow.org, 2015.
- [13] “4 Strategies for Multi-Step Time Series Forecasting”. [Online] Available: <https://machinelearningmastery.com/multi-step-time-series-forecasting/> [Accessed October, 2021]