

การศึกษาการประยุกต์เทคโนโลยีการเรียนรู้เชิงลึกสำหรับการจัดการจราจรบน
เครือข่ายเอสดีเอ็น

A STUDY ON DEEP LEARNING APPLICATION FOR SDN-BASED
TRAFFIC MANAGEMENT

โดย

บุญญพัฒน์ แปลงพระเนตร

Punyapat Plangpranet

ศรัณญ์ รื่นรวย

Sarunyu Ruenruay

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

สาขาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ภาคเรียนที่ 1 ปีการศึกษา 2564

การศึกษาการประยุกต์เทคโนโลยีการเรียนรู้เชิงลึกสำหรับการจัดการจราจรบน
เครือข่ายเอสดีเอ็น

A STUDY ON DEEP LEARNING APPLICATION FOR SDN-BASED
TRAFFIC MANAGEMENT

โดย

บุญญพัฒน์ แปลงพระเนตร

ศรัณญ์ญ รื่นรวย

อาจารย์ที่ปรึกษา

ผู้ช่วยศาสตราจารย์ ดร. สุเมธ ประภาวัต

รองศาสตราจารย์ ดร. กิตติสุชาติ พสุภา

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

สาขาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ภาคเรียนที่ 1 ปีการศึกษา 2564

**A STUDY ON DEEP LEARNING APPLICATION FOR SDN-BASED
TRAFFIC MANAGEMENT**

PUNYAPAT PLANGPRANET

SARUNYU RUENRUAY

**A PROJECT SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
BACHELOR OF SCIENCE PROGRAM IN INFORMATION TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY
KIING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

1/2021

COPYRIGHT 2021

FACULTY OF INFORMATION TECHNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ใบรับรองปริญญาโท ประจำปีการศึกษา 2564

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การศึกษาการประยุกต์เทคโนโลยีการเรียนรู้เชิงลึกสำหรับการจัดการจราจร
บนเครือข่ายเอสดีเอ็น

A STUDY ON DEEP LEARNING APPLICATION FOR
SDN-BASED TRAFFIC MANAGEMENT

ผู้จัดทำ

1. นายบุญญพัฒน์ แปลงพระเนตร รหัสนักศึกษา61070122
2. นายศรัณญ์ รื่นรวย รหัสนักศึกษา61070215

..... อาจารย์ที่ปรึกษา

(ผศ.ดร. สุเมธ ประภาวัต)

..... อาจารย์ที่ปรึกษาร่วม

(รศ.ดร. กิติ์สุชาติ พสุภา)

ใบรับรองโครงการ (PROJECT)

เรื่อง

การศึกษาการประยุกต์เทคโนโลยีการเรียนรู้เชิงลึกสำหรับการจัดการจราจรบน

เครือข่ายเอสดีเอ็น

**A STUDY ON DEEP LEARNING APPLICATION FOR SDN-BASED
TRAFFIC MANAGEMENT**

นายบุญญพัฒน์

แปลงพระเนตร

รหัสประจำตัว 61070122

นายศรัณญ์

รินรว

รหัสประจำตัว 61070215

ขอรับรองว่ารายงานฉบับนี้ ข้าพเจ้าไม่ได้คัดลอกมาจากที่ใด
รายงานฉบับนี้ได้รับการตรวจสอบและอนุมัติให้เป็นส่วนหนึ่งของ
การศึกษาวิชาโครงการ หลักสูตรวิทยาศาสตรบัณฑิต (เทคโนโลยีสารสนเทศ)
ภาคเรียนที่ 1 ปีการศึกษา 2564

.....

(นายบุญญพัฒน์ แปลงพระเนตร)

.....

(นายศรัณญ์ รินรว)

หัวข้อวิทยานิพนธ์	การศึกษาการประยุกต์เทคโนโลยีการเรียนรู้เชิงลึกสำหรับการจัดการ จราจรบนเครือข่ายเอสดีเอ็น		
นักศึกษา	นายบุญญพัฒน์ แปลงพระเนตร	รหัสนักศึกษา	61070122
	นายศรัณญ์ รื่นรวย	รหัสนักศึกษา	61070215
สาขาวิชา	เทคโนโลยีสารสนเทศ		
ปีการศึกษา	2564		
อาจารย์ที่ปรึกษา	ผศ.ดร. สุเมธ ประภาวัต		
อาจารย์ที่ปรึกษาร่วม	รศ.ดร. กิติ์สุชาติ พสุภา		

บทคัดย่อ

ระบบเครือข่ายในปัจจุบันได้มีการขยายตัวจนมีขนาดใหญ่ และมีความซับซ้อนมากขึ้นส่งผลให้การบริหารจัดการเป็นไปด้วยความยุ่งยาก และหนึ่งในปัญหาสำคัญคือ ความแออัดในระบบเครือข่าย อันเนื่องมาจากการเลือกเส้นทางที่ไม่มีประสิทธิภาพประกอบกับการขยายตัวของผู้ใช้งานที่เพิ่มขึ้น ซึ่งระบบเครือข่ายแบบดั้งเดิมไม่สามารถเลียงปัญหานี้ได้ โครงการนี้จึงได้เสนอการประยุกต์ใช้การเรียนรู้เชิงลึกในการจัดการจราจรบนเครือข่ายที่กำหนดด้วยซอฟต์แวร์ เพื่อหลีกเลี่ยงความคับคั่งและสามารถใช้งานเครือข่ายได้อย่างมีประสิทธิภาพมากที่สุด

Project Title	A Study on Deep Learning Application for SDN-Based Traffic Management	
Student	Mr. Punyapat Plangpranet	Student ID 61070122
	Mr. Saruny Ruenruay	Student ID 61070215
Degree	Bachelor of Science	
Program	Information Technology	
Advisor	Asst. Prof. Dr. Sumet Prabhavat	
Co-Advisor	Assoc. Prof. Dr. Kitsuchart Pasupa	

ABSTRACT

In these days, The network system are growing and complicated, so it is difficult for the network management. Network congestion is one of the main problems because of inefficient routing and increasing users. However a traditional network is not solve these problems. In this research presented Deep Learning Application for SDN-Based Traffic Management to avoid network congestion and use network efficiently

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สามารถสำเร็จสมบูรณ์ได้เป็นอย่างดีด้วยความกรุณาจากอาจารย์ที่ปรึกษาปริญญานิพนธ์ ผู้ช่วยศาสตราจารย์ ดร.สุเมธ ประภาวัต และรองศาสตราจารย์ ดร.กิติ์สุชาติ พสุภา ซึ่งเป็นผู้ให้คำปรึกษาในทุก ๆ ข้อสงสัยให้คำแนะนำ แนวคิดและแนวทางการดำเนินงาน ตลอดจนการทำปริญญานิพนธ์ฉบับนี้ อีกทั้งยังตรวจสอบแก้ไขข้อบกพร่องต่าง ๆ ด้วยความเอาใจใส่เป็นอย่างยิ่ง และคอยให้กำลังใจผู้จัดทำในการทำปริญญานิพนธ์ ผู้จัดทำขอขอบคุณเป็นอย่างสูง

ขอขอบพระคุณคณาจารย์คณะเทคโนโลยีสารสนเทศ ผู้เชี่ยวชาญและผู้ทรงคุณวุฒิทุกท่าน ที่ช่วยอบรมสั่งสอนวิชาความรู้ทั้งทางด้านวิชาการและด้านคุณธรรมตั้งแต่อดีตจนถึงปัจจุบัน ให้สามารถนำมาประยุกต์ใช้ในการทำปริญญานิพนธ์ฉบับนี้และสามารถนำไปต่อยอดการทำงานในอนาคตได้

ขอขอบคุณนายสุภวิชญ์ ศิริสวัสดิ์วัฒนา ที่ได้ให้ความช่วยเหลือ คำแนะนำ และอำนวยความสะดวก ตลอดจนคอยให้กำลังใจในการทำปริญญานิพนธ์ฉบับนี้ให้สำเร็จลุล่วงได้เป็นอย่างดี

สุดท้ายนี้ขอกราบขอบพระคุณ คุณพ่อ คุณแม่ และสมาชิกในครอบครัว ที่คอยอบรมให้คำแนะนำและเป็นแรงสนับสนุนในทุก ๆ ด้าน ตลอดจนขอขอบคุณบุคคลที่ผู้จัดทำมิได้กล่าวถึงที่ได้ให้ความช่วยเหลือในการดำเนินการให้ปริญญานิพนธ์ฉบับนี้สามารถสำเร็จสมบูรณ์ได้เป็นอย่างดี

บุญญพัฒน์ แปลงพระเนตร

ศรัณญ์ญู รื่นราย

สารบัญ

หน้า

บทคัดย่อ.....	I
ABSTRACT.....	II
กิตติกรรมประกาศ.....	III
สารบัญ	IV
สารบัญรูป	VI
สารบัญตาราง.....	VII
บทที่ 1 บทนำ	1
1.1 ที่มาและความสำคัญ	1
1.2 วัตถุประสงค์.....	2
1.3 ขอบเขตของโครงการ.....	2
1.4 ขั้นตอนการดำเนินงาน	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	2
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	3
2.1 ระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์ (Software Defined Networks : SDN)	3
2.2 โอเพนวิสวิตช์ (Open vSwitch)	4
2.3 การจัดการจราจรบนเครือข่าย.....	4
2.4 การเรียนรู้เชิงลึก (Deep Learning).....	5
2.5 โครงข่ายความจำระยะสั้นขนาดยาว (Long-Short Term Memory : LSTM).....	5
2.6 หน่วยเวียนกลับแบบมีประตู (Gated Recurrent Unit: GRU).....	9
2.7 มาตรวัดประสิทธิภาพ.....	10

บทที่ 3 วิธีการดำเนินงาน.....	12
__3.1 การปรับปรุงองค์ประกอบภายในคอนโทรลเลอร์	12
__3.2 อัลกอริทึมในการหาเส้นทางสำหรับการย้ายโฟลว์บนระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์ ..	14
__3.3 ข้อมูลที่ใช้ในการวิจัย.....	18
__3.4 การจัดการข้อมูล	18
__3.5 กำหนดโมเดลการเรียนรู้เชิงลึก.....	19
บทที่ 4 การทดลองและผลการทดลอง.....	21
__4.1 การทดลองคอนโทรลเลอร์	21
__4.2 ผลการทดสอบทดสอบโมเดลการเรียนรู้เชิงลึก	22
__4.3 ผลการทดสอบการทำงานของระบบ	23
บทที่ 5 สรุปผลการทดลอง	25
__5.1 สรุปผล	25
__5.2 แนวทางในการพัฒนาต่อ	25
บรรณานุกรม	26

สารบัญรูป

รูปที่	หน้า
รูปที่ 2.1 การเปรียบเทียบสถาปัตยกรรมของระบบเครือข่ายแบบดั้งเดิมกับระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์	3
รูปที่ 2.2 Two-Tier Mechanism.....	4
รูปที่ 2.3 โครงสร้างของการเรียนรู้เชิงลึกหรือโครงข่ายประสาทเทียม.....	5
รูปที่ 2.4 โครงสร้างของโครงข่ายความจำระยะสั้นขนาดยาว.....	6
รูปที่ 3.1 องค์ประกอบภายในคอนโทรลเลอร์.....	12
รูปที่ 3.2 แผนภาพขั้นตอนของอัลกอริทึมในการหาเส้นทางสำหรับย้ายโฟลว์.....	16
รูปที่ 3.3 Flow Table ภายในสวิตช์.....	17
รูปที่ 3.4 Sliding window.....	19
รูปที่ 3.5 รูปแบบข้อมูลนำเข้าและส่งออกสำหรับโมเดล.....	19
รูปที่ 4.1 โทโพโลยีแบบกำหนดเอง.....	21
รูปที่ 4.2 กราฟเปรียบเทียบเส้นโค้งการเรียนรู้ของแต่ละโมเดล.....	22
รูปที่ 4.3 กราฟเปรียบเทียบค่าจริงและค่าทำนายของแต่ละโมเดล.....	23
รูปที่ 4.4 กราฟเปรียบเทียบแสดงจำนวนการเกิดความแออัดในระยะเวลาหนึ่งชั่วโมง.....	24
รูปที่ 4.4 กราฟแสดงปริมาณการใช้งานของลิงก์ระหว่างสวิตช์ที่ 1 และสวิตช์ที่ 3.....	17

สารบัญตาราง

ตารางที่	หน้า
ตารางที่ 3.1 ตารางค่าพารามิเตอร์ของแต่ละโมเดล.....	20
ตารางที่ 4.1 ตารางเปรียบเทียบความคลาดเคลื่อนของแต่ละโมเดล.....	23

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญ

ระบบเครือข่ายในปัจจุบันได้มีการขยายตัวจนมีขนาดใหญ่ และประกอบไปด้วยอุปกรณ์ที่มีความหลากหลายส่งผลให้ระบบเครือข่ายมีความซับซ้อนมากขึ้น หนึ่งในปัญหาสำคัญคือ อุปกรณ์จะสามารถเลือกเส้นทางที่มีประสิทธิภาพภายใต้เครือข่ายที่มีความซับซ้อนได้อย่างไร การเลือกเส้นทางที่ไม่มีประสิทธิภาพประกอบกับการขยายตัวของผู้ใช้งานที่เพิ่มขึ้นนั้นอาจส่งผลให้เกิดความแออัดในเครือข่าย ซึ่งระบบเครือข่ายแบบดั้งเดิมไม่สามารถเล็งปัญหานี้ได้เพราะการหาเส้นทางในระบบเครือข่ายแบบดั้งเดิมจะต้องขึ้นอยู่กับอัลกอริทึมที่ใช้ในการหาเส้นทางในอุปกรณ์เครือข่าย ซึ่งใช้อัลกอริทึมที่แตกต่างกันขึ้นอยู่กับระบบเครือข่ายนั้น ๆ ซึ่งในบางครั้งจะได้เส้นทางที่ไม่เหมาะสมกับการส่งข้อมูลบางประเภท ปัจจุบันได้มีรูปแบบการจัดการระบบเครือข่ายที่มีประสิทธิภาพในการจัดการที่มีชื่อว่าระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์

เครือข่ายที่กำหนดโดยซอฟต์แวร์ (Software Defined Network : SDN) เป็นการแยกส่วนของการทำงานออกเป็น 2 ส่วน คือ ส่วนของการควบคุม (Control Plane) และส่วนของข้อมูล (Data Plane) ซึ่งส่วนของการควบคุมจะอยู่ในส่วนที่เรียกว่าคอนโทรลเลอร์ (Controller) และส่วนของข้อมูลจะอยู่ในส่วนของสวิตช์ (Switch) โดยที่เครือข่ายที่กำหนดโดยซอฟต์แวร์สามารถแก้ปัญหาดังกล่าวได้ด้วยการกระจายการจราจรจากการเลือกเส้นทางเพิ่มเติมที่เหมาะสม โดยใช้ความสามารถของตัวคอนโทรลเลอร์ที่สามารถมองภาพรวมทั้งหมดของเครือข่ายได้

การกระจายการจราจรโดยใช้เครือข่ายที่กำหนดโดยซอฟต์แวร์จะสามารถทำได้ก็ต่อเมื่อเกิดความคับคั่งขึ้นแล้วเท่านั้นทำให้อาจมีการสูญเสียแพ็คเก็ตบางส่วนไป จึงได้มีแนวคิดในการใช้การเรียนรู้เชิงลึก (Deep Learning) มาช่วยในการพยากรณ์การเกิดความคับคั่งในเครือข่ายแล้วปรับเปลี่ยนเส้นทางก่อนที่ความคับคั่งจะเกิดขึ้น ซึ่งจะช่วยให้เกิดการใช้เครือข่ายอย่างมีประสิทธิภาพและลดการเกิดปัญหาต่าง ๆ ลงไป

1.2 วัตถุประสงค์

1. เพื่อพัฒนากลไกจัดการจราจรบนเครือข่ายเอสดีเอ็น ให้สามารถกระจายการใช้เส้นทางก่อนที่จะเกิดความคับคั่ง
2. เพื่อศึกษาแนวทางการใช้เทคโนโลยีการเรียนรู้เชิงลึก เพื่อพยากรณ์การเกิดปัญหาความคับคั่งบนระบบเครือข่าย

1.3 ขอบเขตของโครงการ

1. ศึกษาการทดลองโดยใช้แบบจำลองระบบเครือข่ายบน Mininet
2. ศึกษาโดยใช้ Ryu SDN Framework เป็นส่วนควบคุมระบบเครือข่าย
3. ก่อนการนำ SDN Application และส่วนควบคุมระบบเครือข่ายไปใช้สำหรับสถาปัตยกรรมระบบเครือข่ายรูปแบบใด ๆ จะต้องมีการฝึกฝน โมเดลจากผลลัพธ์ที่ได้จากการปล่อยทราฟฟิกที่หลากหลาย

1.4 ขั้นตอนการดำเนินงาน

1. ศึกษาเทคโนโลยี SDN ในส่วนของส่วนควบคุมระบบเครือข่ายและแอปพลิเคชัน
2. ศึกษาการใช้งานการเรียนรู้เชิงลึกในการพยากรณ์ความคับคั่งของอุปกรณ์ในระบบเครือข่าย
3. ศึกษาการจำลองระบบเครือข่ายและการสร้างแบบจำลองบน Mininet
4. สร้างเครือข่ายที่มีการทำงานแบบ SDN ที่สามารถกระจายทราฟฟิกได้เมื่อเกิดความคับคั่ง
5. สร้างเครือข่ายที่มีการทำงานแบบ SDN ที่สามารถกระจายทราฟฟิกได้ก่อนเกิดความคับคั่ง
6. สร้างสถานการณ์จำลองสำหรับประเมินประกอบไปด้วย Traffic Condition, Network Topology และออกแบบการทดลอง
7. ทดลองและเก็บผลการทดลอง
8. สรุปการทดลอง

1.5 ประโยชน์ที่คาดว่าจะได้รับ

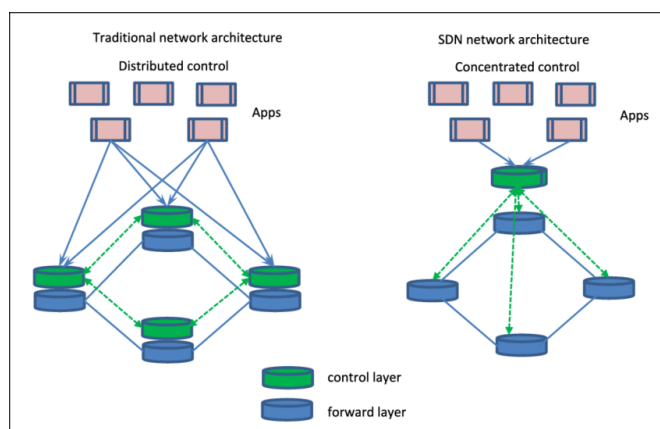
1. แอปพลิเคชัน SDN ที่ใช้ Deep Learning มาช่วยในการเลือกเส้นทาง
2. เส้นทางต่างๆ ถูกใช้อย่างมีประสิทธิภาพ

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์ (Software Defined Networks : SDN)

ระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์หรือ Software Defined Network [1] เป็นระบบเครือข่ายที่มีออกแบบมาเพื่อควบคุม และสั่งการจากส่วนกลางเพียงส่วนเดียว แต่เดิมอุปกรณ์เครือข่ายมีส่วนที่เรียกว่าส่วนควบคุม (Control Plane) และส่วนส่งข้อมูล (Data Plane) อยู่ภายในอุปกรณ์เดียวกัน ซึ่งสร้างความลำบากในการตั้งค่าต่าง ๆ ให้กับอุปกรณ์แต่ละตัวและอาจมีปัญหาดังต่าง ๆ เช่น การหาเส้นทางในการส่งข้อมูลจะต้องขึ้นอยู่กับอัลกอริทึมที่ใช้ในการหาเส้นทางในอุปกรณ์เครือข่ายแต่ละตัว ซึ่งในบางครั้งอาจไม่เหมาะสมกับข้อมูลบางประเภท ทำให้เกิดการสูญเสียแพ็กเก็ตและต้องทำการส่งใหม่เป็นต้น โดยแนวคิดของระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์จะทำการแยกส่วนของการควบคุมและส่วนของการส่งข้อมูลออกจากกัน ซึ่งส่วนของการควบคุมจะอยู่ในส่วนที่เรียกว่าคอนโทรลเลอร์ (Controller) และส่วนของการส่งข้อมูลจะอยู่ที่สวิตช์ (Switch) โดยที่ SDN นั้นสามารถแก้ปัญหาต่าง ๆ ของสถาปัตยกรรมเครือข่ายแบบเดิมได้ เพราะตัวคอนโทรลเลอร์มีซอฟต์แวร์ที่สามารถจัดการเครือข่ายทั้งหมดมารวมอยู่ในที่เดียว โดยคอนโทรลเลอร์จะสื่อสารกับสวิตช์โดยใช้โปรโตคอลมาตรฐานที่มีชื่อว่า OpenFlow



รูปที่ 2.1 การเปรียบเทียบสถาปัตยกรรมของระบบเครือข่ายแบบดั้งเดิมกับระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์

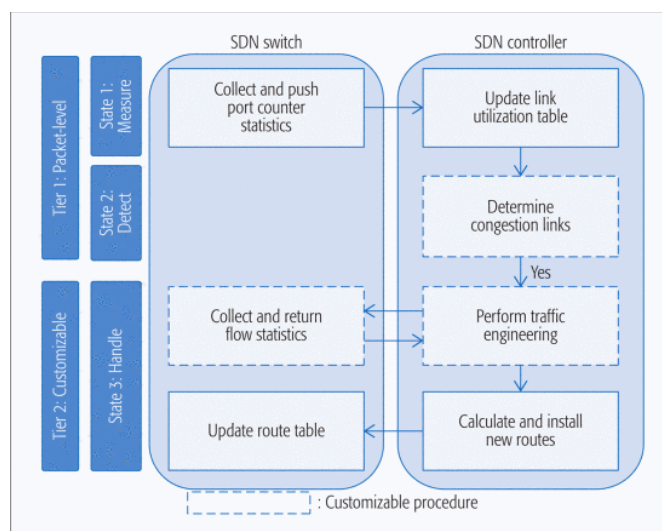
2.2 โอเพนวิสวิตช์ (Open vSwitch)

โอเพนวิสวิตช์ (Open vSwitch) เป็นเทคโนโลยีที่จำลองการทำงานของสวิตช์แบบมัลติเลเยอร์ (Multilayer Switch) โดยถูกออกแบบมาให้สามารถจัดการเครือข่ายแบบอัตโนมัติผ่านการเขียนโปรแกรม และยังคงซัพพอร์ตโปรโตคอลมาตรฐาน อื่น ๆ เช่น NetFlow, sFlow, LACP

2.3 การจัดการจราจรบนเครือข่าย

การจัดการจราจรบนเครือข่ายหมายถึงการคงไว้ซึ่งความพร้อมใช้งานของระบบเครือข่าย แนวคิดวิศวกรรมจราจรบนเครือข่ายกำหนดด้วยซอฟต์แวร์[2] เป็นแนวคิดที่ออกแบบ และควบคุมการจราจรของข้อมูลในเครือข่าย เพื่อให้การจราจรในเครือข่ายนั้นสามารถกระจายการทำงานได้อย่างสมดุลและมีประสิทธิภาพ โดยส่วนใหญ่จะเกี่ยวข้องกับการกระจายการส่งข้อมูลผ่านเส้นทางหลายเส้นทางด้วยการใช้อัลกอริทึมต่าง ๆ เพื่อแบ่งปริมาณการจราจรอย่างเหมาะสม

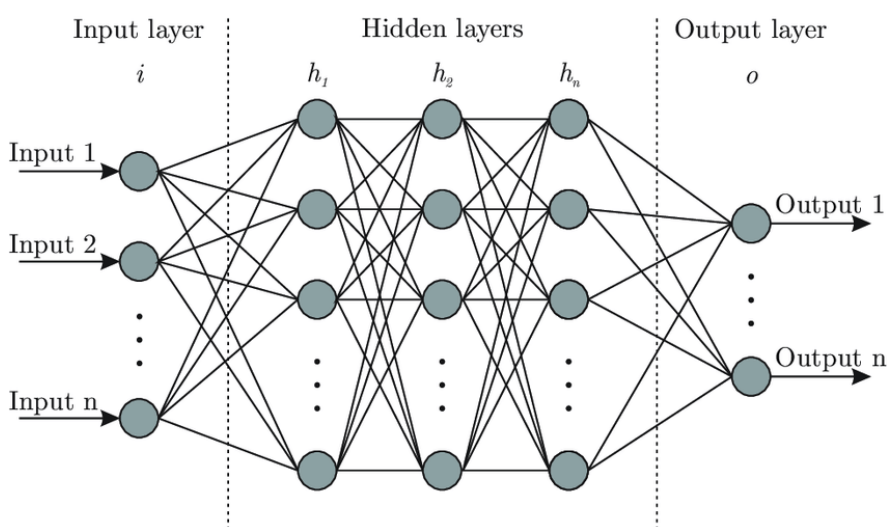
แนวคิด Two-tier mechanism [3] เป็นแนวคิดการออกแบบระบบจัดการจราจรรูปแบบหนึ่งที่จะแบ่งการทำงานเองเอสดีเอ็นคอนโทรลเลอร์ออกเป็นสองส่วน โดยมีส่วนที่ตรวจสอบสถานะการงานและส่วนที่เป็นกลไกการวัดและส่วนของการจัดการเครือข่าย ดังแสดงในรูปที่ 2.2



รูปที่ 2.2 Two-Tier Mechanism

2.4 การเรียนรู้เชิงลึก (Deep Learning)

การเรียนรู้เชิงลึก เป็นวิธีหนึ่งของการเรียนรู้ของเครื่องซึ่งพัฒนามาจากโครงข่ายประสาทเทียม (Artificial Neural Network : ANN) เป็นการจำลองรูปแบบการประมวลผลของสมองมนุษย์ โดยมีโครงสร้างแบบลำดับชั้นเชื่อมต่อกันต่าง ๆ ผลลัพธ์ของชั้นก่อนหน้าจะกลายเป็นข้อมูลนำเข้าของชั้นถัดไป จากโครงสร้างสถาปัตยกรรมข้อมูลดังรูปที่ 2.3 ซึ่งจะประกอบไปด้วยโครงข่ายย่อย ๆ หลายชั้น สามารถใช้งานได้หลากหลายรูปแบบ เช่น การประมวลผลภาพ การรู้จำใบหน้า หรือการประมวลผลภาษาธรรมชาติ

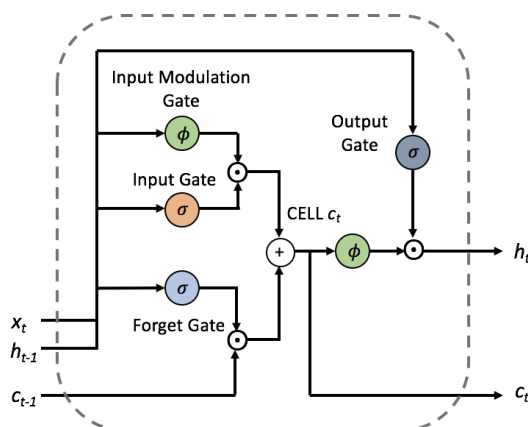


รูปที่ 2.3 โครงสร้างของการเรียนรู้เชิงลึกหรือโครงข่ายประสาทเทียม

2.5 โครงข่ายความจำระยะสั้นขนาดยาว (Long-Short Term Memory : LSTM)

โครงข่ายความจำระยะสั้นขนาดยาวเป็นส่วนหนึ่งของการเรียนรู้เชิงลึกที่สร้างขึ้นมาเพื่อแก้ปัญหาการลืมข้อมูลของ RNN ในกรณีที่มีข้อมูลนำเข้ายาวมาก ทำให้การส่งค่าผลลัพธ์ฟีดฟอร์เวิร์ดไม่สามารถที่จะเก็บรักษาข้อมูลในช่วงเวลาที่ผ่านมาได้ทั้งหมด เอกลักษณะสำคัญของโครงข่ายชนิดนี้คือ เซลล์ความจำ (Cell หรือ Memory cell) มีหน้าที่เก็บข้อมูลเข้าที่สำคัญเอาไว้ไม่ให้เลือนหายไปในช่วงระยะเวลาหนึ่ง เป็นการแก้ปัญหาการพึ่งพาระยะยาว ซึ่งเหมาะสำหรับการพยากรณ์ปริมาณการใช้งานเครือข่ายเอสดีเอ็นทีที่มีชุดข้อมูลปริมาณมาก หรือพยากรณ์ปริมาณการใช้แบนด์วิธในเครือข่าย [4]

LSTM ประกอบด้วยหน่วยความจำ (Cell), ประตูสัญญาณรับเข้า (Input Gate), ประตูสัญญาณสำหรับกันลืม (forget gate), ประตูสัญญาณสำหรับผลลัพธ์ (Output Gate) และผลคูณแบบฮาดามาร์ด (Hadamard Product) ดังแสดงในรูปที่ 2.5



รูปที่ 2.4 โครงสร้างของโครงข่ายความจำระยะสั้นขนาดยาว

Input Gate เป็นหน่วยย่อยในการกำหนดข้อมูลที่จะนำเข้ามาวิเคราะห์ใน Cell โดยรับข้อมูลเข้ามาเพื่อทำการเขียนค่าลงไปในแต่ละ Cell โดยสามารถกำหนดได้จากสมการดังนี้

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i)$$

โดย i_t คือค่าที่ได้จากการ Input Gate

σ คือฟังก์ชัน Sigmoid

W_{xi} คือค่าน้ำหนักสำหรับคำนวณ Input ใน Input Gate

x_t คือค่า Input ที่นำเข้ามาคำนวณ

W_{hi} คือค่าน้ำหนักสำหรับคำนวณ Hidden State ใน Input Gate

h_{t-1} คือ Hidden State ที่ได้มาจากการคำนวณในหน่วยเวลาก่อนหน้า

W_{ci} คือค่าน้ำหนักสำหรับคำนวณ Memory Cell State ใน Input Gate

c_{t-1} คือ Memory Cell State ที่ได้จากการคำนวณในหน่วยเวลาก่อนหน้า

b_i คือค่า Bias ที่ใช้ในการคำนวณใน Input Gate

Forget Gate เป็นหน่วยย่อยที่ใช้ในการกำหนดข้อมูลที่จะนำเข้ามาวิเคราะห์ใน Cell โดยกำหนดว่าข้อมูลนั้นควรที่จะถูกบันทึกหรือถูกลืม โดยสามารถกำหนดได้จากสมการดังนี้

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f)$$

โดย f_t คือค่าที่ได้จากการ Forget Gate

σ คือฟังก์ชัน Sigmoid

W_{xf} คือค่าน้ำหนักสำหรับคำนวณ Input ใน Forget Gate

x_t คือค่า Input ที่นำเข้ามาคำนวณ

W_{hf} คือค่าน้ำหนักสำหรับคำนวณ Hidden State ใน Forget Gate

h_{t-1} คือ Hidden State ที่ได้มาจากการคำนวณในหน่วยเวลาก่อนหน้า

W_{cf} คือค่าน้ำหนักสำหรับคำนวณ Memory Cell State ใน Forget Gate

c_{t-1} คือ Memory Cell State ที่ได้จากการคำนวณในหน่วยเวลาก่อนหน้า

b_i คือค่า Bias ที่ใช้ในการคำนวณใน Forget Gate

Memory Cell State Gate เป็นหน่วยย่อยในการกำหนดข้อมูลที่จะนำเข้ามาวิเคราะห์ใน Cell และทำการคำนวณค่าสถานะ เพื่อใช้ในการคำนวณรอบถัดไป โดยมีสมการดังนี้

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c)$$

โดย c_t คือค่า Memory Cell ในช่วงหน่วยเวลา

f_t คือค่าผลลัพธ์ที่ได้จาก Forget Gate

c_{t-1} คือค่า Memory Cell State ในช่วงเวลาก่อนหน้า

\tanh คือฟังก์ชัน Hyperbolic Tangent

W_{xc} คือค่าน้ำหนักสำหรับคำนวณค่า input จาก Memory Cell State Gate

x_t คือค่า Input ที่นำเข้ามาคำนวณ

W_{hc} คือค่าน้ำหนักสำหรับคำนวณ Hidden State ใน Memory Cell State Gate

h_{t-1} คือ Hidden State ที่ได้มาจากการคำนวณในหน่วยเวลาก่อนหน้า

b_c คือค่า Bias ที่ใช้ในการคำนวณใน Memory Cell State Gate

Output gate เป็นหน่วยย่อยสำหรับคำนวณ Output ของ Cell ซึ่งผลลัพธ์ที่ได้จาก Cell นี้จะมีอยู่ 2 อย่าง ได้แก่ Output และ Hidden State สำหรับใช้ในการคำนวณครั้งถัดไป โดยมีสมการดังนี้

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_{t-1} + b_o)$$

โดย o_t คือค่าที่ได้จากการ Output Gate

σ คือฟังก์ชัน Sigmoid

W_{xo} คือค่าน้ำหนักสำหรับคำนวณ Input ใน Output Gate

x_t คือค่า Input ที่นำเข้ามาคำนวณ

W_{ho} คือค่าน้ำหนักสำหรับคำนวณ Hidden State ใน Output Gate

h_{t-1} คือ Hidden State ที่ได้มาจากการคำนวณในหน่วยเวลาก่อนหน้า

W_{co} คือค่าน้ำหนักสำหรับคำนวณ Memory Cell State ใน Output Gate

c_{t-1} คือ Memory Cell State ที่ได้จากการคำนวณในหน่วยเวลาก่อนหน้า

b_i คือค่า Bias ที่ใช้ในการคำนวณใน Forget Gate

$$h_t = o_t \cdot \tanh(c_t)$$

โดย h_t คือค่า Hidden State จากการคำนวณ

o_t คือค่าที่ได้จากการ Output Gate

\tanh คือฟังก์ชัน Hyperbolic Tangent

C_t คือค่า Memory Cell ในช่วงหน่วยเวลา

2.6 หน่วยเวียนกลับแบบมีประตู (Gated Recurrent Unit: GRU)

ถูกพัฒนาต่อออกมา จาก LSTM ซึ่งพัฒนาในส่วนของการลดความซับซ้อนในการทำงานของโครงข่ายประสาทแบบ LSTM เนื่องจากจำนวนหน่วยย่อยใน Cell จำนวนมาก ซึ่งมีผลต่อประสิทธิภาพในการวิเคราะห์และ ทำนายผล โดย GRU ได้ทำการลดความซับซ้อนในการทำงานของโครงข่ายประสาทแบบ LSTM โดยการลดหน่วยย่อยใน Cell เหลือเพียง 2 ส่วน ได้แก่ Update Gate และ Reset Gate

Update Gate เป็นหน่วยย่อยที่ทำการนำข้อมูลไปคำนวณเพื่อกำหนดสถานะของ Cell สำหรับใช้ในการคำนวณในขั้นถัดไป โดยทำการคำนวณในทุก ๆ รอบที่มีข้อมูลเข้ามา

$$z_t = \sigma(W_{xz}x_t + W_{hz}h_{t-1} + b_z)$$

โดย z_t คือค่าที่ได้จากการ Update Gate

σ คือฟังก์ชัน Sigmoid

W_{xz} คือค่าน้ำหนักสำหรับคำนวณ Input ใน Update Gate

x_t คือค่า Input ที่นำเข้ามาคำนวณ

W_{hz} คือค่าน้ำหนักสำหรับคำนวณ Hidden State ใน Update Gate

h_{t-1} คือ Hidden State ที่ได้มาจากการคำนวณในหน่วยเวลาก่อนหน้า

b_z คือค่า Bias ที่ใช้ในการคำนวณใน Update Gate

Reset Gate เป็นหน่วยย่อยที่ใช้ในการกำหนดข้อมูลว่าควรที่จะเก็บค่าสถานะที่ได้จากการคำนวณในครั้งที่ผ่านมาหรือไม่เพียงใด

$$r_t = \sigma(W_{xr}x_t + W_{hr}h_{t-1} + b_r)$$

โดย r_t คือค่าที่ได้จากการ Rest Gate

σ คือฟังก์ชัน Sigmoid

W_{xr} คือค่าน้ำหนักสำหรับคำนวณ Input ใน Reset Gate

x_t คือค่า Input ที่นำเข้ามาคำนวณ

W_{hr} คือค่าน้ำหนักสำหรับคำนวณ Hidden State ใน Reset Gate

h_{t-1} คือ Hidden State ที่ได้มาจากการคำนวณในหน่วยเวลาก่อนหน้า

b_r คือค่า Bias ที่ใช้ในการคำนวณใน Reset Gate

2.7 มาตรวัดประสิทธิภาพ

ค่าความถูกต้อง คือค่าสถิติใช้สำหรับเปรียบเทียบข้อมูล ผลระหว่างผลลัพธ์เป้าหมาย กับ ผลลัพธ์ที่ทำนายได้ว่ามีความสัมพันธ์กันอย่างไร โดยการคำนวณค่าความถูกต้องนั้น มีวิธีการประเมินค่า 3 วิธี

กำหนดให้

ตัวแปร a คือ ค่าข้อมูลที่แท้จริง

ตัวแปร y คือ ค่าผลลัพธ์ที่ได้จากการพยากรณ์

ตัวแปร n คือจำนวนข้อมูลทั้งหมด

2.7.1 ค่าเฉลี่ยความผิดพลาดกำลังสอง (Mean Square Error: MSE)

$$MSE = \frac{(a_1 - y_1)^2 + \dots + (a_n - y_n)^2}{n}$$

2.7.2 รากที่สองของค่าเฉลี่ยความผิดพลาดกำลังสอง (Root Mean Square Error: RMSE)

$$RMSE = \sqrt{\frac{(a_1 - y_1)^2 + \dots + (a_n - y_n)^2}{n}}$$

2.7.3 ค่าเฉลี่ยของค่าสัมบูรณ์ความผิดพลาด (Mean Absolute Error: MAE)

$$MAE = \frac{|a_1 - y_1| + \cdots + |a_n - y_n|}{n}$$

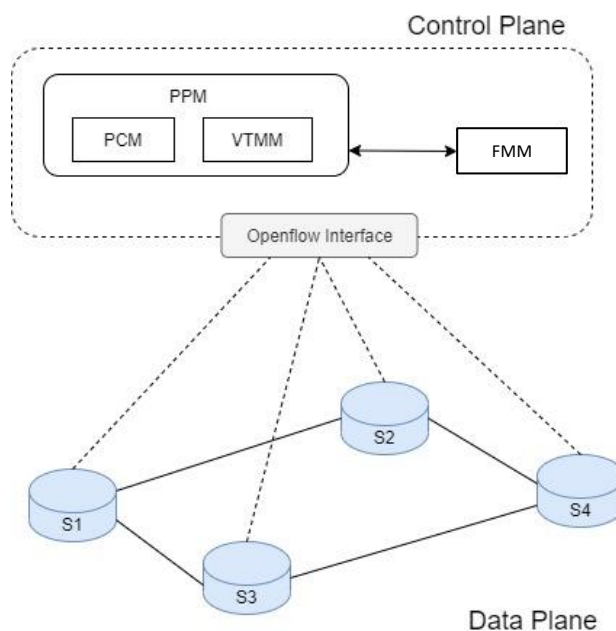
บทที่ 3

วิธีการดำเนินการวิจัย

ในบทนี้จะกล่าวถึงวิธีการดำเนินงานโครงการการศึกษาการประยุกต์เทคโนโลยีการเรียนรู้เชิงลึกสำหรับการจัดการจราจรบนเครือข่ายเอสดีเอ็น โดยกลุ่มผู้วิจัยได้ดำเนินงานตามขั้นตอนดังนี้

- 3.1 การปรับปรุงองค์ประกอบภายในคอนโทรลเลอร์
- 3.2 อัลกอริทึมในการหาเส้นทางสำหรับการย้ายโฟลว์บนระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์
- 3.3 การสร้างชุดข้อมูลอนุกรมเวลาสำหรับฝึกฝนและทดสอบ
- 3.4 การฝึกฝนและทดสอบโมเดลการเรียนรู้เชิงลึก

3.1 การปรับปรุงองค์ประกอบภายในคอนโทรลเลอร์



รูปที่ 3.1 องค์ประกอบภายในคอนโทรลเลอร์

การปรับปรุงองค์ประกอบภายในของคอนโทรลเลอร์ จะนำองค์ประกอบมาตรฐานภายในคอนโทรลเลอร์ของ RYU Controller มาปรับปรุงและเพิ่มเติม โดยที่องค์ประกอบมาตรฐานภายในคอนโทรลเลอร์ประกอบไปด้วยส่วนต่าง ๆ ดังนี้

3.1.1 Path Provision Module (PPM)

ทำหน้าที่จัดการข้อความจากโอเพนโฟลว์ทั้งหมดจากองค์ประกอบภายในคอนโทรลเลอร์ โดยภายในจะมีองค์ประกอบย่อยคือ Path Computation เมื่อ Path Provision Module ได้รับแพ็คเกจมาจากสวิตช์ Path Provision Module จะทำการส่งต่อข้อความไปยัง Path Computation Module เพื่อคำนวณหาเส้นทางเพื่อไปยังปลายทางต่อไป

3.1.2 Path Computation Module

ทำหน้าที่ในการหาเส้นทางที่มีระยะที่สั้นที่สุดและมีแบนด์วิดท์มากที่สุดในการส่งข้อมูล โดยใช้ข้อมูลโทโพโลยีที่ได้จาก Virtual Topology Management Module (VTMM) จากนั้นจะส่งผลการคำนวณหาเส้นทางกลับไปยัง Path Provision Module (PPM) และทำการสร้าง FlowMod Message เพื่อใช้ในการติดตั้งกฎการส่งออก จากนั้นจะทำการติดตั้งกฎการส่งออกไปยังทุกสวิตช์ที่อยู่ในเส้นทางที่ได้คำนวณไว้

3.1.3 Virtual Topology Management Module (VTMM)

ทำหน้าที่ในการปรับปรุงแก้ไขโทโพโลยีของระบบเครือข่าย เมื่อได้รับการแจ้งเตือนจาก Flow Monitoring Module โดย VTMM จะปรับเปลี่ยนในโทโพโลยีจำลอง ซึ่งสร้างจากไลบรารี Networkx โดยใช้ Link Layer Discovery Protocol (LLDP)

โดยการปรับปรุงองค์ประกอบภายในคอนโทรลเลอร์จะเพิ่มองค์ประกอบในการตรวจสอบปริมาณการใช้งานของลิงก์ตามเวลาจริง โดยจะเรียกว่า Flow Monitoring Module (FMM) ซึ่งค่าปริมาณการใช้งานของลิงก์ จะถูกนำไปให้โมเดลทำนายปริมาณการใช้งานในเวลาข้างหน้าและใช้เป็นค่าในการลบสวิตช์ที่เกิดความแออัดบนเส้นทางที่พร้อมใช้งานถ้าหากมีปริมาณการใช้งานของลิงก์ที่มากกว่า 70 เปอร์เซ็นต์ของแบนด์วิดท์ เมื่อโมเดลทำนายว่ามีปริมาณการใช้งานของลิงก์มากกว่า 70 เปอร์เซ็นต์

TMM จะแจ้งเตือนจากนั้นจะส่งข้อมูลไปยัง VTMM เพื่อปรับปรุงโทโพโลยีแลจากนั้น Path Provision Module ก็จะคำนวณหาเส้นทางที่ดีที่สุดใหม่อีกครั้งจากโทโพโลยีจำลองที่ถูกปรับปรุงแล้ว

3.2 อัลกอริทึมในการหาเส้นทางสำหรับการย้ายโพล์บนระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์

สำหรับอัลกอริทึมในโครงการนี้ จะใช้อัลกอริทึมในการหาเส้นทางสำหรับการย้ายโพล์โดยพิจารณาจากขนาดของโพล์และความพร้อมใช้ของแบนด์วิธที่คงเหลือต่ำสุดของเส้นทางนั้น โดยปกติแล้วการส่งข้อมูล (รูปแบบของ TCP) ในระบบเครือข่ายจะมีกลไกในการควบคุมความแออัดเพื่อลดความเสี่ยงที่แพ็คเกจจะไปถึงผู้รับเนื่องจากเกิดการสูญหายของข้อมูล เรียกว่า Congestion Control เมื่อมีความแออัดเกิดขึ้นกลไกนี้จะทำการลดขนาดของ Congestion Window (CWND) ลง ทำให้ส่งได้ในปริมาณที่น้อยลง เวลาที่ผู้รับจะได้รับข้อมูลครบทั้งหมดก็จะนานขึ้น ดังนั้นอัลกอริทึมที่ใช้ในโครงการนี้จะเลือกเส้นทางที่มีความพร้อมใช้ของแบนด์วิธมากที่สุดและเหมาะสมกับขนาดของโพล์เพื่อลดความแออัดของเส้นทางเดิม คอนโทรลเลอร์ต้องการคำนวณหาปริมาณการใช้งานแบนด์วิธที่ในขณะนั้นในแต่ละลิงก์ของแต่ละสวิตช์เพื่อตรวจสอบความแออัดและเทียบเกณฑ์ประเมินความพร้อมใช้ 2 ระดับ โดยที่คอนโทรลเลอร์จะสอบถามปริมาณการใช้งานของลิงก์ ณ ปัจจุบันในทุก ๆ 2 วินาทีเพื่อทำการลบลิงก์ที่มีปริมาณการใช้งานที่มากกว่า 40 เปอร์เซ็นต์ จากนั้นจะนำไปอัปเดตให้กับโทโพโลยีจำลองเพื่อนำไปใช้หาเส้นทางที่จะย้ายโพล์ สำหรับการคำนวณอัตราส่วนปริมาณการใช้งานในแต่ละครั้งต้องทำการคำนวณหาปริมาณการใช้งานแบนด์วิธในปัจจุบันก่อนจากสมการดังนี้

$$b_i = \frac{n_i(t) - n_i(t - T)}{T}$$

โดย b_i คือ ค่าปริมาณการใช้งานแบนด์วิธในปัจจุบัน ณ เวลา t

$n_i(t)$ คือ ค่าปริมาณการส่งผ่าน ณ เวลา t

$n_i(t - T)$ คือ ค่าปริมาณการส่งผ่าน ณ เวลา $t - T$

T คือ เวลาการเรียกตรวจสอบในแต่ละครั้ง

เมื่อได้ค่าปริมาณการใช้งานของลิงก์จะถูกนำไปให้โมเดลหาปริมาณการใช้งานในระยะเวลาข้างหน้า เมื่อได้ค่าปริมาณการใช้งานในระยะเวลาข้างหน้าแล้ว จะนำมาตรวจสอบว่ามีการใช้งานเกินเกณฑ์ที่กำหนดไว้หรือไม่ ถ้าหากว่าเกินเกณฑ์จะมีการปรับปรุงโทโพโลยีจำลองและ PCM จะนำไปหาเส้นทางสำหรับการย้ายโพล์โดยพิจารณาตามลำดับขนาดของโพล์ จากนั้นจะต้องนำไปตรวจสอบว่าเส้นทางที่คำนวณออกมาจะสามารถย้ายโพล์ไปที่เส้นทางนั้นได้หรือไม่โดยไม่ก่อให้เกิดความแออัดซ้ำซ้อน ซึ่งต้องคำนวณหาปริมาณแบนด์วิดท์คงเหลือต่ำสุดที่พร้อมใช้งานของเส้นทางนั้น ในขั้นแรกต้องทำการหาเส้นทางที่ดีที่สุดสำหรับโพล์นั้นก่อนซึ่งสามารถหาได้จากสมการดังนี้

$$BP = \min_{P \in P_{A \rightarrow B}} \sum_{e_i \in P} c_i$$

โดย BP คือ เส้นทางที่ดีที่สุดสำหรับโพล์นั้น

$P_{A \rightarrow B}$ คือ เส้นทางทั้งหมดระหว่าง A ไป B

c_i คือ ค่าใช้จ่ายของลิงก์ (e_i) นั้น ๆ

e_i คือ ลิงก์ของเส้นทางนั้น ๆ

เมื่อได้เส้นทางที่ดีที่สุดแล้ว จากนั้นจะหาปริมาณแบนด์วิดท์คงเหลือที่ต่ำที่สุดจากทุกลิงก์ของเส้นทางนั้นจากสมการดังนี้

$$ABW = \min_{e_i \in BP} a_i$$

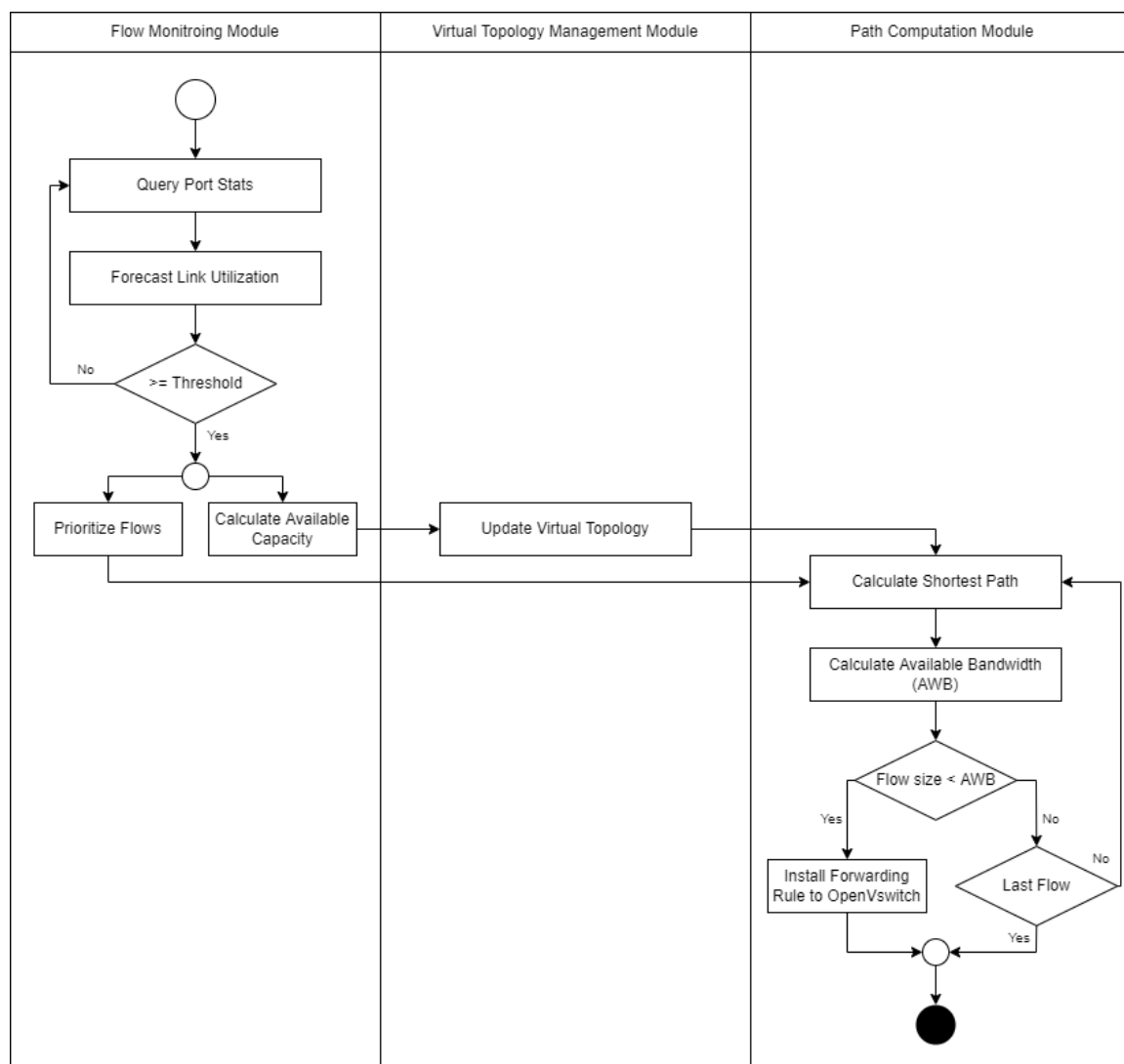
โดย ABW คือ ค่าแบนด์วิดท์ต่ำสุดที่พร้อมใช้งานของเส้นทาง

BP คือ เส้นทางที่ดีที่สุดจาก A ไป B

e_i คือ ลิงก์ของเส้นทาง

a_i คือ ค่าความจุคงเหลือที่พร้อมใช้ของลิงก์ (e_i) นั้น ๆ ซึ่งหาได้จากการนำค่าแบนด์วิดท์ลบกับค่าปริมาณการใช้งานแบนด์วิดท์ในปัจจุบัน

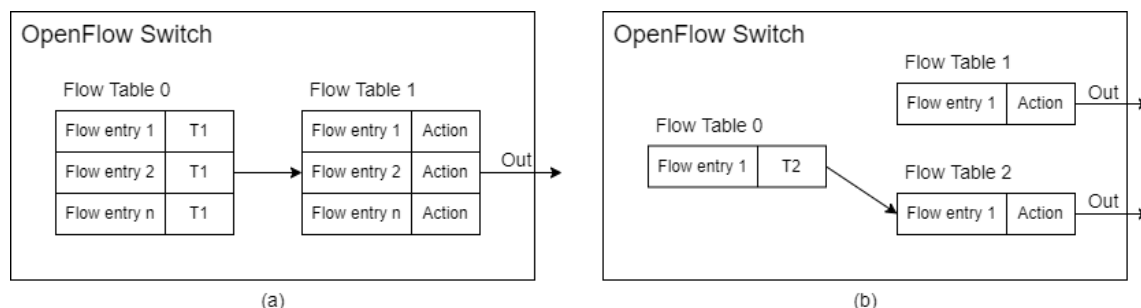
เมื่อได้ค่าแบนด์วิดท์ต่ำสุดที่พร้อมใช้งานแล้ว จากนั้นจะนำมาเปรียบเทียบกับขนาดของโฟลว์ที่จะทำการย้าย การเลือกเส้นทางที่เหมาะสมที่สุดโดยพิจารณาจากค่าแบนด์วิดท์ต่ำสุดที่พร้อมใช้งานและขนาดของโฟลว์ สามารถอธิบายการทำงานตามแผนภาพได้ดังนี้



รูปที่ 3.2 แผนภาพขั้นตอนของอัลกอริทึมในการหาเส้นทางสำหรับย้ายโฟลว์

โดยปกติสวิตช์จะสามารถส่งข้อมูลไปได้โดยอาศัยข้อมูลจาก Flow Table กล่าวคือเมื่อมีโฟลว์หรือแพ็กเก็ตถูกส่งมาที่สวิตช์ สวิตช์จะนำมาเปรียบเทียบกับข้อมูลใน Flow Table ถ้าหากว่ามีข้อมูลที่ตรงกันก็จะส่งออกตามพอร์ตที่กำหนด แต่ถ้าหากไม่มีข้อมูลที่ตรงกันสวิตช์จะร้องขอข้อมูลเส้นทางจากคอนโทรลเลอร์ สภาวะปกติสวิตช์จะ Flow Table ทั้งหมด 2 ตารางดังรูปที่ 3.3 (a) โดยตารางที่ 0 จะส่ง

โพล์ไปตรวจสอบกับตารางที่ 1 เพื่อหาว่าข้อมูลสำหรับการส่งออก แต่ถ้าหากมีการย้ายเส้นทางเกิดขึ้น คอนโทรลเลอร์จะสร้างตารางที่ 2 ที่มีข้อมูลเส้นทางใหม่ให้กับสวิตช์ที่อยู่ในเส้นทางใหม่และเปลี่ยนให้ตารางที่ 0 ส่งโพล์มาที่ตารางที่ 2 ดังรูปที่ 3.3 (b) ถ้าหากโพล์ที่ถูกย้ายเส้นทางส่งข้อมูลครบแล้ว คอนโทรลเลอร์จะเปลี่ยนให้ตารางที่ 0 ส่งโพล์ให้กับตารางที่ 1 เพื่อกลับมาใช้เส้นทางเดิม



รูปที่ 3.3 Flow Table ภายในสวิตช์

เมื่อเริ่มการทำงานของระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์ คอนโทรลเลอร์จะสอบถามปริมาณการใช้แบนด์วิดท์ในทุกลิงก์ของแต่ละสวิตช์ในทุกช่วงเวลา และส่งให้โมเดลเพื่อหาค่าปริมาณการใช้งานในระยะเวลายาวหน้าและตรวจสอบว่ามีความแออัดเกิดขึ้นหรือไม่ ถ้ามีความแออัดเกิดขึ้น FMM จะทำการหาโพล์ที่ผ่านลิงก์นั้นแล้วจัดลำดับขนาดของโพล์โดยเรียงจากโพล์ที่มีขนาดใหญ่ที่สุด จากนั้นจะทำการตรวจสอบว่ามีลิงก์ไหนบ้างที่มีการใช้งานแบนด์วิดท์ที่เกินกว่า 40 เปอร์เซ็นต์ FMM จะส่งข้อมูลไปให้ VTMM ลบการเชื่อมโยงระหว่าง สวิตช์นั้นออกจากโทโพโลยีจำลอง จากนั้น PCM จะทำการหาเส้นทางโดยเลือกจากโพล์ลำดับแรกที่มีขนาดใหญ่ที่สุด และตรวจสอบว่าโพล์มีขนาดเล็กกว่าปริมาณแบนด์วิดท์ต่ำสุดที่พร้อมใช้งานของเส้นทางสำหรับโพล์นั้นหรือไม่ ถ้าโพล์มีขนาดเล็กกว่าก็เข้ากระบวนการส่งกฎการส่งออก (Forwarding Rule) ไปยังสวิตช์ที่เกี่ยวข้อง แต่ ถ้าโพล์มีขนาดที่ใหญ่กว่าก็จะทำการพิจารณาโพล์ในลำดับถัดไปแทน เมื่อโพล์ที่ได้ทำการย้ายออกไปได้ทำการส่งข้อมูลจนครบแล้วโพล์นั้นจะถูกย้ายกลับมาสู่เส้นทาง

3.3 ข้อมูลที่ใช้ในการวิจัย

ข้อมูลที่ใช้ที่ใช้ในการวิจัยจะเป็นข้อมูลประเภทอนุกรมเวลาที่ได้มาจากการเก็บค่าปริมาณการใช้งานของแต่ละลิงค์ในทุกช่วงเวลา โดยจะเก็บข้อมูลจากการสร้างการจราจรในเครือข่ายที่มีการผสมระหว่างรูปแบบการจราจรที่มีการบังคับให้เกิดความแออัดและรูปแบบการจราจรแบบสุ่ม ซึ่งจัดเก็บในรูปแบบไฟล์นามสกุล .csv

3.4 การจัดการข้อมูล

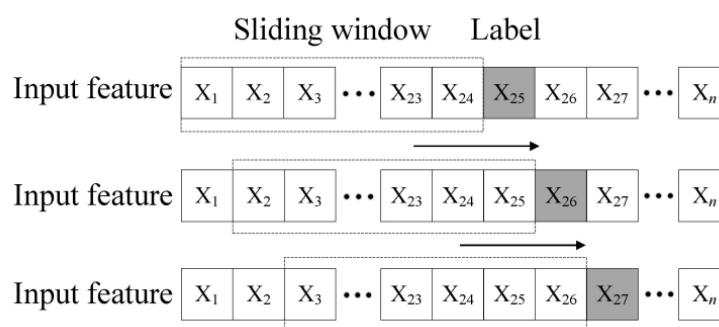
การทำความสะอาดข้อมูล (Data Cleansing) ข้อมูลที่ได้มามีจำนวนไม่มากนัก แต่มีความผิดพลาดอยู่จึงจำเป็นต้องทำการตรวจสอบ แก้ไข หรือลบเพื่อให้รายการข้อมูลที่ไม่ถูกต้องออกไปจากชุดข้อมูลจึงจะนำไปใช้ประโยชน์ได้อย่างไม่เกิดความผิดพลาดความสับสน หากนำข้อมูลที่มีความผิดพลาดไปใช้ในโครงข่ายประสาทเทียม จะทำให้ผลลัพธ์แย่กว่าที่ควรจะเป็น

หลังจากทำความสะอาดข้อมูลเสร็จแล้วจากนั้นจะต้องทำการปรับขอบเขตของข้อมูล (Feature Scaling) ให้อยู่ในช่วงเดียวกัน ดังนั้นผู้วิจัยจึงเลือกใช้วิธี Rescaling (Min – Max Normalization) เพื่อทำการปรับช่วงข้อมูลให้อยู่ในช่วง [0, 1] จากสมการดังนี้

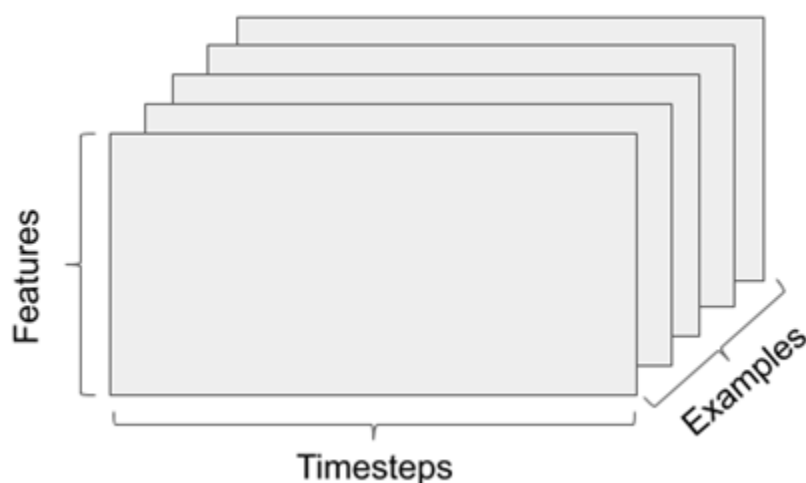
$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

โดย x คือข้อมูลนำเข้าและ x' คือข้อมูลที่ถูกปรับช่วงแล้ว หลังจากปรับช่วงของข้อมูลแล้วจะทำการแบ่งข้อมูล (Data Splitting) เป็นข้อมูลสำหรับการฝึกฝน (Training set) ข้อมูลสำหรับทดสอบ (Testing set) และข้อมูลสำหรับตรวจสอบ (Validation set)

หลังจากปรับช่วงข้อมูลแล้วจะต้องแบ่งข้อมูลฝึกฝนและข้อมูลทดสอบเป็นข้อมูลนำเข้าและข้อมูลส่งออกด้วยเทคนิค Sliding window โดยกำหนดขนาดของขนาดของกรอบที่จะใช้ระบุข้อมูลนำเข้า (Window Size) และลำดับถัดไปของกรอบข้อมูลจะถูกนำมาเป็นข้อมูลส่งออกดังรูปที่ 3.4 และขั้นตอนท้ายคือการแปลงข้อมูลให้อยู่ในรูปแบบ 3 มิติ ดังรูปที่ 3.5 เพื่อนำไปใช้กับโมเดล



รูปที่ 3.4 Sliding window



รูปที่ 3.5 รูปแบบข้อมูลนำเข้าและส่งออกสำหรับโมเดล

3.5 กำหนดโมเดลการเรียนรู้เชิงลึก

หลังจากที่ได้มีการสร้างชุดข้อมูลแล้วจากนั้นทางผู้พัฒนาจึงได้ทำการเลือกโมเดลในการทำการทดสอบทั้งหมด 3 โมเดลดังนี้

1. Long Short-term Memory (LSTM)
2. Bidirectional Long Short-term Memory (BiLSTM)
3. Gated Recurrent Unit (GRU)

ซึ่งจะแบ่งการทดสอบออกเป็นสองรูปแบบคือแบบหลายอินพุตหลายเอาต์พุต (Multiple Input Multiple Output) เพื่อให้ทำนายปริมาณการใช้งานลิ้งค์ของทุกพอร์ตในการทำนายหนึ่งครั้ง และแบบ

หลายอินพุตหนึ่งเอาต์พุต (Multiple Input Single Output) เพื่อให้ทำนายปริมาณการใช้งานลิ้งค์ครั้งละหนึ่งพอร์ต โดยก่อนการฝึกฝนโมเดลผู้วิจัยได้ทำการกำหนดพารามิเตอร์ต่าง ๆ ดังตารางที่ 3.1

Hyperparameter	LSTM	BiLSTM	GRU
Activation Function	ReLu	ReLu	ReLu
Optimizer	Adam	Adam	Adam
Learning rate	0.001	0.001	0.001
Dropout rate	0.2	0.2	0.2
Batch size	16	16	16
Number of neurons	100	75	100
Epoch	36	27	30
Loss	MSE	MSE	MSE

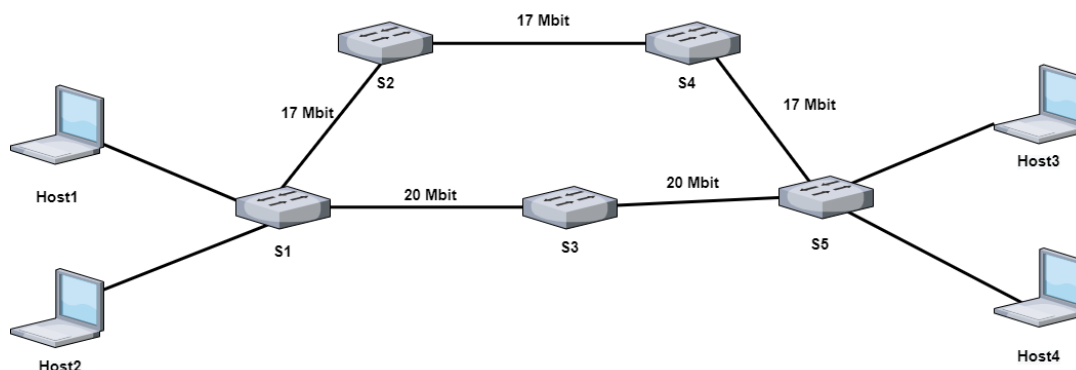
ตารางที่ 3.1 ตารางค่าพารามิเตอร์ของแต่ละ โมเดล

บทที่ 4

การทดลองและผลการทดลอง

4.1 การทดลองคอนโทรลเลอร์

การทดลองนี้จะเป็นการนำคอนโทรลเลอร์ที่ได้ออกแบบไว้ในบทที่ 3 มาทดลองและวัดประสิทธิภาพโดยทำการสร้างระบบเครือข่ายเสมือนจริงจากโปรแกรมเลียนแบบพฤติกรรมการทำงานของระบบเครือข่ายหรือ Mininet ซึ่งโปรแกรมนี้จะช่วยสร้างระบบเครือข่ายที่เลียนแบบการทำงานของระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์ โดยคอนโทรลเลอร์จะใช้ซอฟต์แวร์ที่มีชื่อว่า Ryu Controller ซึ่งเป็นซอฟต์แวร์ที่มีประสิทธิภาพและมีการใช้งานอย่างแพร่หลาย โดยโครงการนี้ได้ดำเนินการสร้างเครือข่ายตามโครงสร้างเครือข่ายที่กำหนดตามรูปที่ 4.1 ระบบที่สร้างขึ้นประกอบไปด้วย Open vSwitch จำนวน 5 ตัว และ PC จำนวน 4 เครื่อง แต่ละเครื่องสลับการรับ-ส่งข้อมูล ใช้โปรแกรม iPerf [5] เป็นเครื่องมือใช้วัดความกว้างของแบนด์วิดท์ โดยสร้างแพ็กเก็ตแบบ UDP ส่งเข้าไปในเครือข่ายเพื่อสร้างการจราจรแบบสุ่มในระบบเป็นเวลาหนึ่งชั่วโมง

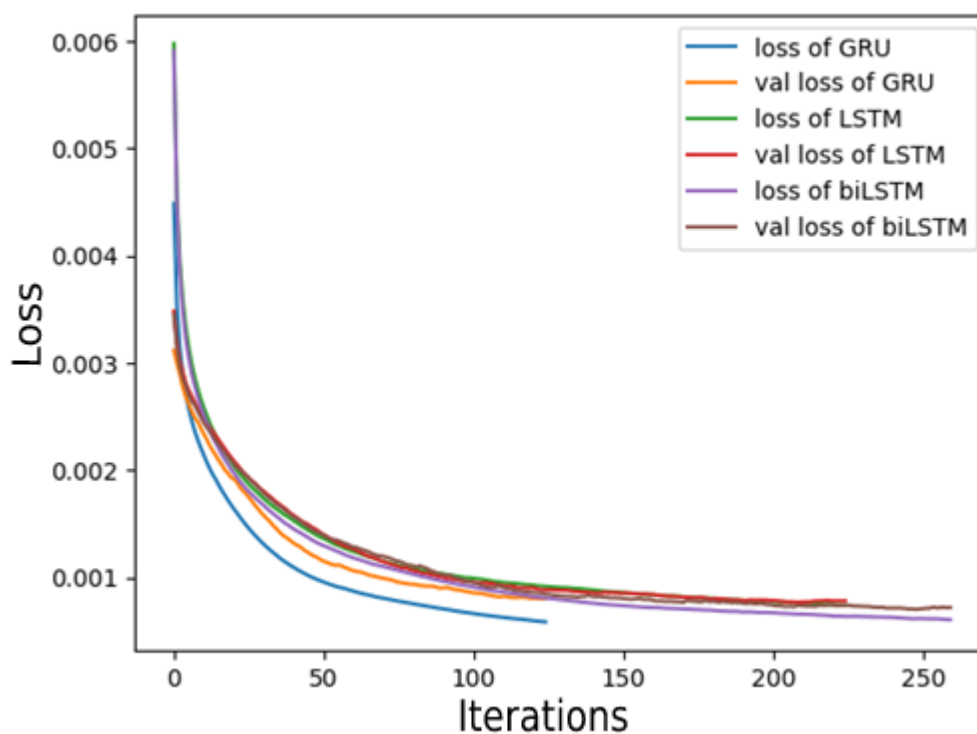


รูปที่ 4.1 โทโพโลยีแบบกำหนดเอง

4.2 ผลการทดสอบทดสอบโมเดลการเรียนรู้เชิงลึก

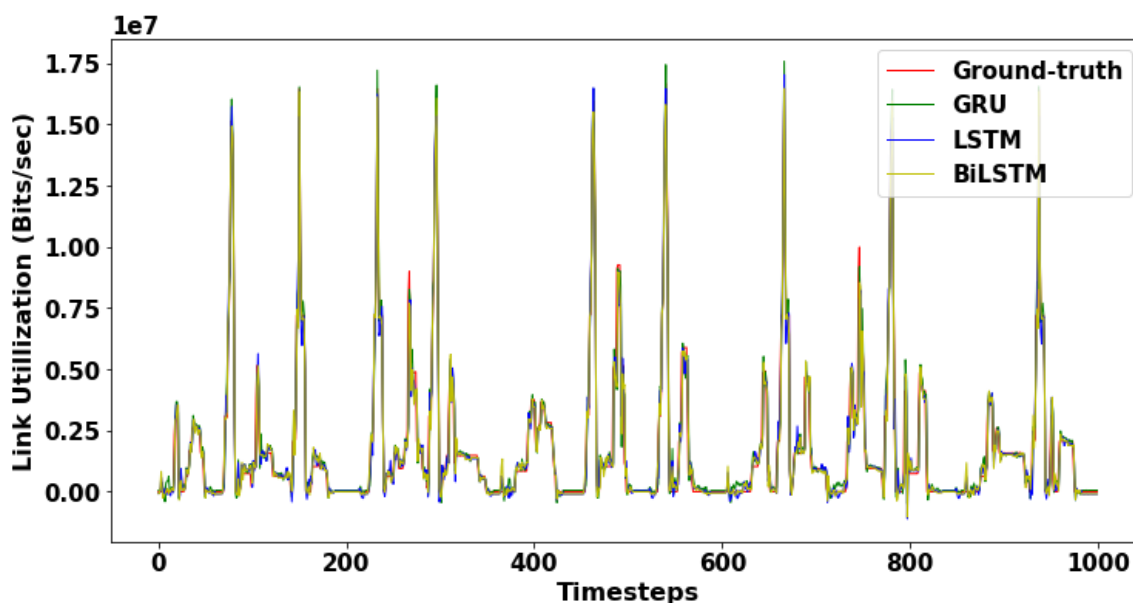
4.2.1 ผลการทดสอบแบบหลายอินพุตหลายเอาต์พุต (Multiple Input Multiple Output)

จากผลการทดสอบสามารถแสดงผลเป็นกราฟเปรียบเทียบเส้นโค้งการเรียนรู้ได้ดังรูปที่ 4.2 โดยจะเห็นได้ว่าโมเดลสามารถเรียนรู้ได้ดีโดยเกิดปัญหา Overfitting หรือ Underfitting



รูปที่ 4.2 กราฟเปรียบเทียบเส้นโค้งการเรียนรู้ของแต่ละโมเดล

ในส่วนของการเปรียบเทียบค่าจริงและค่าทำนายสามารถแสดงผลเป็นกราฟได้ดังรูปที่ 4.3 โดยเลือกใช้ข้อมูลของสวิตช์ที่ 1 พอร์ตที่ 2 ในการเปรียบเทียบ



รูปที่ 4.3 กราฟเปรียบเทียบค่าจริงและค่าทำนายของแต่ละ โมเดล

ผลค่าคลาดเคลื่อนจากการทดสอบ โมเดล โดยในขั้นตอนนี้มีค่าคลาดเคลื่อนที่คำนวณจาก MSE, RMSE และ MAE เพื่อเปรียบเทียบความแม่นยำของแต่ละ โมเดลดังแสดงในตารางที่ 4.1

Model	MSE	RMSE	MAE
LSTM	0.00064	0.02548	0.01011
BiLSTM	0.00054	0.02335	0.00878
GRU	0.00061	0.02477	0.01002

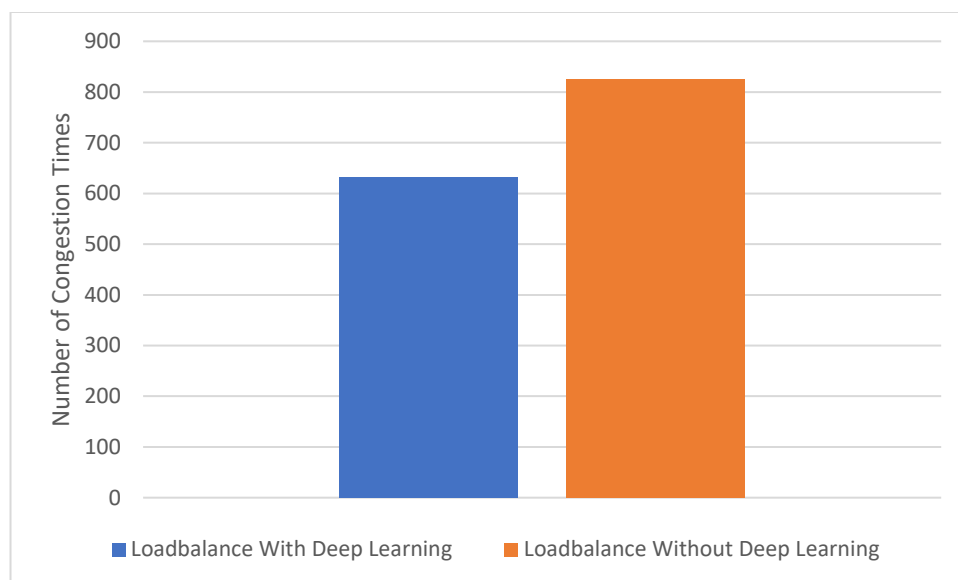
ตารางที่ 4.1 ตารางเปรียบเทียบความคลาดเคลื่อนของแต่ละ โมเดล

4.3 ผลการทดสอบการทำงานของระบบ

ในการวัดประสิทธิภาพการทำงานของระบบที่พัฒนาในโครงงานนี้บนโทโพโลยีที่กำหนดจะใช้ตัวชี้วัดประสิทธิภาพการทำงานคือ จำนวนครั้งการเกิดความแออัด

จำนวนครั้งการเกิดความแออัดในเครือข่ายที่กำหนดด้วยซอฟต์แวร์โดยใช้คอนโทรลเลอร์ที่พัฒนาขึ้นจัดการการจราจรในเครือข่าย ดังภาพที่ 4.4 แสดงแกน Y ของกราฟเป็นจำนวนครั้ง โดยแท่งสี

น้ำเงินเป็นจำนวนการเกิดความแออัดของคอนโทรลเลอร์ที่นำการเรียนรู้เชิงลึกมาใช้ ส่วนแท่งสีส้มเป็นจำนวนการเกิดความแออัดของคอนโทรลเลอร์ที่มีการกระจายกราฟฟิกแบบปกติ



รูปที่ 4.4 กราฟเปรียบเทียบแสดงจำนวนการเกิดความแออัดในระยะเวลาหนึ่งชั่วโมง

หลังจากมีการใช้การเรียนรู้เชิงลึก จำนวนการเกิดความแออัดมีจำนวนครั้งลดน้อยลง ซึ่งสอดคล้องกับรูปที่ 4.4 โดยหมายความว่าคอนโทรลเลอร์สามารถกระจายการจราจรในเครือข่ายได้อย่างมีประสิทธิภาพ

บทที่ 5

สรุปผลการทดลอง

5.1 สรุปผล

ในปัจจุบันระบบเครือข่ายในปัจจุบันได้มีการขยายตัวจนมีขนาดใหญ่ การจัดการปัญหาความแออัดแบบเดิมตามกลไกของระบบเครือข่ายนั้นยังไม่สามารถจัดการได้มีประสิทธิภาพดีพอ ดังนั้นโครงการนี้จึงได้พัฒนาคอนโทรลเลอร์ของระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์ที่สามารถกระจายการจราจรในเครือข่ายเมื่อเกิดความแออัด โดยใช้อัลกอริทึมการหาเส้นทางสำหรับการย้ายโฟลว์ ซึ่งเส้นทางที่สารภีใช้งานได้นั้น จะต้องสอดคล้องกับเกณฑ์ที่กำหนดไว้ โดยผลลัพธ์ที่ได้ทำการทดลองแสดงให้เห็นว่าคอนโทรลเลอร์สามารถกระจายการจราจรได้อย่างมีประสิทธิภาพบนโทโพโลยีที่กำหนด โดยในแง่จำนวนครั้งการแออัดสามารถลดลงได้ถึง 23.04 เปอร์เซ็นต์

5.2 แนวทางในการพัฒนาต่อ

คอนโทรลเลอร์ที่พัฒนาในโครงการนี้สามารถทำนายปริมาณการใช้งานของลิงก์ภายใต้โทโพโลยีและรูปแบบการจราจรที่กำหนดไว้แล้วเท่านั้น การพัฒนาโครงการในอนาคตจะพัฒนาเทคโนโลยีการเรียนรู้เชิงลึกให้สามารถพยากรณ์ปริมาณการใช้งานของลิงก์ให้ใช้ได้กับโทโพโลยีและรูปแบบการจราจรที่หลากหลาย

บรรณานุกรม

1. Xenofon Foukas, Mahesh K. Marina, Kimon Kontovasilis “Software Defined Networking Concepts,” The University of Edinburgh & NCSR "Demokritos.
2. I ZHAOGANG SHU, JIAFU WAN, JIAXIANG LIN, SHIYONG WANG, DI LI, SEUNGMIN RHO, CHANGCAI YANG, "Traffic Engineering in Software-defined Networking: Measurement and Management," IEEE Access, vol. 4, pp.3242-3256 (2016).
3. Ming-Hung Chen, Yen-Chen Tien, Yuan-Ting Huang, “A Low-Latency Two-Tier Measurement and Control Platform for Commodity SDN,” IEEE Communications Magazine, pp. 0163-6804 September (2016).
4. Maxime Labonne; Charalampos Chatzinakis; Alexis Olivereau, “Predicting Bandwidth Utilization on Network Links Using Machine Learning,” 2020 European Conference on Networks and Communications (EuCNC), September (2020).
5. iPerf. Available from: <https://iperf.fr>
6. Keras. Available from: <https://keras.io/>
7. Pumrapee Poomka, “Deep Learning Techniques For Sentiment Analysis From Product User”, (วิทยานิพนธ์ปริญญาโทบริหารธุรกิจ มหาวิทยาลัยเทคโนโลยีสุรนารี, 2562).
8. Péter Megyesi, Alessio Botta, Giuseppe Aceto, Antonio Pescapè, Sándor Molnár, “Available Bandwidth Measurement in Software Defined Network,” 31st ACM/SIGAPP Symposium on Applied Computing, April (2016)

ภาคผนวก

โค้ดและการทำงานของโปรแกรมที่เกี่ยวข้อง


```

df_1 = pd.read_csv('https://raw.githubusercontent.com/Punyapatt/FinalProject/main/dataset/csv_final.csv', index_col=0)
df_2 = pd.read_csv('https://raw.githubusercontent.com/Punyapatt/FinalProject/main/dataset/csv_final2.csv', index_col=0)
df = pd.concat([df_1, df_2], ignore_index=True)
df.drop(['5_port3', '5_port4', '1_port3', '1_port4'], axis=1, inplace=True)
df = df.reindex(sorted(df.columns), axis=1)

```

รูปที่ 5.1 โค้ดการดึงข้อมูล

```

dataset = df.to_numpy()
scaler = MinMaxScaler()
dataset = scaler.fit_transform(dataset)
dataset = dataset[:-85]
train_size = int(len(dataset)*0.8)
test_size = len(dataset) - train_size
train, test = dataset[0:train_size,:], dataset[train_size:len(dataset),:]
print(len(dataset))
print(train.shape, test.shape)
print(len(train), len(test))

```

รูปที่ 5.2 โค้ดการแบ่งข้อมูลการเรียนรู้และการทดสอบ

```

timestep = 8
X_train = []
y_train = []
for i in range(timestep, train_size):
    X_train.append(train[i-timestep:i, :])
    y_train.append(train[i, :])
X_train, y_train = np.array(X_train), np.array(y_train)
n_features = X_train.shape[2]

```

รูปที่ 5.3 โค้ดการ block cross validation time series

```

model_gru = Sequential()
model_gru.add(GRU(100, activation='tanh', input_shape=(timestep, n_features)))
model_gru.add(Dense(n_features))
opt = keras.optimizers.Adam(learning_rate=0.001)
model_gru.compile(optimizer=opt, loss='mse')

model_lstm = Sequential()
model_lstm.add(LSTM(100, activation='tanh', input_shape=(timestep, n_features)))
model_lstm.add(Dropout(0.2))
model_lstm.add(Dense(n_features))
opt = keras.optimizers.Adam(learning_rate=0.001)
model_lstm.compile(optimizer=opt, loss='mse')

Bilstm_model = Sequential()
Bilstm_model.add(Bidirectional(LSTM(100, activation='tanh', input_shape=(timestep, n_features))))
Bilstm_model.add(Dropout(0.2))
Bilstm_model.add(Dense(n_features))
opt = keras.optimizers.Adam(learning_rate=0.001)
Bilstm_model.compile(optimizer=opt, loss='mse')

```

รูปที่ 5.4 โค้ดการสร้างโมเดล

```
model_gru.fit(X_train, y_train, epochs=125, batch_size=32, verbose=2, validation_split=0.3)
```

```
...
```

```
model_lstm.fit(X_train, y_train, epochs=225, batch_size=32, verbose=2, validation_split=0.3)
```

```
...
```

```
Bilstm_model.fit(X_train, y_train, epochs=260, batch_size=32, verbose=2, validation_split=0.3)
```

รูปที่ 5.5 โค้ดการฝึกฝนโมเดล

```
model_history = vars(model_gru.history)
plt.style.use('default')
plt.plot(model_history['history']['loss'], label='loss of GRU')
plt.plot(model_history['history']['val_loss'], label='val loss of GRU')
plt.legend()
plt.show()
```

...

```
model_history = vars(model_lstm.history)
plt.style.use('default')
plt.plot(model_history['history']['loss'], label='loss of LSTM')
plt.plot(model_history['history']['val_loss'], label='val loss of LSTM')
plt.legend()
plt.show()
```

...

```
model_history = vars(Bilstm_model.history)
plt.style.use('default')
plt.plot(model_history['history']['loss'], label='loss of biLSTM')
plt.plot(model_history['history']['val_loss'], label='val loss of biLSTM')
plt.legend()
plt.show()
```

รูปที่ 5.6 โค้ดการปรับแต่งโมเดล

```
pre_gru = model_gru.predict(X_test)
#pre_gru = scaler.inverse_transform(pre_gru)
pre_lstm = model_lstm.predict(X_test)
#pre_lstm = scaler.inverse_transform(pre_lstm)
pre_bilstm = Bilstm_model.predict(X_test)
#pre_bilstm = scaler.inverse_transform(pre_bilstm)
#y_test = scaler.inverse_transform(y_test)
```

รูปที่ 5.7 โค้ดการทดสอบโมเดล

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import mean_absolute_error as mae
from sklearn.metrics import mean_squared_error as mse

labels = ['MAE', 'RMSE', 'MSE']
predic_list = [pre_gru, pre_lstm, pre_bilstm]
effi = [[mae(ground, i), math.sqrt(mse(ground, i)), mse(ground, i)] for i in predic_list]
effi
print(labels)
print("GRU " + str(effi[0]))
print("LSTM " + str(effi[1]))
print("BiLSTM " + str(effi[2]))
```

รูปที่ 5.8 โค้ดการวัดประสิทธิภาพ

ประวัติผู้เขียน

ชื่อ - นามสกุล	บุญญพัฒน์ แปลงพระเนตร
รหัสนักศึกษา	61070122
สาขาวิชา	เทคโนโลยีสารสนเทศ
เบอร์โทร	098-383-6276
อีเมล	61070122@it.kmitl.ac.th
ปีการศึกษา	2564



ชื่อ - นามสกุล	ศรัณญ์ รื่นรวบ
รหัสนักศึกษา	61070215
สาขาวิชา	เทคโนโลยีสารสนเทศ
เบอร์โทร	095-952-0876
อีเมล	61070215@it.kmitl.ac.th
ปีการศึกษา	2564

