แบบจำลองการทำนายระยะเวลาในการเข้าเทียบท่าของเรือโดยสารสาธารณะ
Arrival Time Prediction Model to a Pier for Public Transportation Boats

ชนะวิชญ์ พัชรเจริญวงษ์[1] กันต์กวี เหิรเมฆ[1] และ วรางคณา กิ้มปาน[1*]
Chanawit Patcharacharoenwong[1] Kankawee Hernmek[1] and Warangkhana Kimpan[1]
[1]ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
[1]Department of Computer Science, Faculty of Science, King Mongkut's Institute of Technology Ladkrabang
วันที่ส่งบทความ : 26 มีนาคม 2563 วันที่แก้ไขบทความ : 19 สิงหาคม 2563 วันที่ตอบรับบทความ : 30 ตุลาคม 2563
Received: 26 March 2020, Revised: 19 August 2020, Accepted: 30 October 2020

บทคัดย่อ

งานวิจัยนี้นำเสนอ การสร้างแบบจำลองเพื่อทำนายระยะเวลาในการเข้าเทียบท่าของเรือโดยสารสาธารณะ โดยใช้ข้อมูลการเดินเรือจากอุปกรณ์อินเทอร์เน็ตประสานสรรพสิ่ง (Internet of Things : IoT) บนเรือโดยสารสาธารณะ ตั้งแต่วันที่ 27 พฤศจิกายน 2561 ถึงวันที่ 31 พฤษภาคม 2562 เพื่อนำข้อมูลของเรือโดยสารสาธารณะมาวิเคราะห์หาความสัมพันธ์ และนำไปสร้างเป็นแบบจำลองการทำนายที่สร้างด้วยอัลกอริทึมการถดถอยต่าง ๆ ทั้ง 6 อัลกอริทึม ประกอบไปด้วย Linear Regression, Random Forest, Gradient Boost, eXtreme Gradient Boost, Light Gradient Boost และ CatBoost จากนั้นทดสอบประสิทธิภาพแบบจำลอง ด้วยการหารากของค่าความคลาดเคลื่อนกำลังสองเฉลี่ย (Root Mean Square Error : RMSE) เพื่อหาแบบจำลองที่ดีที่สุด และวัดผลโดยนำระยะเวลาที่แบบจำลองทำนายได้ ไปทดสอบเทียบกับระยะเวลาการเดินเรือของเรือโดยสาร ซึ่งผลการวิจัยพบว่าแบบจำลองของการเดินเรือที่สร้างจากอัลกอริทึม CatBoost มีค่า RMSE ดีที่สุด ในการเดินเรือจากท่าเรือต้นทางไปยังท่าเรือปลายทาง อยู่ที่ 88.25 วินาที และในเที่ยวการเดินเรือปลายทางไปยังต้นทาง มีค่าความคลาดเคลื่อนโดยเฉลี่ยในการเดินเรือดีที่สุดอยู่ที่ 90.06 วินาที เมื่อเทียบกับอัลกอริทึมอื่น

คำสำคัญ : แบบจำลองการทำนาย เรือโดยสารสาธารณะ การเรียนรู้ของเครื่อง อัลกอริทึม CatBoost

Abstract

This paper proposed a time prediction model of public transportation boats arrival to a pier using navigational data from the Internet of Things (IoT) devices attached to public boats from 27 November 2018 to 31 May 2019 in order to analyze the relationship of public transportation boats data used in creating a prediction model. Six algorithms

*ที่อยู่ติดต่อ E-mail address: warangkhana.ki@kmitl.ac.th

which are Linear Regression, Random Forest, Gradient Boost, eXtreme Gradient Boost, Light Gradient Boost, and CatBoost, were created into models. After that, the performance of the models were tested by Root Mean Square Error in order to find the optimal model and evaluated by testing the predicted arrival time models then compared to the arrival time from IoT devices at transportation boats. The results show that the model created by CatBoost algorithm performed the optimal of Root Mean Square Error values at 88.25 seconds of the travelling time from the origin to the destination pier and at 90.06 seconds of the return trip compared to other algorithms.

Keywords: Prediction model, Public transportation boat, Machine learning, CatBoost algorithm

1. บทนำ

กรุงเทพมหานครเป็นเมืองหลวงและจุดศูนย์กลางของเศรษฐกิจในประเทศไทย ผู้คนส่วนใหญ่จึงเดินทางเข้ามาทำงานที่กรุงเทพมหานครกันเป็นจำนวนมาก โดยมีจำนวนกว่า 5.5 ล้านคน ส่งผลให้การจราจรหนาแน่นติดขัดตามไปด้วย เนื่องจากจำนวนประชากรที่มีมากทำให้ผู้คนที่ใช้รถยนต์ส่วนตัวเป็นพาหนะในการเดินทางก็มีปริมาณมากตามไปด้วย ด้วยเหตุนี้ประชากรส่วนใหญ่ที่อาศัยอยู่ภายในกรุงเทพมหานครจึงหันมาใช้งานบริการขนส่งสาธารณะ เนื่องจากมีความสะดวก ประหยัด และรวดเร็วกว่า เพื่อหลีกเลี่ยงปัญหาการจราจรที่ติดขัดอันเนื่องมาจากผู้ใช้รถใช้ถนนที่มีจำนวนมาก การคมนาคมทางน้ำก็เป็นอีกทางเลือกหนึ่ง ที่สะดวกรวดเร็วไม่แพ้การคมนาคมทางรางหรือทางถนน

ในปัจจุบันการคมนาคมทางน้ำด้วยเรือโดยสารสาธารณะ มีผู้ใช้บริการเฉลี่ยราว 48,982 คนต่อเดือน ซึ่งผู้ใช้บริการเหล่านี้ ไม่สามารถทราบได้ว่าเรือโดยสารตอนนี้อยู่ตรงจุดใด และจะมาถึงท่าเทียบเรือตรงจุดที่ตนเองรออยู่เมื่อใด การวางแผนการเดินทางในเวลาเร่งด่วนก็อาจจะทำได้ยาก ซึ่งปัญหานี้สามารถใช้กระบวนการทางวิทยาการข้อมูลในการวิเคราะห์ข้อมูลการเดินเรือของเรือโดยสารสาธารณะ และนำไปสร้างเป็นแบบจำลองเพื่อหาระยะเวลาที่เรือจะเข้ามาเทียบท่าในแต่ละท่าเรือได้ เป็นการช่วยอำนวยความสะดวกในการวางแผนการเดินทางให้กับผู้ใช้บริการเรือโดยสารได้อีกทางหนึ่ง ดังนั้นงานวิจัยนี้เป็นการนำเสนอการสร้างแบบจำลอง เพื่อทำนายระยะเวลาในการเข้าเทียบท่าของเรือโดยสารสาธารณะ ให้ใกล้เคียงกับสภาพความเป็นจริงมากที่สุด โดยแบ่งข้อมูลออกเป็นการเดินเรือในเที่ยวการเดินเรือต้นทางไปยังปลายทาง และเที่ยวการเดินเรือปลายทางไปยังต้นทาง จากนั้นนำข้อมูลการเดินเรือทั้งหมดไปสร้างเป็นแบบจำลอง โดยใช้อัลกอริทึมจำนวน 6 อัลกอริทึม ประกอบไปด้วย Linear Regression, Random Forest, Gradient Boost, eXtreme Gradient Boost, Light Gradient Boost และ CatBoost

2. ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 การเรียนรู้ของเครื่อง (Machine Learning)

เป็นสาขาหนึ่งของปัญญาประดิษฐ์ เกี่ยวข้องกับการศึกษาและการสร้างอัลกอริทึมที่สามารถเรียนรู้ข้อมูล และทำนายผลข้อมูลได้ อัลกอริทึมจะทำงานโดยอาศัยแบบจำลองที่สร้างมาจากชุดข้อมูลตัวอย่างขาเข้าเพื่อการทำนายหรือตัดสินใจในภายหลัง แทนที่จะทำงานตามลำดับของคำสั่งโปรแกรมคอมพิวเตอร์ การเรียนรู้ของเครื่องแบ่งออกเป็นการเรียนรู้ของเครื่องแบบมีผู้สอน (Supervised Learning) และการเรียนรู้ของเครื่องแบบไม่มีผู้สอน (Unsupervised Learning) ซึ่งการเรียนรู้แบบมีผู้สอนจำเป็นต้องมีข้อมูลในส่วนสำหรับฝึกสอน (Training Data) จากที่มนุษย์ป้อนเข้ามา เพื่อให้อัลกอริทึมสร้างแบบจำลองในการเรียนรู้ความสัมพันธ์ระหว่างข้อมูลแล้วทำนายผลลัพธ์ออกมา แต่การเรียนรู้แบบนี้แตกต่างจากการเรียนรู้แบบไม่มีผู้สอน คือจะไม่มีการระบุผลที่ต้องการไว้ก่อน เนื่องด้วยการเรียนรู้แบบนี้จะพิจารณาวัตถุเป็นเซตของตัวแปรสุ่ม เพียงทำการป้อนข้อมูลเข้าไปแล้วบอกความต้องการ จากนั้นจึงให้เครื่องทำนายออกมา [1]

2.2 การวิเคราะห์การถดถอย (Regression Analysis)

การวิเคราะห์การถดถอยเป็นสถิติวิเคราะห์ชนิดหนึ่งที่ใช้ในการศึกษา และตรวจสอบความสัมพันธ์ระหว่างตัวแปร ตั้งแต่ 2 ตัวแปรขึ้นไป โดยแบ่งเป็นตัวแปรอิสระ (Independent Variable) และตัวแปรตาม (Dependent Variable) ตัวแปร อิสระ มักเรียกว่า “ตัวแปรพยากรณ์ (Predicted Variable)” ส่วนตัวแปรตามมักเรียกว่า “ตัวแปรตอบสนอง (Response Variable)” โดยสามารถนำมาสร้างเป็นแบบจำลองการถดถอยเชิงเส้นอย่างง่าย (Simple Linear Regression Model) ได้โดย สมมติว่า Y และ X มีความสัมพันธ์กันดังสมการ (1) [2]

$$Y = \alpha + \beta X + \varepsilon \quad (1)$$

โดยที่ α และ β เป็นสัมประสิทธิ์การถดถอยของประชากร (Population Regression Coefficient) จะถือว่าเป็นค่าคงที่และไม่ทราบค่า เรียกว่า ตัวแบบการถดถอยเชิงเส้น ในเบื้องต้นจะสามารถตรวจสอบความสัมพันธ์ระหว่าง Y กับ X ว่ามีความสัมพันธ์เชิงเส้นตรงหรือไม่โดยการนำค่าของ Y กับ X พลอตเป็นจุดแผนภาพที่ได้ เรียกว่า แผนภาพการกระจาย (Scatter Diagram) เมื่อทราบว่า Y กับ X ว่ามีความสัมพันธ์เชิงเส้นตรง จากนั้นประมาณตัวแบบด้วยสมการการถดถอย (Regression Equation) ดังสมการที่ (2)

$$\hat{Y} = a + bX \quad (2)$$

โดยที่ a และ b เป็นตัวประมาณแบบกำลังสองน้อยสุด (Least Square Method) ของ α และ β ตามลำดับ กล่าวคือ เราจะหา a และ b ที่ทำให้ SSE = $\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$ = $\sum_{i=1}^{n} e_i^2$ มีค่าน้อยที่สุด ซึ่งค่าของ a และ b จะเป็นค่าประมาณของ α และ β ตามลำดับ เรียก a และ b ว่า สัมประสิทธิ์การถดถอยของตัวอย่าง (Sample Regression Coefficient) โดยที่

$$a = \bar{Y} - b\bar{X} \tag{3}$$

และ $$b = \frac{\sum_{i=1}^{n} X_i Y_i - \frac{(\sum_{i=1}^{n} X_i)(\sum_{i=1}^{n} Y_i)}{n}}{\sum_{i=1}^{n} X_i^2 - \frac{(\sum_{i=1}^{n} X_i)^2}{n}} \tag{4}$$

เมื่อกำหนดสัญลักษณ์

$$S_{XX} = \sum_{i=1}^{n}(X_i - \bar{X})^2 = \sum_{i=1}^{n} X_i^2 - \frac{(\sum_{i=1}^{n} X_i)^2}{n} \tag{5}$$

$$S_{XY} = \sum_{i=1}^{n} (X_i - \bar{X})(Y_i - \bar{Y}) = \sum_{i=1}^{n} X_i Y_i - \frac{(\sum_{i=1}^{n} X_i)(\sum_{i=1}^{n} Y_i)}{n} \tag{6}$$

$$S_{YY} = \sum_{i=1}^{n}(Y_i - \bar{Y})^2 = \sum_{i=1}^{n} Y_i^2 - \frac{(\sum_{i=1}^{n} Y_i)^2}{n} \tag{7}$$

สามารถเขียนได้เป็น

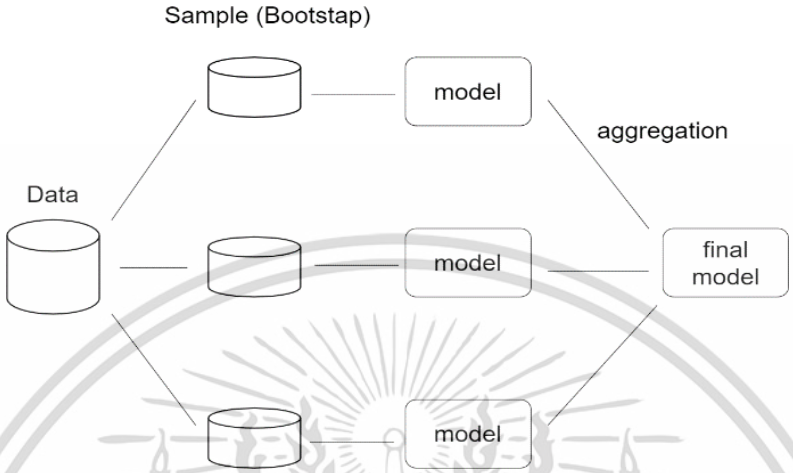
$$b = \frac{S_{XY}}{S_{XX}} \tag{8}$$

2.3 Ensemble Learning

เป็นการนำแบบจำลองมาเรียนรู้หลาย ๆ ครั้ง เพื่อเพิ่มประสิทธิภาพของแบบจำลอง ซึ่งเทคนิคที่ใช้กันบ่อยได้แก่ Bagging และ Boosting [3] มีรายละเอียดดังนี้

2.3.1 Bagging

Bagging หรือ Bootstrap Aggregation เป็นการสร้างแบบจำลองออกมาหลาย ๆ แบบจำลอง โดยใช้การสุ่มข้อมูลตัวอย่างจากข้อมูลฝึกสอนออกมาเป็นหลาย ๆ ชุด สำหรับวิธีการสุ่มข้อมูลออกมาเป็นวิธีสุ่มแบบแทนที่ (Random with Replacement) หมายความว่าข้อมูลที่เรามียังคงเดิมไม่ได้ลดลงหลังจากการสุ่ม ซึ่งสามารถสุ่มข้อมูลหลาย ๆ รอบเพื่อให้ได้ classifier หลาย ๆ ตัว เวลาทำนายจะใช้ classifier ทุกตัวที่สร้างขึ้นมาเพื่อทำนายชุดข้อมูลใหม่ที่พบ ลักษณะการทำงานของ Bagging แสดงดังรูปที่ 1

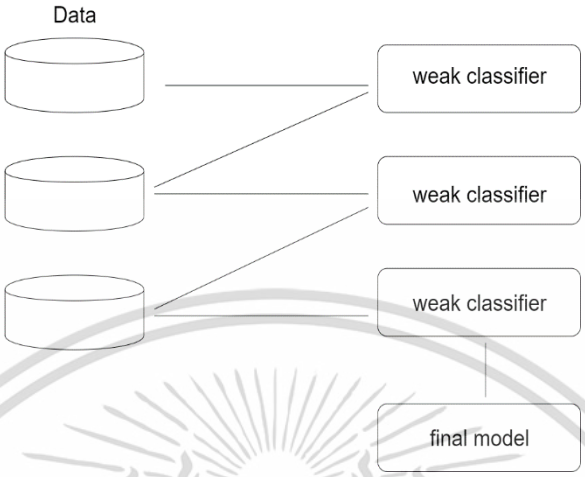


รูปที่ 1. ลักษณะการทำงานของ Bagging

Bagging เป็นพื้นฐานของ Random Forest Classifier ซึ่งเป็นแบบจำลองที่นำต้นไม้ตัดสินใจ (Decision Tree) หลาย ๆ ต้นมาฝึกสอนร่วมกัน ตั้งแต่ 10 ต้นขึ้นไปจนถึง 1000 ต้น หรือมากกว่านั้น โดยที่ต้นไม้ตัดสินใจแต่ละต้นจะได้รับข้อมูลที่เป็น subset ของ feature และข้อมูลทั้งหมดแบบสุ่ม จากนั้นก็ให้ต้นไม้ตัดสินใจแต่ละต้นทำการทำนายและเลือกคำตอบจากค่าที่ทำนายที่ได้รับการโหวตมากที่สุด

2.3.2 Boosting

เป็นการนำ classifier ที่มีความแม่นยำต่ำ (Weak Classifier) มาทำนายข้อมูล จากนั้นจะให้ classifier ที่มีความแม่นยำต่ำตัวใหม่มาแก้ไข error โดยผลรวมของ classifier จะเกิดเป็น classifier ใหม่ขึ้น และจะทำแบบนี้ไปจนแบบจำลองที่ได้ไม่มีค่าคลาดเคลื่อนเกิดขึ้น ซึ่งจะเป็นแบบจำลองที่ดีที่สุด ลักษณะการทำงานของ Boosting แสดงดังรูปที่ 2 ตัวอย่างโมเดล ได้แก่ Gradient boosting (GBM), AdaBoost เป็นต้น และยังมีโมเดลที่ถูกพัฒนาในลักษณะที่คล้ายกันที่น่าสนใจ และผู้วิจัยได้นำมาสร้างแบบจำลอง ได้แก่ XGBoost (eXtreme Gradient Boosting), Light GBM (Light Gradient Boosting) และ CatBoost



รูปที่ 2. ลักษณะการทำงานของ Boosting

2.3.2.1 GBM (Gradient Boosting)

Gradient Boosting เป็นเทคนิคการเรียนรู้ของเครื่องสำหรับแก้ปัญหาการถดถอย (Regression) และการจำแนกประเภท (Classification) GBM จะสร้างโครงสร้างการถดถอยตามลำดับ ซึ่ง GBM ใช้เทคนิคการเพิ่มการรวมจำนวน classifier ที่มีความแม่นยำต่ำ เพื่อสร้างเป็น classifier ใหม่โดยต้นไม้ในลำดับต่อไปจะถูกสร้างจากข้อผิดพลาดจากการคำนวณต้นไม้ก่อนหน้าโดยใช้อัลกอริทึม Level-wise ในการสร้างต้นไม้ [1]

2.3.2.2 XGBoost (eXtreme Gradient Boosting)

XGBoost เป็นเทคนิคที่พัฒนามาจาก Gradient Boosting ซึ่ง XGBoost เป็นแบบจำลองที่นำเอาต้นไม้ตัดสินใจมาฝึกสอนต่อกันหลาย ๆ ต้น โดยที่ต้นไม้ตัดสินใจแต่ละต้นจะเรียนรู้จากค่าความผิดพลาดของต้นก่อนหน้า ซึ่งทำให้ความแม่นยำในการทำนายจะมากขึ้นเรื่อย ๆ เมื่อมีการเรียนรู้ของต้นไม้ตัดสินใจต่อเนื่องกันจนมีความลึกมากพอ แบบจำลองจะหยุดเรียนรู้เมื่อไม่เหลือค่าความผิดพลาดจากต้นไม้ตัดสินใจต้นก่อนหน้าให้เรียนรู้แล้ว [4]

2.3.2.3 Light GBM (Light Gradient Boosting)

Light GBM เป็น GBM เฟรมเวิร์กที่มีประสิทธิภาพสูง ใช้อัลกอริทึมต้นไม้ตัดสินใจสำหรับทำ Classification หรือ Regression โดยที่ Light GBM ใช้อัลกอริทึม Leaf-wise ซึ่งสามารถลดการสูญเสียได้มากกว่าอัลกอริทึม Level-wise ดังนั้นจึงให้ผลลัพธ์ที่แม่นยำและมีความเร็วมากกว่า [1]

2.3.2.4 CatBoost (Gradient boosting with categorical features support)

CatBoost มาจากคำว่า “**Cat**egory” และ “**Boost**ing” เป็นอัลกอริทึมการเรียนรู้ของเครื่องจาก Yandex ที่เปิดให้ใช้งานแบบโอเพนซอร์ส CatBoost สามารถใช้งานร่วมกับเฟรมเวิร์กการ

เรียนรู้เชิงลึก เช่น Google's TensorFlow และ Apple's Core ML ได้ ซึ่งอัลกอริทึม CatBoost ได้เพิ่มประสิทธิภาพการทำงานจาก GBM โดยสามารถจัดการกับตัวแปรโดยอัตโนมัติ และไม่ต้องการการแปลงชุดข้อมูลให้เป็นรูปแบบเฉพาะ นอกจากนี้ CatBoost ยังสามารถจัดการกับตัวแปรและค่าข้อมูลบางส่วนที่หายไป (Missing Values) ได้อย่างมีประสิทธิภาพอีกด้วย [5]

2.4 ความถูกต้องของการทำนาย

แบบจำลองการทำนายทุกวิธีจะมีค่าความคลาดเคลื่อนเกิดขึ้นโดยที่ความถูกต้องของค่าที่ทำนายได้จะมากหรือน้อยขึ้นอยู่กับค่าความคลาดเคลื่อนของการทำนาย ซึ่งสามารถหาค่าความคลาดเคลื่อนของการทำนายได้จากวิธีดังต่อไปนี้ [6]

กำหนดให้ A คือ ค่าข้อมูลจริง

P คือ ค่าผลลัพธ์ที่ได้จากการพยากรณ์

n คือ จำนวนข้อมูลทั้งหมด

- ค่าความคลาดเคลื่อนกำลังสองเฉลี่ย (Mean Square Error : MSE) ดังสมการที่ (9)

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(A-P)^2 \tag{9}$$

- รากของค่าความคลาดเคลื่อนกำลังสองเฉลี่ย (Root Mean Square Error : RMSE) หรือความคลาดเคลื่อนมาตรฐาน (Standard Error : SE) ดังสมการที่ (10)

$$\mathrm{RMSE} = \mathrm{SE} = \sqrt{\mathrm{MSE}} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(A-P)^2} \tag{10}$$

2.5 งานวิจัยที่เกี่ยวข้อง

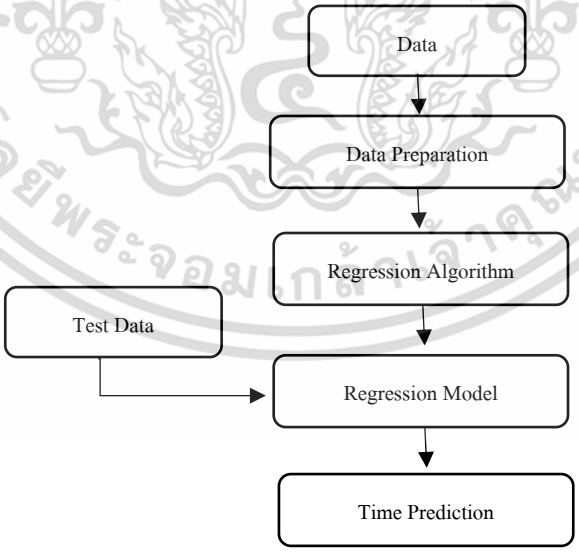
ในปี ค.ศ. 2015 A. Gal และคณะ [7] นำเสนอการสร้างแบบจำลองในการทำนายระยะเวลาในการเดินทางของรถบัส ด้วยวิธี Snapshot predictor และด้วยอัลกอริทึม เช่น Random Forest, Extremely Randomized Tree, AdaBoost, Gradient boost และ GBLAD อีกทั้งยังสร้างแบบจำลองโดยใช้การ-บูรณาการระหว่างวิธี Snapshot predictor กับอัลกอริทึมดังกล่าว ซึ่งแบบจำลองที่สามารถทำนายได้แม่นยำมากที่สุดเป็นแบบจำลองที่สร้างจากการผสมผสานกันระหว่าง Snapshot predictor and และ GBLAD และในปีเดียวกัน Y. Zhang และ A. Haghani [8] ได้ทำการเปรียบเทียบประสิทธิภาพของแบบจำลองในการทำนายระยะเวลาการเดินทางบนถนนที่สร้างจากอัลกอริทึม 3 ตัว ประกอบไปด้วย ARIMA, Random Forest และ Gradient Boost จากการวัดประสิทธิภาพแบบจำลองที่สร้างจาก Gradient Boost, ARIMA และ Random Forest ค่าเฉลี่ยเปอร์เซ็นต์ความคลาดเคลื่อนของแบบจำลองทั้ง 3 มีความใกล้เคียงกันมากในช่วงเวลาที่ไม่เร่งด่วน สำหรับช่วงเวลาที่เร่งด่วน แบบจำลองที่สร้างจาก Gradient Boost สามารถทำนายออกมาได้แม่นยำกว่า

ต่อมาในปี ค.ศ. 2017 A. Dorogush และคณะ [9] นำเสนอประสิทธิภาพการทำงานของอัลกอริทึม CatBoost โดยทำการเปรียบเทียบการทำงานกับไลบรารีของ Gradient Boosting ผลที่ได้คืออัลกอริทึม CatBoost มีประสิทธิภาพการทำงานดีกว่า ไม่ว่าจะเป็นในด้านของระยะเวลาที่ใช้ในการฝึกสอนหรือการใช้ทรัพยากรของตัวเครื่อง

ในปี ค.ศ. 2019 J. Cheng และคณะ [10] ได้นำเสนองานวิจัยเกี่ยวกับ การสร้างแบบจำลองในการทำนายระยะเวลาในการเดินทางบนถนน โดยใช้ Gradient Boosting Decision Tree แล้วนำไปเปรียบเทียบประสิทธิภาพการทำงานกับแบบจำลองที่สร้างจากอัลกอริทึมอื่น เช่น Backpropagation Neural Network และ Support Vector Machine (SVM) ซึ่งจากการวัดประสิทธิภาพแบบจำลองด้วยการหาค่าเฉลี่ยเปอร์เซ็นต์ความคลาดเคลื่อนสมบูรณ์ (MAPE) พบว่าแบบจำลองที่สร้างจาก Gradient Boosting Decision Tree มีค่า MAPE น้อยกว่าแบบจำลองที่สร้างจาก BP Neural Network และ SVM แสดงให้เห็นว่าแบบจำลองที่สร้างจาก Gradient Boosting Decision Tree สามารถทำนายระยะเวลาการเดินทางได้ใกล้เคียงมากที่สุด

3. วิธีการทดลอง

การดำเนินงานวิจัยมุ่งเน้นที่การค้นหาเทคนิคทางด้านการเรียนรู้ของเครื่อง เพื่อสร้างแบบจำลองในการทำนายระยะเวลาที่คาดว่าเรือโดยสารสาธารณะจะเข้ามาจอดเทียบท่า เพื่อค้นหาอัลกอริทึมที่เหมาะสมที่สุดที่สามารถทำนายค่าได้ใกล้เคียง รวมถึงศึกษาวิธีลดคุณลักษณะ และการวัดประสิทธิภาพที่เหมาะสม โดยเริ่มจากการจัดการข้อมูลด้วยกระบวนการเตรียมข้อมูล (Data Preparation) สร้างแบบจำลองในการทำนายระยะเวลาที่เรือโดยสารสาธารณะจะเข้ามาเทียบท่าในแต่ละท่าเรือ และทำการทดสอบประสิทธิภาพของแบบจำลอง แสดงดังรูปที่ 3 ซึ่งมีรายละเอียดการดำเนินการดังนี้



รูปที่ 3. ขั้นตอนการสร้างแบบจำลอง

3.1 การเตรียมข้อมูล

เนื่องจากข้อมูลที่มีอยู่เป็นจำนวนมาก ทำให้มีข้อมูลหลายส่วนที่ไม่จำเป็นในการนำมาประมวลผล เนื่องจากไม่มีความสัมพันธ์กันดังนั้นการคัดกรองข้อมูลจึงเป็นสิ่งที่จำเป็น โดยต้องทำให้ข้อมูลมีขนาดเล็กลง ให้เหลือแต่ข้อมูลที่จำเป็นต่อการวิเคราะห์ เพื่อให้สอดคล้องกับการนำไปสร้างเป็นแบบจำลอง สำหรับการทำนายระยะเวลาที่เรือโดยสารสาธารณะจะเข้ามาจอดเทียบท่า โดยรายละเอียดของข้อมูลการเดินเรือของเรือโดยสารสาธารณะ แสดงดังตารางที่ 1

3.1.1 การคัดเลือกข้อมูล (Data Selection)

เป็นขั้นตอนที่ต้องวิเคราะห์ และมองหาความสัมพันธ์กันของข้อมูลของเรือโดยสารว่า มีข้อมูลตัวใดบ้างที่สัมพันธ์กัน และสัมพันธ์กันอย่างไร หรือสามารถนำไปใช้ทำอะไรต่อไปได้

3.1.2 การแปลงข้อมูล (Data Transformation)

จากการตรวจสอบข้อมูลเรือที่ได้จากอุปกรณ์ IoT พบว่า ยังมีข้อมูลบางอย่างที่ยังไม่สามารถนำไปใช้งานได้ เนื่องจากข้อมูลอยู่ในรูปที่ยังไม่เหมาะสมที่จะนำไปใช้งาน จึงได้ทำการแปลงค่าข้อมูลเพื่อให้สามารถนำไปใช้ตรวจสอบ และวิเคราะห์หาความสัมพันธ์กับระยะเวลาที่ใช้ในการเดินเรือของเรือโดยสารได้ง่ายและสะดวกมากยิ่งขึ้น

ตารางที่ 1. ข้อมูลการเดินเรือของเรือโดยสารสาธารณะ

ข้อมูล	คำอธิบาย	ชนิด
midlat	ข้อมูลตำแหน่งละติจูด (Latitude) ของเรือโดยสาร ณ ขณะนั้น	float
midlon	ข้อมูลตำแหน่งลองจิจูด (Longitude) ของเรือโดยสาร ณ ขณะนั้น	float
device	ข้อมูลชื่อของอุปกรณ์ที่ติดตั้งอยู่บนเรือโดยสารแต่ละลำ	string
ts	ข้อมูลระบุ timestamp (วันที่/เดือน/ปี และเวลา) ของเรือโดยสาร ณ ขณะนั้น	datetime

3.1.3 การสร้างข้อมูลใหม่ (Construct Data)

จากที่ได้ทำการวิเคราะห์ข้อมูลที่มีอยู่แล้วพบว่า ข้อมูลที่เรือส่งกลับมาเพียงอย่างเดียวนั้นไม่สามารถทำให้ทราบได้ถึงระยะเวลาของการเดินเรือทั้งหมด เนื่องจากข้อมูลเหล่านี้ไม่ได้แสดงถึงความสัมพันธ์ระหว่างตำแหน่ง และระยะเวลาการเดินเรือโดยตรง

3.1.4 การทำความสะอาดข้อมูล (Data Cleaning)

จากการตรวจสอบข้อมูล พบว่าข้อมูลส่วนใหญ่อยู่ในสภาพสมบูรณ์ แต่อย่างไรก็ตามยังมีข้อมูลบางส่วนที่ขาดหายไปและมีค่าที่ผิดปกติที่ไม่สอดคล้องกับความเป็นจริง รวมถึงมีข้อมูลที่ไม่มีความสัมพันธ์กับข้อมูลตัวอื่น ๆ ซึ่งไม่จำเป็นต่อการนำมาใช้งาน

3.2 การสร้างแบบจำลอง (Modelling)

สร้างแบบจำลองการถดถอย ในการทำนายระยะเวลาที่เรือโดยสารจะเข้ามาจอดเทียบท่า โดยการนำข้อมูลที่ผ่านกระบวนการเตรียมข้อมูลมาทำการแบ่งเป็นข้อมูล midlat (ค่าตำแหน่งละติจูดของเรือ) midlon (ค่าตำแหน่งลองจิจูดของเรือ) และ ts (ระยะเวลาการเดินเรือ) โดยที่ข้อมูลการเดินเรือที่นำมาใช้ในการสร้างแบบจำลอง จะใช้ข้อมูลการเดินเรือของเที่ยวการเดินทางเดียว เนื่องด้วยทิศทางการเดินเรือของเที่ยวการเดินเรือของเรือโดยสารทั้งสองเที่ยวนั้น เคลื่อนที่ไปในทิศทางที่ตรงข้ามกัน ทำให้เกิดการทับซ้อนกันของข้อมูลการเดินเรือ จึงทำการแบ่งข้อมูลเที่ยวการเดินเรือทั้ง 2 เที่ยวออกจากกันเป็น 2 ส่วน มีสัดส่วน 70:30 ซึ่งข้อมูลส่วนแรกเป็นข้อมูลสำหรับฝึกสอนในการเรียนรู้ของแบบจำลอง และส่วนที่สองสำหรับการทดสอบ

3.3 การวัดประสิทธิภาพแบบจำลอง

วิธีการทดสอบเพื่อเปรียบเทียบประสิทธิภาพการทำงานของแบบจำลองการถดถอย ที่สร้างจากแต่ละอัลกอริทึมสามารถพิจารณาได้จากค่าความคลาดเคลื่อนที่เกิดขึ้น โดยที่ความถูกต้องของค่าที่ทำนายได้จะมีค่ามากหรือน้อยขึ้นอยู่กับค่าความคลาดเคลื่อนของการทำนาย ซึ่งเป็นผลต่างของค่าจริงกับค่าที่ทำนาย

4. ผลการทดลองและวิจารณ์

เมื่อเสร็จสิ้นกระบวนการเตรียมข้อมูล จะได้ข้อมูลที่จะนำไปสร้างเป็นแบบจำลอง ประกอบไปด้วยข้อมูลระบุตำแหน่ง เป็นค่าละติจูด และค่าลองจิจูด ของเรือโดยสาร และข้อมูลระยะเวลาการเดินเรือจากท่าเรือเริ่มต้นไปยังท่าเรือสุดท้าย ซึ่งกรณีศึกษาคือ คลองภาษีเจริญ จังหวัดกรุงเทพมหานคร โดยมีจำนวนข้อมูลเมื่อเสร็จสิ้นกระบวนการทั้งหมด 136,460 รายการ แบ่งข้อมูลตามเที่ยวการเดินเรือออกมาได้เป็นข้อมูลของเที่ยวการเดินเรือจากท่าเรือต้นทาง ไปยังท่าเรือปลายทาง มีจำนวนข้อมูลทั้งหมด 45,261 รายการ และข้อมูลของเที่ยวการเดินเรือจากท่าเรือปลายทางไปยังท่าเรือต้นทาง ซึ่งมีจำนวนข้อมูลทั้งหมด 91,199 รายการ

การสร้างแบบจำลองจะใช้ข้อมูลของตำแหน่งละติจูด ลองจิจูด และระยะเวลาการเดินเรือ ข้อมูลแบ่งเป็น ข้อมูลในการสร้างชุดข้อมูลฝึกสอน และชุดข้อมูลทดสอบ โดยแบ่งข้อมูลออกเป็นสัดส่วน 70:30 ทำการสร้างแบบจำลองด้วยอัลกอริทึมการถดถอยทั้งหมด 6 อัลกอริทึม ประกอบไปด้วย Linear Regression, Random Forest, Gradient Boosting, eXtreme Gradient Boosting, Light Gradient Boosting และ CatBoost ผ่านโปรแกรม Jupyter Notebook

สำหรับการวัดประสิทธิภาพของแบบจำลองที่สร้างจากอัลกอริทึมการถดถอย สามารถวัดได้จากค่ารากของค่าความคลาดเคลื่อนกำลังสองเฉลี่ย (Root Mean Square Error : RMSE) โดยค่าความ

คลาดเคลื่อนเฉลี่ย จะขึ้นอยู่กับค่าความต่างระหว่างค่าที่แบบจำลองทำนายได้ กับข้อมูลที่นำไปสร้างแบบจำลอง

ตารางที่ 2 เป็นการแสดงผลการวัดประสิทธิภาพแบบจำลอง จากตารางจะเห็นได้ว่า แบบจำลองของการเดินเรือทั้ง 2 เที่ยวการเดินเรือ จากต้นทาง - ปลายทาง ที่สร้างจากอัลกอริทึม CatBoost มีค่า RMSE ในการเดินเรือจากท่าเรือต้นทางไปยังท่าเรือปลายทางอยู่ที่ 88.25 วินาที และในเที่ยวการเดินเรือปลายทางไปยังต้นทาง มีค่า RMSE ในการเดินเรืออยู่ที่ 90.06 วินาที เมื่อเทียบกับชุดข้อมูลการเดินเรือที่ใช้ทดสอบ

ตารางที่ 2. ผลการวัดประสิทธิภาพแบบจำลอง

เที่ยวการเดินเรือ	อัลกอริทึม	RMSE
ต้นทาง – ปลายทาง	Linear Regression	101.6627
	Random Forest	88.9860
	Gradient Boosting	88.2753
	eXtreme Gradient Boosting	88.2877
	Light Gradient Boosting	88.7557
	Catboost	88.2531
ปลายทาง – ต้นทาง	Linear Regression	103.5393
	Random Forest	91.1341
	Gradient Boosting	90.2080
	eXtreme Gradient Boosting	90.0723
	Light Gradient Boosting	90.2512
	Catboost	90.0687

จากนั้นนำค่าระยะเวลาที่แบบจำลองที่สร้างจากอัลกอริทึม CatBoost ทำนายมาทดสอบโดยการเทียบกับระยะเวลาในการเดินเรือ ระหว่างท่าเรือของเรือโดยสารโดยเฉลี่ยอ้างอิงจากข้อมูลการเดินเรือของเรือโดยสารสาธารณะ ตั้งแต่วันที่ 9-12 สิงหาคม พ.ศ. 2562 เวลา 9.00 -19.00 น. ซึ่งจากการตรวจสอบโดยอิงจากข้อมูลเดินเรือ ปัจจัยในด้านของวัน และเวลาเดินเรือ ไม่มีผลต่อระยะเวลาการเดินเรือของเรือโดยสารสาธารณะ

ซึ่งจากการนำระยะเวลาการเดินเรือที่แบบจำลองทำนายได้กับระยะเวลาการเดินเรือระหว่างท่าโดยเฉลี่ยทั้งในเที่ยวการเดินเรือจากท่าต้นทาง–ท่าปลายทาง และจากท่าปลายทาง-ท่าต้นทาง มาทำการเปรียบเทียบกัน เพื่อทดสอบว่าแบบจำลองสามารถทำนายระยะเวลาที่เรือโดยสารจะเข้ามาเทียบท่าได้ใกล้เคียงมากน้อยเพียงใด ผลการเปรียบเทียบที่ได้แสดงดังตารางที่ 3 และตารางที่ 4 ตามลำดับ โดยผลลัพธ์ที่ได้จากการนำระยะเวลาการเดินเรือโดยเฉลี่ยของเรือโดยสารมาเทียบกับระยะเวลาที่แบบจำลองทำนายได้ มีค่าความคลาดเคลื่อนค่อนข้างต่ำ ซึ่งแสดงให้เห็นว่าระยะเวลาการเดินเรือที่แบบจำลองที่สร้างจากอัลกอริทึม CatBoost สามารถทำนายได้ มีความใกล้เคียงกับระยะเวลาการเดินเรือเฉลี่ย ณ ช่วงวันเวลาดังกล่าว

ตารางที่ 3. ผลการนำระยะเวลาการเดินเรือที่ทำนายได้ ทดสอบในเที่ยวท่าต้นทางถึงท่าปลายทาง

ตำแหน่งต้นทาง	ตำแหน่งปลายทาง	ระยะเวลาเดินเรือโดยเฉลี่ย (วินาที)	แบบจำลองทำนายได้ (วินาที)	ระยะเวลาเดินเรือที่คลาดเคลื่อน (วินาที)
0	1	69.61	56.88	+ 12.73
1	2	82.39	95.14	- 12.75
2	3	182.31	171.60	+ 10.71
3	4	112.27	133.20	- 20.93
4	5	287.17	278.36	+ 8.81
5	6	178.97	183.70	- 4.73
6	7	193.20	196.95	- 3.75
7	8	39.81	33.25	+ 6.56
8	9	55.54	65.86	- 10.32
9	10	70.65	75.54	- 4.89
10	11	188.52	157.12	+ 31.40
11	12	43.98	62.72	- 18.74
12	13	71.94	65.38	+ 6.56
13	14	356.92	359.69	- 2.77
14	15	101.93	111.07	- 10.86
15	16	98.12	92.29	+ 5.83
16	17	268.88	280.08	- 11.20
17	18	256.80	273.30	- 16.50
18	19	111.13	105.38	+ 5.75

5. สรุปผลการทดลอง

งานวิจัยนี้นำเสนอการสร้างแบบจำลอง เพื่อทำนายระยะเวลาในการเข้าเทียบท่าของเรือโดยสาร ซึ่งสามารถทำนายระยะเวลาในการเข้าเทียบท่าของเรือโดยสารได้ใกล้เคียงกับสภาพความเป็นจริงมากที่สุด โดยการนำข้อมูลของเรือโดยสารมาเข้าสู่กระบวนการเตรียมข้อมูล ซึ่งแบ่งออกได้เป็นข้อมูลการเดินเรือในเที่ยวการเดินเรือต้นทาง - ปลายทาง และเที่ยวการเดินเรือปลายทาง - ต้นทาง จากนั้นนำข้อมูลการเดินเรือไปสร้างเป็นแบบจำลอง โดยใช้อัลกอริทึมต่าง ๆ ทั้ง 6 อัลกอริทึม ประกอบไปด้วย Linear Regression, Random Forest, Gradient Boost, eXtreme Gradient Boost, Light Gradient Boost และ CatBoost ซึ่งเมื่อนำแบบจำลองไปทดสอบประสิทธิภาพโดยการหาค่า RMSE พบว่า แบบจำลองที่สร้างจากอัลกอริทึม CatBoost มีค่า RMSE ต่ำที่สุดทั้ง 2 เที่ยวการเดินเรือ โดยในเที่ยวการเดินเรือท่าเรือต้นทาง - ท่าเรือปลายทาง มีค่า RMSE อยู่ที่ 88.25 วินาที และในเที่ยวการเดินเรือท่าเรือปลายทาง - ท่าเรือต้นทาง มีค่า RMSE อยู่ที่ 90.06 วินาที

ตารางที่ 4. ผลการนำระยะเวลาการเดินเรือที่ทำนายได้ ทดสอบในเที่ยวท่าปลายทางถึงท่าต้นทาง

ตำแหน่งต้นทาง	ตำแหน่งปลายทาง	ระยะเวลาเดินเรือโดยเฉลี่ย (วินาที)	แบบจำลองทำนายได้ (วินาที)	ระยะเวลาเดินเรือที่คลาดเคลื่อน (วินาที)
0	1	72.89	69.86	+ 3.03
1	2	85.40	88.25	- 2.85
2	3	123.61	112.47	+ 11.14
3	4	154.88	185.63	- 30.75
4	5	312.36	284.58	+ 27.78
5	6	191.05	198.03	- 6.95
6	7	204.75	188.51	+ 16.24
7	8	38.54	45.72	- 7.18
8	9	65.64	61.34	+ 4.30
9	10	83.62	71.35	+ 12.27
10	11	176.84	195.37	- 18.53
11	12	55.60	62.29	- 6.69
12	13	69.3	72.88	- 3.58
13	14	370.96	367.30	+ 3.66
14	15	85.66	96.96	- 11.30
15	16	124.61	118.87	+ 5.74
16	17	270.78	277.10	- 6.32
17	18	268.65	261.25	+7.40
18	19	93.39	111.94	- 18.55

ผลลัพธ์ที่ได้จากการนำแบบจำลองมาทดสอบกับข้อมูลการเดินเรือของเรือโดยสาร พบว่าค่าความคลาดเคลื่อนที่เกิดจากการนำค่าระยะเวลาการเดินเรือที่แบบจำลองที่สร้างจากอัลกอริทึม CatBoost ทำนายได้เทียบกับค่าระยะเวลาการเดินเรือของเรือโดยสารทั้งในเที่ยวการเดินเรือจากท่าเรือต้นทางไปยังท่าเรือปลายทาง และจากท่าเรือปลายทางไปยังท่าเรือต้นทาง มีค่าความคลาดเคลื่อนต่ำจัดว่าอยู่ในเกณฑ์ที่ดี ซึ่งแสดงให้เห็นว่า ระยะเวลาการเดินเรือที่แบบจำลองทำนายได้นั้น มีความใกล้เคียงกับระยะเวลาการเดินเรือและสามารถนำแบบจำลองไปใช้งานได้จริง

อย่างไรก็ตาม เนื่องจากแบบจำลองจำเป็นต้องใช้ข้อมูลตำแหน่งของเรือ (ละติจูดและลองจิจูด) เพื่อใช้ในการทำนาย ทำให้แบบจำลองจะไม่สามารถทำนายค่าระยะเวลาการเดินเรือในการเข้าเทียบท่าที่ถูกต้องได้ ในกรณีที่เรือโดยสารจอดนิ่งอยู่กับที่เป็นเวลานาน ซึ่งสาเหตุที่ทำให้เรือหยุดนิ่งอาจเกิดจากเหตุขัดข้องกับเครื่องยนต์ของเรือโดยสาร หรือเกิดเหตุขัดข้องกับอุปกรณ์ IoT บนเรือ ที่ทำให้ไม่สามารถส่งตำแหน่งของเรือโดยสารได้ หรืออาจเกิดจากกรณีที่เรือโดยสารอยู่นอกเหนือเส้นทางการเดินเรือของเรือโดยสารสาธารณะ

กิตติกรรมประกาศ

ผู้วิจัยขอขอบคุณ คุณวิภู ประจันตะเสน และ คุณสิงหา วงษ์ดีไทย สำหรับการสนับสนุนข้อมูลตัวอย่างที่ใช้ในการวิจัยนี้

เอกสารอ้างอิง (References)

[1] Vijite, P. 2018. Machine Learning. Available at: https://medium.com/coeffest/table-of-contents-machine-learning-theory-103315c4afa9. Retrieved January 17, 2019.

[2] Gallo, A. 2015. A Refresher on Regression Analysis. Available at: https://hbr.org/2015/11/a-refresher-on-regression-analysis. Retrieved February 5, 2019.

[3] Singh, A. 2018. A Comprehensive Guide to Ensemble Learning (with Python codes). Available at: https://www.analyticsvidhya.com/blog/2018/06/comprehensive-guide-for-ensemble-models/. Retrieved February 13, 2019.

[4] Tseng, G. 2018. Gradient Boosting and XGBoost. Available at: https://medium.com/@gabrieltseng/. Retrieved February 20, 2019.

[5] Ray, S. 2017. CatBoost: A machine learning library to handle categorical (CAT) data automatically. Available at: https://www.analyticsvidhya.com/blog/2017/08/catboost -automated-categorical-data/. Retrieved March 19, 2019.

[6] Smriti, S. 2019. What is Mean Squared Error, Mean Absolute Error, Root Mean Squared Error and R Squared?. Available at: https://www.studytonight.com/post/what-is-mean-squared-error-mean-absolute-error-root-mean-squared-error-and-r-squared. Retrieved April 4, 2019.

[7] Gal, A., Mandelbaum, A., Schnitzler, F., Senderovich, A. and Weidlich, M. 2017. Traveling time prediction in scheduled transportation with journey segments. *Inf. Syst.*, (64), 266–280.

[8] Zhang, Y. and Haghani, A. 2015. A gradient boosting method to improve travel time prediction. *Transportation Research Part C : Emerging Technologies*, (58), 308-324.

[9] Dorogush, A.V., Ershov, V. and Gulin, A. 2018. CatBoost: Gradient boosting with categorical features support. In ML Systems Workshop at NIPS.

[10] Cheng, J., Li, G. and Chen, X. 2019. Research on Travel Time Prediction Model of Freeway Based on Gradient Boosting Decision Tree. *IEEE Access*, (7), 7466-7480.