การวิเคราะห์พฤติกรรมในระบบขนส่ง
เพื่อการตรวจจับเหตุการณ์ผิดปกติบนถนนทางหลวง
TRANSPORTATION BEHAVIOR ANALYSIS
FOR INCIDENT DETECTION IN HIGHWAY

คมนภัค พรทรัพย์ไพศาล

ศุษิล ภาณุวรรณ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2561

การวิเคราะห์พฤติกรรมในระบบขนส่ง
เพื่อการตรวจจับเหตุการณ์ผิดปกติบนถนนทางหลวง

TRANSPORTATION BEHAVIOR ANALYSIS
FOR INCIDENT DETECTION IN HIGHWAY

คมนภัค พรทรัพย์ไพศาล

ศุษิล ภาณุวรรณ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2561

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปี การศึกษา 2561
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เรื่อง การวิเคราะห์พฤติกรรมในระบบขนส่งเพื่อการตรวจจับเหตุการณ์ผิดปกติบนถนนทางหลวง
TRANSPORTATION BEHAVIOR ANALYSIS FOR INCIDENT DETECTION IN HIGHWAY

ผู้จัดทำ

1. นายคมนภัค พรทรัพย์ไพศาล รหัสนักศึกษา 58010148
2. นายศุบิล ภาณุวรรณ รหัสนักศึกษา 58011262

____________________ อาจารย์ที่ปรึกษา

(ผศ.อัครเดช วัชระภูพงษ์)

____________________ อาจารย์ที่ปรึกษาร่วม

(ผศ.ดร.ธนัญชัย ตรีภาค)

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การวิเคราะห์พฤติกรรมในระบบขนส่ง
เพื่อการตรวจจับเหตุการณ์ผิดปกติบนถนนทางหลวง

นายคมนภัค	พรทรัพย์ไพศาล	58010148
นายศุขิล	ภาณุวรรณ	58011262
ผศ.อัครเดช	วัชระภูพงษ์	อาจารย์ที่ปรึกษา
ผศ.ดร.ธนัญชัย	ตรีภาค	อาจารย์ที่ปรึกษา

ปีการศึกษา 2561

บทคัดย่อ

การเดินทางและการคมนาคมขนส่งระหว่างจังหวัดหรือระหว่างประเทศนั้นมีได้หลากหลายเส้นทาง อาทิเช่น รถยนต์ส่วนบุคคล, รถบริการสาธารณะ, รถไฟ, เรือโดยสาร, เครื่องบินพาณิชย์, ฯลฯ ซึ่งในการคมนาคมทางบกทั้งการเดินทางหรือการขนส่งสินค้าที่ไม่รวมทางรถไฟ ล้วนต้องใช้ถนนทางหลวงทั้งสิ้นไม่ว่าจะเป็น ทางหลวงพิเศษ (Superhighway) จำพวกทางด่วนต่าง ๆ หรือทางหลวงท้องถิ่น (Local road) ที่เป็นถนนในเมืองทั่วไป ทว่าทางหลวงพิเศษที่เชื่อมระหว่างจังหวัดหลายแห่งหรือแม้กระทั่งทางหลวงท้องถิ่นในตัวจังหวัดบางแห่งเองกลับไม่มีระบบที่สามารถตรวจจับเหตุการณ์ผิดปกติบนท้องถนนต่าง ๆ เช่น อุบัติเหตุ, รถติดโดยไม่ทราบสาเหตุ, ทางขาด, ฯลฯ ได้อย่างแม่นยำและมีประสิทธิภาพ ยังคงต้องใช้คนที่อยู่ในเหตุการณ์หรือเดินทางผ่านมาในบริเวณที่เกิดเหตุเป็นผู้แจ้งกลับไปยังหน่วยงานที่เกี่ยวข้องอยู่

ทางผู้จัดทำโครงงานจึงมีความคิดว่าจะนำ Machine Learning เข้ามาช่วยในการพัฒนาระบบเพื่อตรวจจับความผิดปกติที่เกิดขึ้นบนท้องถนน โดยอ้างอิงจากข้อมูล GPS log ของรถยนต์หรือรถบรรทุกที่วิ่งอยู่บนทางหลวงต่าง ๆ

เพื่อเป็นการศึกษาขั้นตอนและวิธีการในการทำวิจัย (Research Method) ในเรื่องของ Machine Learning เพื่อแก้ปัญหาเกี่ยวกับการคมนาคมดังที่กล่าวข้างต้นทางผู้จัดทำจึงเริ่มจากเรื่องของการตรวจจับความผิดปกติบน Computer Network Traffic ซึ่งทางผู้จัดทำโครงงานที่อยู่ในสายงานคอมพิวเตอร์มีความรู้พื้นฐานอยู่ระดับหนึ่ง นอกจากนั้นการนำ Machine Learning มาใช้ตรวจจับความผิดปกติบน Computer Network Traffic ดังกล่าวยังมีงานวิจัยรองรับหลายฉบับทำให้สามารถนำมาใช้อ้างอิงในการศึกษา Machine Learning เพื่อนำไปประยุกต์ใช้ได้ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Transportation Behavior Analysis for Incident Detection in Highway

Mr. Komnapak	Pornsupphisan	58010203
Mr. Susin	Phanuwan	58011221
Asst.Prof. Akkradach	Watcharapupong	Advisor
Asst.Prof.Dr. Thanunchai	Threepak	Advisor

Academic Year 2018

ABSTRACT

Transportation between provinces or country have many options these days e.g., personal car, public bus, train, passenger boat, commercial airplane, etc. Land transportation both traveling and cargo's transporting excluding train, all have to use highway whether superhighway or local highway (local road) for the trip. However, many of the superhighways connecting between provinces or even some of the local roads don't have efficient incident detection system for automatically detect abnormally or incident on the road, such as car accident, traffic jam with unknown causes, damaged road, etc. Reporting about incident to the related organization still have to depend on witness or people passing by.

We have an idea about using Machine Learning method to help develop incident detection system base on GPS log of public transportation vehicle

For purpose of learning about research method and how to apply Machine Learning to solve transportation problem as mentioned before, we start from learning and develop model for anomaly detection on computer traffic network. Because we are in field of computer and have average base knowledge about computer network. Furthermore, using Machine Learning for develop anomaly detection on computer network traffic have many study that we can learn from and use as reference.

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานเล่มนี้สำเร็จได้ด้วยความกรุณาของอาจารย์อัครเดช วัชระภูพงษ์และอาจารย์ธนัญชัย ตรีภาค อาจารย์ประจำภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ซึ่งได้ให้คำแนะนำและช่วยเหลือการจัดทำโครงงานทุกอย่างจนลุล่วงไปได้ด้วยดี ผู้วิจัยขอกราบขอบพระคุณเป็นอย่างสูงมา ณ ที่นี้

สุดท้ายนี้ขอกราบขอบพระคุณผู้ที่เกี่ยวข้องทุกท่านที่ได้ให้การสนับสนุนและให้กำลังใจที่ดีแก่ผู้วิจัยตลอดมา

คมนภัค พรทรัพย์ไพศาล

ศุษิล ภาณุวรรณ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย I

บทคัดย่อภาษาอังกฤษ II

สารบัญ IV

สารบัญตาราง VI

สารบัญรูป VII

บทที่ 1 บทนำ 1

1.1 ความเป็นมาของปัญหา 1

1.2 แนวคิดของโครงงาน 1

1.3 วัตถุประสงค์ของโครงงาน 2

1.4 ขอบเขตของโครงงาน 2

1.5 ประโยชน์ที่คาดว่าจะได้รับ 2

บทที่ 2 ทฤษฎีและเครื่องมือที่เกี่ยวข้อง 3

2.1 ทฤษฎีที่เกี่ยวข้อง 3

2.1.1 Machine Learning 3

2.1.2 artificial neural network 5

2.2 เครื่องมือที่เกี่ยวข้อง 13

2.2.1 TensorFlow 13

2.2.2 Python 13

2.2.3 Keras 15

บทที่ 3 การออกแบบและการพัฒนา 17

3.1 PROBLEM STATEMENT 17

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดักแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

3.2 NORM SPEED/UNITS MODEL 17
3.2.1 Cell Data 18
3.2.2 Preprocessing 19
3.2.3 Training Phase 19
3.2.4 Testing Phase 20

บทที่ 4 การดำเนินงานและการทดลอง 23
4.1 ขั้นตอนการทดลอง 23
4.1.1 การนำข้อมูลเข้าและ การประมวลผลเบื้องต้น 23
4.1.2 การเทรนโมเดล 25
4.1.3 การทดสอบโมเดล 27
4.1.4 การตรวจหาความผิดปกติ 28
4.2 ผลการทดลอง 31
4.2.1 ผลจากโมเดล Speed และ Units 31
4.2.2 ผลการตรวจหาความผิดปกติ 39

บทที่ 5 สรุปผลและปัญหาที่พบ 41

บรรณานุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตาราง หน้า

1 รายละเอียดของข้อมูล CELL DATA 19

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูป หน้า

2.1 MACHINE LEARNING 3
2.2 SUPERVISED LEARNING 4
2.3 UNSUPERVISED LEARNING 5
2.4 REINFORCEMENT LEARNING 5
2.5 MODEL OF ARTIFICIAL NEURAL NETWORK 6
2.6 MODEL OF MULTI-LAYER PERCEPTRON 7
2.7 MODEL OF HOPFIELD NETWORK 8
2.8 MODEL OF SINGLE-LAYER PERCEPTRON 9
2.9 MODEL OF RECURRENT NETWORK 10
2.10 MODEL OF SELF-ORGANIZING MAP 10
2.11 MODEL OF BOLTZMANN MACHINE 11
2.12 MODEL OF ASSOCIATIVE NEURAL NETWORK 11
2.13 MODEL OF INSTANTANEOUSLY TRAINED NETWORKS 12
2.14 MODEL OF SPIKING NEURAL NETWORKS 12
2.15 TENSORFLOW 13
2.16 PYTHON 14
2.17 KERAS 15
3.1 BLOCK DIAGRAM FOR NN MODEL 18
3.2 เวคเตอร์ของเอาท์พุตในเลเยอร์ชั้นถัดไป 20
3.3 RELU 20
3.4 INCIDENT DETECTION DIAGRAM 21
3.5 สมการหาค่าสัมประสิทธิ์ 22
4.1 ตัวอย่างโปรแกรมการอ่านค่าจากไฟล์ .CSV 24
4.2 ตัวอย่างโปรแกรมในการเทรนโมเดล 26
4.3 ตัวอย่างโปรแกรมในการทดสอบโมเดล 27
4.4 ตัวอย่างโปรแกรมในการทดสอบโมเดล 28
4.5 ตัวอย่างโปรแกรมในการทดสอบโมเดลเพื่อตรวจหาความผิดปกติ 29
4.6 ตัวอย่างโปรแกรมในการทดสอบโมเดลเพื่อตรวจหาความผิดปกติ 30

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูป **หน้า**

4.7 กราฟฮีทแมพค่าเฉลี่ยความเร็วรถของถนนทางหลวงหมายเลข 2 ขาเข้าภายในเดือนกรกฎาคม 2561 31
4.8 กราฟฮีทแมพค่าเฉลี่ยความเร็วรถของถนนทางหลวงหมายเลข 2 ขาเข้าภายในเดือนกรกฎาคม 2561 32
4.9 กราฟฮีทแมพค่าเฉลี่ยจำนวนรถของถนนทางหลวงหมายเลข 2 ขาเข้าภายในเดือนกรกฎาคม 2561 33
4.10 กราฟฮีทแมพค่าเฉลี่ยจำนวนรถของถนนทางหลวงหมายเลข 2 ขาเข้าภายในเดือนกรกฎาคม 2561 34
4.11 กราฟฮีทแมพค่าเฉลี่ยความเร็วรถของถนนทางหลวงหมายเลข 2 ขาออกภายในเดือนกรกฎาคม 2561 35
4.12 กราฟฮีทแมพค่าเฉลี่ยความเร็วรถของถนนทางหลวงหมายเลข 2 ขาออกภายในเดือนกรกฎาคม 2561 36
4.13 กราฟฮีทแมพค่าเฉลี่ยจำนวนรถของถนนทางหลวงหมายเลข 2 ขาออกภายในเดือนกรกฎาคม 2561 37
4.14 กราฟฮีทแมพค่าเฉลี่ยจำนวนรถของถนนทางหลวงหมายเลข 2 ขาออกภายในเดือนกรกฎาคม 2561 38
4.15 กราฟแสดงค่าสัมประสิทธิ์ในวันที่ 30 กรกฎาคม 2561 39
4.16 ข่าวอุบัติเหตุจากเว็บจส.100ในวันที่ 30 กรกฎาคม 2561 39
4.17 กราฟแสดงค่าสัมประสิทธิ์ในวันที่ 22 กรกฎาคม 2561 40
4.18 ข่าวอุบัติเหตุจากเว็บจส.100ในวันที่ 22 กรกฎาคม 2561 40

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาของปัญหา

มนุษย์มีการสร้างสิ่งอำนวยความสะดวกสะบายให้กับตน โดยการสร้างเครื่องไม้เครื่องมือมาตั้งแต่สมัยยุคหินจนถึงปัจจุบัน ได้มีวิทยาการเพิ่มขึ้นมาก เช่นการสร้างเครื่องจักร หรือการสร้างระบบขนส่งสาธารณะขึ้นมา โดยสามารถขนส่งได้ทั้งทางบก ทางน้ำ และทางอากาศ

โดยระบบขนส่งสาธารณะ มีมาเพื่อตอบสนองต่อความต้องการความสะดวกสะบาย เนื่องจากปัญหาด้านการเดินทางที่ต้องใช้ระยะเวลานาน และความเหนื่อยล้าจากการเดินทาง โดยที่ระบบขนส่งสาธารณะนั้น ถือได้ว่าเป็นระบบที่ผ่านขั้นตอนของวิวัฒนาการมาแล้วเป็นเวลานานหลายยุคหลายสมัย เริ่มจาก การขนส่งสาธารณะครั้งแรกสุดที่พบในประวัติศาสตร์นั้น เริ่มขึ้นโดยชาวโรมัน ผู้ซึ่งก่อตั้งให้บริการระบบยานพาหนะที่รับจ้างขนส่งผู้คนและสัมภาระ ต่อมาได้มีการสร้างรถไฟที่ขับเคลื่อนด้วยเครื่องจักร ไอน้ำ และได้มีการพัฒนาระบบขนส่งสาธารณะขึ้นมาตลอดจนถึงปัจจุบัน

ระบบขนส่งสาธารณะในปัจจุบันนั้น ถ้าเทียบกับในอดีตมีการทำให้การขนส่งดีขึ้นมาก และทำให้มีการกระจายตัวของประชากรเพิ่มขึ้น แต่ก็ทำให้เกิดปัญหาหลายๆปัญหาตามมา เช่น อุบัติเหตุที่เกิดจากความประมาท หรือการจราจรที่ติดขัด ทว่าหลายๆครั้งที่เกิดเหตุการณ์ไม่พึงประสงค์ขึ้นบนท้องถนน กลับต้องรอคนที่อยู่ในการณ์เป็นผู้แจ้งเหตุไปยังหน่วยงานที่เกี่ยวข้องซึ่งอาจเกิดความล้าช้าในการเข้ามาแก้ปัญหาดังกล่าวได้

1.2 แนวคิดของโครงงาน

เนื่องจากข้อมูลของกรมขนส่งฯที่มีอยู่ มีลักษณะเป็น GPS log และความเร็วของรถขนส่งสาธารณะต่างๆที่วิ่งอยู่บนถนนทางหลวง ทำให้เป็นไปได้ยากที่จะใช้มนุษย์ในการวิเคราะห์หาความผิดปกติของการจราจรที่เกิดขึ้น ทางผู้จัดทำจึงมีความคิดในการนำ Maching Learning หรือการเรียนรู้ของเครื่องเข้ามาช่วยในการเรียนรู้รูปแบบที่ควรจะเป็นของการจราจรบนทางหลวงและสามารถบอกได้ว่ามีความผิดปกติของการจราจรเกิดขึ้นที่จุดไหน เมื่อไหร่เช่น หากว่าความเร็วของ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รถขนส่งสาธารณะในบริเวณหนึ่ง ๆลดลงต่ำกว่าที่ควรจะเป็นในเวลานาน ทำให้สามารถคาดเดาได้ว่าบริเวณดังกล่าวอาจเกิดอุบัติเหตุหรือเหตุการณ์ผิดปกติบางอย่างเกิดขึ้น ส่งผลให้การจราจรติดขัด

1.3 วัตถุประสงค์ของโครงงาน

นำ Machine Learning มาประยุกต์ใช้ในการสร้างโมเดลเพื่อใช้ตรวจจับเหตุการณ์ผิดปกติที่อาจจะเกิดขึ้นบนถนนทางหลวง โดยสามารถบอกได้ว่าเหตุการณ์ดังกล่าวเกิดขึ้นที่ช่วงเวลาไหน และระหว่างหลักกิโลเมตรใดบนถนนทางหลวง

1.4 ขอบเขตของโครงงาน

เพื่อศึกษาการประยุกต์ใช้ Machine Learning ในเรื่องของการตรวจจับเหตุการณ์อุบัติเหตุที่เกิดขึ้นบนถนนทางหลวงหมายเลข 2 (ถนนมิตรภาพ)

1.5 ประโยชน์ที่คาดว่าจะได้รับ

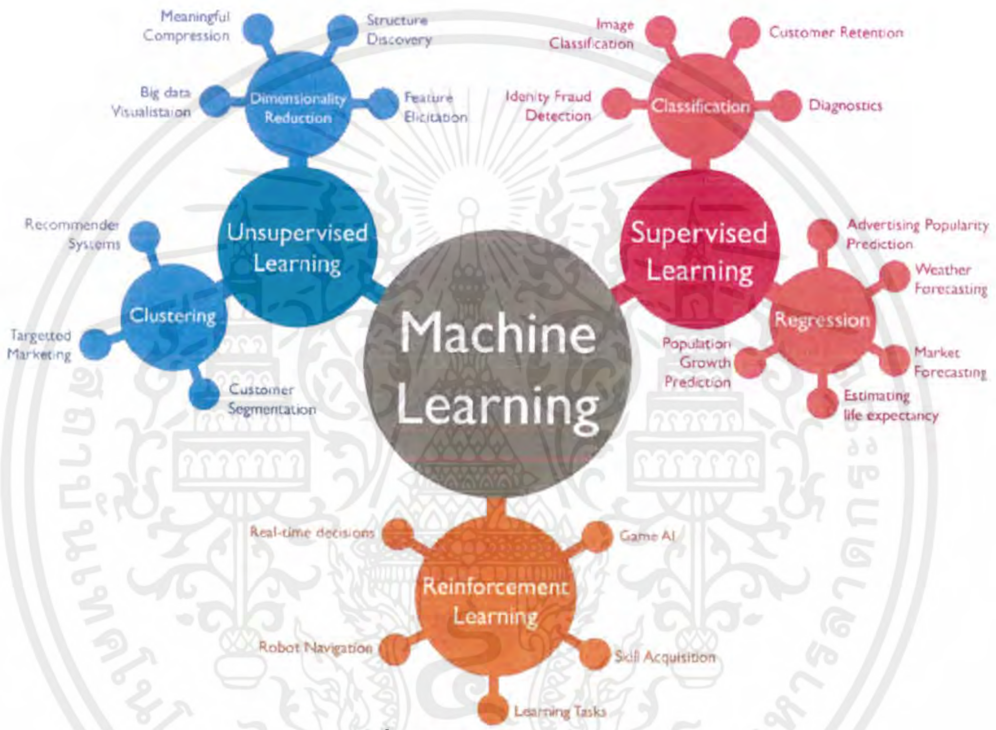
1. ได้รับความรู้เกี่ยวกับ Machine learning
2. เพื่อพัฒนาโมเดลเพื่อการตรวจจับเหตุการณ์ผิดปกติบนถนนทางหลวง
3. เพื่อปรับปรุงแก้ไขและประยุกต์ใช้อัลกอริทึมในการทำนาย (prediction) หรือจำแนกประเภท (classification) ให้มีประสิทธิภาพและความแม่นยำมากขึ้น
4. เข้าใจขั้นตอนและวิธีการทำงานวิจัย (research method)

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและเครื่องมือที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 Machine Learning



รูปที่ 2.1 Machine Learning

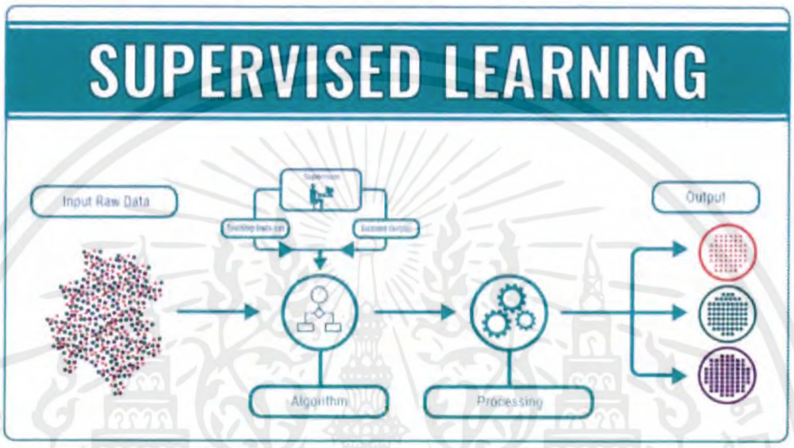
Machine Learning เป็นสาขาหนึ่งของปัญญาประดิษฐ์ที่พัฒนามาจากการศึกษาการรู้จำแบบ เกี่ยวข้องกับการศึกษาและการสร้างอัลกอริทึมที่สามารถเรียนรู้ข้อมูลและทำนายข้อมูลได้ อัลกอริทึมนั้นจะทำงานโดยอาศัยโมเดลที่สร้างมาจากชุดข้อมูลตัวอย่างขาเข้าเพื่อการทำนายหรือตัด สินใจในภายหลัง แทนที่จะทำงานตามลำดับของคำสั่งโปรแกรมคอมพิวเตอร์

การเรียนรู้ของเครื่องมีเกี่ยวข้องอย่างมากกับสถิติศาสตร์ เนื่องจากทั้งสองสาขาศึกษาการวิเคราะห์ข้อมูลเพื่อการทำนายเช่นกัน นอกจากนี้ยังมีความสัมพันธ์กับสาขาการหาค่าเหมาะที่สุดในทางคณิตศาสตร์ที่แงของวิธีการ ทฤษฎี และการประยุกต์ใช้ การเรียนรู้ของเครื่องสามารถนำไปประยุกต์ใช้งานได้หลากหมาย ไม่ว่าจะเป็นการกรองอีเมล์ขยะ การรู้จำตัวอักษร เครื่องมือค้นหา และคอมพิวเตอร์วิทัศน์

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Machine Learning สามารถแบ่งได้เป็น 3 ประเภท ตามประเภทของ "ข้อมูลฝึก" หรือ "ข้อมูลขาเข้า" ได้ดังนี้

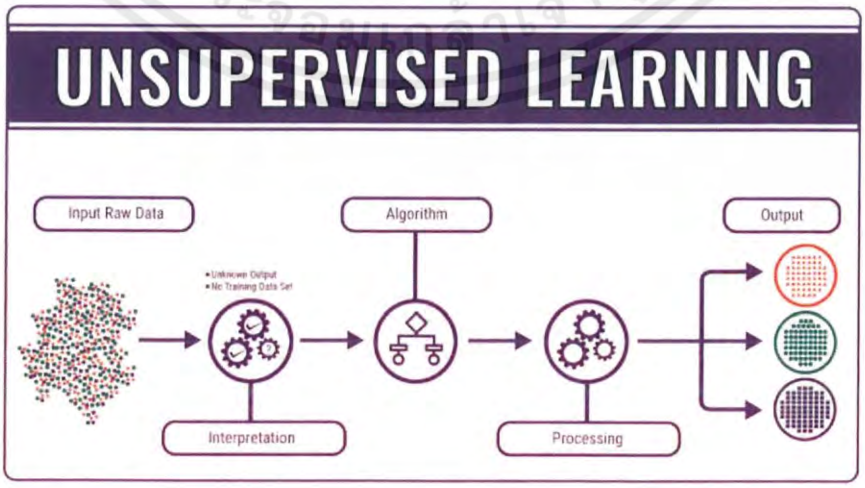
1. supervised learning คือ การเรียนรู้แบบมีผู้สอน โดยที่ผู้สอนจะป้อนข้อมูลตัวอย่างและผลลัพธ์เข้าสู่คอมพิวเตอร์ เป้าหมายคือการสร้างกฎทั่วไปที่สามารถเชื่อมโยงข้อมูลขาเข้ากับขาออกได้



รูปที่ 2.2 supervised learning

Model ส่วนใหญ่ที่ใช้ คือ Support vector machine, Naive Bayes, Gradient boosting, Classification trees / random forest

2. unsupervised learning คือ การเรียนรู้แบบไม่มีผู้สอน โดยจะให้คอมพิวเตอร์หาโครงสร้างของข้อมูลขาเข้าเอง

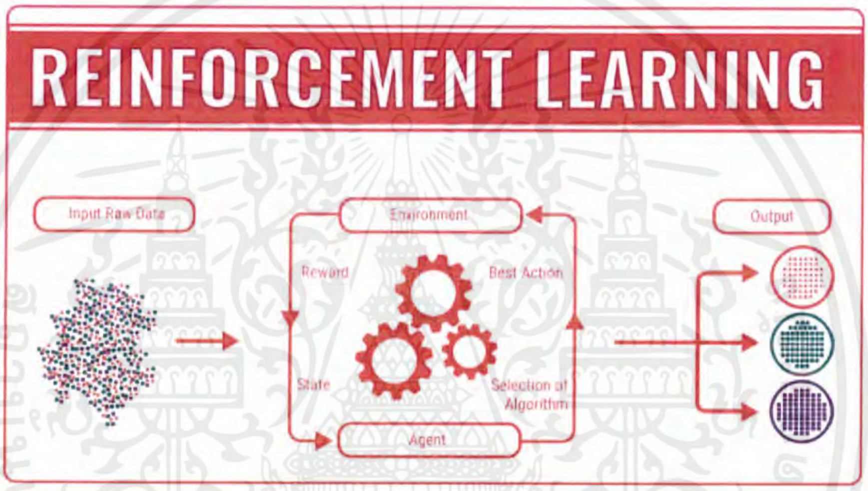


เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.3 unsupervised learning

Model ส่วนใหญ่ที่ใช้ คือ K Nearest Neighbour, K Mean

3. reinforcement learning คือ การเรียนรู้แบบเสริมกำลัง โดยให้คอมพิวเตอร์มีปฏิสัมพันธ์กับสิ่งแวดล้อมที่เปลี่ยนไปตลอดเวลาโดยคอมพิวเตอร์จะต้องทำงานบางอย่าง (เช่น ขับรถ) โดยที่ไม่มีผู้สอนคอยบอกอย่างจริงจังว่าวิธีการที่ทำอยู่นั้นเข้าใกล้เป้าหมายแล้วหรือไม่ ตัวอย่างเช่น การเรียนรู้เพื่อเล่นเกม



รูปที่ 2.4 reinforcement learning

Model ส่วนใหญ่ที่ใช้ คือ Markov Decision Processes (MDP), Q-learning

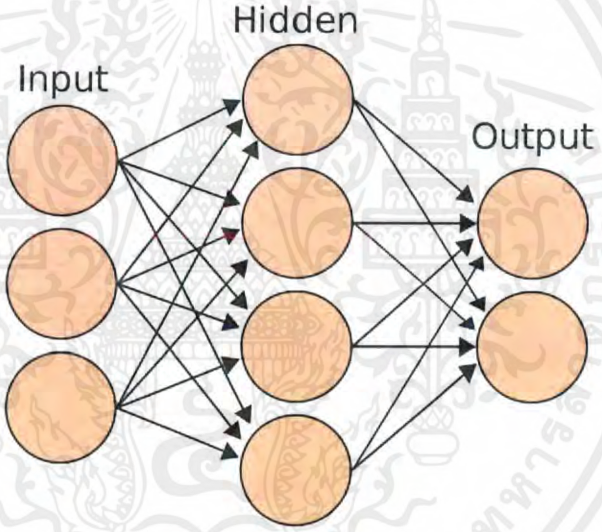
2.1.2 artificial neural network

โครงข่ายประสาทเทียมหรือ artificial neural network คือ โมเดลทางคณิตศาสตร์หรือโมเดลทางคอมพิวเตอร์สำหรับประมวลผลสารสนเทศด้วยการคำนวณแบบคอนเนคชันนิสต์ (connectionist) แนวคิดเริ่มต้นของเทคนิคนี้ได้มาจากการศึกษาโครงข่ายไฟฟ้าชีวภาพ (bioelectric network) ในสมอง ซึ่งประกอบด้วย เซลล์ประสาท (neurons) และ จุดประสานประสาท (synapses) ตามโมเดลนี้ ข่ายงานประสาทเกิดจากการเชื่อมต่อระหว่างเซลล์ประสาท จนเป็นเครือข่ายที่ทำงานร่วมกัน

โมเดลของ artificial neural network ข่ายงานประสาทแบบป้อนไปหน้า (feedforward) ประกอบด้วยเซตของบัพ (node) ซึ่งอาจจะถูกกำหนดให้เป็นบัพอินพุต (input nodes) บัพเอาต์พุต (output nodes) หรือ บัพอยู่ระหว่างกลางซึ่งเรียกว่า บัพฮิดเดน (hidden nodes) มีการเชื่อมต่อระหว่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บัพ (หรือนิวรอน) โดยกำหนดค่าน้ำหนัก (weight) กำกับอยู่ที่เส้นเชื่อมทุกเส้น เมื่อข่ายงานเริ่มทำงาน จะมีการกำหนดค่าให้แก่บัพอินพุต โดยค่าเหล่านี้ อาจจะได้มาจากการกำหนดโดยมนุษย์ จากเซนเซอร์ที่วัดค่าต่างๆ หรือผลจากโปรแกรมอื่นๆ จากนั้นบัพอินพุต จะส่งค่าที่ได้รับ ไปตามเส้นเชื่อมขาออก โดยที่ค่าที่ส่งออกไปจะถูกคูณกับค่าน้ำหนักของเส้นเชื่อม บัพในชั้นถัดไปจะรับค่า ซึ่งเป็นผลรวมจากบัพต่างๆ แล้วจึงคำนวณผลอย่างง่าย โดยทั่วไปจะใช้ฟังก์ชันซิกมอยด์ (sigmoid function) แล้วส่งค่าไปยังชั้นถัดไป การคำนวณเช่นนี้จะเกิดขึ้นไปเรื่อยๆทีละชั้น จนถึงบัพเอาต์พุต โดยในยุคแรก (ราว ค.ศ. 1970) จำนวนชั้นจะถูกกำหนดไว้เป็นค่าคงที่ แต่ในปัจจุบันมีการนำขั้นตอนวิธีเชิงพันธุกรรม มาช่วยออกแบบโครงสร้างของข่ายงาน ดู นิวโรอีโวลูชัน (Neuroevolution)



รูปที่ 2.5 Model of Artificial Neural Network

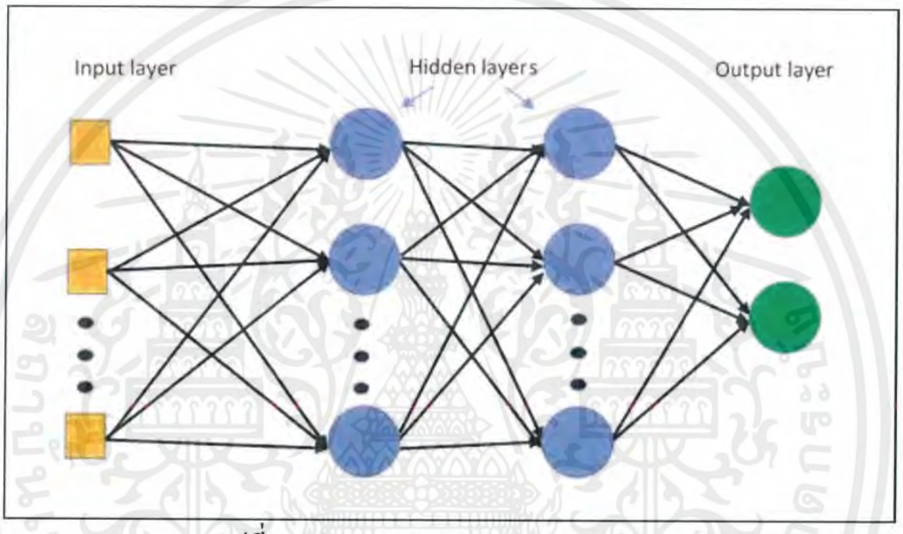
ประเภทของข่ายงานประสาทเทียม

2.1.2.1 เพอร์เซ็ปตรอนหลายชั้น (multi-layer perceptron)

โครงข่ายประสาทเทียมแบบ MLP เป็นรูปแบบหนึ่งของโครงข่ายประสาทเทียมที่มีโครงสร้างเป็นแบบหลายๆชั้น ใช้สำหรับงานที่มีความซับซ้อนได้ผลเป็นอย่างดี โดยมีกระบวนการฝึกฝนเป็นแบบมีผู้สอน (Supervise) และใช้ขั้นตอนการส่งค่าย้อนกลับ (Backpropagation) สำหรับการฝึกฝนกระบวนการส่งค่าย้อนกลับ ประกอบด้วย 2 ส่วนย่อยคือ การส่งผ่านไปข้างหน้า (Forward Pass) การส่งผ่านย้อนกลับ (Backward Pass) สำหรับการส่งผ่านไป

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้างหน้า ข้อมูลจะผ่านเข้าโครงข่ายประสาทเทียมที่ชั้นข้อ มูลเข้า และจะส่งผ่าน จากอีกชั้นหนึ่งไปสู่อีกชั้นหนึ่งจนกระทั่งถึงชั้นข้อมูลออก ส่วนการส่งผ่านย้อนกลับค่าน้ำหนักการเชื่อมต่อจะถูกปรับเปลี่ยนให้สอดคล้องกับกฎการแก้ข้อผิดพลาด (Error-Correction) คือผลต่างของผลตอบที่แท้จริง (Actual Response) กับผลตอบเป้าหมาย (Target Response) เกิดเป็นสัญญาณผิดพลาด (Error Signal) ซึ่งสัญญาณผิดพลาดนี้จะถูกส่งย้อนกลับเข้าสู่โครงข่ายประสาทเทียมในทิศทางตรงกันข้ามกับการเชื่อมต่อ และค่าน้ำหนักของการเชื่อมต่อจะถูกปรับจนกระทั่งผลตอบที่แท้จริงเข้าใกล้ผลตอบเป้าหมาย



รูปที่ 2.6 Model of multi-layer perceptron

สัญญาณที่มีโครงข่ายประสาทเทียมแบบ MLP มี 2 ประเภทคือ Function Signal และ Error Signal

2.1.2.1.1 Function Signal เป็นสัญญาณเข้าที่มาจากโหนดในชั้นก่อนหน้าและจะส่งผ่านไปข้างหน้าจากโหนดหนึ่งไปสู่อีกโหนดหนึ่ง

2.1.2.1.2 Error Signal เป็นสัญญาณย้อนกลับที่เกิดขึ้นที่โหนดในชั้นข้อมูลออกของโครงข่ายประสาทเทียม และถูกส่งผ่านย้อนกลับจากชั้นหนึ่งไปสู่อีกชั้นหนึ่ง

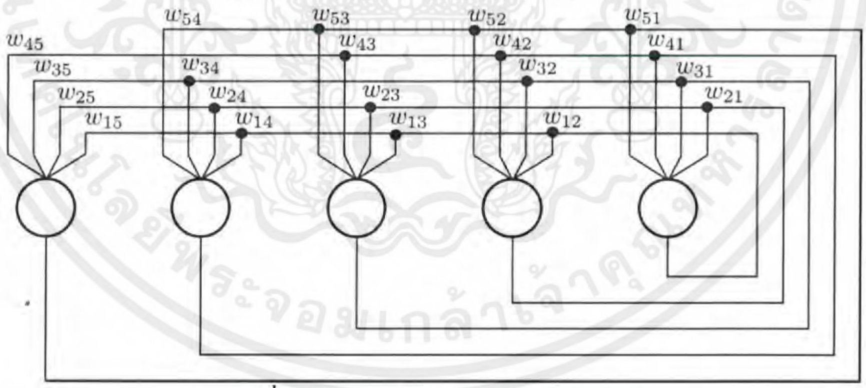
หลักการทำงานของ MLP คือในแต่ละชั้นของชั้นซ่อนตัว (Hidden Layer) จะมีฟังก์ชันสำหรับคำนวณเมื่อได้รับสัญญาณ (Output) จากโหนดในชั้นก่อนหน้านี้ เรียกว่า Activation Function โดยในแต่ละชั้นไม่จำเป็นต้องเป็นฟังก์ชันเดียวกันก็ได้ ชั้นซ่อนตัวนั้นมีหน้าที่สำคัญคือจะพยายามแปลงข้อมูลที่เข้ามาในชั้น (Layer) นั้นๆให้สามารถแยกแยะความแตกต่างโดยใช้เส้นตรงเส้นเดียว (Linearly Separable) และก่อนที่ข้อมูลจะถูกส่งไปถึงชั้นข้อมูลออก (Output Layer) ใน

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บางครั้งอาจจำเป็นต้องใช้ชั้นซ่อนตัวมากกว่า 1 ชั้นในการแปลงข้อมูลให้อยู่ในรูป Linearly Separable

ในการคำนวณหา Output ในปัญหาการจำแนกทำได้โดยการใส่ข้อมูล Input เข้าไปในโครงข่ายประสาทเทียมที่เราได้ทำการหาไว้แล้ว จากนั้นให้ทำการเปรียบเทียบค่าของ Output ใน Output Layer และให้ทำการเลือกค่าของ Output ที่มีค่าสูงกว่า (Neuron ที่มีค่าสูงกว่า) และทำการรับค่าของพยากรณ์ที่ตรงกับ Neuron ที่เลือก และให้นำค่าของ มาเปรียบเทียบกับค่าที่ยอมรับได้ หากค่าของ อยู่ในช่วงที่รับได้ (Error น้อยกว่า Error ที่เรากำหนด) ก็ให้ทำการรับข้อมูลชุดถัดไป แต่หากค่าของ มากกว่าค่าที่ยอมรับได้ ให้ทำการปรับค่าน้ำหนักและ Biased ตามขั้นตอนที่ได้กล่าวไว้ข้างต้น เมื่อทำการปรับน้ำหนักเรียบร้อยแล้ว ให้ทำการรับข้อมูลชุดถัดไปและทำตามขั้นตอนซ้ำอีกรอบจนกระทั่งถึงข้อมูลชุดสุดท้าย และเมื่อทำข้อมูลชุดสุดท้ายเสร็จจะนับเป็น 1 รอบของการคำนวณ (1 Epoch) จากนั้นจะทำการหาค่าผิดพลาดรวมเฉลี่ย จากค่าเฉลี่ยของ ที่ได้เก็บค่าเอาไว้ เพื่อใช้ในการตรวจสอบว่าค่า โดยเฉลี่ยในการจำแนกนั้น มีค่าน้อยกว่าค่าผิดพลาดที่ยอมรับได้หรือไม่ ถ้าใช่แสดงว่าโครงข่ายประสาทเทียมที่สร้างขึ้นนั้นสามารถให้ผลลัพธ์ที่ถูกต้องของทุกๆข้อมูลแล้ว จึงทำการจบการเรียนรู้ได้ แต่ถ้าไม่ใช่ ให้กลับไปทำตามขั้นตอนแรก โดยเริ่มรับข้อมูลชุดที่ 1 ใหม่

2.1.2.2 ข่ายงานฮอปฟิลด์ (Hopfield network)



รูปที่ 2.7 Model of Hopfield network

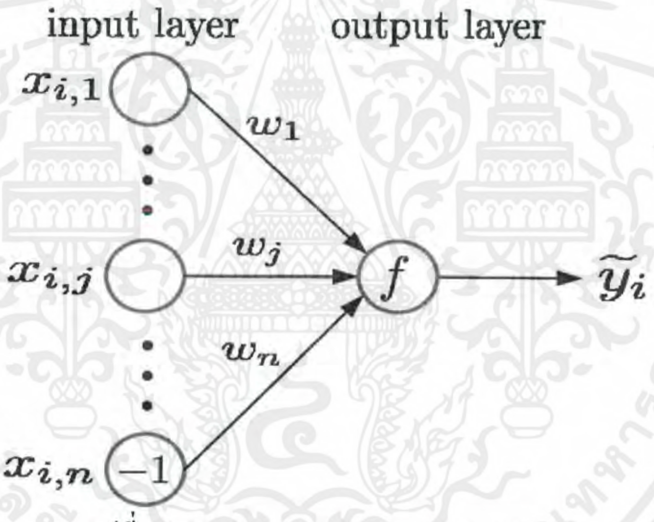
ข่ายงานฮอปฟิลด์ บางครั้งเรียกว่า Hebb's Rule เป็นวิธีการหรือทฤษฎีที่เกี่ยวข้องของกับการจัดกลุ่มของข้อมูลโดยอาศัยต้นแบบมาจากระบบประสาท คือ เซลล์ใดๆที่อยู่ใกล้กันและสามารถกระตุ้นเซลล์ที่อยู่ใกล้เคียงแบบซ้ำๆ ต่อเนื่องกันจนเกิดการเปลี่ยนแปลงขึ้นกับเซลล์ตนเองและเซลล์ใกล้เคียงจะถือว่า เซลล์ทั้งสองมีปฏิสัมพันธ์ที่สนับสนุนกัน ส่งผลให้ทั้งคู่สามารถจะเจริญเติบโตไปด้วยกันได้ Hebb's Rule จะเกี่ยวของกับความสัมพันธ์ของชุดข้อมูล 2 ชุดที่สนับสนุน

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กันจนทำให้น้ำหนักของข้อมูลมีความน่าเชื่อถือมากขึ้น ช่วยให้เกิดความมั่นใจว่าค่าน้ำหนักดังกล่าวสามารถนำมาใช้งานได้อย่างมีประสิทธิภาพใน Hebb's Rule แต่ละ Node ของชั้น Input Layer จะเชื่อมโยงกับทุก Node ของชั้น Output Layer อย่างสมบูรณ์ดังนั้นการทำงานของวิธีนี้จึงอาศัยรูปแบบจาก Input เป็นหลักโดยการ Mapping ระหว่าง Input Pattern กับ Recalled Pattern (Output Pattern ที่สามารถกลับสู่ Input Pattern ได้) ซึ่งได้จากการพิจารณาข้อมูลที่อยู่ใกล้เคียงแต่การ Mapping มีข้อเสียคือสามารถ Mapping ได้เฉพาะข้อมูลที่สัมพันธ์กันในแบบ Orthogonal (แบบตั้งฉาก) ผลลัพธ์ในการ Mapping ด้วย Hebb's Rule

2.1.2.3 ตัวอย่างประเภทอื่น

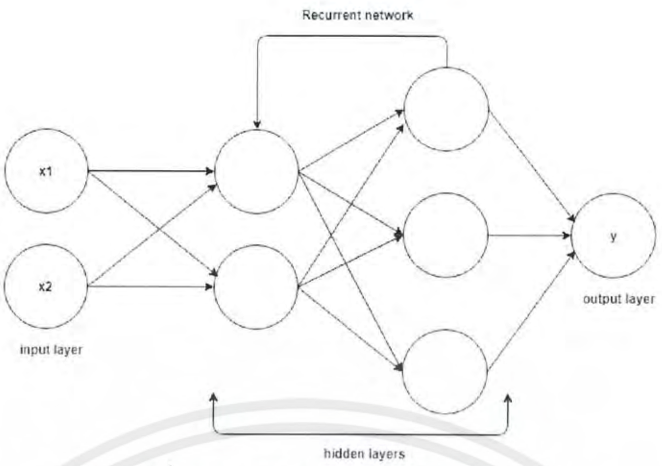
- เพอร์เซ็ปตรอนชั้นเดียว (single-layer perceptron)



รูปที่ 2.8 Model of single-layer perceptron

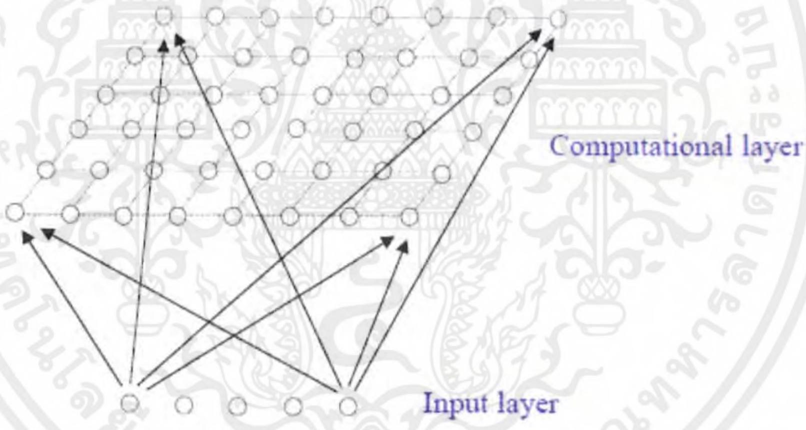
- ข่ายงานแบบวนซ้ำ (recurrent network)

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 Model of recurrent network

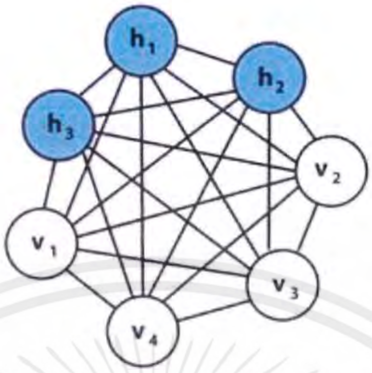
- แผนผังจัดระเบียบเองได้ (self-organizing map)



รูปที่ 2.10 Model of self-organizing map

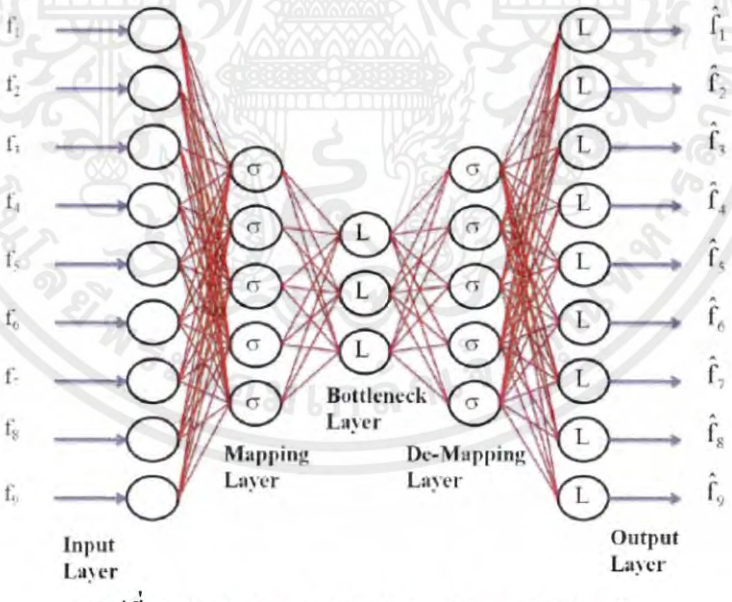
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เครื่องจักรโบลทซ์แมน (Boltzmann machine)



รูปที่ 2.11 Model of Boltzmann machine

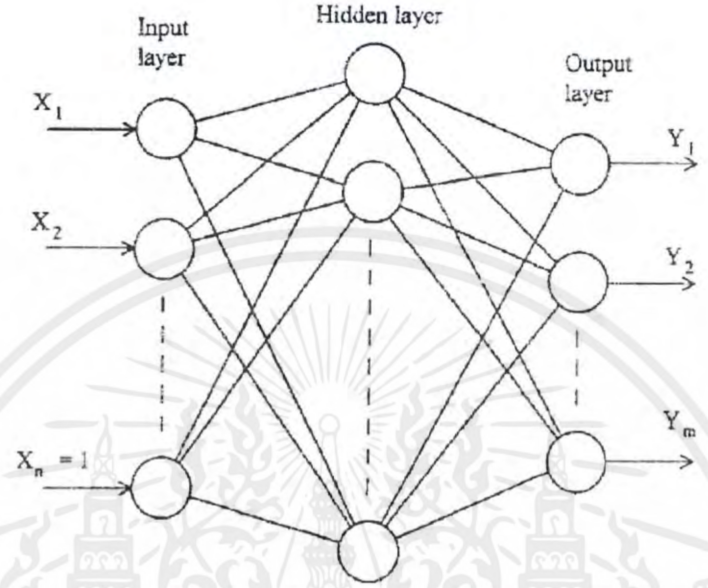
- กลไกแบบคณะกรรมการ (committee of machines)
- ข่ายงานความสัมพันธ์ (associative Neural Network-ASNN)



รูปที่ 2.12 Model of Associative Neural Network

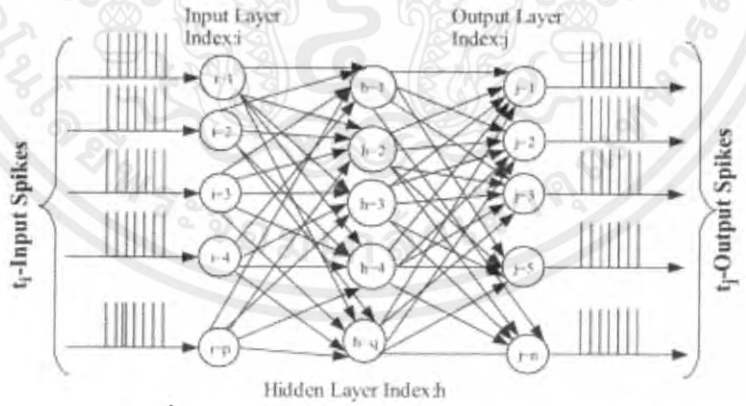
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ข่ายงานกึ่งสำเร็จรูป (instantaneously trained networks)



รูปที่ 2.13 Model of instantaneously trained networks

- ข่ายงานแบบยิงกระตุ้น (spiking neural networks)



รูปที่ 2.14 Model of spiking neural networks

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 เครื่องมือที่เกี่ยวข้อง

2.2.1 TensorFlow

TensorFlow คือ ไลบรารีที่ช่วยในการเขียนหรือพัฒนา Machine Learning ที่ทาง Google ได้ทำการพัฒนาไลบรารีซอฟต์แวร์โอเพนซอร์ส สำหรับการเขียนโปรแกรม dataflow ในงานหลายประเภท ที่จะใช้ภาษา python ในการเขียนโปรแกรม รองรับเวอร์ชั่นทั้ง python2 และ Python3 โดย TensorFlow สามารถทำงานบน CPU และ GPUs รองรับระบบปฏิบัติการ Linux, macOS, Windows และ Android

```
import sys
from pySageDecoder import PySageDecoder, PySageDecodable
import tensorflow_wrapper as tf_wp


if __name__ == '__main__':
    if len(sys.argv) < 2:
        sys.exit("expected arguments: lbn-decoder-args tf_model_file")

    tf_model_file = sys.argv.pop()
    tf_model_op = tf_wp.load_model(tf_model_file)
    decoder = PySageDecoder()
    decoder.setup(sys.argv)

    while not decoder.done():
        features = decoder.get_features()
        log_posterior = tf_wp.get_log_posterior(features, tf_model_op)
        decodable = decoder.get_matrix_decodable(log_posterior)
        decoder.decode(decodable)
        decoder.next()

    decoder.finalize()
```

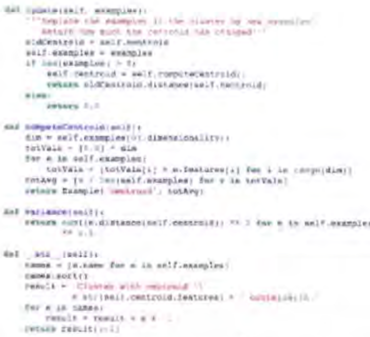
รูปที่ 2.15 TensorFlow

2.2.2 Python

Python คือชื่อภาษาที่ใช้ในการเขียนโปรแกรมภาษาหนึ่ง ซึ่งถูกพัฒนาขึ้นมาโดยไม่ยึดติดกับแพลตฟอร์ม กล่าวคือสามารถรันภาษา Python ได้ทั้งบนระบบ Unix, Linux , Windows NT, Windows 2000, Windows XP หรือแม้แต่ระบบ FreeBSD อีกอย่างหนึ่งภาษาตัว นี้เป็น OpenSource เหมือนอย่าง PHP ทำให้ทุกคนสามารถที่จะนำ Python มาพัฒนาโปรแกรมของเราได้ฟรีๆโดยไม่ต้องเสียค่าใช้จ่าย และความเป็น Open Source ทำให้มีคนเข้ามาช่วยกันพัฒนาให้ Python มีความสามารถสูงขึ้น และใช้งานได้ครบคุมกับทุกลักษณะงาน

โค้ดของ Python ถูกสร้างขึ้นมาจากภาษาซี การประมวลผลจะทำในแบบอินเทอร์พรีเตอร์ คือจะประมวลผลไปทีละบรรทัดและปฏิบัติตามคำสั่งที่ได้รับ Python เวอร์ชันแรกคือเวอร์ชัน 0.9.0 ออกมาเมื่อปี 2533 และเวอร์ชันปัจจุบันคือ 3.7.3 สำหรับระบบปฏิบัติการวินโดว์

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.16 Python

คุณลักษณะเด่นของภาษา Python

1. สนับสนุนแนวแบบคิดออปเจกต์โอเรียนเทด หรือ OOP (Object Oriented Programming)
2. เป็น Open Source
3. โค้ดที่เขียนด้วย Python สามารถนำไปรันบนระบบปฏิบัติการ ได้หลากหลาย
4. สนับสนุนเทคโนโลยี COM ของ Ms-windows
5. Python รวมมาตรฐานการอินเตอร์เฟส Tkinter ซึ่งสนับสนุนบนระบบ X windows, Ms-windows และ Macintosh การใช้คำสั่ง Tkinter API ช่วยให้โปรแกรมเมอร์ไม่ต้องแก้ไขโค้ดเมื่อนำไปรันบนระบบปฏิบัติการอื่นๆ
6. เป็น Dynamic typing คือ สามารถเปลี่ยนชนิดข้อมูลได้ง่ายและสะดวก
7. มี Buil-in Object Types คือ โครงสร้างของข้อมูลที่สามารถใช้ได้ใน Python ประกอบด้วย ลิสต์, ดิกชันนารี, สตริง ที่ง่ายต่อการใช้งานและมีประสิทธิภาพสูง
8. มีเครื่องมือต่างๆ มากมาย เช่น การประมวลผลเท็กซ์ไฟล์ การเรียงข้อมูล การเชื่อต่อสตริง การตรวจสอบเงื่อนไขของข้อความ การแทนคำ เป็นต้น
9. มีมอดูลสำหรับจัดการ Regular Expresion
10. มีมอดูลที่สร้างขึ้นจากนักพัฒนาสนับสนุนมากมาย ได้แก่ COM, Image, CORBA, ORBs, XML เป็นต้น
11. จัดการหน่วยความจำอย่างอัตโนมัติ สามารถจักการพื้นที่หน่วยความจำที่ไม่ต่อเนื่องให้ทำงานได้อย่างมีประสิทธิภาพ
12. อนุญาตให้ฝังชุดคำสั่งของ Python เอาไว้ภายในโค๊ดภาษา C/C++ ได้
13. อนุญาตให้โปรแกรมเมอร์สร้าง Dynamic Link Libray (DLL) เพื่อใช้ร่วมกับ Python

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

14. มีมอดูลสนับสนุนเกี่ยวกับเน็ตเวิร์ก โปรเซส เธรด regular, expression, xml, GUI และอื่นๆ
15. ประกอบด้วยมอดูลสำหรับสร้าง Internet Script และติดต่อกับอินเทอร์เน็ตผ่าน Sockets, และทำหน้าที่เป็น CGI Script ตรอดจนใช้งานคำสั่ง FTP , Glopher, XML และอื่นๆอีกมาก
16. สามารถประมาลผมทางด้านวิยาศาสตร์ และวิศวกรรมศาสตร์ได้อย่างมีประสิทธิภาพ
17. มีฟังก์ชันสนับสนุนฐานข้อมูล เช่น MySQL, Sybase, Oracle , Informix, ODBC และอื่นๆ
18. มีไลบรารีสนับสนุนด้านการสร้างภาพกราฟฟิก เช่น ทำภาพเบลอ หรือภาพชัด หรือเขียนข้อความบนภาพ ตลอดจนบันถึกไฟล์ในรูปแบบต่างๆ ได้อย่างสะดวกและมีประสิทธิภาพ
19. มีไลบรารีสนับสนุนด้านปัญญาประดิษฐ์
20. มีไลบรารีสำหรับสร้างเอกสาร PDF โดยไม่ต้องติดตั้ง Acrobat Writer
21. มีไลบรารีสำหรับสร้าง Shockwaves Flash (SWF) โดยไม่ต้องติดตั้ง Macromedia Flash

2.2.3 Keras

```
import pandas as pd
import urllib.request as ur
from tqdm import tqdm
import io
import requests
import numpy
from keras.models import Sequential
from keras.layers import Dense
from keras.wrappers.scikit_learn import KerasClassifier
from sklearn.preprocessing import LabelEncoder

#Read csv file from UCI Website
def read_csv_from_url(url_data):
    uci_dataset = url_data
    s=requests.get(uci_dataset).content
    the_dataframe=pd.read_csv(io.StringIO(s.decode('utf-8')))
    dataset = the_dataframe.values
    return dataset

# Random seed for reproducibility
seed_val = 7
numpy.random.seed(seed_val)

# split into input (X) and output (Y) variables
def split_my_data(ratio,dataset):
    X = dataset[:,0:ratio].astype(float)
    Y = dataset[:,ratio]
    return X, Y
```

รูปที่ 2.17 Keras

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Keras เป็น deep-learning library ที่ใช้งานง่ายที่สุดในขณะนี้ และในขณะเดียวกันก็มีประสิทธิภาพในการรันโมเดลสูง เนื่องจาก back-end ของ Keras มีทั้ง Tensorflow และ Theano ซึ่งจัดเป็น deep-learning library ที่สมรรถนะสูงทั้งคู่ โดยผู้สร้างคือ Francois Chollet ซึ่งเป็น Data Scientists ที่ Google

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบและการพัฒนา

3.1 Problem Statement

ในช่วงศตวรรษที่ผ่านมา ถนนทางหลวงที่เชื่อมต่อระหว่างจังหวัดได้รับการพัฒนาไปมากตามการขยายตัวของบ้านเมือง ทั้งการพัฒนาถนนที่มีอยู่เดิมและการตัดถนนเส้นใหม่เพื่อให้สะดวกต่อการเดินทางระหว่างจังหวัดมากยิ่งขึ้น และด้วยยุคสมัยและเทคโนโลยีที่เปลี่ยนไปทำให้บุคคลทั่วไปเข้าถึงรถยนต์ได้มากขึ้นประกอบกับถนนต่าง ๆที่ได้รับการพัฒนาส่งผลให้มีผู้ใช้รถใช้ถนนมากขึ้น ในขณะเดียวกันเมื่อมีผู้ใช้รถใช้ถนนมากขึ้นก็ไม่อาจจะหลีกเลี่ยงอุบัติเหตุบนท้องถนนที่เพิ่มมากขึ้นด้วยเช่นเดียวกัน ซึ่งถ้าหากเกิดอุบัติเหตุเกิดขึ้นแล้วควรจะมีการแจ้งเหตุไปยังหน่วยงานที่เกี่ยวข้องเพื่อให้ความช่วยเหลือโดยเร็วที่สุด

ทว่าหลายครั้งที่อุบัติเหตุซึ่งเกิดบนถนนทางหลวงระหว่างจังหวัดถูกแจ้งไปยังหน่วยงานที่เกี่ยวข้องล่าช้าส่งผลให้ความช่วยเหลือทั้งในด้านการปฐมพยาบาลหรือ การจัดการพื้นที่ล่าช้า อีกทั้งยังทำให้การจราจรบริเวณที่เกิดอุบัติเหตุดังกล่าวติดขัดเช่นกัน

เพื่อแก้ไขปัญหาดังกล่าว โครงงานชิ้นนี้จึงได้นำเสนอ "ระบบตรวจจับความผิดปกติที่บนถนนทางหลวง" ซึ่งแบ่งออกเป็น 2 ส่วนดังนี้

- Norm Speed/Units Model คือการ ประยุกต์ใช้ Machine Learning เพื่อค้นหา "รูปแบบทั่วไป" ของความเร็วและจำนวนรถบนถนนทางหลวงในกรณีปกติซึ่งไม่มีเหตุการณ์หรืออุบัติเหตุใด ๆ เกิดขึ้น
- Incident Detection คือการใช้ "รูปแบบทั่วไป" เพื่อเปรียบเทียบกับข้อมูลจริงเพื่อหาตำแหน่งและช่วงเวลาที่รถติดผิดปกติไปจากปกติวิสัยซึ่งสามารถสื่อถึงการเกิดอุบัติเหตุหรือเหตุการณ์ไม่ปกติอื่น ๆ ได้

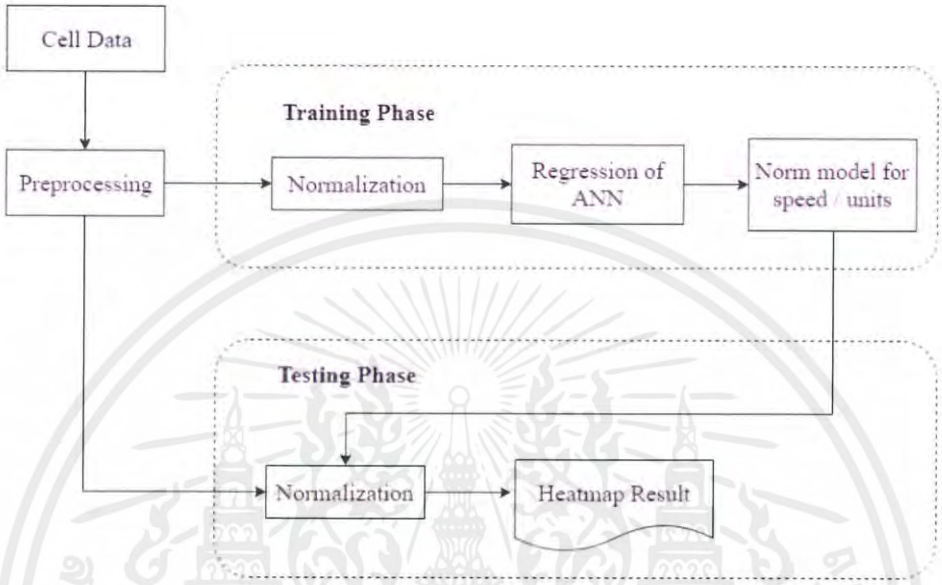
หมายเหตุ: ระบบดังกล่าวทดสอบด้วยข้อมูลของถนนทางหลวงหมายเลข 2 หรือถนนมิตรภาพที่มีจุดเริ่มต้นที่จังหวัดสระบุรี และสิ้นสุดที่จังหวัดหนองคาย

3.2 Norm Speed/Units Model

ในส่วนนี้จะเป็นการใช้ Machine Learning "Artificial Neural Network" ในการเรียนรู้รูปแบบปกติของความเร็วและจำนวนรถบนถนนทางหลวงสาย 2 ว่า ณ เวลาหนึ่ง ๆ ที่หลักกิโลใด ๆ ควรจะ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีความเร็วและจำนวนรถเท่าใดในกรณีที่ไม่มีเหตุการณ์ไม่ปกติเกิดขึ้น ซึ่งกระบวนการเรียนรู้เป็นไปตามรูปที่ 3.1



รูปที่ 3.1 Block Diagram for NN Model

3.2.1 Cell Data

ชุดข้อมูลที่ใช้ทดสอบนั้นมาจากการรวมข้อมูลจาก GPS ที่อยู่บนรถขนส่งสาธารณะแต่ล่ะคันซึ่งวิ่งบนทางหลวงสายต่าง ๆ (อยู่ในรูปของละติจูดและลองจิจูด) มาประมวลผลร่วมกับข้อมูลหลักกิโลเมตรทำให้สามารถบอกได้ว่า ณ เวลาหนึ่ง ๆที่ช่วงระหว่างหลักกิโลเมตรใดบนเส้นทางหลวงสายใด มีจำนวนรถและมีความเร็วเป็นเท่าไร

feature name	*description*	*type*
datetime	วันเวลาที่เก็บข้อมูล อยู่ในรูปของ ปี:เดือน:วัน ชั่วโมง:นาที:วินาที	continuous
road_number	หมายเลขถนนทางหลวง	discrete
km	ช่วงระหว่างหลักกิโล เช่น เลข 1 หมายถึงช่วงระหว่างหลักกิโลที่ 0 ถึง 1	discrete
direction	ทิศทางของถนนที่เก็บข้อมูล เช่น ถนนขาเข้าหรือถนนขาออก	discrete

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

all_units	จำนวนรถทั้งหมดที่อยู่ในช่วงระหว่างหลักกิโลดังกล่าว	continuous
inflow_units	จำนวนรถที่กำลังเข้ามาในช่วงระหว่างหลักกิโล	continuous
outflow_unit	จำนวนรถที่กำลังออกจากช่วงระหว่างหลักกิโล	continuous
avg_speed	ความเร็วเฉลี่ยของรถในช่วงระหว่างหลักกิโล	continuous
max_speed	ความเร็วสูงสุดของรถในช่วงระหว่างหลักกิโล	continuous

ตารางที่ 3.1 รายละเอียดของข้อมูล Cell Data

3.2.2 Preprocessing

เนื่องจากสิ่งที่เราต้องการคือรูปแบบของความเร็วของรถบนถนนทางหลวงในแต่ละช่วงเวลาเทียบกับช่วงของหลักกิโลเมตร ดังนั้นจึงต้องทำการแปลงวันเวลาในข้อมูลให้อยู่ในรูปของนาทีใน 1 วันโดยไม่สนใจวันที่ๆเก็บข้อมูลดังกล่าวเช่น เก็บข้อมูลเรคคอร์ดนี้ในวันที่ 1 เดือนกรกฎาคม ปี 2017 (2017:07:01) เวลา 2 นาฬิกา 14 นาที 52 วินาที (02:14:52) จะทำการแปลงให้อยู่ในรูปนาทีที่ 134 ของวันและตัดวันที่ออก

นอกจากนี้ตัว Neural Network นั้นเหมาะกับอินพุตที่อยู่ในรูป 0 ถึง 1 จึงต้องทำการ Normalize ข้อมูลอินพุตทั้งที่ใช้ในการเทรนโมเดลและที่ใช้ทดสอบโมเดลให้อยู่ในช่วง 0 ถึง 1

เนื่องจากโครงงานชิ้นนี้มีขอบเขตอยู่ที่ถนนทางหลวงสาย 2 แต่ข้อมูลดิบที่มีนั้นเป็นข้อมูลของถนนทางหลวงทุกสายอยู่รวมกันจึงต้องทำการกรองข้อมูลให้เหลือเฉพาะเรคคอร์ดที่เป็นของถนนทางหลวงสาย 2 เท่านั้น

3.2.3 Training Phase

ในการเทรนโมเดลจะใช้ข้อมูล Cell Data ในช่วงระยะเวลา 1 เดือนโดยเทรนในรูปแบบ "การวิเคราะห์การถดถอย" (Regression Analysis) ซึ่งเป็นการหาความสัมพันธ์ระหว่างตัวแปรอิสระ (Independent Value หรือ X ในที่นี้คือ เวลา และ หมายเลขหลักกิโล) กับตัวแปรตาม (Dependent Value หรือ Y ในที่นี้คือ ความเร็ว และ จำนวนรถโดยเทรนเป็น 2 โมเดลแยกจากกัน) และโมเดลที่เลือกใช้งานคือ ANN (Artificial Neural Network) ซึ่งในแต่ละเลเยอร์ แต่ละโหนด จะมีเอาท์พุตเป็นตัวเลขจำนวนจริง 1 ตัวที่มาจากการนำอินพุตผ่านกระบวนการ weighting และ biasing ก่อนจะผ่าน activation function แล้วจึงส่งต่อไปให้ทุก ๆ โหนดที่อยู่ในเลเยอร์ถัดไป เพราะฉะนั้นทุกๆ โหนดในเลเยอร์ถัดไปจะมีชุดของ weight และค่า bias ของตัวเอง

สมมุติว่าเอาท์พุตของแต่ละโหนดในหนึ่งเลเยอร์ถูกเก็บอยู่ในรูปของเวคเตอร์ a ดังนั้นเวคเตอร์ของเอาท์พุตในเลเยอร์ชั้นถัดไปจะอยู่ในรูปที่ 3.2

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดักแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\sigma(Wa + b).$$

รูปที่ 3.2 เวคเตอร์ของเอาท์พุตในเลเยอร์ชั้นถัดไป

เมื่อ W คือ เมทริกซ์ที่เก็บค่า weight และ b คือเวคเตอร์ของ bias โดยจำนวนหลักของ W จะเท่ากับจำนวนโหนดในเลเยอร์ก่อนหน้าและ จำนวนแถวของ W จะเท่ากับจำนวนโหนดในเลเยอร์ปัจจุบัน และ b จะมีจำนวนเท่ากับจำนวนโหนดของเลเยอร์ปัจจุบันเช่นกัน

Activation function ที่ใช้ทดสอบ คือ rectified linear unit หรือ ReLU ซึ่งอยู่ในรูปที่ 3.3

$$\sigma(x) = \begin{cases} 0, & \text{for } x \le 0, \\ x, & \text{for } x > 0. \end{cases}$$

รูปที่ 3.3 ReLU

โครงสร้าง Neural Network ที่ใช้ทดสอบมีดังนี้

1. Input layer: 2 node (ความเร็วและช่วงเวลา)
2. Hidden layer 1: 128 node, activation function: rectified linear unit (ReLU)
3. Hidden layer 2: 256 node, activation function: rectified linear unit (ReLU)
4. Hidden layer 3: 256 node, activation function: rectified linear unit (ReLU)
5. Hidden layer 4: 256 node, activation function: rectified linear unit (ReLU)
6. Output layer 1: 1 node, activation function: linear

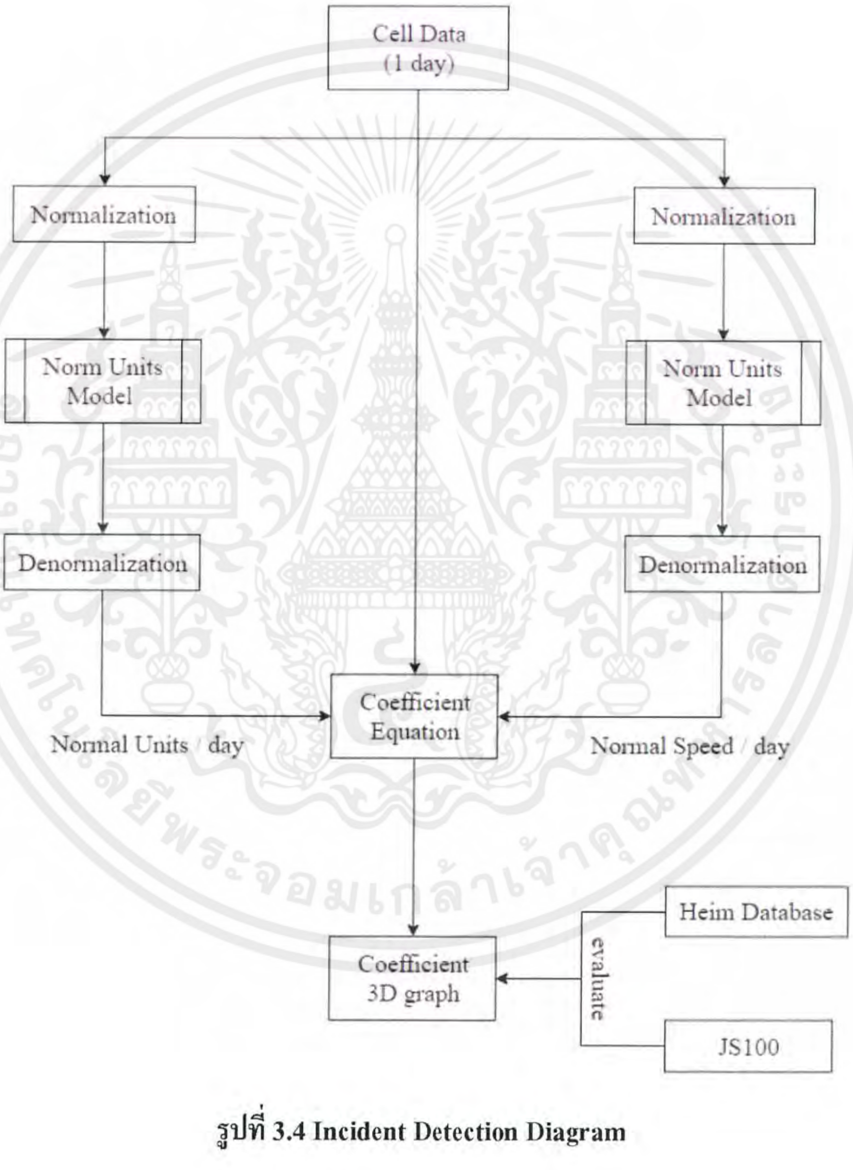
3.2.4 Testing Phase

ในการทดสอบโมเดลจะใช้ข้อมูล Cell Data ของ 1 วันหรือมากกว่าที่ไม่อยู่ในช่วงเวลาของข้อมูลที่ใช้เทรนโมเดล เช่น ข้อมูลที่ใช้เทรนโมเดลเป็นข้อมูลของเดือน 7 ดังนั้นข้อมูลที่ใช้ทดสอบจะเป็นข้อมูลเวลา และหมายเลขหลักกิโลของ 1 วันหรือมากกว่า ในเดือน 1 – 6 หรือ 8 – 12

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และทดสอบประสิทธิภาพของโมเดลด้วยค่าเฉลี่ยของค่าสัมบูรณ์ของความคลาดเคลื่อน (Mean Absolute Error) และนำผลที่ได้จากโมเดลมาแสดงด้วย แผนภาพความหนาแน่น (Heatmap)

3.25 Application (Incident Detection)



รูปที่ 3.4 Incident Detection Diagram

ในการตรวจจับเหตุการณ์ผิดปกติที่อาจเกิดขึ้นบนถนน จะใช้ค่าความเร็วและจำนวนรถจากข้อมูลจริงในหนึ่งวันเป็นอินพุตยังโมเดลที่ได้ทำการเทรนไว้เรียบร้อยแล้วเพื่อหาค่าความเร็ว

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และจำนวนรถ "ที่ควรจะเป็น" แล้วนำผลได้จากการทำนายของโมเดลมาเทียบอัตราส่วนกับความเร็วและจำนวนรถที่เกิดขึ้นจริงเพื่อหาค่า "สัมประสิทธิ์" ตามสมการ

$$coefficient = \frac{\#speed}{real_speed} \times \frac{real_units}{\#units}$$

รูปที่ 3.5 สมการหาค่าสัมประสิทธิ์

จากนั้นนำค่าสัมประสิทธิ์ที่ได้มาพล็อตกราฟ 3 มิติ โดย X คือหลักกิโล Y คือเวลาใน 1 วันและ Z คือค่าสัมประสิทธิ์ แล้วตรวจสอบดูว่าที่หลักกิโลเมตรไหนเวลาเท่าไรที่ค่าสัมประสิทธิ์มีการเปลี่ยนแปลงอย่างกะทันหัน จะทำให้สามารถบอกได้ว่าบริเวณดังกล่าวมีโอกาสที่จะเกิดอุบัติเหตุหรือมีเหตุการณ์ผิดปกติเกิดขึ้นสูง

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การดำเนินงานและการทดลอง

4.1 ขั้นตอนการทดลอง

4.1.1 การนำข้อมูลเข้าและ การประมวลผลเบื้องต้น

เนื่องจากข้อมูลดิบที่มีนั้นอยู่ในรูปของไฟล์ .CSV โดยหนึ่งไฟล์คือข้อมูล Cell Data ในเวลา 1 ชั่วโมงและมีเรคคอร์ดเฉลี่ยประมาณ 15,000 – 20,000 เรคคอร์ดต่อไฟล์และในการเทรนโมเดลจะใช้ข้อมูลรวมทั้งสิ้นเป็นเวลา 1 เดือนเต็มซึ่งมีทั้งหมด 24 x 30 = 720 ไฟล์จึงมีขนาดใหญ่มาก ดังนั้นในการอ่านข้อมูลเข้ามาในโปรแกรมจึงใช้ library multiprocessing ในการสร้างโพสเซสใหม่เพื่ออ่านข้อมูลหลาย ๆ ไฟล์พร้อมกันแบบขนาน (parallel) เพื่อช่วยลดเวลาในการนำเข้าข้อมูลเข้ามาในโปรแกรม แล้วนำข้อมูลที่นำเข้าจากแต่ละโพรเซสมารวมกันอยู่ในรูปของ Data Frame ซึ่งเป็นตัวแปรชนิดหนึ่งจาก library Pandas ที่มีความสามารถในการจัดการข้อมูลรูปแบบตารางได้ดี

ในขณะเดียวกันก็ใช้โพรเซสแต่ละโพสเซสที่สร้างขึ้นมาเพื่ออ่านข้อมูลในการทำ Preprocessing ของแต่ละไฟล์ด้วยเช่นกัน โดยการคัดกรองข้อมูลดิบที่มีให้เหลือเฉพาะข้อมูลของถนนสายที่ต้องการ (ณ ที่นี้คือถนนสาย 2) และทิศทางที่ต้องการเนื่องจากในการเทรนโมเดล จะเทรนถนนขาเข้า และ ถนนขาออกแยกจากกันเป็นคนละโมเดล หลังจากนั้นจะนำ Time stamp ที่มีในข้อมูลดิบของแต่ละเรคคอร์ดมาทำการตัดวันที่ออกให้เหลือเพียงเวลา แล้วนำเวลาดังกล่าวมาแปลงให้อยู่ในรูปนาที สุดท้ายนำผลลัพธ์ที่ได้ (นาทีของเรคคอร์ดนั้น ๆ) มาเพิ่มเป็นอีกหนึ่งคอลัมของเรคคอร์ดดังกล่าว โดยตัวอย่างโปรแกรมที่ใช้ดังรูปที่ 4.1

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
import pandas as pd
import numpy as np
import multiprocessing as mp
from datetime import datetime, timedelta
import sys

output = mp.Queue()
def read(date, hr, output, header, month,  direction):

    fileName = 'data_2018-' + month + '-'
    try:
        df = pd.read_csv('./Data/Transportation Traffic Data/2018-' + month + '/CellData/' + fileName + date + '_' + hr + '.csv', header=None)
    except FileNotFoundError:
        output.put(None)
        sys.exit()
    df.columns = header
    df = df[(df['road_number'] == 2) & (df['direction'] == direction )].reset_index(drop = True)
    df['datetime'] = df['datetime'].apply(lambda x: x.replace('2018-' + month + '-' + date + ' ', ''))
    df['min'] = pd.to_timedelta(df['datetime']).dt.total_seconds().div(60).astype(int).astype(float)
    df['avg_speed'] =  df.avg_speed.astype(float)
    df['max_speed'] =  df.max_speed.astype(float)
    df['km'] = df.km.astype(float)
    df['all_units'] = df.all_units.astype(float)
    df['datetime'] = df['datetime'].apply(lambda x: x.replace(x, '2018-' + month + '-' + date + ' ' + x))
    df['datetime'] = pd.to_datetime(df['datetime'])

    output.put(df)

def extract_data(startdate, enddate, month, direction='in'):
    header = 'datetime,road_number,km,direction,all_units,inflow_units,outflow_unit,samecell_units,avg_speed,max_speed,avg_traveltime,max_traveltime,min'.split(',')
    df = pd.DataFrame(columns=header)
    header.remove('min')
    date = '01,02,03,04,05,06,07,08,09,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31'.split(',')

    hr = '00,01,02,03,04,05,06,07,08,09,10,11,12,13,14,15,16,17,18,19,20,21,22,23'.split(',')
    print('2018-' + month + '_' + direction)
    for d in date[startdate - 1:enddate]:

        processes = [mp.Process(target=read, args=(d, h, output, header, month, direction)) for h in hr]
        for p in processes:
            p.start()
        try:
            newdf = pd.concat([output.get() for p in processes])
        except ValueError:
            newdf = None
        for p in processes:
            p.join()

        if newdf is None:
            print ('Day ' + d +  ' is not exits')
            continue
        df = pd.concat([df, newdf])
        print ('Day ' + d +  ' finished')

    return df.sort_values(['datetime', 'km']).reset_index(drop = True)
```

Page 1 of 1

รูปที่ 4.1 ตัวอย่างโปรแกรมการอ่านค่าจากไฟล์ .CSV

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2 การเทรนโมเดล

หลังจากได้ข้อมูลเข้ามาอยู่ในตัวแปร Data Frame และพร้อมงานแล้ว จึงเลือกเฉพาะ attribute ที่ต้องการในการเทรนโมเดลซึ่งในที่นี้จะใช้ หมายเลขหลักกิโล และเวลาซึ่งอยู่ในรูปนาที ของเรคคอร์ดที่ได้จากการ Preprocess เป็นตัวแปร X และ attribute ที่ต้องการทำนาย คือ avg_speed และall_units (ทั้งสอง attributes นั้นเทรนแยกกันคนละโมเดล) เป็นตัวแปร Y จากนั้นทำการ Normalization กับทั้ง X และ Y เพื่อลดขอบเขตของ attribute ต่าง ๆ ลงให้อยู่ในช่วง 0 – 1

และใช้ library Keras ช่วยในการสร้าง Neural Network ตามที่ออกแบบไว้ในบทที่ 3 ใช้ ฟังชั่นในการวัดผลความแม่นยำเป็น Mean Absolute Error และใช้ตัวเพิ่มประสิทธิภาพ (Optimizer) เป็นแบบ Adam โดยกำหนดขนาดของชุดข้อมูลในการเทรน (Batch size) เป็น 250,000 แถว เทรนทั้งหมด 120 รอบ หลังจากเสร็จสิ้นการเทรนจะทำการบันทึกโมเดลที่เทรนเรียบร้อยเป็นไฟล์นามสกุล .hdf5 โดยตัวอย่างโปรแกรมที่ใช้ดังรูปที่ 4.2

ทำการเทรน 4 ครั้ง ดังนี้

1. ถนนสาย 2 ขาเข้า กำหนดค่า Y เป็น Avg_speed (speed model)
2. ถนนสาย 2 ขาเข้า กำหนดค่า Y เป็น All_units (units model)
3. ถนนสาย 2 ขาออก กำหนดค่า Y เป็น Avg_speed (speed model)
4. ถนนสาย 2 ขาออก กำหนดค่า Y เป็น All_units (units model)

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดักแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
import pandas as pd
import numpy as np
from datetime import datetime, timedelta
from read_data import extract_data, plotHeatMap
import tensorflow as tf

from keras.callbacks import ModelCheckpoint, TensorBoard
from keras.models import Sequential
from keras.layers import Dense, Activation, Flatten
from keras.models import load_model, model_from_json

from sklearn.preprocessing import MinMaxScaler

def getunique(a):
    unique, counts = np.unique(a, return_counts=True)
    return dict(zip(unique, counts))

def denormalize(df, norm_data, Y):
    df = df[Y].values.reshape(-1,1)
    norm_data = norm_data.reshape(-1,1)
    scl = MinMaxScaler()
    a = scl.fit_transform(df)
    new = scl.inverse_transform(norm_data)
    return new

def trainDeep(filename, Y, df=None):

    if Y == 'all_units':
        df = df.drop(df[df[Y] == 0].index)
    scaler = MinMaxScaler(feature_range=(0, 1))

    # Normalization :
    X_train = scaler.fit_transform(df[['km', 'min']].values)
    y_train = scaler.fit_transform(df[[Y]].values)

    model = Sequential()
    model.add(Dense(128, kernel_initializer='normal',input_dim = 2, activation='relu'))
    # The Hidden Layers :
    model.add(Dense(256, kernel_initializer='normal',activation='relu'))
    model.add(Dense(256, kernel_initializer='normal',activation='relu'))
    model.add(Dense(256, kernel_initializer='normal',activation='relu'))
    # The Output Layer :
    model.add(Dense(1, kernel_initializer='normal',activation='linear'))

    model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['accuracy'])
    model.summary()

    model.fit(X_train, y_train, epochs=120, batch_size=250000, validation_split = 0.2)
    model.save('./Transport_Deep/road1/' + filename + '.hdf5')

def main(df=None):

    df = extract_data(1, 31, '07', 'in')
    trainDeep('road2_in_avg_speed', 'avg_speed', df)
    trainDeep('road2_in_units', 'all_units', df)

    df = extract_data(1, 31, '07', 'out')
    trainDeep('road2_out_avg_speed', 'avg_speed', df)
    trainDeep('road2_out_units', 'all_units', df)


if __name__ == "__main__":
    main()
```

Page 1 of 1

รูปที่ 4.2 ตัวอย่างโปรแกรมในการเทรนโมเดล

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.3 การทดสอบโมเดล

เมื่อเสร็จสิ้นการเทรนโมเดลแล้วจะกลับไปทำการ Preprocessing อีกครั้งหนึ่งกับข้อมูลในเดือนอื่น ที่ไม่ใช่เดือนที่ใช้เทรนเพื่อนำมาใช้ในการทดสอบโมเดล

ในการทดสอบ จะทำการโหลดโมเดลที่บันทึกไว้ในไฟล์ .hdf5 ขึ้นมา จากนั้นทำการ Normalization ข้อมูลหมายเลขหลักกิโลและเวลาซึ่งอยู่ในรูปนาที (ค่า X) แล้วใช้ค่า X ที่ผ่านการ Normalization แล้วป้อนเข้าสู่โมเดลเพื่อให้โมเดล ทำนายค่า Y (avg_speed หรือ all_units) ตามโมเดลที่ใช้ จากนั้นนำผลลัพธ์ที่ได้มาทำการ Denormalization แล้วนำมาวัดผลกับค่า Y จริงจากข้อมูลดิบ สุดท้ายใช้ผลลัพธ์ที่ได้จากโมเดลมาพล็อตเป็นกราฟ Heatmap เพื่อดูแนวโน้มของข้อมูลที่ได้จากการทำนายเทียบกับแนวโน้มของข้อมูลดิบด้วยตาอีกครั้งหนึ่ง โดยตัวอย่างโปรแกรมที่ใช้ดังรูปที่ 4.3 และ 4.4

```
import pandas as pd
import numpy as np
from datetime import datetime, timedelta
from read_data import extract_data, plotHeatMap
import tensorflow as tf

from keras.callbacks import ModelCheckpoint, TensorBoard
from keras.models import Sequential
from keras.layers import Dense, Activation, Flatten
from keras.models import load_model, model_from_json

from sklearn.preprocessing import MinMaxScaler

def getunique(a):
    unique, counts = np.unique(a, return_counts=True)
    return dict(zip(unique, counts))

def denormalize(df, norm_data, Y):
    df = df[Y].values.reshape(-1,1)
    norm_data = norm_data.reshape(-1,1)
    scl = MinMaxScaler()
    a = scl.fit_transform(df)
    new = scl.inverse_transform(norm_data)
    return new

def testDeep(data, filename, Y, month):

    df = pd.DataFrame(columns='datetime,km,min'.split(',')
)
    df[['datetime', 'km', 'min']] = data[['datetime', 'km'
, 'min']]

    model = load_model('./Transport_Deep/' + filename + '.
hdf5')

    # Normalization :
    scaler = MinMaxScaler()
    X_test = scaler.fit_transform(df[['km', 'min']].values
)
    y_test = scaler.fit_transform(data[[Y]].values)

    # Prediction :
    pred = model.predict(X_test)
    # Denormalization :
    pred = denormalize(data, pred, Y)
```

รูปที่ 4.3 ตัวอย่างโปรแกรมในการทดสอบโมเดล

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    score = model.evaluate(X_test, y_test, verbose=1
    print(model.metrics_names)
    print(score)

    df[:] = pred
    df.to_csv('./Transport_Deep/result/2018-' + month + '/
data_result_' + filename + '.csv', sep='\t', encoding='utf
-8')

    plotHeatMap(df,'min','km', Y,  filename, month)

def main(df=None):

    df = extract_data(1, 31, '08', 'in')
    testDeep(df, 'road2_in_avg_speed', 'avg_speed', '08')
    testDeep(df, 'road2_in_units', 'all_units', '08')

    df = extract_data(1, 31, '08', 'out')
    testDeep(df, 'road2_out_avg_speed', 'avg_speed', '08')
    testDeep(df, 'road2_out_units', 'all_units', '08')

if __name__ == "__main__":
    main()
```

รูปที่ 4.4 ตัวอย่างโปรแกรมในการทดสอบโมเดล

4.1.4 การตรวจหาความผิดปกติ

ในส่วนนี้จะเป็นการใช้โมเดลที่มีในการตรวจหาความผิดปกติที่มีความเป็นไปได้ว่าอาจเป็นอุบัติเหตุบนข้อมูลดิบที่มี โดยจะแบ่งออกเป็น 2 ส่วนดังนี้

ส่วนแรกเป็นส่วนของการทำนาย "ความปกติ" โดยเริ่มจากการโหลดโมเดลที่บันทึกไว้ทั้ง Speed model และ Units model ของถนนฝั่งใดฝั่งหนึ่ง (ขาเข้าหรือขาออก) จากนั้นใช้โมเดลทั้งสองทำนาย "รูปแบบทั่วไป" ของความเร็วและจำนวนรถในแต่ละวันจากข้อมูลดิบ โดยทำการ Preprocessing และ Normalize ข้อมูลดิบก่อนป้อนเข้าโมเดลเหมือนกับตอนทดสอบโมเดล สุดท้ายนำผลลัพธ์ที่ได้จากโมเดลมาทำ Denormalize อีกครั้งหนึ่ง

ส่วนที่สองเป็นส่วนของการตรวจหา "ความผิดปกติ" ซึ่งเมื่อมีทั้งข้อมูลดิบ และข้อมูลที่ใกล้เคียงกับสภาพปกติของความเร็วและจำนวนรถบนท้องถนนแล้ว นำข้อมูลทั้งสองชุดมาทำการคำนวนตามสมการ Incident Coefficient ในบทที่ 3 ในลักษณะจุดต่อจุด กล่าวคือจะใช้ข้อมูลดิบและผลทำนายจากโมเดลที่อยู่ในหลักกิโลเมตรและช่วงเวลาเดียวกันมาเข้าสมการ เช่น ในเวลา 20.21 ของวันที่ 22 เดือนกรกฎาคม ที่หลักกิโลเมตรที่ 12 บนถนนสาย 2 ข้อมูลดิบบอกว่ามีความเร็วเฉลี่ย 15 กม/ชม. และมีจำนวนรถ 20 คัน และผลทำนายจากโมเดลบอกว่า ณ เวลาและหลักกิโลเมตรดังกล่าวควรมีความเร็วเฉลี่ย 40 กม/ชม และมีจำนวนรถ 5 คัน ดังนั้นค่า incident coefficient ณ เวลาและหลักกิโลเมตรดังกล่าวจะเท่ากับ $\frac{40}{15} \times \frac{20}{5}$

เมื่อคำนวณค่า incident coefficient ครบทุกจุดที่มีบนชุดข้อมูล (ครบทุกแถว) แล้วจะนำข้อมูลมาพล็อตอยู่ในรูปกราฟ 3 มิติโดย X คือเวลา Y คือหลักกิโลเมตร และ Z คือค่า coefficient จากนั้นตรวจสอบว่ามีบริเวณไหนเกิดการเปลี่ยนแปลงของค่า coefficient อย่างกะทันหันบ้าง จึงนำวันเวลา และหลักกิโลเมตร ณ จุด ๆนั้นไปค้นหาจากแหล่งข้อมูลอุบัติเหตุที่เชื่อถือได้ว่า ณ จุด

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังกล่าวที่ค่า coefficient เปลี่ยนแปลงกะทันหันมีเหตุการณ์ไม่ปกติใดเกิดขึ้นจริงหรือไม่ โดยตัวอย่างโปรแกรมที่ใช้ดังรูปที่ 4.5 และ 4.6

```
import pandas as pd
import numpy as np
from read_data import extract_data, plotHeatMap
from datetime import datetime, timedelta, time, date
from keras.models import load_model
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from sklearn.preprocessing import MinMaxScaler

def denormalize(df, norm_data, Y):
    df = df[Y].values.reshape(-1,1)
    norm_data = norm_data.reshape(-1,1)
    scl = MinMaxScaler()
    a = scl.fit_transform(df)
    new = scl.inverse_transform(norm_data)
    return new

def findIncidentThreshold(data):

    date = '01,02,03,04,05,06,07,08,09,10,11,12,13,14,15,
  16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31'.split(','
  )
    speed_model = load_model('./Transport_Deep/
  road2_in_avg_speed.hdf5')
    units_model = load_model('./Transport_Deep/
  road2_in_units.hdf5')
    for d in date:
        data = extract_data(int(d), int(d), '07', 'out')
        df = pd.DataFrame(columns='datetime,km,min,
  speedAct,unitsAct'.split(','))
        df[['datetime', 'km', 'min', 'speedAct', 'unitsAct
  ']] = data[['datetime', 'km', 'min', 'avg_speed', '
  all_units']]

        scaler = MinMaxScaler()
        X_test = scaler.fit_transform(df[['km', 'min']].
  values)

        # Speed Prediction
        speed = speed_model.predict(X_test)
        speed = denormalize(data, speed, 'avg_speed')

        # Units Prediction
        units = units_model.predict(X_test)
        units = denormalize(data, units, 'all_units')
```

รูปที่ 4.5 ตัวอย่างโปรแกรมในการทดสอบโมเดลเพื่อตรวจหาความผิดปกติ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        df['avg_speed'] = np.abs(speed)
        df['all_units'] = np.ceil(np.abs(units))

        def calculate(row):
            if row['speedAct'] !=0 and row['all_units'] != 0:
                return (row['avg_speed'] / row['speedAct']) * (row['unitsAct'] / row['all_units'] )
            else:
                return -1

        # Calculate Coefficient for each row
        df['threshold'] = df.apply(calculate, axis=1)
        df = df[(df['threshold'] != -1) & (df['threshold'] > 10)]

        # Plot Coefficient 3D Graph
        fig = plt.figure(figsize = (20,12))
        ax = fig.add_subplot(111, projection='3d')
        ax.plot_trisurf(df['min']
                        , df['xm']
                        , df['threshold']
                        , color=(0,1,1,1)
        )
        fig.canvas.draw()
        labels = [item.get_text() for item in ax.get_xticklabels()]

        for i in range(len(labels)):
            if labels [i] != '':
                labels[i] = df['datetime'].iloc[0].strftime('%Y-%m-%d ') + str(timedelta(minutes=int(labels[i])))

        ax.set_xticklabels(labels)
        plt.savefig('./Application/2018-07/pic/out/Day_' + d + '.png', format='png', dpi=300)
    # plt.show()

def main(df=None):
    findIncidentThreshold(df)

if __name__ == "__main__":
    main()
```

รูปที่ 4.6 ตัวอย่างโปรแกรมในการทดสอบโมเดลเพื่อตรวจหาความผิดปกติ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ผลการทดลอง

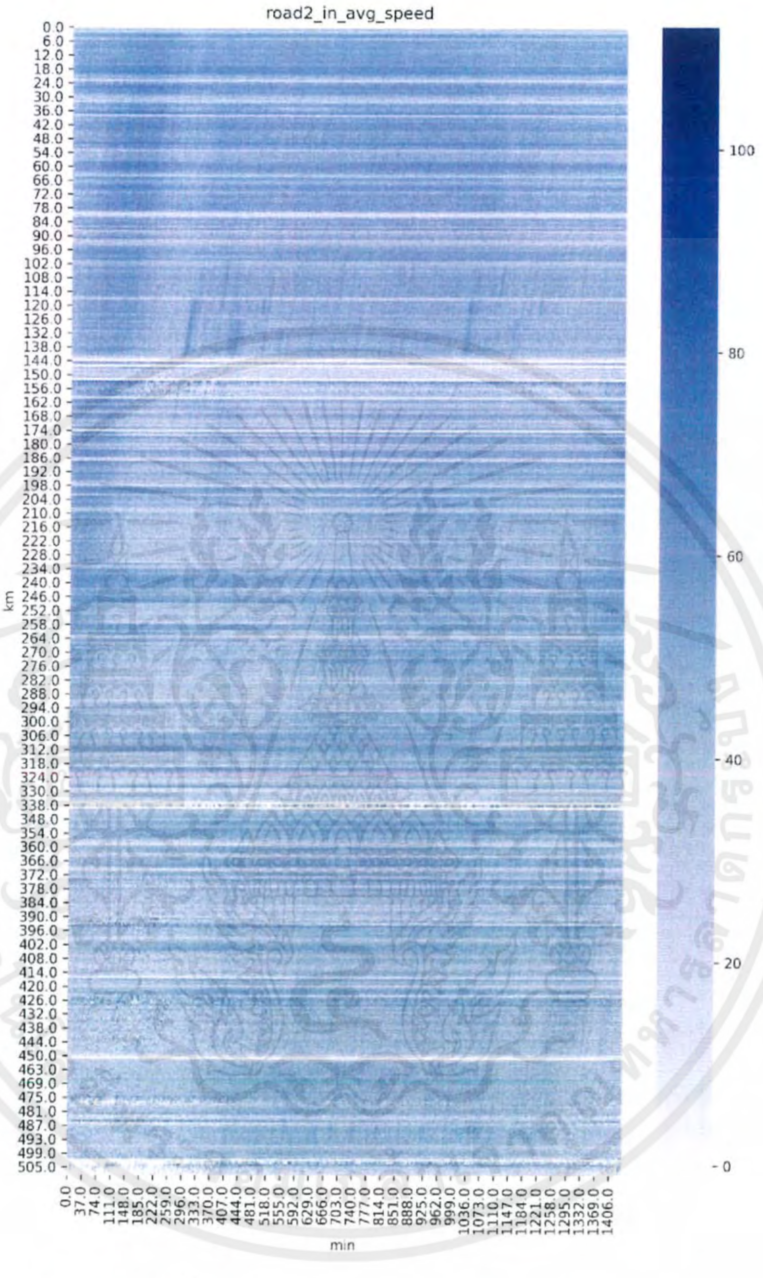
4.2.1 ผลจากโมเดล Speed และ Units

ผลการทดสอบโมเดล Speed และ Units โมเดลในรูปของกราฟฮีทแมพ เปรียบเทียบกับค่าเฉลี่ยที่คำนวณจากข้อมูลดิบและเปอร์เซ็นต์ความผิดพลาดที่วัดด้วยอัลกอริทึม Mean Absolute Error

ผลการทดลองโมเดลความเร็วของรถบนถนนทางหลวงหมายเลข 2 ขาเข้าซึ่งมีเปอร์เซ็นต์ความผิดพลาด (Error rate) : 0.288 ดังรูปที่ 4.7 เมื่อเทียบกับการหาความเร็วเฉลี่ยของรถบนถนนทางหลวงหมายเลข 2 ขาเข้า ดังรูปที่ 4.8

รูปที่ 4.7 กราฟฮีทแมพค่าเฉลี่ยความเร็วรถของถนนทางหลวงหมายเลข 2 ขาเข้าภายในเดือนกรกฎาคม 2561 ที่ผ่านโมเดล

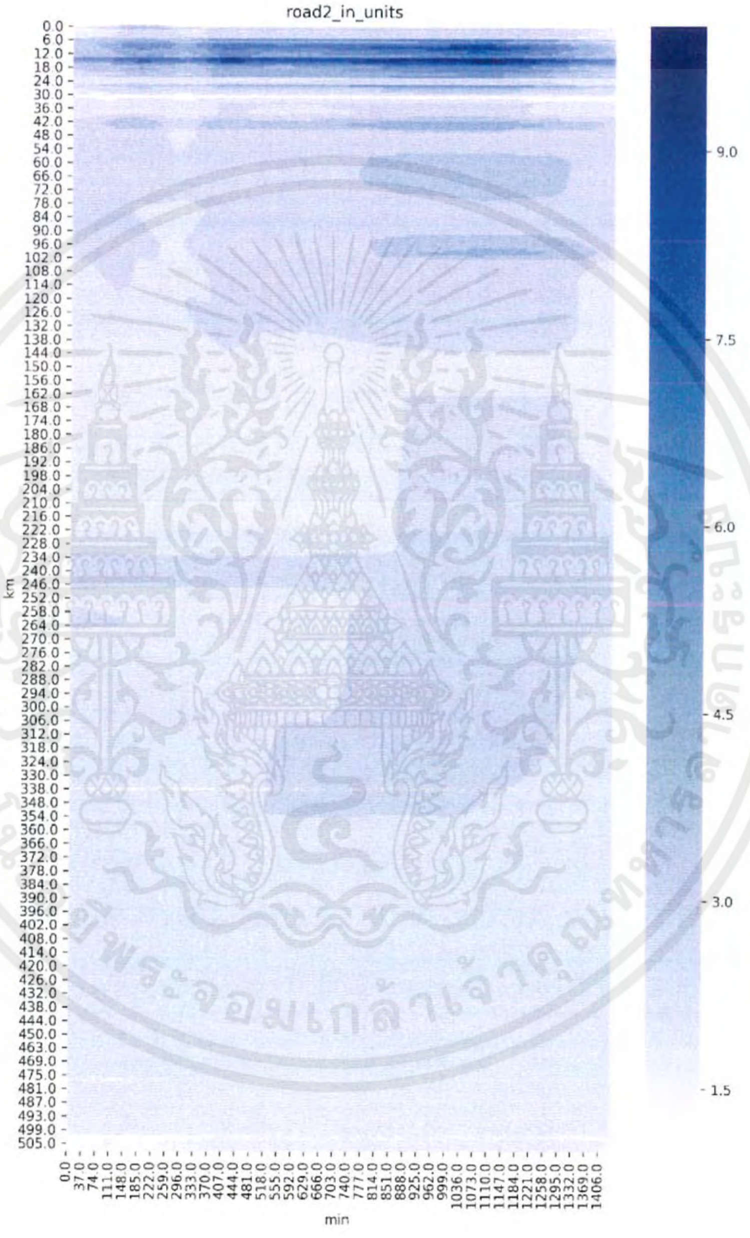
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.8 กราฟฮีทแมพค่าเฉลี่ยความเร็วรถของถนนทางหลวงหมายเลข 2 ขาเข้าภายในเดือนกรกฎาคม 2561 จากการหาค่าเฉลี่ย

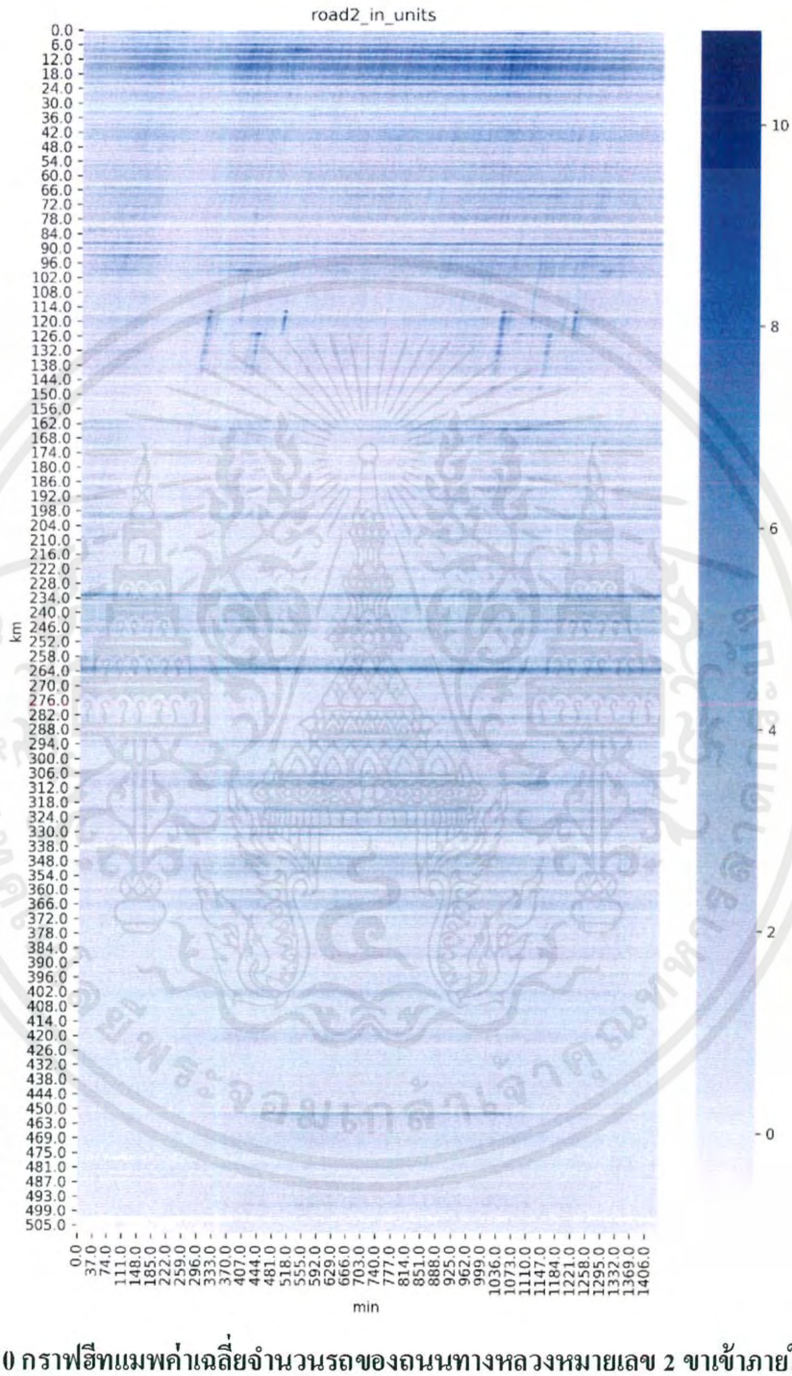
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลองโมเดล Units ถนนทางหลวงหมายเลข 2 ขาเข้าซึ่งมีเปอร์เซ็นต์ความผิดพลาด (Error rate) : 0.342 ดังรูปที่ 4.9 เมื่อเทียบกับการหาจำนวนเฉลี่ยของรถบนถนนทางหลวงหมายเลข 2 ขาเข้าดังรูปที่ 4.10



รูปที่ 4.9 กราฟฮีทแมพค่าเฉลี่ยจำนวนรถของถนนทางหลวงหมายเลข 2 ขาเข้าภายในเดือนกรกฎาคม 2561 ที่ผ่านโมเดล

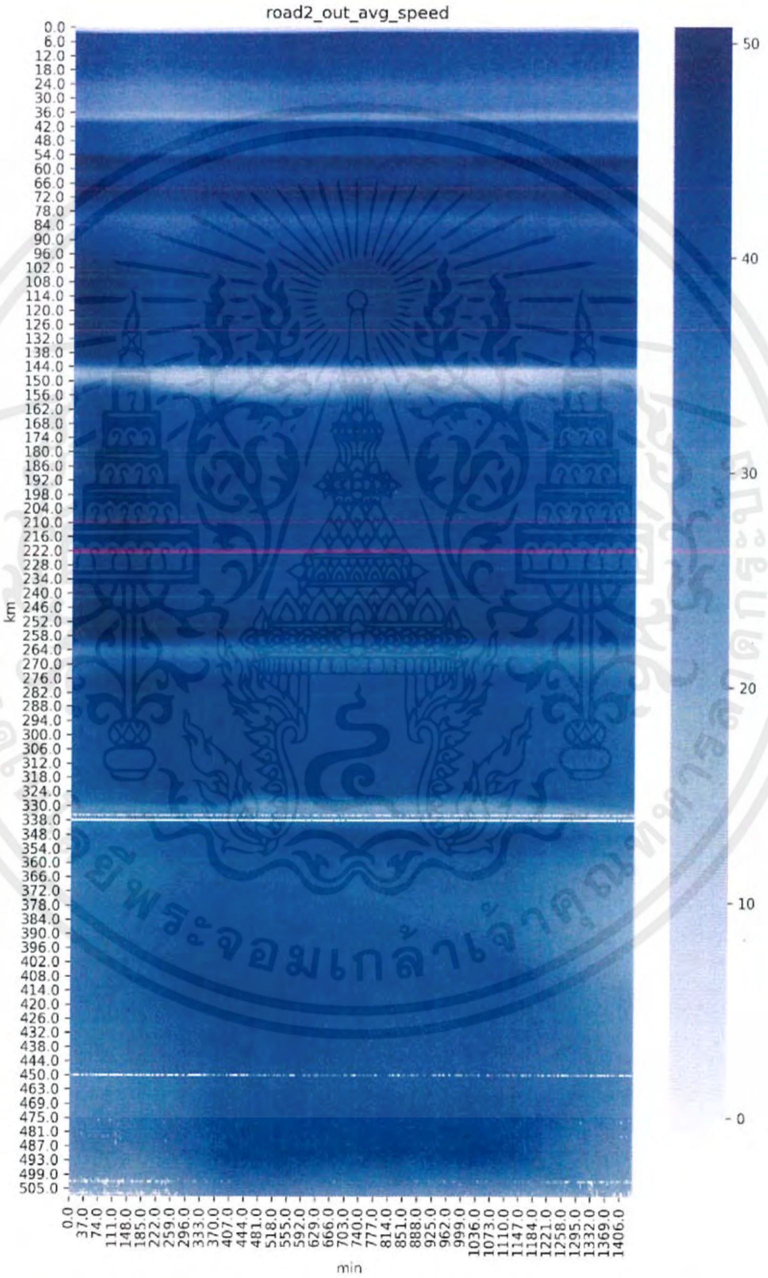
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 กราฟฮีทแมพค่าเฉลี่ยจำนวนรถของถนนทางหลวงหมายเลข 2 ขาเข้าภายในเดือนกรกฎาคม 2561 จากการหาค่าเฉลี่ย

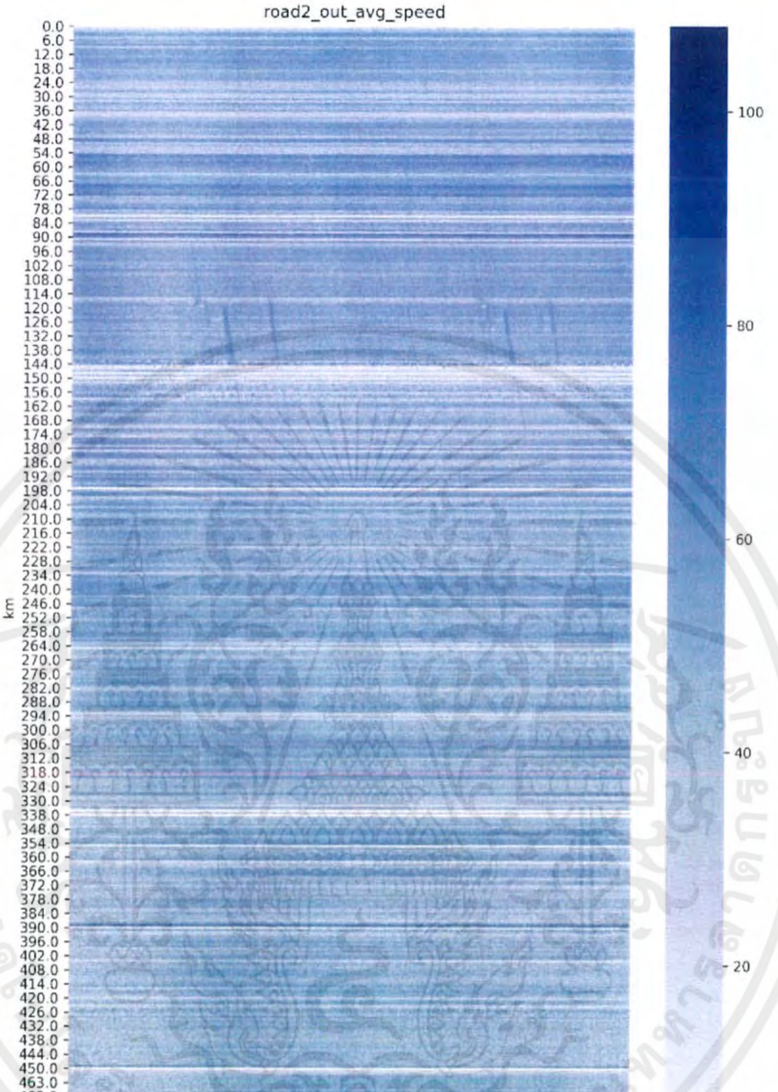
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลองโมเดล Speed ถนนทางหลวงหมายเลข 2 ขาออกซึ่งมีเปอร์เซ็นต์ความผิดพลาด (Error rate) : 0.208 ดังรูปที่ 4.11 เมื่อเทียบกับการหาความเร็วเฉลี่ยของรถบนถนนทางหลวงหมายเลข 2 ขาออกดังรูปที่ 4.12



รูปที่ 4.11 กราฟฮีทแมพค่าเฉลี่ยความเร็วรถของถนนทางหลวงหมายเลข 2 ขาออกภายในเดือนกรกฎาคม 2561 ที่ผ่านโมเดล

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



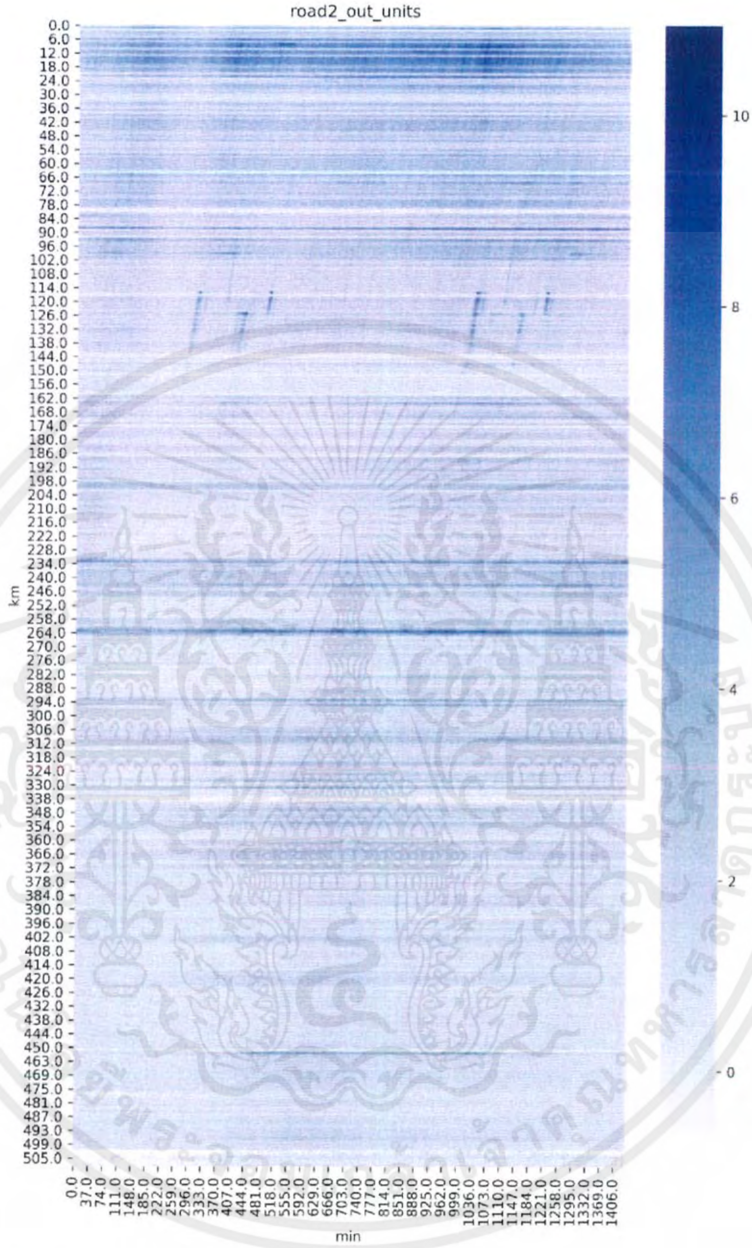
รูปที่ 4.12 กราฟฮีทแมพค่าเฉลี่ยความเร็วรถของถนนทางหลวงหมายเลข 2 ขาออกภายในเดือนกรกฎาคม 2561 จากการหาค่าเฉลี่ย

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลองโมเดล Units ถนนทางหลวงหมายเลข 2 ขาออกซึ่งมีเปอร์เซ็นต์ความผิดพลาด (Error rate) : 0.195 ดังรูปที่ 4.13 เมื่อเทียบกับการหาจำนวนเฉลี่ยของรถบนถนนทางหลวงหมายเลข 2 ขาออกดังรูปที่ 4.14

รูปที่ 4.13 กราฟฮีทแมพค่าเฉลี่ยจำนวนรถของถนนทางหลวงหมายเลข 2 ขาออกภายในเดือนกรกฎาคม 2561 ที่ผ่านโมเดล

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

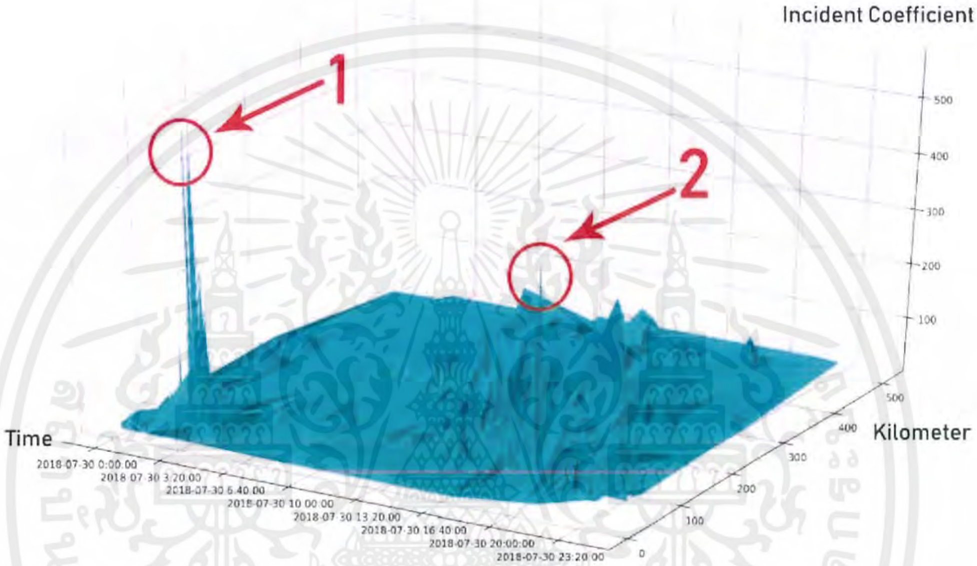


รูปที่ 4.14 กราฟฮีทแมพค่าเฉลี่ยจำนวนรถของถนนทางหลวงหมายเลข 2 ขาออกภายในเดือนกรกฎาคม 2561 จากการหาค่าเฉลี่ย

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 ผลการตรวจหาความผิดปกติ

ผลการทดลองการตรวจหาความผิดปกติด้วยผลลัพธ์จากการทำนายของโมเดลและข้อมูลจริงโดยเปรียบเทียบแบบวันต่อวันกับแหล่งข่าวที่น่าเชื่อถือ เช่น จส.100 ,สวพ.91 , HAIMS (Highway Accident Information Management System)



รูปที่ 4.15 กราฟแสดงค่าสัมประสิทธิ์ในวันที่ 30 กรกฎาคม 2561

ผลการตรวจสอบอุบัติเหตุจากเว็บของจส.100

ข้อมูลจราจร ถนนมิตรภาพ
30 กรกฎาคม 2561, 11:06น.
อุบัติเหตุ ถนนมิตรภาพ ขาเข้า จากอำเภอสีคิ้ว จังหวัดนครราชสีมา มุ่งหน้าเขื่อนลำตะคอง จุดเกิดเหตุช่วงกิโลเมตรที่93 รถชนกัน 6 คัน กีดขวางช่องทางขวา

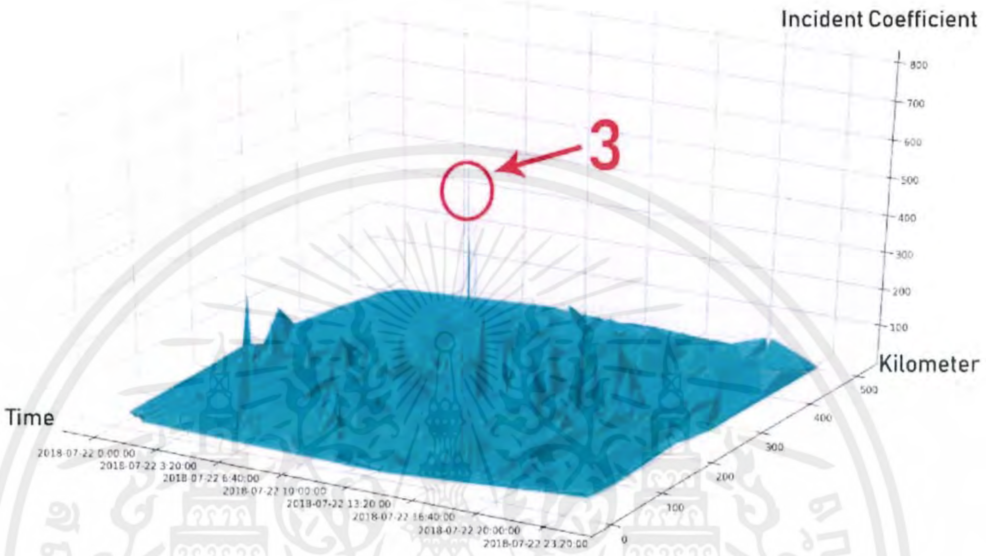
ข้อมูลจราจร ถนนมิตรภาพ
30 กรกฎาคม 2561, 04:42น.
อุบัติเหตุ ถนนมิตรภาพ ขาเข้า จากนครราชสีมามุ่งหน้าสระบุรี กิโลเมตรที่8 เยื้องกับบริษัท ซีพีเอฟ อำเภอแก่งคอย จังหวัดสระบุรี รถทัวร์ชนท้ายกัน 3 คัน มีผู้บาดเจ็บ 1 คน การจราจรเคลื่อนตัวช้า

รูปที่ 4.16 ข่าวอุบัติเหตุจากเว็บจส.100ในวันที่ 30 กรกฎาคม 2561

จุดที่ 1 30 กรกฎาคม 2561 เวลา 04.42 น. เกิดอุบัติเหตุบริเวณ ถนนมิตรภาพ ขาเข้า จากนครราชสีมามุ่งหน้าสระบุรี หลักกิโลเมตรที่ 8 เยื้องกับบริษัท ซีพีเอฟ อำเภอแก่งคอย จังหวัดสระบุรี รถทัวร์ชนท้ายกัน 3 คัน

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จุดที่ 2 30 กรกฎาคม 2561 เวลา 19.38 น เกิดอุบัติเหตุบริเวณ ถนนมิตรภาพ ขาเข้า ช่วงปากช่อง มุ่งหน้านครราชสีมา ที่กิโลเมตร 82 มีรถทัวร์เสียหลักพลิกตะแคง



รูปที่ 4.17 กราฟแสดงค่าสัมประสิทธิ์ในวันที่ 22 กรกฎาคม 2561

ผลการตรวจสอบอุบัติเหตุจากเว็บของจส.100

ข้อมูลจราจร ถนนมิตรภาพ
22 กรกฎาคม 2561, 16:36น.
ถนนมิตรภาพ ช่วง อ.ปากช่อง จ.นครราชสีมา มุ่งหน้าเขื่อนลำตะคอง ที่เลยฟาร์มโชคชัย 4กม. รถพ่วงบรรทุกน้ำมันเสียหลักพลิกตะแคง ขวางทุกช่องทาง การจราจรติดขัด

รูปที่ 4.18 ข่าวอุบัติเหตุจากเว็บจส.100ในวันที่ 22 กรกฎาคม 2561

จุดที่ 3 22 กรกฎาคม 2561 เวลา 16.36 น. เกิดอุบัติเหตุบริเวณ ถนนมิตรภาพ อ.ปากช่อง จ.นครราชสีมุ่งหน้าเขื่อนลำตะคอง ที่เลยฟาร์มโชคชัย 4 กม. รถพ่วงบรรทุกน้ำมันเสียหลักพลิกตะแคง ขวางทุกช่องทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดักแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลและปัญหาที่พบ

5.1 สรุปผลการทดลอง

จากสมมติฐานข้างต้นได้พิสูจน์แล้วว่าโมเดลนี้สามารถตรวจพบว่ามีอุบัติเหตุเกิดขึ้นจริง แต่ก็ยังมีการตรวจพบในกรณีที่มีการจราจรติดขัดโดยไม่มีอุบัติเหตุเกิดขึ้น แต่มีสาเหตุมาจากสาเหตุอื่น

5.2 ปัญหาที่พบระหว่างการทดลอง

- ในการเทรน Neural Network นั้นใช้เวลาในกระบวนการเทรนโมเดลทั้งหมดค่อนข้างนาน ประมาณ 30 นาที - 1 ชั่วโมงต่อหนึ่งโมเดล ซึ่งในโครงการนี้จะต้องใช้ทั้งหมด 4 โมเดล และในบางครั้งอาจมีการปรับพารามิเตอร์ต่าง ๆ ของโมเดลแล้วทำการเทรนใหม่เช่นกัน
- ไลบรารี่บางตัวที่ใช้งานมี Dependency ค่อนข้างสูง ทำให้มีความซับซ้อนในการติดตั้งและในบางครั้งที่ ไลบรารี่มีการอัพเดตส่งผลให้ไม่รองรับ Dependency ตัวเก่า จึงต้องทำการอัพเดต Dependency ดังกล่าวตามหรือ ในกรณีที่แย่ยิ่งกว่าคืออาจต้องการทำติดตั้งไลบรารี่ใหม่ทั้งหมด
- ยังคงมี False Positive หรือผลบวกเทียมในกรณีที่ ค่าสัมประสิทธิ์ (Incident Coefficient) มีการเปลี่ยนแปลงขึ้นสูงอย่างกระทันหันซึ่งตามสมมติฐานจะสรุปว่าเกิดเหตุการณ์ไม่ปกติขึ้น ณ เวลาและหลักกิโลเมตรนั้น ๆ ทว่าเมื่อตรวจสอบกับแหล่งข่าวที่เชื่อถือได้หลายแห่งกลับไม่พบเหตุการณ์ใด ๆ

5.3 แนวทางการศึกษาต่อ

- มีการกรองชุดข้อมูลที่ใช้เทรนโมเดลโดยแยกข้อมูลของวันหยุดวันหยุดนักขัตฤกษ์และวันเทศกาลต่าง ๆ ออกเพื่อให้ได้ชุดข้อมูลที่บ่งบอกถึง "สภาพปกติ" ของถนนจริง ๆ และข้อมูลที่ถูกกรองออกอาจนำไปสร้างอีกหนึ่งโมเดลสำหรับวันเทศกาลโดยเฉพาะ
- เปลี่ยนระบบทั้งหมดจากที่ตรวจหาเหตุการณ์ผิดปกติบนชุดข้อมูล (Dataset) ซึ่งถูกเก็บมาในอดีตมาเป็นระบบที่สามารถตรวจจับเหตุการณ์ผิดปกติบนข้อมูลที่ถูกส่งมาจาก GPS ของรถที่กำลังวิ่งอยู่แบบเรียลไทม์

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดักแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

Natthawat P. 2560. **มาทำความรู้จัก Machine Learning เบื้องต้น**

Available : https://medium.com/convolab/machine-learning-basics-2b38700cb10b

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดักแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้