ปริญญานิพนธ์ปีการศึกษา 2565

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง กลไกการเพิ่มประสิทธิภาพในเครือข่ายที่กำหนดด้วยซอฟต์แวร์

PERFORMANCE IMPROVEMENT MECHANISM IN SOFTWARE-DEFINED NETWORK

ผู้จัดทำ

1. นายธีรัตม์ บุตราภาศ รหัสนักศึกษา 62010448
2. นายภัทรพัทธิ์ ชัยอมรเวทย์ รหัสนักศึกษา 62010684

_____อาจารย์ที่ปรึกษา

(รศ.ดร. ศักดิ์ชัย ทิพย์จักษุรัตน์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กลไกการเพิ่มประสิทธิภาพในเครือข่ายที่กำหนดด้วยซอฟต์แวร์

นาย ธีรัตม์ บุตราคาศ 62010448
นาย ภัทรพัทธิ์ ชัยอมรเวทย์ 62010684
รศ. ดร. ศักดิ์ชัย ทิพย์จักษุรัตน์ อาจารย์ที่ปรึกษา
ปีการศึกษา 2565

บทคัดย่อ

ปัจจุบัน การควบคุมเครือข่ายหรือการคอนฟิกเครือข่ายขึ้นอยู่กับอุปกรณ์ ซึ่งอาจจะเป็นอุปกรณ์จากผู้ผลิตที่แตกต่างกัน ทำให้กระบวนการคอนฟิกเครือข่ายอาจจะมีความแตกต่างกัน ดังนั้นเพื่อแก้ปัญหาดังกล่าว จึงได้มีการพัฒนาเทคโนโลยี “เครือข่ายที่กำหนดโดยซอฟต์แวร์” หรือ “เอสดีเอ็น” เพื่อเข้ามาช่วยแก้ปัญหาในเรื่องดังกล่าวและสามารถเรียนรู้ในเรื่องของเครือข่ายได้ง่ายขึ้น โดยโครงสร้างของเอสดีเอ็นจะมีส่วนการควบคุมหรือคอนโทรลเลอร์ ที่สามารถกำหนดการทำงานของเครือข่ายทั้งโครงสร้างโดยใช้ภาษาโปรแกรมมิ่งระดับสูงได้ ไม่ว่าจะเป็น ภาษาไพทอน หรือ ภาษาจาวา รวมถึงช่วยให้สามารถนำการเรียนรู้ของเครื่องเชิงลึกมาประยุกต์ใช้งานในด้านเครือข่าย

โครงงานนี้ผู้จัดทำได้นำเสนอวิธีการเพิ่มประสิทธิภาพในเครือข่ายที่กำหนดด้วยซอฟต์แวร์ โดยนำโมเดลการเรียนรู้เชิงลึกโครงข่ายประสาทเทียม Gated Recurrent Units (GRU) มาใช้ในการคาดการณ์การใช้ประโยชน์แบนด์วิดท์ของแต่ละลิงค์ที่จะเกิดขึ้นในอนาคต ผู้จัดทำได้นำโปรแกรม D-IGT มาใช้ในการจำลองสร้างแพ็กเก็ตเพื่อที่จะส่งข้อมูลจากอุปกรณ์ต้นทางไปยังอุปกรณ์ปลายทางที่ต้องการโดยผ่านสวิตช์และมีการเดินทางของข้อมูลในหลากหลายเส้นทาง ในโครงงานนี้เราได้นำริวคอนโทรลเลอร์มาใช้เป็นตัวควบคุมเครือข่ายหรือเรียกได้อีกอย่างว่าคอนโทรลเลอร์ ริวคอนโทรลเลอร์จะทำหน้าที่เก็บสถิติในการส่งข้อมูลแต่ละครั้งและนำไปส่งให้กับโมเดลการเรียนรู้เชิงลึกเพื่อทำการคาดการณ์การใช้ประโยชน์แบนด์วิดท์ในแต่ละลิงค์ในอนาคต เมื่อคาดการณ์เสร็จก็จะส่งข้อมูลกลับมาให้ที่ริวคอนโทรลเลอร์ ถ้าเกิดข้อมูลที่คาดการณ์มาได้มีค่าเกินเกณฑ์ที่กำหนดไว้ ก็จะทำการสกัดกั้นสวิตช์นั้น และทำการจัดเปลี่ยนเส้นทางใหม่ เพื่อหลีกเลี่ยงการเกิดการสูญหายของแพ็กเก็ต ตลอดจนลดระดับความคับคั่งในการส่งข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Performance Improvement Mechanism in Software-defined Network

Mr. Theerat	Budtrakart	62010448
Mr. Phattaraphat	Chaiamornvate	62010684
Assoc. Prof. Dr. Sakchai	Thipchaksurat	Advisor

Academic Year 2022

Abstract

At present, controlling a network or network configurations depends on devices, which might come from different vendors. This may cause various network configuration processes. Therefore, to overcome those problems, the "Software Defined Network" (SDN) has been developed. SDN can help to solve these problems and be able to acknowledge about the network system more easily. Software Defined Network's structure consists of control plane which can be adjusted a whole network system by using high-level programming language, such as python or java. Moreover, deep learning can be implemented in this technology too.

In this project, we propose a deep learning model, work based on Gated Recurrent Units (GRU) for predicting bandwidth utilization on each link in the future. We used D-ITG for generate a packets traffic in a network by sending from end-to-end devices, which passed through a switch. In this project, we used RYU controller for centralized a network system. Ryu controller's collecting packets statistic between links from each switch in a network and send to the GRU model for predicting bandwidth utilization in advance, then pass data back to a controller. If predicted bandwidth utilization's greater than threshold, we are going to block that path and do a Re-routing algorithm. We expect that our proposed method may reduce a congestion between links and avoid of packet loss.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานกลไกการเพิ่มประสิทธิภาพในเครือข่ายที่กำหนดโดยซอฟต์แวร์นี้ ผู้จัดทำขอขอบพระคุณอาจารย์ที่ปรึกษาปริญญานิพนธ์ รองศาสตราจารย์ ดร. ศักดิ์ชัย ทิพย์จักษุรัตน์ ที่กรุณาให้ความรู้ การแก้ปัญหา การบริหารโครงงาน และ คำปรึกษาต่างๆ ที่เกี่ยวกับแนวทางในการดำเนินโครงงานนี้ ตั้งแต่ริเริ่มจนกระทั่งพัฒนาเสร็จสมบูรณ์ พร้อมทั้งกรรมการปริญญานิพนธ์ ทุกท่านที่ร่วมให้คำแนะนำในการทำปริญญาพนธ์ให้สัมฤทธิ์ผลได้ด้วยดี ตลอดจนคณาจารย์ที่ให้ความรู้มาโดยตลอด

ขอกราบขอบพระคุณบิดามารดา ที่ให้การอุปการะเลี้ยงดู พร้อมทั้งสนับสนุน ส่งเสริมการศึกษา แนะนำแนวทางการแก้ไขอุปสรรคต่างๆ และให้กำลังใจมาโดยตลอด ผู้จัดทำหวังว่าปริญญานิพนธ์นี้จะเป็นประโยชน์กับท่านผู้ที่สนใจและผู้เกี่ยวข้อง รวมไปถึงเป็นประโยชน์ต่อสังคมในอนาคต

ธีรัตม์ บุตรากาศ

ภัทรพัทธิ์ ชัยอมรเวทย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย I

บทคัดย่อภาษาอังกฤษ II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญตาราง VII

สารบัญภาพ VIII

บทที่ 1 บทนำ 1

1.1 ความเป็นมาและความสำคัญ 1

1.2 วัตถุประสงค์ของโครงงาน 2

1.3 ขอบเขตการวิจัย 3

1.3.1 ขอบเขตด้านเนื้อหา 3

1.3.2 ขอบเขตด้านแหล่งข้อมูล 4

1.4 ประโยชน์ที่ได้รับ 4

1.5 แผนการดำเนินงาน 5

บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง 6

2.1 ทฤษฎีที่เกี่ยวข้อง 6

2.1.1 เครือข่ายที่กำหนดโดยซอฟต์แวร์ 6

2.1.2 โอเพนโฟลว์โพรโทคอล 10

2.1.3 การเรียนรู้เชิงลึก และ โครงข่ายประสาทเทียม 11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

2.1.4 Bidirectional Long-Short Term Memory (Bi-LSTM) และ Gated Recurrent Units (GRU) 13
2.1.5 อัลกอริทึมของไดก์สตรา 14
2.1.6 การกำหนดเส้นทาง 15
2.1.7 การสูญหายของข้อมูลแพ็กเก็ต 16
2.1.8 แบนด์วิดท์ 17
2.1.9 ทรูพุต 17
2.1.10 ดีเลย์ 18
2.2 งานวิจัยที่เกี่ยวข้อง 18
บทที่ 3 วิธีการนำเสนอ 18
3.1 องค์ประกอบของโครงงาน 20
3.1.1 ชั้นโครงสร้างพื้นฐาน 20
3.1.2 ริวเอสดีเอ็นคอนโทรลเลอร์ 21
3.1.3 แอปพลิเคชันคาดการณ์ค่าการใช้ประโยชน์แบนด์วิดท์ GRU 26
3.1.4 ขั้นตอนการสร้างแอปพลิเคชันในโครงงาน 27
3.2 ขั้นตอนการทำงานของระบบทั้งหมด 32
บทที่ 4 ผลการทดลอง 35
4.1 การติดตั้งโปรแกรม 35
4.1.1 การออกแบบโครงสร้างเครือข่าย 35
4.1.2 D-ITG (Distributed Internet Traffic Generator) 36

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

4.1.3 ริวคอนโทรลเลอร์ 38

4.2 การรวบรวมข้อมูล 39

4.2.1 การเทรนข้อมูล 40

4.3 ผลการทดลอง 43

4.3.1 พารามิเตอร์ของการทดลอง 43

บทที่ 5 สรุปผล 47

5.1 บทสรุป 47

5.1.1 การออกแบบเครือข่าย 48

5.1.2 ริวคอนโทรลเลอร์ 48

5.1.3 โมเดลการเรียนรู้เชิงลึก 48

5.2 ปัญหาและอุปสรรคที่พบ 48

5.2.1 ข้อจำกัดของเครื่องเสมือน 48

5.2.1 ปัญหาของซอฟต์แวร์ 49

5.3 แนวทางในการพัฒนาต่อ 51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตาราง **หน้า**

ตารางที่ 1.1 แผนการดำเนินงานในช่วงเทอมที่ 1 ปีการศึกษา 2565.....5
ตารางที่ 1.2 แผนการดำเนินงานในช่วงเทอมที่ 2 ปีการศึกษา 2565.....6
ตารางที่ 2.1 ฟิล์ดของ OFPPortStatusReply.....11
ตารางที่ 3.1 ตัวอย่างชุดข้อมูลที่เก็บมาจาก OFPPortStatsReply ด้วยภาษาไพทอน.....32
ตารางที่ 3.2 ค่าพารามิเตอร์ที่ใช้ในการทดลองของโมเดล Bi-LSTM และ GRU 37
ตารางที่ 4.1 พารามิเตอร์ที่ได้นำมาใช้จากเอกสารงานวิจัย.....40
ตารางที่ 4.2 แสดงค่าความผิดพลาดจากการทำนายของแต่ละโมเดล.....41
ตารางที่ 4.3 พารามิเตอร์ที่ใช้ในการทดลองเพื่อทำการจำลองเครือข่าย.....43
ตารางที่ 4.4 พารามิเตอร์ที่ใช้ในการทดลองส่งข้อมูลด้วยซฮฟต์แวร์ D-ITG 44

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

รูป **หน้า**

รูปที่ 2.1 โครงสร้างพื้นฐานของเครือข่ายทั่วไป 7
รูปที่ 2.2 โครงสร้างพื้นฐานของเครือข่ายที่กำหนดโดยซอฟต์แวร์ 8
รูปที่ 2.3 สถาปัตยกรรมของเอสดีเอ็นที่ถูกแบ่งออกเป็น 3 ชั้น 9
รูปที่ 2.4 การเปรียบเทียบระหว่างการเรียนรู้ด้วยเครื่องและการเรียนรู้เชิงลึก 12
รูปที่ 2.5 หลักการทำงานระหว่างโมเดล Uni-LSTM และ Bi-LSTM 13
รูปที่ 2.6 องค์ประกอบของโครงข่ายประสาทเทียมระหว่าง GRU และ LSTM 14
รูปที่ 2.7 การประยุกต์ใช้ออลกอริทึมไดก์สตรากับระบบเอสดีเอ็น 14
รูปที่ 2.8 หลักการส่งข้อมูลภายในเครือข่าย 15
รูปที่ 2.9 การเปรียบเทียบระหว่างแบนด์วิดท์ และ ทรูพุต 17
รูปที่ 3.1 ภาพรวมของระบบเอสดีเอ็นของโครงงาน 18
รูปที่ 3.2 การเปรียบเทียบโครงสร้างระหว่าง SDN มาตรฐาน และโครงสร้าง SDN ของโครงงาน 19
รูปที่ 3.3 องค์ประกอบของเอสดีเอ็นอย่างละเอียด 20
รูปที่ 3.4 โครงสร้างเครือข่ายที่จะลองขึ้นผ่าน MiniEdit 21
รูปที่ 3.5 Ryu SDN Framework แบบทั่วไป 22
รูปที่ 3.6 Ryu SDN Framework ของโครงงาน 22
รูปที่ 3.7 โมดูลสังเกตการณ์การส่งข้อมูลภายในเครือข่ายในแต่ละวินาที 23
รูปที่ 3.8 การร้องขอข้อมูลเชิงสถิติระหว่างริวคอนโทรลเลอร์และโอเพนโฟลว์สวิตช์ 24
รูปที่ 3.9 การทำงานระหว่างชั้นแอปพลิเคชัน และ ชั้นควบคุม 27
รูปที่ 3.10 การรวบรวมข้อมูลจากเอสดีเอ็นไปยังแพลตฟอร์มจูปิเตอร์ 28
รูปที่ 3.11 กระบวนการเตรียมข้อมูลก่อนทำการเทรนโมเดล Bi-LSTM และ GRU 29
รูปที่ 3.12 โครงสร้างของโครงข่ายประสาทเทียมชนิด Bi-LSTM และ GRU ของโครงงาน 31
รูปที่ 3.13 ภาพรวมขั้นตอนทำการทำงานของระบบทั้งหมด 32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

รูป **หน้า**

รูปที่ 3.14 โฟลว์ชาร์ตการทำงานของระบบอย่างละเอียด....33
รูปที่ 3.15 โฟลว์ชาร์ตการทำงานของโมดูลการจัดเส้นทางใหม่....34
รูปที่ 4.1 การออกแบบเครือข่ายโดยใช้ซอฟต์แวร์มินิเน็ต....36
รูปที่ 4.2 โครงสร้างของซอฟต์แวร์ D-ITG....37
รูปที่ 4.3 ตัวอย่างข้อมูลทางสถิติจากซอฟต์แวร์ D-ITG....37
รูปที่ 4.4 โมดูลสังเกตการณ์เครือข่าย (Traffic Monitor Module)....38
รูปที่ 4.5 ขั้นตอนการนำข้อมูลจาก Virtual Machine สู่ระบบปฏิบัติการวินโดว์ผ่าน Git....39
รูปที่ 4.6 ผลลัพธ์ของการเรียนรู้ระหว่างโมเดล LSTM และ Bi-LSTM....41
รูปที่ 4.7 ผลลัพธ์ของการเรียนรู้ระหว่างโมเดล LSTM และ GRU....42
รูปที่ 4.8 เวลาที่ใช้ในการคาดการณ์ของโมเดล GRU....43
รูปที่ 4.9 เวลาที่ใช้ในการคาดการณ์ของโมเดล LSTM....43
รูปที่ 4.10 การประเมินประสิทธิภาพเครือข่ายที่วัดจากดีเลย์....44
รูปที่ 4.11 การประเมินประสิทธิภาพเครือข่ายที่วัดจากอัตราสูญหายของแพ็กเก็ต....45
รูปที่ 4.12 การประเมินประสิทธิภาพเครือข่ายที่วัดจากอัตราสูญหายของทรูพุต....45
รูปที่ 5.1 องค์ประกอบของซฮฟต์แวร์ sFlow-RT....49
รูปที่ 5.2 ตัวอย่างการตรวจสอบโฟลว์แบบเรียลไทม์ของซอฟต์แวร์ sFlow-RT....50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญ

เครือข่ายที่กำหนดโดยซอฟต์แวร์ (Software-Defined Network) หรือ เอสดีเอ็น (SDN) เป็นสถาปัตยกรรมของเครือข่ายที่ออกแบบมาเพื่อที่จะสามารถควบคุมเครือข่ายทั้งหมดมารวมอยู่ที่ซอฟต์แวร์เพียงจุดเดียว เพื่อแก้ปัญหาการทำงานร่วมกันของอุปกรณ์ที่มีการจัดการเป็นไปลักษณะ “ปิด” หรือขึ้นอยู่กับอุปกรณ์ของแต่ละผู้ผลิต ความสามารถของเอสดีเอ็นมีความหลากหลาย เพราะ เป็นสถาปัตยกรรมที่สามารถปรับเปลี่ยนเพื่อปรับปรุงการทำงานทั้งระบบเครือข่ายได้ ซึ่งทำให้บริหารการทำงานของเครือข่ายให้มีประสิทธิภาพมากขึ้น

ปัจจุบันการตั้งค่าอุปกรณ์เครือข่ายทั่วไปนั้น จะแตกต่างกันออกไปตามผู้ผลิต (Vendor Specific Command) ซึ่งเป็นปัญหาหลักๆ ของเครือข่ายแบบเก่าที่ผู้ดูแลหรือผู้ทำงานด้านเครือข่ายต้องเผชิญอยู่เป็นประจำ ทุกๆ วันนี้มีการใช้งานเครือข่ายกันอย่างมาก ทำให้เกิดการขยายเครือข่าย (Scaling) ในที่สุด นอกจากนี้ผู้ผลิตอุปกรณ์เครือข่ายก็มีการแข่งขันสูง ทำให้ผลิตอุปกรณ์เครือข่ายที่มีการตั้งค่าที่แตกต่างกันออกไปเป็นอย่างมาก ในการติดตั้งเอุปกรณ์เครือข่ายร่วมกันจากหลายๆ ผู้ผลิตนั้น ทำไปได้ค่อนข้างยาก ซึ่งเกิดจากปัจจัยหนึ่งก็คือความสามารถของแต่ละอุปกรณ์ไม่เท่ากัน (Vendor Limitations) ประเด็นนี้เป็นสาเหตุที่ทำให้การดูแลเครือข่ายรวมเป็นไปในทางที่ค่อนข้างยาก และ ลำบากต่อการจัดการอุปกรณ์ทุกเครื่องโดยตรง

การเรียนรู้เชิงลึก (Deep Learning) เป็นส่วนหนึ่งของปัญญาประดิษฐ์ (Artificial Intelligence) และการเรียนรู้ด้วยเครื่อง (Machine Learning) ซึ่งเป็นการเรียนรู้ด้วยตัวเองหรือแบบอัตโนมัติ และถูกจำลองขึ้นเป็นโครงข่ายประสาทเทียม (Artificial Neural Network) ในโครงข่ายประสาทเทียมจะประกอบไปด้วย 3 ชั้นหลักๆ ได้แก่ ชั้นอินพุต ชั้นประมวลผล ชั้นเอาท์พุต ซึ่งการเรียนรู้เชิงลึกจะมีความซับซ้อนกว่าการเรียนรู้ด้วยเครื่องตรงชั้นประมวลผล ในชั้นประมวลผลจะประกอบไปด้วยหลายๆ ชั้น เพื่อเรียนรู้ข้อมูลจากชั้นอินพุตให้เป็นแพทเทิร์น นอกจากนี้ยังทำการคัดกรองข้อมูลแล้วสรุปผลผ่านทางชั้นเอาท์พุต ซึ่งไม่มีความแน่นอนว่าจะผิดหรือถูก เมื่อมีการเรียนรู้เชิงลึกที่มีความแม่นยำมากเท่าไหร่ประสิทธิภาพในการประยุกต์เข้ากับเอสดีเอ็นก็จะมากขึ้นไปด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โพรโทคอลการจัดการเส้นทาง (Routing Protocol) คือ โพรโทคอลที่ใช้ในการแลกเปลี่ยนข้อมูลเส้นทาง เพื่อที่จะสร้างตารางข้อมูลเส้นทาง (Routing Table) ของแต่ละอุปกรณ์ ระหว่างอุปกรณ์เครือข่ายต่างๆ ที่ทำงานในชั้นการสื่อสารที่ 3 ของโมเดลโอเอสไอ (OSI Model) ซึ่งก็คือชั้นเครือข่าย (Network Layer) เมื่ออุปกรณ์ทราบเส้นทางภายในเครือข่ายแล้ว อุปกรณ์นั้นก็จะสามารถส่งข้อมูลหรือแพ็กเก็ต (Packet) ไปยังอุปกรณ์เป้าหมายได้อย่างถูกต้อง เพื่อที่จะให้การส่งข้อมูลมีประสิทธิภาพที่ดี เส้นทางที่จะทำการส่งข้อมูลนั้นต้องเป็นการส่งข้อมูลที่เร็วที่สุดที่จะเป็นไปได้ หรือ เส้นทางที่สั้นที่สุด (Shortest Path) ซึ่งเส้นทางจะถูกเลือกด้วยอัลกอริทึมการจัดการเส้นทาง (Routing Algorithm) ในการคำนวณก็จะประกอบไปด้วยจำนวนข้อมูลที่ถูกส่งหรือเปลี่ยนแปลงภายในลิงค์ (Link) หรือ กรณีที่อุปกรณ์หรือโหนด (Node) ไม่สามารถทำงานได้ เป็นต้น นอกจากนี้ยังมีกรณีการสูญหายของแพ็กเก็ต (Packet Loss) ซึ่งเกิดจากความสามารถในการรับข้อมูลมีมากกว่าการส่งข้อมูล เราเตอร์ (Router) จะมีวิธีการจัดเรียงการส่งข้อมูล (Queuing) และจะถูกจัดเก็บอยู่ในหน่วยความจำ (Buffer) เมื่อหน่วยความจำเก็บข้อมูลจนเต็มที่แล้วและไม่สามารถรับข้อมูลเข้ามาอีกได้ จึงเป็นเหตุที่ทำให้เราเตอร์ต้องเลือกที่จะละทิ้งข้อมูลบางส่วนออกไปและทำให้เกิดการสูญหายของแพ็กเก็ตในที่สุด

เนื่องจากเอสดีเอ็น (SDN) เป็นสถาปัตยกรรมเครือข่ายที่ถูกออกแบบมาเพื่อให้สามารถถูกควบคุมด้วยซอฟต์แวร์ได้นั้น เราจึงสามารถนำเอาหลักการของการเรียนรู้เชิงลึก (Deep Learning) มาประยุกต์ใช้กับเอสดีเอ็น เพื่อทำให้ประสิทธิภาพของเครือข่ายเพิ่มขึ้นได้ เช่น การทำ Intrusion detection และ Protection systems การทำ Programmable firewall การทำ Routing Optimization เป็นต้น สำหรับโครงงานนี้เราได้นำโครงข่ายประสาทเทียมมาประยุกต์ใช้ในด้านการจัดการเพิ่มประสิทธิภาพในการเลือกเส้นทางที่ดีที่สุดของอุปกรณ์ในเครือข่าย โดยใช้โมเดล Gated Recurrent Units ซึ่งเราจะใช้โมเดลคาดการณ์การใช้ประโยชน์แบนด์วิดท์ (Bandwidth Utilization) สำหรับอนาคตของแต่ละลิงค์ของเครือข่าย หากการคาดการณ์สื่อความหมายได้ว่าจะเกิดการส่งข้อมูลมากจนเกิดความคับคั่งข้อมูลได้ ตัวคอนโทรลเลอร์ก็จะทำการเปลี่ยนเส้นทางที่ดีที่สุดในการส่งข้อมูลชุดใหม่ต่อไป ดังนั้นการประยุกต์ใช้การเรียนรู้เชิงลึกกับเอสดีเอ็นจะสามารถทำให้การทำงานของเอสดีเอ็นมีประสิทธิภาพที่สูงขึ้นได้

1.2 วัตถุประสงค์ของโครงงาน

1) เพื่อศึกษาหลักการทำงานของเอสดีเอ็น (SDN) เนื่องจากเป็นเทคโนโลยีที่น่าสนใจ โดยทั่วไปแล้ว อุปกรณ์ทางด้านเครือข่ายจะไม่สามารถเข้าไปคอนฟิกได้อิสระและซอฟต์แวร์ของเครื่องถูกกำหนดโดยบริษัทชั้นนำ เช่น Cisco, Juniper เป็นต้น ซึ่งเอสดีเอ็นสามารถควบคุมเครือข่ายโดยใช้ซอฟต์แวร์เป็นตัวกำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) เพื่อนำการเรียนรู้เชิงลึกเข้ามาช่วยยกระดับประสิทธิภาพในด้านเครือข่ายคอมพิวเตอร์
3) เพื่อนำเสนออัลกอริทึมในการปรับปรุงประสิทธิภาพของกระบวนการจัดการเส้นทางในเอสดีเอ็น โดยนำโมเดลการเรียนรู้เชิงลึกเข้ามาประยุกต์ใช้
4) เพื่อเปรียบเทียบอัลกอริทึม หรือ โมเดลที่ใช้ในเครือข่ายจำลอง และ นำเสนอถึงข้อดีและข้อเสียของแต่ละวิธี
5) เพื่อศึกษากระบวนการทำวิจัยอย่างเป็นระบบมีแบบแผน

1.3 ขอบเขตการวิจัย

ขอบเขตการวิจัยในโครงงานนี้ถูกแบ่งออกเป็น 2 ด้าน ได้แก่ ด้านเนื้อหา และ ด้านแหล่งข้อมูล โดยมีรายละเอียดดังนี้

1.3.1 ขอบเขตด้านเนื้อหา

โครงงานนี้มีวัตถุประสงค์ในการเพิ่มประสิทธิภาพของเครือข่ายที่กำหนดโดยซอฟต์แวร์ โดยนำโครงข่ายประสาทเทียมมาใช้สำหรับการคาดการณ์แบนด์วิดท์ของลิงค์ภายในเครือข่าย เพื่อทำการจัดการเส้นทางให้มีประสิทธิภาพเพิ่มขึ้น และ หลีกเลี่ยงการเกิดการสูญหายของแพ็กเก็ตจากอุปกรณ์ในเครือข่ายที่มีความจุบัฟเฟอร์ไม่เพียงพอต่อการรับข้อมูล รวมไปถึงการลดระยะเวลาในการส่งข้อมูลเล็กน้อยและหลีกเลี่ยงความคับคั่งในการเดินทางของการส่งข้อมูล

1) มินิเน็ต (Mininet) เป็นอิมูเลเตอร์จำลองเครือข่าย (Network Emulator) ซึ่งสามารถใช้งานบนระบบปฏิบัติการได้หลายรูปแบบ ซึ่งโปรแกรมนี้จะใช้งานสำหรับการจำลองระบบเครอข่ายขึ้นมา ซึ่งในเครือข่ายจะประกอบไปด้วย โอเพนโฟลว์สวิตช์ (OpenFlow switch), โฮสท์ (Host) และลิงค์ (Link) ขึ้นมา โดยอิมูเลเตอร์ในโครงงานนี้ทำงานบนระบบปฏิบัติการอุบุนตู (Ubuntu) ซึ่งทำงานผ่านเครื่องเสมือน (Virtual Machine) อีกทีหนึ่ง

2) ริว (RYU) เป็นเฟรมเวิร์คของเครือข่ายที่กำหนดโดยซอฟต์แวร์ ซึ่งทำหน้าที่เป็นส่วนควบคุมระบบเครือข่าย (Network System) และสามารถสั่งงานผ่านภาษาระดับสูงได้อย่างภาษาไพทอน นอกเหนือจากการควบคุมระบบ RYU ทำหน้าที่รวบรวมข้อมูลภายในเครือข่ายจากอุปกรณ์ OpenFlow ที่ถูกมินิเน็ตสร้างขึ้น ส่งให้กับเน็ตเวิร์คแอปพลิเคชันที่สร้างขึ้นโดยใช้โมเดลของการเรียนรู้เชิงลึกแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Gated Recurrent Units ในการคาดการณ์ค่าการใช้ประโยชน์แบนด์วิดท์ที่จะเกิดขึ้นในอนาคตภายในลิงค์ของเครือข่าย

3) การเรียนรู้เชิงลึก (Deep Learning) เป็นอัลกอริทึมหรือกระบวนการที่ใช้การเรียนรู้และเลียนแบบเครือข่ายเซลล์ประสาท (Neuron Network) เป็นส่วนที่จะทำให้ประสิทธิภาพของเครือข่ายที่กำหนดโดยซอฟต์แวร์เพิ่มขึ้นจากการคาดการณ์การใช้ประโยชน์แบนด์วิดท์ เพื่อลดความคับคั่งของข้อมูลและป้องกันการเกิดการสูญหายของแพ็กเก็ตได้

1.3.2 ขอบเขตด้านแหล่งข้อมูล

การวิจัยในโครงงานนี้เราได้จำลองเครือข่ายผ่านเครื่องเสมือน (Virtual Machine) โดยใช้โปรแกรม Oracle VM VirtualBox ซึ่งภายในเครื่องเสมือนจะมีการติดตั้งระบบปฏิบัติการอุบุนตู (Ubuntu) และ มีองค์ประกอบโปรแกรมต่างๆ ได้แก่ ริวคอนโทรลเลอร์, มินิเน็ตอีมูเลเตอร์ และโปรแกรมดีไอทีจี (D-ITG) โดยโปรแกรมสามารถทำงานร่วมกันได้ภายในระบบปฏิบัติการนี้

1.4 ประโยชน์ที่ได้รับ

การจัดเส้นทางเป็นกระบวนการหนึ่งที่สำคัญในเครือข่าย ซึ่งเป็นการจัดเส้นทางให้กับข้อมูลที่ต้องการส่งจากโฮสต์ต้นทางไปยังโอสต์ปลายทาง แต่อาจจะมีบางเครือข่ายมีกระบวนการจัดการเส้นทางไม่มีประสิทธิภาพเพียงพอต่อการรองรับปริมาณข้อมูลจำนวนมากในเครือข่าย เนื่องจากเส้นทางการส่งข้อมูลจะถูกพิจารณาด้วย Shortest Path Algorithm เพียงอย่างเดียว โดยไม่ได้มีการคาดการณ์ล่วงหน้าว่าจะมีกรณีที่หน่วยความจำถูกใช้งานเกินความจุหรือไม่ ซึ่งส่งผลให้อุปกรณ์เครือข่ายนั้นทำการละทิ้งแพ็กเก็ตที่เข้ามาหลังจากนั้นทั้งหมดจนกว่าพื้นที่หน่วยความจำว่างพอที่จะรับข้อมูลชุดใหม่ได้ เหตุดังกล่าวทำให้เกิดการสูญหายของแพ็กเก็ต (Packet Loss)

โครงงานชิ้นนี้จะมุ่งเน้นไปที่การแก้ไขปัญหาดังกล่าว โดยการใช้โครงข่ายประสาทเทียมแบบวนกลับชนิด Gated Recurrent Units (GRU) มาทำเป็นแอปพลิเคชันสำหรับการคาดการณ์ค่าการใช้ประโยชน์แบนด์วิดท์ (Bandwidth Utilization) ที่กำลังจะเกิดขึ้นบนลิงค์ โดยแอปพลิเคชันนี้จะทำงานอยู่บนคอนโทรลเลอร์ ซึ่งเป็นเสมือนสมองของเครือข่าย ที่ควบคุมการทำงานระหว่างโครงสร้างพื้นฐานเครือข่าย (Network Infrastructure) และเน็ตเวิร์กแอปพลิเคชัน ค่าการใช้ประโยชน์แบนด์วิดท์ที่ถูกคาดการณ์โดยโมเดล GRU จะถูกนำมาตัดสินใจว่าลิงค์ใดที่จะเกิดความคับคั่งของข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Congestion) แล้วจึงตัดสินใจเปลี่ยนเส้นทางการส่งแพ็กเก็ตด้วยการออกคำสั่งผ่าน ริวเอสดีเอ็น คอนโทรลเลอร์ ไปยังโครงสร้างพื้นฐานเครือข่าย โดยการเปลี่ยนเส้นทางการส่งแพ็กเก็ตจะช่วยลดอัตราการสูญหายของแพ็กเก็ตเนื่องจาการเปลี่ยนเส้นทางข้อมูล ก่อนที่จะเกิดความคับคั่งขึ้นมาจริง

1.5 แผนการดำเนินงาน

ขั้นตอน	สิงหาคม				กันยายน				ตุลาคม					พฤศจิกายน			
การดำเนินงาน	6	13	20	27	3	10	17	24	1	8	15	22	29	5	12	19	26
1.เริ่มทดสอบซอฟต์แวร์พื้นฐานที่ใช้สำหรับโครงงานนี้																	
2.นำทฤษฎีเครือข่ายนำมาประยุกต์เข้าร่วมกับเอสดีเอ็น																	
3.ศึกษาเอกสารที่มีหัวข้อเกี่ยวข้องกับโครงงาน																	
4.ศึกษาคำสั่งรวมถึงออกแบบโครงสร้างเครือข่าย																	
5.นำข้อมูลที่ได้จากการจำลองมาเทรนผ่านโมเดล Bi-LSTM																	
6.ทำการวิเคราะห์ผลลัพธ์ของโมเดล และออกแบบคอนโทรลเลอร์																	
7.จัดเตรียมรายงานและไฟล์การนำเสนอโครงงาน 1																	

ตารางที่ 1.1 แผนการดำเนินงานในช่วงเทอมที่ 1 ปีการศึกษา 2565

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการดำเนินงาน	มกราคม				กุมภาพันธ์				มีนาคม				เมษายน			
	7	14	21	28	4	11	18	25	4	11	18	25	1	8	15	22
1.ปรับปรุงโครงงานหลังสอบโครงงาน 1 และ ศึกษาทฤษฎีเพิ่มเติม																
2.นำโมเดลที่ศึกษาได้เพิ่มเติมมาประยุกต์ใช้เข้ากับโครงงาน																
3.นำโมเดลการเรียนรู้เชิงลึกเข้ามาทำงานบนระบบ SDN																
4.ทำการประเมินประสิทธิภาพของแต่ละโมเดล																
5.จัดเตรียมรายงานและไฟล์การนำเสนอโครงงาน 2																
6.จัดทำปริญญานิพนธ์																

ตารางที่ 1.2 แผนการดำเนินงานในช่วงเทอมที่ 2 ปีการศึกษา 2565

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในบทที่ 2 เราได้นำเสนอทฤษฎีที่เกี่ยวข้องกับเครือข่ายที่กำหนดโดยซอฟต์แวร์ (Software Defined Network) หรือ เอสดีเอ็น (SDN) การเรียนรู้เชิงลึก (Deep Learning) และตัวชี้วัดต่างๆ สำหรับวัดประสิทธิภาพเครือข่ายที่กำหนดโดยซอฟต์แวร์ (Performance Evaluation) รวมไปถึงงานวิจัยที่ได้นำมาประยุกต์ใช้และปรับปรุง งานวิจัยที่ศึกษา ได้นำการเรียนรู้เชิงลึกเข้ามาประยุกต์ใช้เพื่อเพิ่มประสิทธิภาพให้กับเครือข่ายเอสดีเอ็น เช่น การคาดการณ์การใช้ข้อมูลแพ็กเก็ต โดยใช้การเรียนรู้ของเครื่อง (Machine Learning) เพื่อทำให้เครือข่ายส่งข้อมูลได้อย่างชาญฉลาดมากขึ้น (Network Intelligence)

2.1 ทฤษฎีที่เกี่ยวข้อง

ในโครงงานนี้ เรามีวัตถุประสงค์ในการศึกษาการเพิ่มประสิทธิภาพของเครือข่ายที่กำหนดโดยซอฟต์แวร์ ซึ่งเราสามารถอธิบายทฤษฎีที่เกี่ยวข้อง (Theoretical Background) ได้ดังนี้

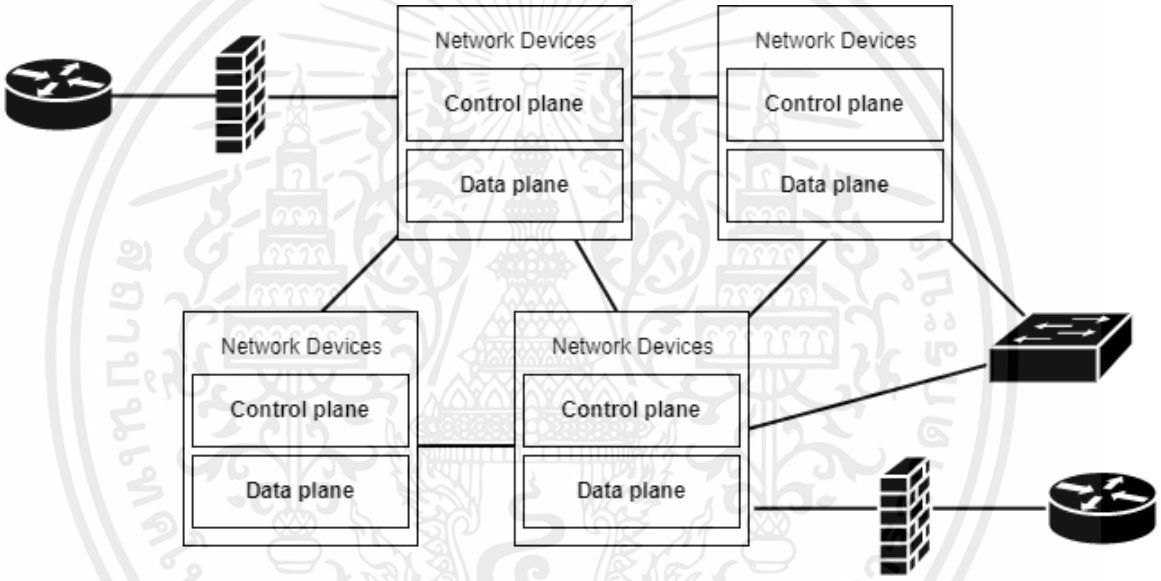
2.1.1 เครือข่ายที่กำหนดโดยซอฟต์แวร์

เครือข่ายที่กำหนดโดยซอฟต์แวร์ (Software-Defined Network) หรือ เอสดีเอ็น เป็นเทคโนโลยีที่ช่วยอำนวยความสะดวกให้กับองค์กรในการเปลี่ยนแปลงการตั้งค่าอุปกรณ์จากเดิม โดยการใช้ซอฟต์แวร์เข้ามาช่วยในการควบคุมแต่ละส่วนภายในระบบเครือข่าย หรือเรียกอีกชื่อหนึ่งได้ว่า “Programmable Network” ซึ่งสามารถควบคุมผ่านภาษาโปรแกรมมิ่งระดับสูงได้ เช่น ไพทอน (Python) และ จาวา (Java) เป็นต้น ซึ่งแตกต่างจากพื้นฐานทั่วไปในปัจจุบัน เนื่องจากการตั้งค่าอุปกรณ์จะขึ้นอยู่กับผู้ผลิตของแต่ละเครือนั้นๆ ไม่ว่าจะเป็น Cisco, Juniper และอื่นๆ

ในส่วนของอุปกรณ์เครือข่ายแบบทั่วไป จะประกอบไปด้วย ส่วนควบคุม (Control Plane) เป็นส่วนประมวลผลของอุปกรณ์ และ ส่วนการส่งข้อมูล (Data Plane) ที่คอยรับมาจากส่วนควบคุมอีกทีหนึ่งและส่งไปยังอุปกรณ์อีกอันหนึ่งภายในเครือข่าย ทั้งสองส่วนนี้จะถูกรวมกันไว้ในอุปกรณ์ทั้งหมดดังแสดงในรูปที่ 2.1

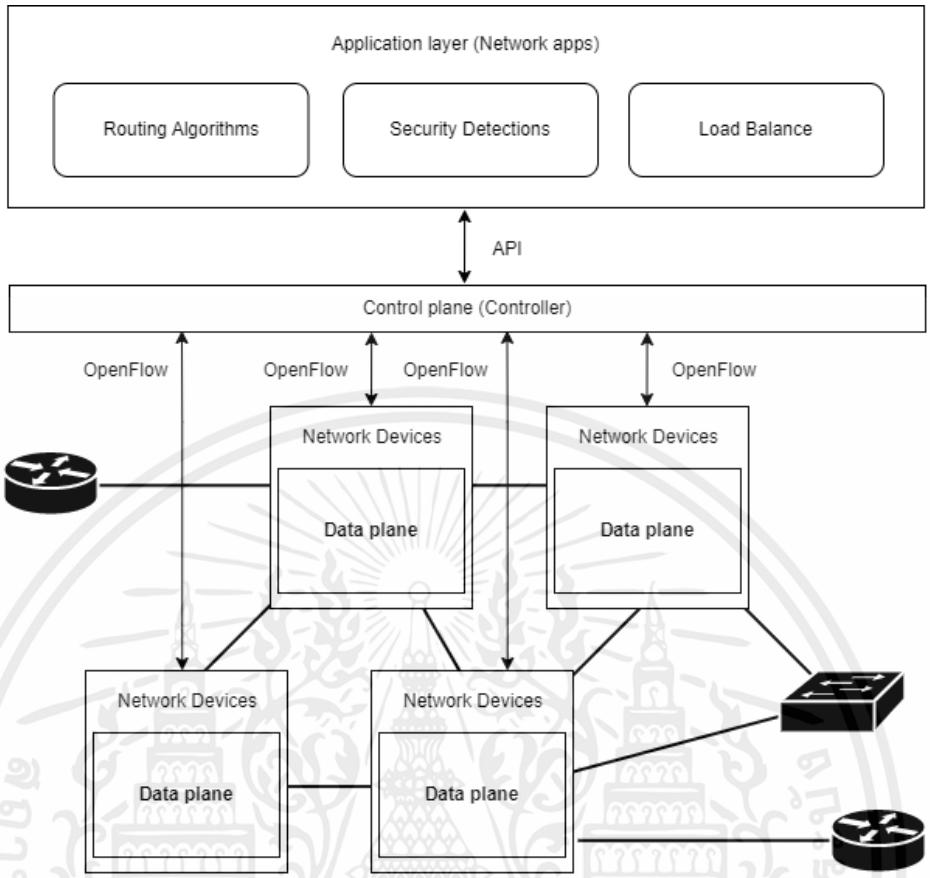
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สถาปัตยกรรมของเครือข่ายที่กำหนดโดยซอฟต์แวร์จะถูกออกแบบเป็นสองส่วนหลักๆ คือ ส่วนควบคุม (Control Plane) และ ส่วนอุปกรณ์ (Data Plane) ซึ่งสองส่วนนี้จะถูกแยกออกจากกัน ดังแสดงในรูปที่ 2.2 ส่วนควบคุมภายในเทคโนโลยีนี้จะทำหน้าที่เป็น "Centralized Control Plane" หรือที่เรียกกันว่าคอนโทรลเลอร์ (Controller) เปรียบเป็นสมองของเอสดีเอ็น เนื่องจากทำหน้าที่ควบคุมการส่งข้อมูลของเครือข่าย ประมวลผลข้อมูล เก็บสถิติ และสามารถตั้งค่าอุปกรณ์ในเครือข่ายได้ โดยคำสั่งต่างๆ จะมาจากคอนโทรลเลอร์และส่งไปยังส่วนอุปกรณ์ (Data Plane) ผ่านโพรโทคอลกลางโอเพนโฟลว์ (OpenFlow Protocol)



รูปที่ 2.1 โครงสร้างพื้นฐานของเครือข่ายทั่วไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



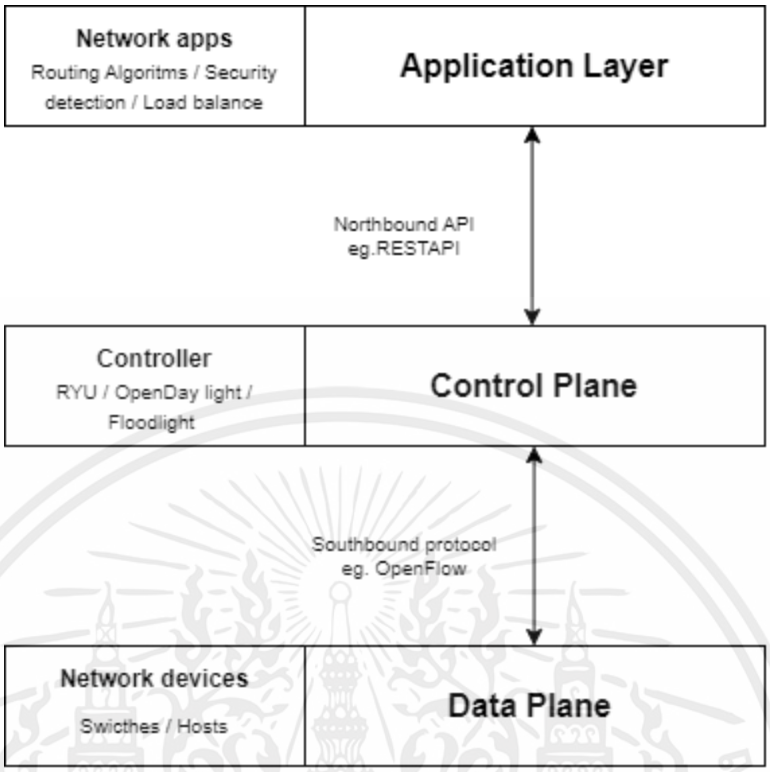
รูปที่ 2.2 โครงสร้างพื้นฐานของเครือข่ายที่กำหนดโดยซอฟต์แวร์

จากหัวข้อ 2.1.1 จะสังเกตได้ว่า เครือข่ายที่กำหนดโดยซอฟต์แวร์มีความสามารถในการควบคุมบริหารจัดการระบบเครือข่ายได้ง่ายกว่าระบบเครือข่ายทั่วไป เนื่องจากไม่ต้องคำนึงถึงอุปกรณ์ที่ถูกผลิตออกมาว่าเป็นอุปกรณ์มาจากบริษัทเดียวกันหรือไม่ การปรับเปลี่ยนขนาดของเครือข่าย (Scalability) วางแผนง่ายขึ้น ประหยัดค่าใช้จ่าย และไม่ยุ่งยาก การตั้งค่าอุปกรณ์เครือข่ายสามารถทำได้โดยใช้ภาษาโปรแกรมมิ่งระดับสูงภาษาเดียวกันได้ ดังนั้นเอสดีเอ็นจึงมียืดหยุ่นในการควบคุมและการเข้าถึงที่ง่ายกว่าเดิม

2.1.1.1 สถาปัตยกรรมของเครือข่ายที่กำหนดโดยซอฟต์แวร์

จากรูปที่ 2.2 ในหัวข้อที่ 2.1.1 จะสังเกตได้ว่าสถาปัตยกรรมของเอสดีเอ็น (SDN Architecture) จะถูกแบ่งออกเป็น 3 ส่วน ได้แก่ ชั้นแอปพลิเคชันเครือข่าย ชั้นการควบคุม และชั้นอุปกรณ์หรือโครงสร้างพื้นฐานเครือข่าย ซึ่งแต่ละชั้นจะทำหน้าที่แตกต่างกันออกไปดังแสดงในรูป 2.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 สถาปัตยกรรมของเอสดีเอ็นที่ถูกแบ่งออกเป็น 3 ชั้น

1) ชั้นแอปพลิเคชันเครือข่าย โดยทั่วไป เครือข่ายที่เป็นรูปแบบทั่วไปจะใช้อุปกรณ์ฮาร์ดแวร์ในการทำแอปต่างๆ เช่น ไฟร์วอลล์ (Firewall) แต่ภายในระบบของเอสดีเอ็นจะเปลี่ยนจากอุปกรณ์ฮาร์ดแวร์เช่นนี้ ให้อยู่ในรูปแบบของซอฟต์แวร์แทน เพื่อที่จะสามารถควบคุมโดยใช้โปรแกรมระดับสูงได้

2) ชั้นการควบคุม (Control Plane) เปรียบเสมือนเป็นส่วนสมองของระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์ คอยทำหน้าที่จัดการและควบคุมระบบเครือข่าย (Network System) การไกลของข้อมูล (Traffic Flow) รวมไปถึงการเก็บสถิติของข้อมูล (Statistic Flow) โดยทั้งหมดนี้จะถูกทำงานภายใต้เครือข่ายเอสดีเอ็นคอนโทรลเลอร์ (SDN Controller) และยังคงทำหน้าที่ควบคุมอุปกรณ์ภายในชั้นอุปกรณ์โครงสร้างพื้นฐานเครือข่าย

3) ชั้นอุปกรณ์โครงสร้างพื้นฐานเครือข่าย (Data Plane) เป็นส่วนที่ประกอบไปด้วยอุปกรณ์เครือข่าย เช่น สวิตช์ (Switch) และ โฮสต์ (Host) โดยชั้นนี้มีหน้าที่หลักๆ คือส่งข้อมูลตามโฟลว์เทเบิล (Flow table) ที่ระบุไว้ และติดต่อกับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอนโทรลเลอร์ผ่านโอเพนโฟลว์โพรโทคอล (OpenFlow Protocol) เพื่อส่งค่าสถิติให้กับคอนโทรลเลอร์เฟรมเวิร์ค (Controller Framework) และจัดเส้นทางให้แพ็กเก็ต (Packet) แต่ละอันว่าควรส่งไปยังเส้นทางไหน เพื่อให้ถึงปลายทางอย่างถูกต้อง

2.1.2 โอเพนโฟลว์โพรโทคอล

โอเพนโฟลว์โพรโทคอล (OpenFlow Protocol) เป็นโพรโทคอลพื้นฐานที่ถูกใช้ภายในเอสดีเอ็น และ เป็นโพรโทคอลที่คอนโทรลเลอร์ใช้ติดต่อสื่อสารกับอุปกรณ์ภายในเครือข่ายที่กำหนดโดยซอฟต์แวร์

2.1.2.1 โอเพนโฟลว์สวิตช์

โอเพนโฟลว์สวิตช์ (OpenFlow Switch) เป็นอุปกรณ์สวิตช์ที่สามารถทำงานร่วมกับโอเพนโฟลว์โพรโทคอล รวมไปถึงสามารถทำงานร่วมกับอุปกรณ์จำลอง (Virtual Machine) เช่น มินิเน็ต ที่สามารถทำงานร่วมกันได้

2.1.2.2 ข้อความโอเพนโฟลว์

ข้อความโอเพนโฟลว์ (OpenFlow Message) เป็นข้อความที่คอนโทรลเลอร์และโอเพนสวิตช์ใช้ติดต่อสื่อสารร่วมกันโดยผ่านโอเพนโฟลว์โพรโทคอล ถูกแบ่งออกเป็น 3 ประเภท ได้แก่

1) Controller-to-switch เป็นข้อความที่ถูกสร้างขึ้นโดยคอนโทรลเลอร์ ซึ่งข้อความแต่ละข้อความจะต้องมีการตอบกลับจากโอเพนโฟลว์สวิตช์หรือไม่ก็ได้ ขึ้นอยู่กับประเภทของข้อความ เช่น Send-Packet ใช้เพื่อบอกว่าแพ็กเก็ตไปยังพอร์ตไหนของสวิตช์
2) Symmetric เป็นข้อความที่ถูกสร้างขึ้นโดยคอนโทรลเลอร์หรือสวิตช์ก็ได้ ซึ่งไม่มีข้อบังคับใดๆ ในการส่ง เช่น Echo ใช้เพื่อเช็คลิงก์ระหว่างฝั่งตรงข้ามกัน
3) Asynchronous เป็นข้อความที่ถูกสร้างขึ้นโดยสวิตช์และส่งไปยังคอนโทรลเลอร์ เพื่อเตือนการทำงานของสวิตช์ให้คอนโทรลเลอร์ได้เรียนรู้ เช่น การบ่งบอกว่าแพ็กเก็ตได้มาถึงแล้ว การเปลี่ยนสถานะของสวิตช์ เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2.3 โอเฟนโฟลว์แชนเนล

โอเพนโฟลว์แชนเนล (OpenFlow Channel) เป็นอินเตอร์เฟสที่เชื่อมต่อระหว่างโอเพนโฟลว์สวิตช์แต่ละเครื่องภายในเครือข่าย และ ทำการเชื่อมต่อกับเอสดีเอ็นคอนโทรลเลอร์อีกด้วย ช่องทางดังกล่าวสามารถมีมากกว่าหนึ่งช่องทางก็ได้ ซึ่งจะเรียกว่ามัลติพาร์ท (Multipath) โดยช่องทางนี้จะทำหน้าที่เป็นเส้นทางผ่านในการส่งข้อความโอเพนโฟลว์ (OpenFlow Message)

2.1.2.4 สถานะพอร์ทของโอเพนโฟลว์สวิตช์

เป็นข้อความที่โอเพนโฟลว์สวิตช์ใช้ตอบกลับไปยังคอนโทรลเลอร์ เพื่อรายงานข้อมูลทางสถิติที่ใช้ในการสื่อสารภายในเครือข่าย เราได้นำข้อความนี้มาประยุกต์ใช้ในโครงงานของเรา และ นำไปใช้ในโมเดลของการเรียนรู้เชิงลึก ซึ่งเราจะอธิบายในบทต่อไป

เลขนัย (Index)	ฟิลด์ (Field)	คำอธิบาย (Description)
1	port_no	หมายเลขพอร์ตของสวิตช์
2	rx_packets	จำนวนแพ็กเก็ตที่ได้รับ
3	tx_packets	จำนวนแพ็กเก็ตที่ส่ง
4	rx_bytes	จำนวนไบต์ที่ได้รับ
5	tx_bytes	จำนวนไบต์ที่ส่ง
6	rx_dropped	จำนวนแพ็กเก็ตที่ถูกดรอปโดยฝั่งรับ (rx)
7	tx_dropped	จำนวนแพ็กเก็ตที่ถูกดรอปโดยฝั่งส่ง (tx)
8	rx_error	จำนวน error ที่ได้รับ
9	tx_error	จำนวน error ที่ส่ง

ตารางที่ 2.1 ฟิล์ดของ OFPPortStatusReply

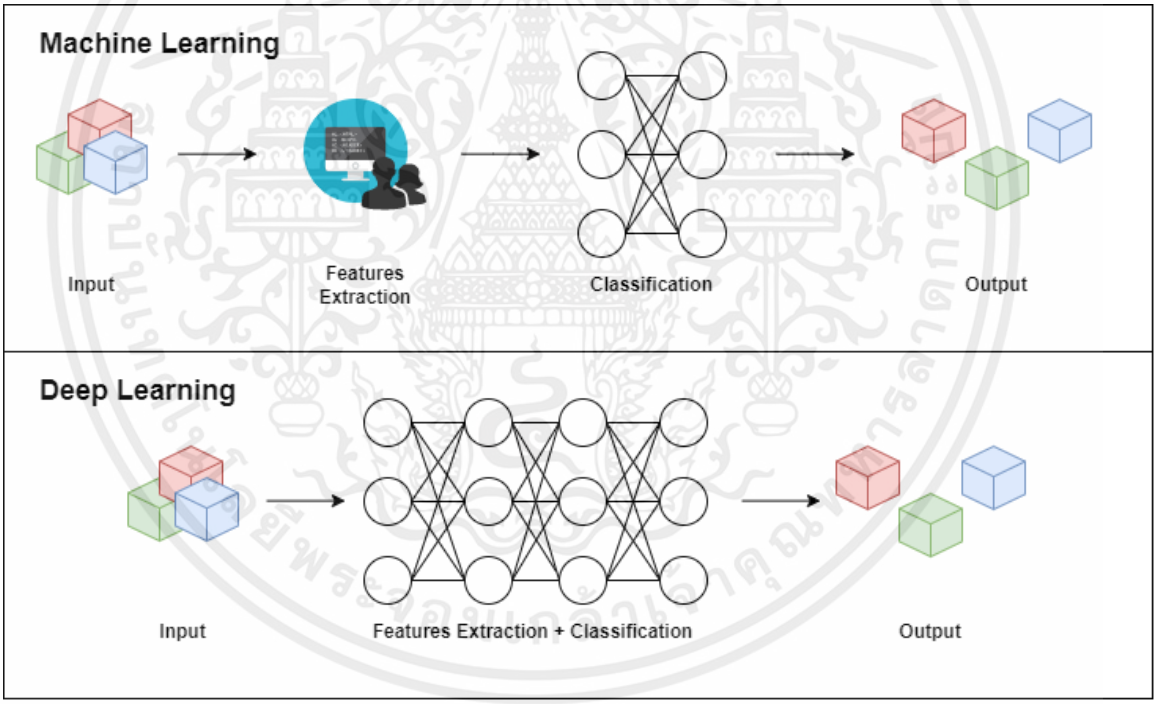
2.1.3 การเรียนรู้เชิงลึกและโครงข่ายประสาทเทียม

การเรียนรู้เชิงลึก (Deep Learning) เป็นซอฟต์แวร์คอมพิวเตอร์ที่มีการลอกเลียนแบบให้คล้ายกับโครงข่ายเซลล์ประสาท (Network of Neuron) ในสมองของมนุษย์ รวมไปถึงเป็นส่วนหนึ่งของการเรียนรู้ด้วยเครื่อง (Machine Learning) และ ปัญญาประดิษฐ์ (Artificial Intelligence) อัลกอริทึมของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเรียนรู้เชิงลึกจะแตกต่างออกไปจากการเรียนรู้ด้วยเครื่องตรงที่ การเรียนรู้เชิงลึกจะสร้างอยู่ในรูปแบบของโครงข่ายประสาทเทียม (Artificial Neural Network หรือ ANN) หลายชั้นเชื่อมต่อกัน ซึ่งจะถูกแบ่งออกเป็น 3 ชั้น ได้แก่ ชั้นอินพุต (Input layer), ชั้นซ่อนตัว (Hidden layer) และ ชั้นเอาท์พุต (Output layer) เมื่ออัลกอริทึมประกอบไปด้วยชั้นประมวลผลสองชั้นขึ้นไป ก็จะถือว่าอัลกอริทึมนี้เป็นโครงข่ายประสาทเทียมเชิงลึก (Deep-Neural Network)

ในทางกลับกัน การเรียนรู้ด้วยเครื่องจะมีโครงสร้างที่ซับซ้อนน้อยกว่าการเรียนรู้เชิงลึก และการเรียนรู้ด้วยเครื่องจำเป็นที่จะต้องจำแนกฟีเจอร์ของข้อมูล (Features Extraction) ก่อนที่จะนำไปแยกหมวดหมู่ (Classification) ส่วนการเรียนรู้เชิงลึกจะนำการจำแนกฟีเจอร์ของข้อมูลพร้อมกับการแยกหมวดหมู่ไปในตัวระหว่างเรียนรู้อัลกอริทึมดังรูปที่ 2.4 ด้านล่าง



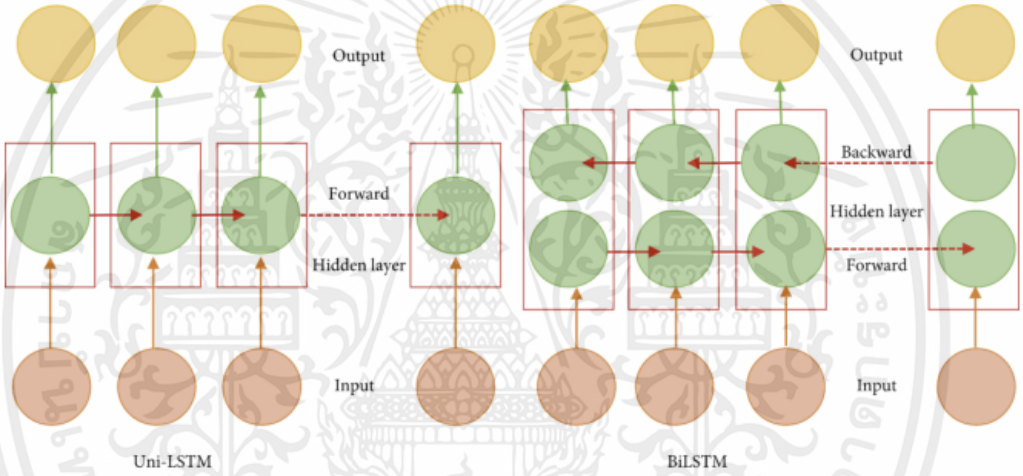
รูปที่ 2.4 การเปรียบเทียบระหว่างการเรียนรู้ด้วยเครื่องและการเรียนรู้เชิงลึก

จากรูปที่ 2.4 จะสังเกตุได้ว่าการทำงานของอัลกอริทึมการเรียนรู้เชิงลึกจะมีความซับซ้อนมากกว่าการเรียนรู้ด้วยเครื่องและมีขั้นตอนที่สั้นกว่าการเรียนรู้ของเครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.4 Bidirectional Long-Short Term Memory และ Gated Recurrent Units

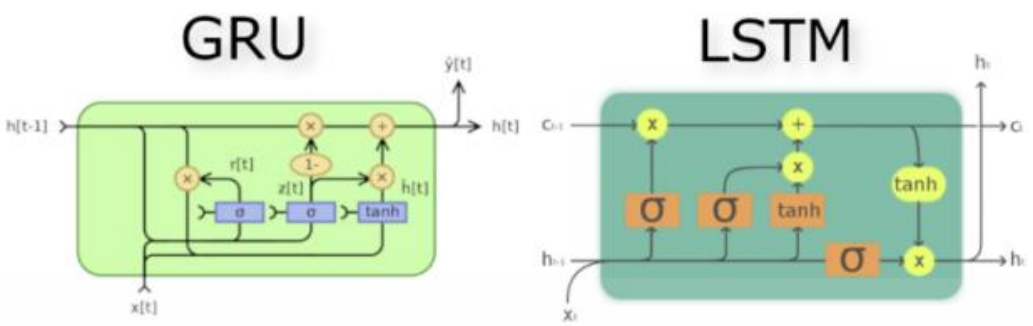
Bidirectional Long-Short Term Memory เป็นโครงข่ายประสาทเทียมแบบวนกลับ (Recurrent Neural Network หรือ RNN) ที่พัฒนาต่อมาจาก Uni Long Short Term Memory ซึ่งหลักการทำงานของ Bidirectional LSTM แตกต่างจาก Uni Long Short Term Memory แบบปกติ ซึ่งจะมีอินพุตอยู่สองทาง ได้แก่ Forward layer และ Backward layer ซึ่ง Uni Long Short Term Memory จะมีอินพุตแค่ Forward layer ดังรูปที่ 2.5 ด้านล่าง ซึ่งสองโมเดลนี้เหมาะสมกับการใช้กับข้อมูลประเภทอนุกรมเวลา (Time Series) และการทำนายสิ่งที่จะเกิดขึ้นในอนาคต



รูปที่ 2.5 หลักการทำงานระหว่างโมเดล Uni-LSTM และ Bi-LSTM

Gated Recurrent Units เป็นโมเดลที่ทำงานที่เกี่ยวกับข้อมูลประเภทอนุกรมเวลาเช่นเดียวกับ LSTM และ Bidirectional Long-Short Term Memory แต่องค์ประกอบของโมเดลนั้นจะมีขนาดที่เล็กที่สุดจากสามโมเดล LSTM จะประกอบไปด้วย 3 เกต ในขณะที่ GRU ประกอบไปด้วย 2 เกต ซึ่งโมเดล GRU ถูกออกแบบขึ้นมาเพื่อแก้ปัญหาประสิทธิภาพในการเทรนข้อมูลที่มีลักษณะ Sequence ที่ยาวและมีการใช้ Parameter ที่น้อยกว่า LSTM รวมไปถึงความเหมาะสมต่อการเทรนที่มีการใช้ข้อมูลที่ไม่เยอะเกินไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

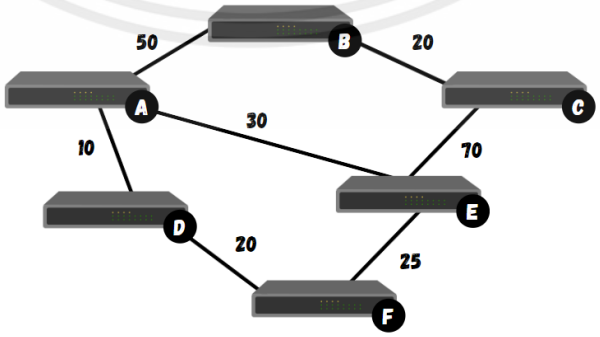


รูปที่ 2.6 องค์ประกอบของโครงข่ายประสาทเทียมระหว่าง GRU และ LSTM

จากรูปที่ 2.6 ด้านบน จะสังเกตุได้ว่าโครงข่ายประสาทเทียมแบบ GRU มีหลักการทำงานที่ซับซ้อนน้อยกว่าโครงข่ายประสาทเทียมแบบ LSTM โดยมีการใช้ประตู (Gate) ที่น้อยกว่าอยู่หนึ่งประตู โครงงานของเราได้ทำการเก็บข้อมูลที่ไม่ได้มากนัก จึงเป็นสาเหตุที่ทำให้นำโมเดล GRU มาทดสอบกับระบบเอสดีเอ็นในที่สุด ซึ่งผลลัพธ์ของการทดสอบจะอธิบายภายในบทที่ 4 ภายหลัง

2.1.5 อัลกอริทึมของไดก์สตรา

อัลกอริทึมของไดก์สตรา (Dijkstra's algorithm) เป็นอัลกอริทึมที่มีขั้นตอนวิธีคิดเพื่อที่จะหาเส้นทางที่สั้นที่สุดระหว่างโหนดต้นทางและโหนดปลายทาง โดยอัลกอริทึมนี้ถูกคิดค้นขึ้นโดยนักวิทยาศาสตร์ แอ็ดส์เคอร์ ไดก์สตรา (Edsger W. Dijkstra) ในปี พ.ศ. 2499 ในหลายปีต่อมา การประยุกต์ใช้อัลกอริทึมนี้ก็ถูกนำไปใช้ในหลากหลายวิธี เช่น โพรโทคอลการกำหนดเส้นทางภายในเครือข่าย หรือ Open Shortest Path First (OSPF)



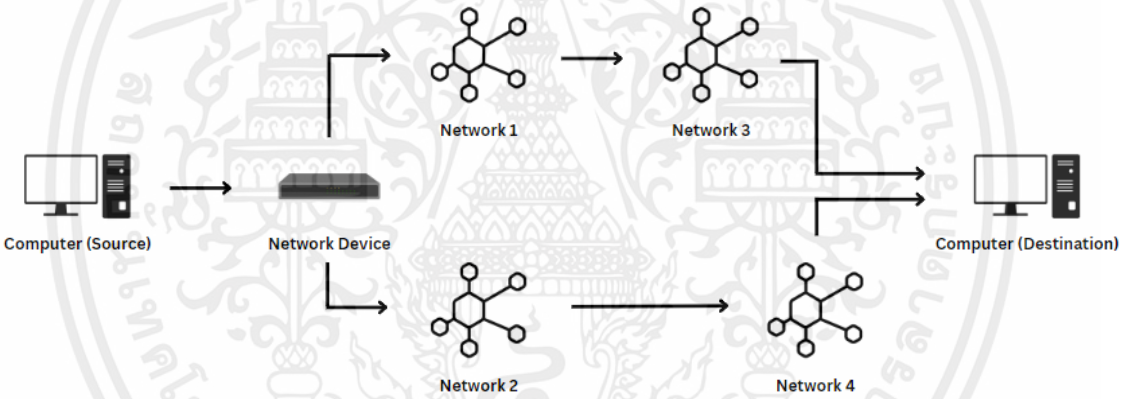
รูปที่ 2.7 การประยุกต์ใช้ออลกอริทึมไดก์สตรากับระบบเอสดีเอ็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.7 เราจะเปรียบเทียบการนำออลกอริทึมไดก์สตรามาใช้กับระบบการจัดการเส้นทางของโครงงานเรา โดยเราจะให้สวิตช์แต่ละตัวเปรียบเป็นโหนด (Node) ลิงก์ที่เชื่อมกันเปรียบเป็นเส้นทาง (Path) และค่าการใช้ประโยชน์แบนด์วิดท์เปรียบเป็นระยะทาง (Cost)

2.1.6 การกำหนดเส้นทาง

การกำหนดเส้นทาง (Routing) เป็นโพรโทคอลที่ใช้ในการแลกเปลี่ยนเส้นทางภายในเครือข่ายระหว่างอุปกรณ์กับอุปกรณ์ ซึ่งทำงานอยู่ในระดับชั้นที่ 3 ของโมเดลโอเอสไอ (OSI Model) หรือชั้นการสื่อสารเครือข่าย (Network layer) เมื่อแลกเปลี่ยนข้อมูลกันเสร็จสิ้นแล้ว จะสร้างตารางเส้นทางของแต่ละอุปกรณ์ (Routing table) ขึ้นมา เพื่อให้อุปกรณ์แต่ละอันสามารถส่งแพ็กเก็ตข้อมูลไปยังเครื่องปลายทางได้อย่างถูกต้องดังรูปที่ 2.8 ด้านล่าง



รูปที่ 2.8 หลักการส่งข้อมูลภายในเครือข่าย

โพรโทคอลการกำหนดเส้นทางถูกแบ่งออกเป็น 2 ประเภท ได้แก่ การกำหนดเส้นทางแบบคงที่ (Static Routing Protocol) และ การกำหนดเส้นทางแบบไดนามิก (Dynamic Routing Protocol)

2.1.6.1 การกำหนดเส้นทางแบบคงที่

การกำหนดเส้นทางแบบคงที่ (Static Routing Protocol) เป็นการกำหนดค่าและเลือกเส้นทางของเครือข่ายด้วยตนเอง ว่าต้องการให้แพ็กเก็ตข้อมูลเดินทางไปที่ใด โหนดใด อินเตอร์เฟสใด จนกว่าจะถึงปลายทาง ซึ่งการใช้การกำหนดเส้นทางแบบคงที่จะเป็นประโยชน์ใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สถานการณ์ที่ต้องการออกแบบเครือข่ายด้วยตนเอง เมื่อเกิดปัญหา จะสามารถแก้ไขเส้นทางได้ง่าย ภายในเครือข่ายขนาดเล็ก แต่ถ้าหากเป็นเครือข่ายที่มีขนาดใหญ่ จะต้องแก้เส้นทางหลายจุดเมื่อเกิดปัญหาขึ้น

จากรูปที่ 2.8 เราสามารถกำหนดเส้นทางแบบคงที่ได้ โดยกำหนดให้เครื่องคอมพิวเตอร์ที่เป็นต้นทาง (Source) ส่งข้อมูลไปยังเครื่องปลายทาง (Destination) ซึ่งผ่านแค่ Network 1 และ Network 3 ดังรูปด้านบน ซึ่งเป็นการกำหนดเส้นทางในการส่งข้อมูลอย่างชัดเจนและจะแตกต่างจากการกำหนดเส้นทางแบบไดนามิกอย่างสิ้นเชิง โดยเราจะอธิบายต่อไปจากนี้

2.1.6.2 การกำหนดเส้นทางแบบไดนามิก

การกำหนดเส้นทางแบบไดนามิก (Dynamic Routing Protocol) เป็นโพรโทคอลที่จะสร้างตาราเส้นทางและปรับปรุงตารางการกำหนดเส้นทางที่เปลี่ยนไปตามสภาพเครือข่ายจริง โดยจะติดต่อกับเราเตอร์เพื่อนบ้านเพื่อสร้างเส้นทางการส่งข้อมูลที่ดีที่สุดด้วยตนเอง ซึ่งในโพรโทคอลประเภทนี้ มนุษย์ไม่จำเป็นที่จะต้องออกคำสั่งให้โดยตรง โพรโทคอลกำหนดเส้นทางแบบไดนามิกจะเหมาะสมกับเครือข่ายที่มีขนาดใหญ่ เนื่องจากมีการเรียนรู้ตารางเส้นทางจำนวนมาก โพรโทคอลกำหนดเส้นทางแบบไดนามิกสามารถแบ่งออกเป็น 2 ประเภท ได้แก่ Interior Gateway Protocol (IGP) และ Exterior Gateway Protocol (EGP) โดย IGP เป็นโพรโทคอลการกำหนดเส้นทางที่มีการเรียนรู้เส้นทางอยู่ภายใต้ Autonomous System (AS) เดียวกัน หรือ ข้อกำหนด (Policy) เดียวกัน แต่ในทางกลับกัน EGP เป็นโพรโทคอลการกำหนดเส้นทางที่มีการเรียนรู้เส้นทางระหว่าง AS ที่มีข้อกำหนดที่แตกต่างกันออกไป

จากรูปที่ 2.8 หากเรามีการกำหนดเส้นทางแบบไดนามิก อุปกรณ์เครือข่ายที่มีการทำงานเกี่ยวกับการกำหนดเส้นทาง เช่น เราท์เตอร์ (Router) อุปกรณ์นั้นก็จะมีการเรียนรู้เครือข่ายด้วยตนเอง และ ทำการตัดสินใจเองว่าจะให้ส่งข้อมูลไปยังที่เครือข่ายใด โดยผ่านเงื่อนไขต่างๆ เช่น OSPF จะส่งโดยใช้เงื่อนไขระยะทางของเส้นทาง (Cost) ถ้าเส้นทางไหนมีค่าระยะทางน้อยที่สุดในตารางเส้นทาง (Routing Table) ก็จะสั่งการให้มีการส่งข้อมูลไปยังเครือข่ายนั้น

2.1.7 การสูญหายของแพ็กเก็ต

การสูญหายของแพ็กเก็ต (Packet loss) เกิดขึ้นเมื่อแพ็กเก็ตของข้อมูลที่ไม่สามารถเดินทางไปยังเป้าหมายปลายทางได้ สาเหตุที่ทำให้เกิดเหตุการณ์เช่นนี้ เช่น มีการส่งข้อมูลจากหลายแหล่งภายในเวลาเดียวกันทำให้เกิดความคับคั่งของเครือข่าย (Network Congestion) ปัญหาอุปกรณ์ฮาร์ดแวร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

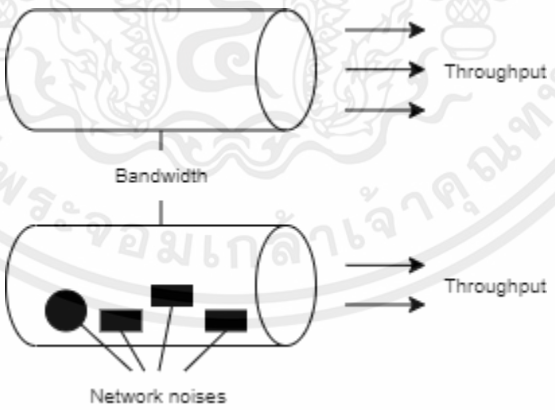
ที่มีความเสียหาย เป็นต้น ตัวอย่างของปัญหาเหล่านี้ที่เกิดขึ้นในชีวิตประจำวัน เช่น การเกิดอาการแลค (Lag) ขณะที่เล่นเกมส์ หรือ รับชมโทรทัศน์ขณะกระตุก

2.1.8 แบนด์วิดท์

แบนด์วิดท์ (Bandwidth) หมายถึง ความกว้างของคลื่นความถี่ หรือ ความกว้างของช่องทางในการรับส่งข้อมูล ในเครือข่ายยิ่งมีขนาดแบนด์วิดท์ที่ค่อนข้างสูง ก็จะได้เปรียบเรื่องการส่งข้อมูล เนื่องจากยิ่งแบนด์วิดท์สูง การรับข้อมูลเข้า-ออก ก็จะสามารถส่งข้อมูลได้มากขึ้น ซึ่งทำให้มีประสิทธิภาพในการส่งข้อมูลดี หรือกล่าวได้ว่าเป็นสัดส่วนโดยตรงของจำนวนข้อมูลทั้งหมดที่สามารถส่งผ่านหรือรับต่อหน่วยเวลา เราสามารถประเมินการใช้งานแบนด์วิดท์ของลิงค์นั้นๆ โดยคิดจากการใช้ค่าประโยชน์แบนด์วิดท์ (Bandwidth Utilization) ซึ่งเป็นค่าเปอร์เซนต์ที่บ่งบอกถึงปริมาณแบนด์วิดท์ที่ถูกใช้งานเมื่อเทียบกับปริมาณแบนด์วิดท์สูงสุดที่มีต่อลิงค์ภายในเครือข่าย

2.1.9 ทรูพุต

ทรูพุต (Throughput) หมายถึง ปริมาณข้อมูลที่สามารถส่งและรับได้ในช่วงเวลาหนึ่ง ทรูพุตมีหน่วยเป็นบิตต่อวินาที (bps หรือ bit per second) เมื่อใดก็ตามที่ทรูพุตมีขนาดน้อยเกินกว่าเหตุแสดงว่าในเครือข่ายมีการสูญหายของแพ็กเก็ตเกิดขึ้น (Packet loss) เรามักพบในชีวิตประจำวัน เช่น ความเร็วอินเตอร์เน็ตที่ใช้ในการดาวน์โหลด (Download speed) หรือ การอัพโหลด (Upload speed)



รูปที่ 2.9 การเปรียบเทียบระหว่างแบนด์วิดท์ และ ทรูพุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.10 ดีเลย์

ดีเลย์ (Delay) หมายถึง เวลาหน่วงที่ใช้ในการส่งแพ็กเก็ตจากต้นทาง (Source) ไปยังปลายทาง (Destination) เวลาหน่วงนี้สามารถเกิดจากสาเหตุดังต่อไปนี้

1) Transmission delay : เป็นเวลาหน่วงของสื่อตัวกลางนำสัญญาณ
2) Queueing delay : เป็นเวลาที่เกิดจากการรอรับส่งแพ็กเก็ตจากความคับคั่งภายในเครือข่าย
3) Processing delay : เป็นเวลาที่เราท์เตอร์ใช้เวลาในการประมวลผลเฮดเดอร์ (Header field)
4) Propagation delay : ระยะทางที่ข้อมูลสามารถเดินทางไปได้บนตัวกลางภายในระยะ 1 วินาที

2.2 งานวิจัยที่เกี่ยวข้อง

จากการศึกษาข้อมูลที่เกี่ยวกับเครือข่ายที่กำหนดโดยซอฟต์แวร์ การเรียนรู้เชิงลึก การเลือกใช้คอนโทรลเลอร์ การจัดเส้นทาง และ โมเดลการเรียนรู้เชิงลึกแบบวนกลับชนิด Bidirectional LSTM เราได้ศึกษาค้นคว้าเอกสารงานวิจัยที่เกี่ยวข้อง (Related Researches) ดังต่อไปนี้

Md. Tariqul Islam, Nazrul Islam และ Md. AI Refat ได้ศึกษางานวิจัยที่เกี่ยวกับการเลือกใช้คอนโทรลเลอร์เพื่อที่จะนำข้อมูลภายในเครือข่ายที่กำหนดโดยซอฟต์แวร์มาทำการประเมินประสิทธิภาพของเครือข่าย (Network Performance Evaluation) เนื่องจากเทคโนโลยีเอสดีเอ็นสามารถใช้คอนโทรลเลอร์ได้หลากหลายรูปแบบ คอนโทรลเลอร์แต่ละอันจะมีประสิทธิภาพที่ต่างกัน ใช้ภาษาในการควบคุมที่แตกต่างกัน เช่น ไพทอน (Python) ซีพลัสพลัส (C++) และ จาวา (Java) ซึ่งผู้วิจัยได้ค้นพบว่า คอนโทรลเลอร์ที่ใช้ภาษาซีพลัสพลัส มีปัญหาในเรื่องของการจัดการหน่วยความจำ (Memory Management) โอเพนเดย์ไลท์คอนโทรลเลอร์ (OpenDay light หรือ ODL) มีปัญหาในเรื่องของการรั่วไหลของหน่วยความจำ (Memory leakage) พอกซ์คอนโทรลเลอร์ (POX) มีปัญหาในเรื่องของการประมวลผลที่ช้ากว่าภาษาจาวาและภาษาซีพลัสพลัส ซึ่งคอนโทรลเลอร์ทุกรูปแบบถูกทดสอบบนอิมูเลเตอร์มินิเน็ต (Mininet Emulator) โดยผู้วิจัยพบว่า ริวคอนโทรลเลอร์ (RYU) มีประสิทธิภาพที่เหมาะสมที่สุดในการดึงข้อมูลจากเครือข่ายเพื่อมาทำการประเมินประสิทธิภาพของเครือข่าย นอกจากนี้ ริวคอนโทรลเลอร์ยังใช้ภาษาไพทอนซึ่งสามารถประยุกต์ใช้กับไลบรารี่ (Library) ได้หลากหลาย และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นิยมใช้บนระบบปฏิบัติการลินุกซ์ (Linux) อีกด้วย ซึ่งส่วนใหญ่อุปกรณ์เครือข่ายทั่วไปจะทำงานร่วมกับระบบปฏิบัติการลินุกซ์

Sima Siami-Namini, Neda Tavakoli and Akbar Siami Namin (2019) [2] งานวิจัยชิ้นนี้ ได้ทำการทดสอบประสิทธิภาพและความแม่นยำของ RNN (Recurrent Neural Network) 2 ประเภท คือ LSTM (Long Short Term Memory) และ Bi-LSTM (Bidirectional LSTM) เนื่องจากงานวิจัยนี้เล็งเห็นถึงความสามารถของ Bi-LSTM ที่มีการรับข้อมูลที่ใช้ในการป้อนจากทั้งในอดีตและอนาคต ในขณะที่ LSTM มีการรับจากแค่ในอดีต โดยจะทดสอบทั้งสองโมเดลนี้ด้วยชุดข้อมูลอนุกรมเวลา (Time Series) ของจุดปิดของตลาดหุ้นรายวัน ผลที่ได้ คือ Bi-LSTM สามารถทำนายจุดปิดของตลาดหุ้นแม่นยำกว่า LSTM อยู่ถึง 37.78% โดยเฉลี่ย ทำให้สามารถสรุปได้ถ้าจะทำการทำนายข้อมูลที่กำลังจะเกิดขึ้นควรจะใช้ Bi-LSTM แทน LSTM

Orman M. A. A. Alssaheli, Z. Zainal Abidin, N.A. Zakaria and Z. Abal Abas (2021) [3] งานวิจัยนี้เป็นการเสนอว่าการตรวจจับข้อมูลภายในเครือข่ายเพื่อการปรับปรุงรักษา รวมไปถึงการพัฒนาในปัจจุบันนั้นเข้าถึงได้ยาก ไม่ว่าจะเป็น การพัฒนาระบบเครือข่ายในคลาวด์ (Cloud), IoT (Internet of Things), บล็อกเชน เนื่อง จากการทำระบบเช่นนี้ต้องมีอุปกรณ์ที่มีความต้องการในการติดตั้งสูง (Hardware-based-network-controller) เพื่อพัฒนาระบบ แต่ในทางกลับกัน เอสดีเอ็น หรือ เครือข่ายจำลองที่กำหนดโดยซอฟต์แวร์ (SDN) เข้าถึงง่ายกว่าการติดตั้งอุปกรณ์ทั่วไป ยกตัวอย่างเช่น การขยายเครือข่าย การบำรุงรักษา การยกระดับความปลอดภัย ในที่นี้ เอสดีเอ็นสามารถทำการตรวจจับข้อมูลภายในเครือข่ายได้ด้วย (Traffic Monitoring) ในงานวิจัยนี้จะเป็นการอธิบายว่า SDN ต้องมีการติดตั้งซอฟต์แวร์อย่างไรบ้าง เพื่อจำลองเครือข่ายเสมือน โดยเครือข่ายนั้นจะถูกสร้างขึ้นได้ด้วย Mininet ซึ่งใน Mininet นั้นมีคอนโทรลเลอร์ หรือ ตัวควบคุมระบบเครือข่ายทั้งหมด (SDN Ryu Controller) รวมถึงการใช้โพรโทคอลโอเพนโฟลว์ในการสื่อสารระหว่างอุปกรณ์ภายในเครือข่าย และตัวแปรสำคัญที่จะใช้ในการตรวจจับข้อมูลเช่น ทรูพุต, จิตเตอร์, แบนด์วิดท์ และระยะเวลาที่ใช้ในการเดินทางจากโหนดไปสู่โหนด (Round-Trip Time between scenarios) ซึ่งสรุปได้ว่า หากผู้ใดสนใจที่จะศึกษาเรื่องการติดตั้งอุปกรณ์เครือข่าย หรือ การศึกษาด้านเครือข่าย เอสดีเอ็นเป็นอีกทางเลือกหนึ่งที่จะทำให้ผู้ใช้งานเข้าถึงเครือข่ายได้ง่ายขึ้นกว่าเดิม

Sowmya Sanagavarapu และ Sashank Sridhar ได้ศึกษาเกี่ยวกับการบริหารเครือข่ายเพื่อทำให้เครือข่ายมีความฉลาดมากขึ้น (Network Intelligence) โดยนำโครงข่ายประสาทเทียม (Recurrent Neural Network) มาประยุกต์ใช้กับเอสดีเอ็นคอนโทรลเลอร์เพื่อทำการทำนายการเดินทางของข้อมูล (Traffic

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

flows) และทำการอัพเดทตารางเส้นทาง (Flow table) ซึ่งโมเดลที่ผู้วิจัยได้นำมาใช้ คือ LSTM (Long Short Term Memory) ซึ่งเป็นโมเดลที่ทำหน้าที่ในรูปแบบลำดับต่อลำดับ (Sequence to Sequence) หลักการทำงานของงานวิจัยชิ้นนี้จะช่วยให้คอนโทรลเลอร์มีความฉลาดมากขึ้นจากการคาดเดาการส่งข้อมูลภายในอนาคต และ สามารถลดระดับความคับคั่งภายในเครือข่ายได้ ผู้วิจัยได้นำ Mean Absolute Percentage Error (MAPE) และ Mean Square Error (MSE) เข้ามาเป็นตัวประเมินในการวัดประสิทธิภาพในการทำนาย โดยการทดลองทั้งหมด ผู้วิจัยได้ทดลองผ่านอิมูเลเตอร์มินิเน็ต (Mininet) และได้ใช้ริวคอนโทรลเลอร์ (RYU Controller) ซึ่งในการคาดการณ์ผู้วิจัยได้นำพารามิเตอร์ (Parameters) เช่น แบนด์วิดท์ (Bandwidth) เพื่อนำมาประเมินคุณภาพในการส่งแพ็กเก็ตข้อมูลภายในเครือข่าย และทำการหลีกเลี่ยงเส้นทางที่มีการใช้แบนด์วิดท์ที่สูง

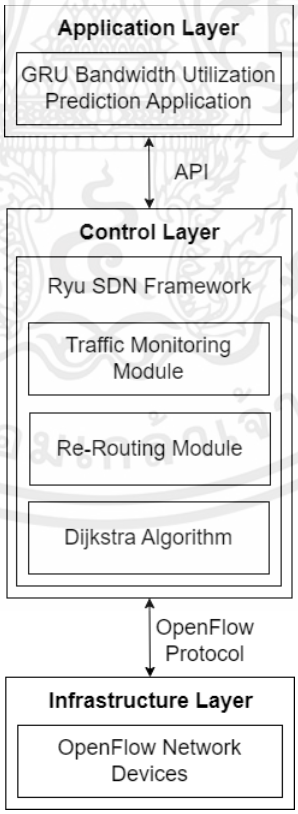
ในโครงงานนี้ เรามุ่งเน้นในการใช้การเรียนรู้เชิงลึกประเภทโครงข่ายประสาทเทียมแบบวนกลับชนิด Bidirectional Long Short Term Memory (Bi-LSTM) และ Gated Recurrent Units (GRU) มาประยุกต์ใช้กับเครือข่ายที่กำหนดโดยซอฟต์แวร์เพื่อคาดการณ์การใช้งานแบนด์วิดท์ในอนาคตของแต่ละลิงค์ภายในเครือข่าย โดยเราได้นำโมเดลการเรียนรู้เชิงลึก Bidirectional LSTM มาเปรียบเทียบกับโมเดล LSTM และ โมเดล GRU เพื่อแสดงการเปรียบเทียบและความแม่นยำในการคาดการณ์ที่แตกต่างกัน เมื่อมีการใช้งานแบนด์วิดท์ที่มีค่าเกินเกณฑ์ที่เราตั้งไว้ (Threshold) ก็จะบังคับให้มีการทำจัดเปลี่ยนเส้นทางใหม่ (Re-Routing) เพื่อลดระดับความคับคั่งภายในเครือข่าย และ หลีกเลี่ยงการเกิดการสูญหายของแพ็กเก็ต โดยประสิทธิภาพของเครือข่าย เราจะทำการประเมินจาก ดีเลย์ (Delay), ทรูพุต (Throughput) และ การสูญหายของแพ็กเก็ต (Packet loss)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

วิธีการนำเสนอ

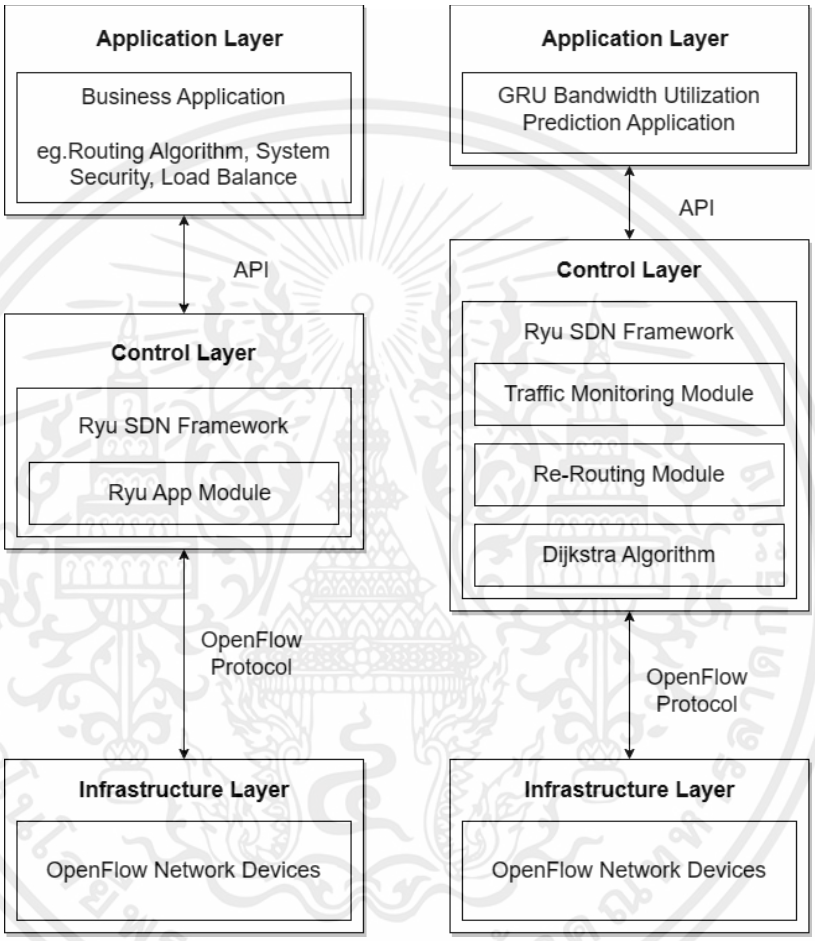
ในบทนี้ เราได้ทำการเสนอโครงสร้าง การออกแบบ องค์ประกอบ และขั้นตอนการทำงานทั้งหมดของโครงงาน ซึ่งประกอบไปด้วย 3 ส่วน ได้แก่ ส่วนของชั้นโครงสร้างพื้นฐาน (Infrastructure Layer) ที่ประกอบไปด้วยโอเพนโฟลว์สวิตช์ (OpenFlow Switches) ที่ถูกจำลองขึ้น รวมไปถึงการออกแบบของเครือข่าย (Topology Design) ส่วนที่สองคือชั้นควบคุม (Control Layer) ที่ประกอบไปด้วยริวเอสดีเอ็นคอนโทรลเลอร์ (Ryu SDN Controller) และ ส่วนสุดท้ายคือชั้นแอปพลิเคชัน (Application Layer) ที่จะประกอบไปด้วยแอปพลิเคชันสำหรับการคาดการณ์ค่าการใช้ประโยชน์แบนด์วิดท์ โดยใช้โมเดลการเรียนรู้เชิงลึก Bi-LSTM ดังแสดงในรูปที่ 3.1



รูปที่ 3.1 ภาพรวมของระบบเอสดีเอ็นของโครงงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราได้นำโครงสร้างโดยทั่วไปของเอสดีเอ็นมาดัดแปลงดังรูปที่ 3.1 และเราได้แสดงการเปรียบเทียบระหว่างโครงสร้างเอสดีเอ็นแบบมาตรฐาน และ โครงส้รางของโครงงานที่เราได้ประยุกต์มาจากโครงสร้างมาตรฐานดังในรูปที่ 3.2



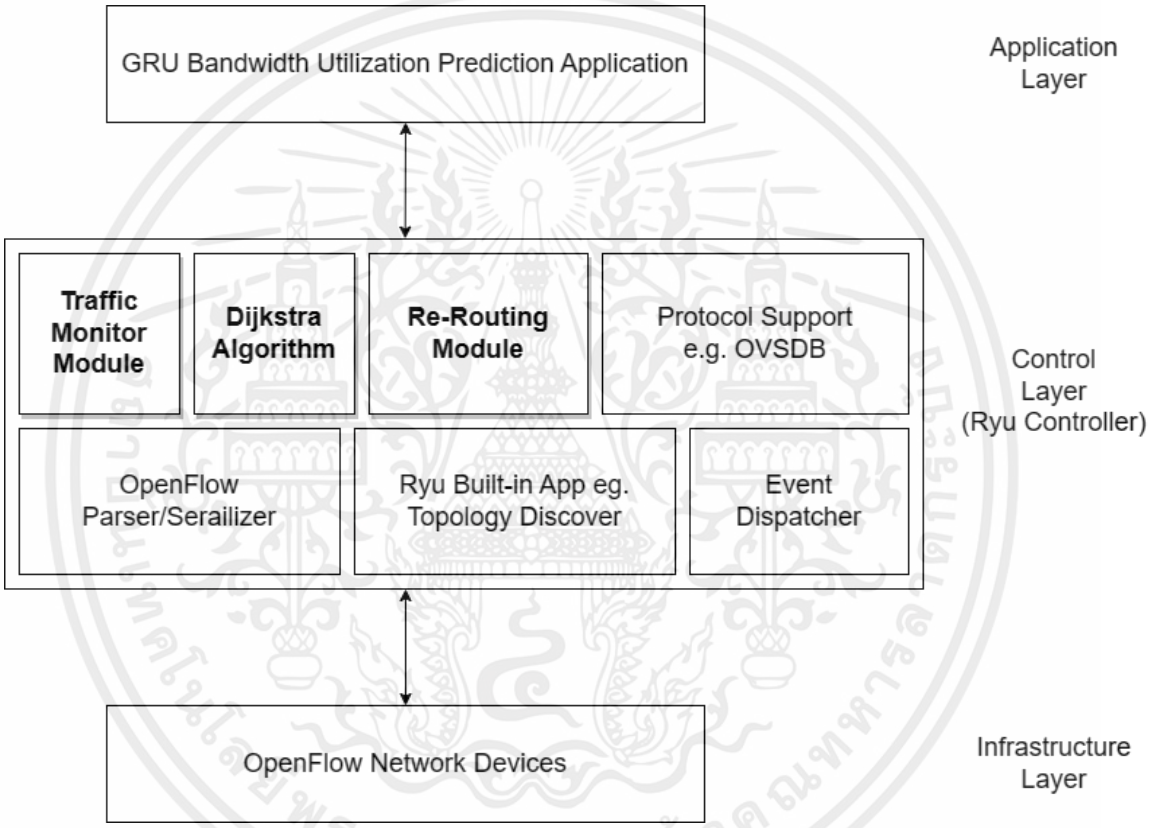
รูปที่ 3.2 การเปรียบเทียบโครงสร้างระหว่าง SDN มาตรฐาน และโครงสร้าง SDN ของโครงงาน

เราได้นำเสนอระบบเอสดีเอ็นของโครงงานผ่านเครื่องเสมือน (Virtual Machine) โดยทำงานผ่านระบบปฏิบัติการอุบุนตู (Ubuntu) และทำการจำลองโอเพนโฟลว์สวิตช์ขึ้นมาด้วยซอฟต์แวร์มินิเน็ต (Mininet) ในชั้นโครงสร้างพื้นฐาน (Infrastructure Layer) เราได้เลือกใช้ริวเอสดีเอ็นคอนโทรลเลอร์ในชั้นการควบคุม (Control Layer) รวมไปถึงการสร้างแอปพลิเคชันสำหรับการคาดการณ์ค่าการใช้ประโยชน์แบนด์วิดท์ในชั้นแอปพลิเคชัน (Application Layer)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1 องค์ประกอบของโครงงาน

ระบบของเอสดีเอ็นภายในโครงงานนี้ประกอบไปด้วย 3 ส่วน ได้แก่ โอเพนโฟลว์สวิตช์และโฮสท์ในชั้นโครงสร้างพื้นฐาน ริวเอสดีเอ็นคอนโทรลเลอร์ และแอปพลิเคชันคาดการณ์ค่าการใช้ประโยชน์แบนด์วิดท์ที่จะเกิดขึ้นในอนาคต โดยใช้โมเดลการเรียนรู้เชิงลึกประเภท Gated Recurrent Units (GRU) ดังแสดงในรูปที่ 3.3

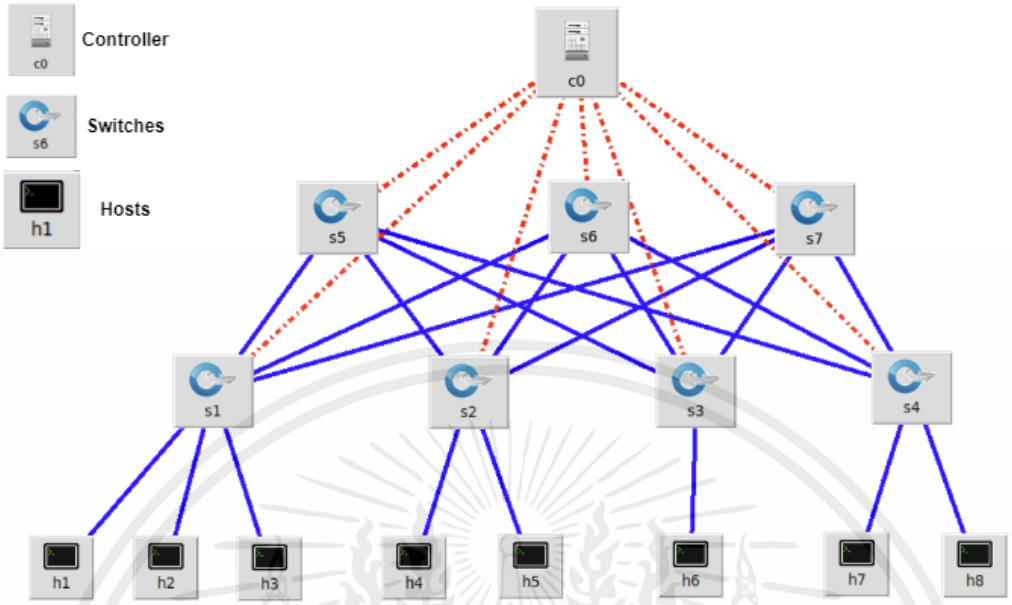


รูปที่ 3.3 องค์ประกอบของเอสดีเอ็นอย่างละเอียด

3.1.1 ชั้นโครงสร้างพื้นฐาน

เราได้ทำการออกแบบเครือข่าย ซึ่งประกอบไปด้วยโอเพนโฟลว์สวิตช์ (switch) 7 เครื่องและโฮสท์ (host) 8 เครื่อง เชื่อมต่อกันดังแสดงในรูป 3.4 อุปกรณ์เครือข่ายทั้งหมดถูกจำลองขึ้นด้วยซอฟต์แวร์มินิเน็ต (Mininet) ซึ่งทำงานผ่านระบบปฏิบัติการอุบุนตู (Ubuntu)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



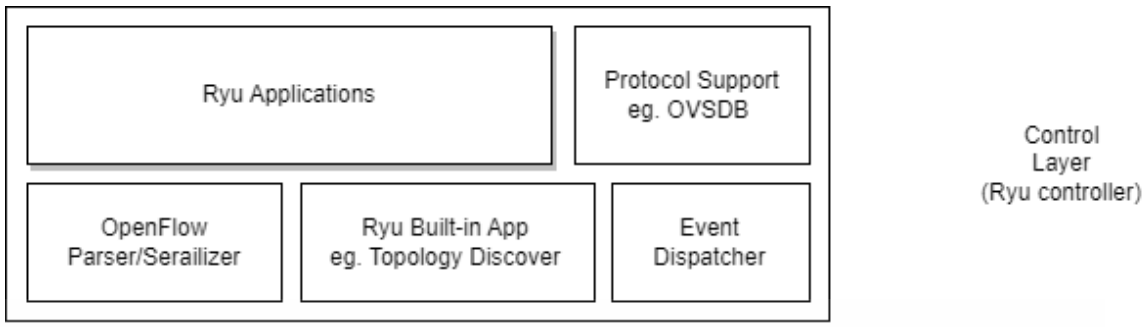
รูปที่ 3.4 โครงสร้างเครือข่ายที่จะลองขึ้นผ่าน MiniEdit

ชั้นโครงสร้างพื้นฐานจะเชื่อมต่อกับริวคอนโทรลเลอร์ โดยจะสื่อสารด้วยโพรโทคอลโอเพนโฟลว์ (OpenFlow Protocol) จากรูปที่ 3.4 มินิเน็ตสามารถแสดงรูปเครือข่ายได้โดยเรียกโปรแกรมผ่านซอฟต์แวร์มินิอิดิต (MiniEdit) ในโครงงานนี้ เราได้กำหนดให้มีการส่งข้อมูลระหว่างโฮสท์ต้นทาง 3 เครื่อง ได้แก่ h1, h2 และ h3 โฮสท์ปลายทาง 5 เครื่อง ได้แก่ h4, h5, h6, h7 และ h8

3.1.2 ริวเอสดีเอ็นคอนโทรลเลอร์

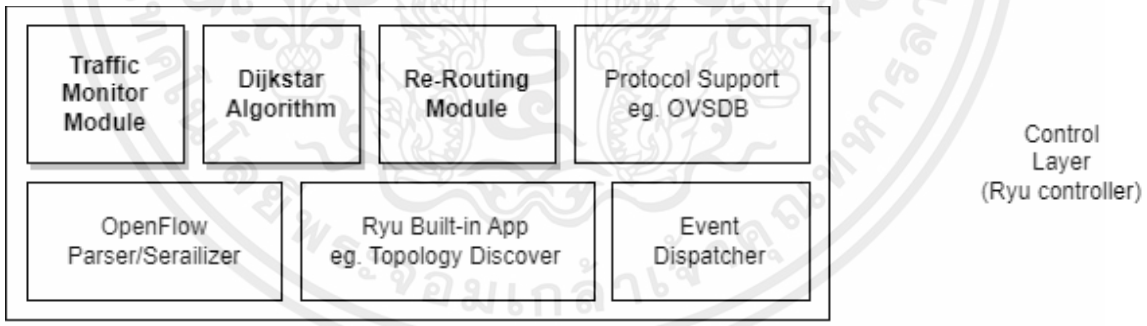
คอนโทรลเลอร์ที่พวกเราได้ตัดสินใจนำมาใช้กับโครงงานได้แก่ ริวเอสดีเอ็นคอนโทรลเลอร์ (Ryu SDN Controller) เป็นหัวใจสำคัญของเครือข่ายที่กำหนดด้วยซอฟต์แวร์ เป็นตัวกลางรวมไปถึงทำหน้าที่สื่อสารระหว่างชั้นแอปพลิเคชันและชั้นโครงสร้างพื้น ริวเอสดีเอ็นคอนโทรลเลอร์สามารถควบคุมการทำงานของระบบเครือข่ายทั้งระบบโดยใช้ภาษาไพทอน (Python) ในการเขียนโปรแกรม ภายในริวคอนโทรลเลอร์จะประอบไปด้วยโมดูลพื้นฐานสำหรับการทำงานร่วมกับโพรโทคอลโอเพนโฟลว์ เช่น Event Dispatcher, OpenFlow Parser/Serializer, Ryu Built-in App ดังแสดงในรูปที่ 3.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 Ryu SDN Framework แบบทั่วไป

จากรูปที่ 3.5 แอปพลิเคชันของริว (Ryu Applications) สามารถนำมาพัฒนาด้วยตนเองต่อได้ ซึ่งผู้พัฒนาส่วนใหญ่จะมีการออกแบบแอปพลิเคชันหรือปรับเปลี่ยนการทำงานเพื่อตอบสนองจุดประสงค์ของการทดลอง ความต้องการของตนเอง โดยโครงงานของเราได้ออกแบบโมดูลไว้ภายในริวเฟรมเวิร์คทั้งหมด 3 โมดูล ได้แก่ โมดูลสังเกตการณ์ส่งข้อมูลภายในเครือข่าย (Traffic Monitor Module) โมดูลการจัดเปลี่ยนเส้นทางใหม่ (Re-Routing Module) และ อัลกอริทึมไดก์สตาร์ (Dijkstra Algorithm) ดังแสดงในรูปที่ 3.6



รูปที่ 3.6 Ryu SDN Framework ของโครงงาน

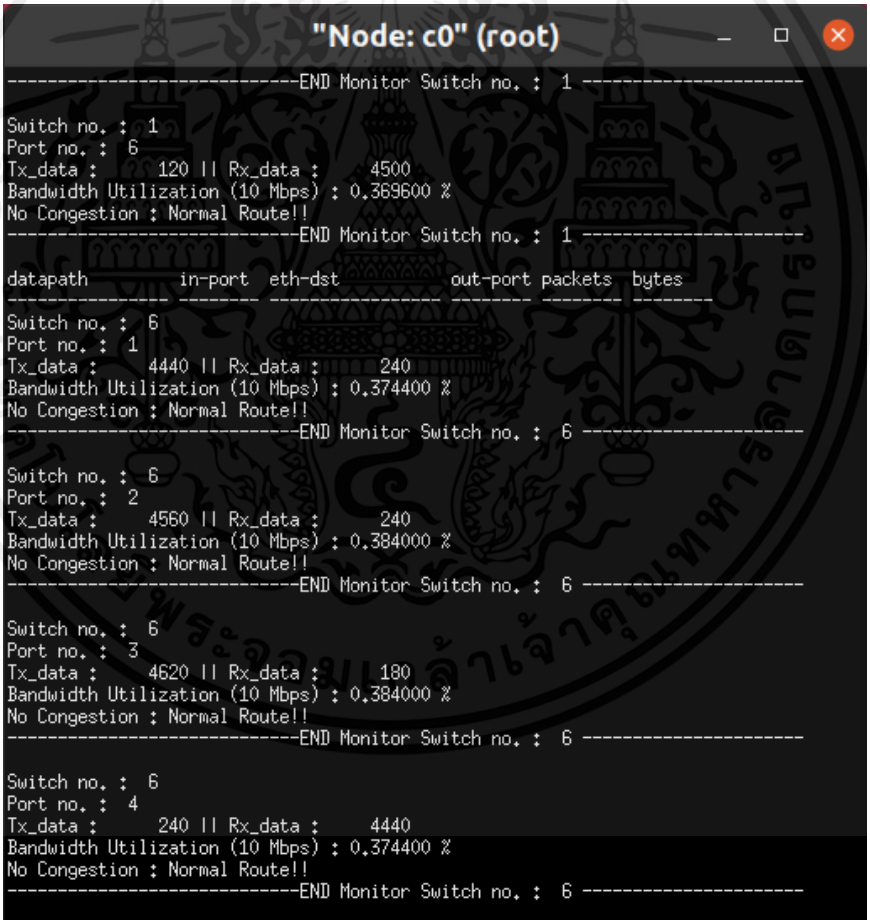
เราได้ทำการติดตั้งริวคอนโทรลเลอร์ให้ทำงานอยู่บนระบบปฏิบัติการอุบุนตูเช่นเดียวกับซอฟต์แวร์มินิเน็ต เราได้เลือกใช้ริวคอนโทรลเลอร์จากคอนโทรลเลอร์หลายๆ ตัว เช่น OpenDaylight

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(ODL), POX, Becon, Floodlight และอื่นๆ ซึ่งสาเหตุในการตัดสินใจและรายละเอียดต่างๆ ของคอนโทรลเลอร์แต่ละชนิด ซึ่งจะอธิบายต่อไปในบทที่ 4

3.1.2.1 โมดูลสังเกตการณ์การส่งข้อมูลภายในเครือข่าย

โมดูลสังเกตการณ์ข้อมูลในภายในเครือข่าย (Traffic Monitor Module) เป็นโมดูลที่ทำหน้าที่แสดงข้อมูลสถิติของการส่งข้อมูลแพ็กเก็ตภายในเครือข่าย เพื่อตรวจสอบว่าในแต่ละลิงค์มีการใช้งานแบนด์วิดท์เท่าไหร่ สวิตช์ที่เท่าไหร่ ขนาดข้อมูลของฝั่งรับ (Receiver_Data) ขนาดข้อมูลของฝั่งส่ง (Sender_Data) ซึ่งเราได้นำข้อมูลภายในเครือข่ายมาแสดงสถานการณ์ในแต่ละลิงค์แบบเรียลไทม์ (Real Time) ดังแสดงในรูปที่ 3.7 เพื่อวิเคราะห์เหตุการณ์ที่เกิดขึ้นภายในระบบ



รูปที่ 3.7 โมดูลสังเกตการณ์การส่งข้อมูลภายในเครือข่ายในแต่ละวินาที

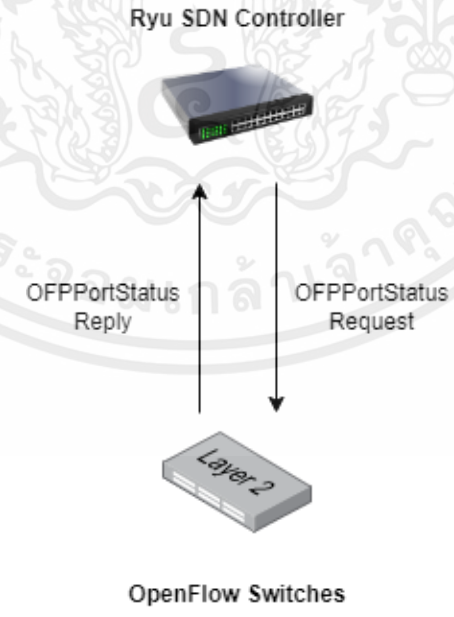
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในโมดูลสังเกตการณ์ภายในเครือข่าย เราได้นำข้อมูลที่ได้มาจากแพ็กเก็ตชนิด OFPPortStatsReply ของโพรโทคอลโอเพนโฟลว์มาคำนวณ ซึ่งจะประกอบไปด้วยฟิลด์ต่างๆ ได้แก่ หมายเลขของสวิตช์ (Switch ID) หมายเลขพอร์ทของสวิตช์ (Port ID) ขนาดข้อมูลของฝั่งรับของพอร์ท (Rx_bytes) ขนาดข้อมูลของฝั่งส่งของพอร์ท (Tx_bytes) และ ค่าการใช้ประโยชน์แบนด์วิดท์ของลิงค์ (Bandwidth Utilization) ในรูปแบบของเปอร์เซ็นต์

3.1.2.2 ค่าการใช้ประโยชน์แบนด์วิดท์

ค่าการใช้ประโยชน์แบนด์วิดท์ (Bandwidth Utilization) คือค่าที่ประเมินประสิทธิภาพของลิงค์นั้นๆ เพื่อให้ทราบว่าในแต่ละลิ้งค์มีการส่งข้อมูลแพ็กเก็ตปริมาณเท่าไหร่ โดยเราได้ใช้ฟิลด์ข้อมูลของแพ็กเก็ตชนิด OFPPortStatusReply ของโพรโทคอลโอเพนโฟลว์ เช่นเดียวกับการทำโมดูลสังเกตการณ์ส่งข้อมูลภายในเครือข่าย

OFPPortStatsReply เป็นแพ็กเก็ตของโพรโทคอลโอเพนโฟลว์ ซึ่งเป็นแพ็กเก็ตที่โอเพนโฟลว์สวิตช์ตอบกลับมายังริวเอสดีเอ็นคอนโทรลเลอร์ โดยริวเอสดีเอ็นคอนโทรลเลอร์ต้องส่งแพ็กเก็ตชนิด OFPPortStatsRequest ไปยังสวิตช์แต่ละตัวก่อนเพื่อร้องขอข้อมูลเชิงสถิติ กระบวนการการร้องขอข้อมูลเชิงสถิติดังแสดงในรูปที่ 3.8



รูปที่ 3.8 การร้องขอข้อมูลเชิงสถิติระหว่างริวคอนโทรลเลอร์และโอเพนโฟลว์สวิตช์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟิลด์ข้อมูลของแพ็กเก็ต OFPPortStatsReply ที่เรานำมาใช้เพื่อคำนวณค่าการใช้ประโยชน์แบนด์วิดท์ได้แก่ หมายเลขสวิตช์ หมายเลขพอร์ท จำนวนไบต์ที่รับ-ส่งในแต่ละลิงค์ ซึ่งสูตรที่ใช้ในการคำนวณค่าการใช้ประโยชน์แบนด์วิดท์สามารถคำนวณได้จากสมการ (1) ดังนี้

$$Bandwidth\ Utilization = \frac{(\text{Tx bytes} + \text{Rx bytes})}{1{,}250{,}000\ \text{bytes}} \quad (1)$$

เราได้กำหนดค่าแบนด์วิดท์สูงสุดในแต่ละลิงค์ให้มีค่าเท่ากับ 10 Megabits ซึ่งมีค่าเท่ากับ 1,2500,000 bytes เพื่อที่จะทำการประเมินค่าการใช้ประโยชน์แบนด์วิดท์ตามจุดประสงค์ของโครงงาน โดยเรากำหนดค่าแบนด์วิดท์ผ่านซอฟต์แวร์มินิเน็ต หลักจากที่คำนวณค่าการใช้ประโยชน์แบนด์วิดท์ของลิงค์ ณ เวลานั้นแล้ว ก็จะนำข้อมูลที่ได้มาเก็บอยู่ในไฟล์ประเภทซีเอสวี (CSV) เพื่อนำไปใช้เป็นชุดข้อมูล ดังแสดงในตารางที่ 3.1 โดยข้อมูลที่ได้มาจะเก็บในแต่ละวินาที หรือ เก็บข้อมูลแบบเรียลไทม์

Tx_bytes	Rx_bytes	Bandwidth Utilization (%)
60303	42345	8.21
92812	113550	16.50
65291	42382	8.61
57818	231450	23.14
54232	48231	8.19

ตารางที่ 3.1 ตัวอย่างชุดข้อมูลที่เก็บมาจาก OFPPortStatsReply ด้วยภาษาไพทอน

จากตางรางที่ 3.1 ชุดข้อมูลที่ได้มานี้จะนำไปใช้สำหรับการเทรนโมเดล (Training Model) หรือ ให้โมเดลการเรียนรู้เชิงลึกประเภท Bidirectional LSTM และ GRU ทำการเรียนรู้ และใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นอินพุท (Input) สำหรับแอปพลิเคชันคาดการณ์ค่าการใช้ประโยชน์แบนด์วิดท์ที่ทำงานอยู่ในชั้นแอปพลิเคชันของโครงงาน เพื่อคาดการณ์ค่าการใช้ประโยชน์แบนด์วิดท์ภายในอนาคตแบบเรียลไทม์

3.1.2.3 อัลกอริทึมไดก์สตาร์

อัลกอริทึมไดก์สตาร์ (Dijkstra Algorithm) เป็นอัลกอริทึมที่ใช้ค้นหาเส้นทางที่สั้นที่สุดระหว่างโหนดต้นทางไปยังโหนดปลายทาง เพื่อใช้เป็นเส้นทางหลักในการส่งข้อมูล

3.1.2.4 โมดูลการจัดเส้นทางใหม่

โมดูลการจัดเส้นทางใหม่ (Re-Routing Module) จะรับค่าการใช้ประโยชน์แบนด์วิดท์ที่ถูกคาดการณ์มาจากเอาท์พุต (Output) ของแอปพลิเคชันการคาดการณ์ค่าการใช้ประโยชน์แบนด์วิดท์ Bi-LSTM มาพิจารณาอีกทีหนึ่ง ซึ่งแบ่งได้เป็น 2 กรณี กรณีแรกกล่าวคือ ถ้าหากได้ค่าที่มากกว่าเกณฑ์ (Threshold) ที่กำหนดไว้ ระบบจะสั่งให้ริวเอสดีเอ็นคอนโทรลเลอร์สั่งการให้อุปกรณ์ที่ได้รับผลกระทบเปลี่ยนเส้นทางในการส่งข้อมูลใหม่ เพื่อป้องกันการเกิดความคับคั่งภายในเครือข่าย (Network Congestion) กรณีที่สองกล่าวคือ ถ้าหากได้ค่าที่ไม่เกินเกณฑ์ที่กำหนดไว้ก็จะไม่มีการเปลี่ยนเส้นทาง

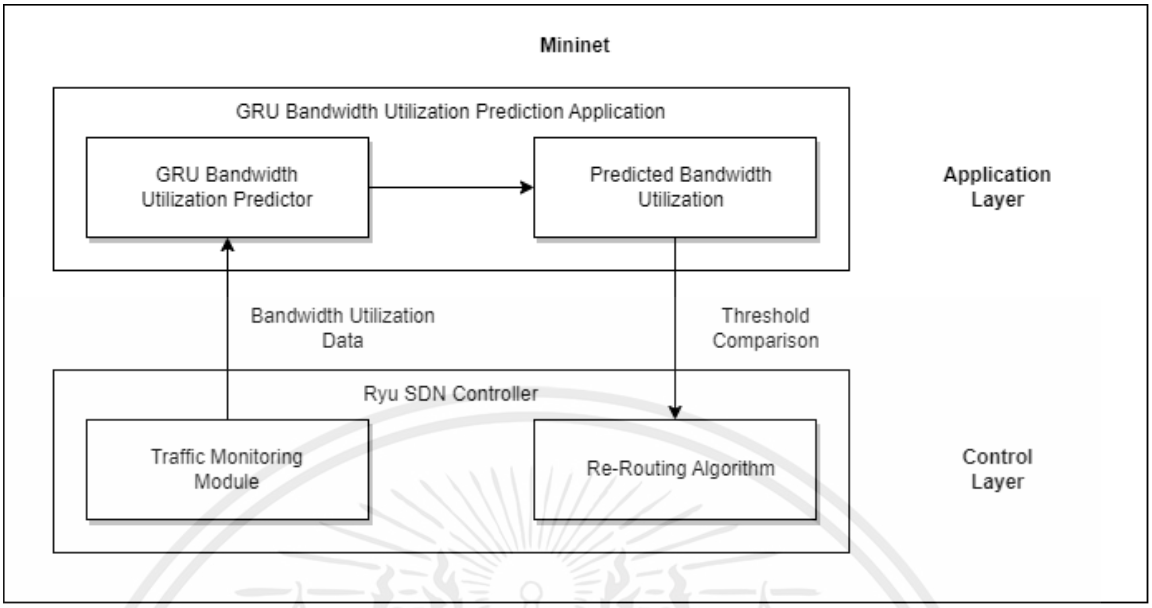
จากกรณีแรก เมื่ออัลกอริทึมการจัดเส้นทางใหม่ทำงาน จะสังเกตในหน้าจอแสดงผล (Terminal) ของเอสดีเอ็นคอนโทรลเลอร์ ว่ามีการตรวจสอบเจอค่าการใช้ประโยชน์แบนด์วิดท์ที่เกินเกณฑ์ที่ตั้งไว้ระหว่างสวิตช์ หลังจากตรวจสอบเจอแล้ว คอนโทรลเลอร์จะทำการจัดเปลี่ยนเส้นทางให้ใหม่

หลังจากที่เอสดีเอ็นได้ทำการจัดเปลี่ยนเส้นทางใหม่แล้ว โมดูลการจัดเส้นทางใหม่จะไม่ได้ทำงานต่อเนื่องเป็นเวลา 30 วินาที และจะไม่มีการคาดการณ์ค่าการใช้ประโยชน์แบนด์วิดท์เนื่องจากเราป้องกันไม่ให้มีการเปลี่ยนแปลงตารางเส้นทาง (Routing Table) อย่างต่อเนื่อง และ ไม่ให้ข้อมูลเปลี่ยนแปลงบ่อยเกินไป

3.1.3 แอปพลิเคชันคาดการณ์ค่าการใช้ประโยชน์แบนด์วิดท์ GRU

แอปพลิเคชันคาดการณ์ค่าการใช้ประโยชน์แบนด์วิดท์ ทำหน้าที่คาดการณ์ข้อมูลแบนด์วิดท์ที่ได้มาจากริวคอนโทรลเลอร์แบบเรียลไทม์ แล้วจะส่งค่าที่คาดการณ์ออกมาเป็นเอาท์พุตกลับไปยังโมดูลการจัดเส้นทางใหม่ที่อยู่ในส่วนของริวคอนโทรลเลอร์ ซึ่งแอปพลิเคชันนี้จะเป็นส่วนประกอบของชั้นแอปพลิเคชันดังแสดงในรูปที่ 3.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 การทำงานระหว่างชั้นแอปพลิเคชัน และ ชั้นควบคุม

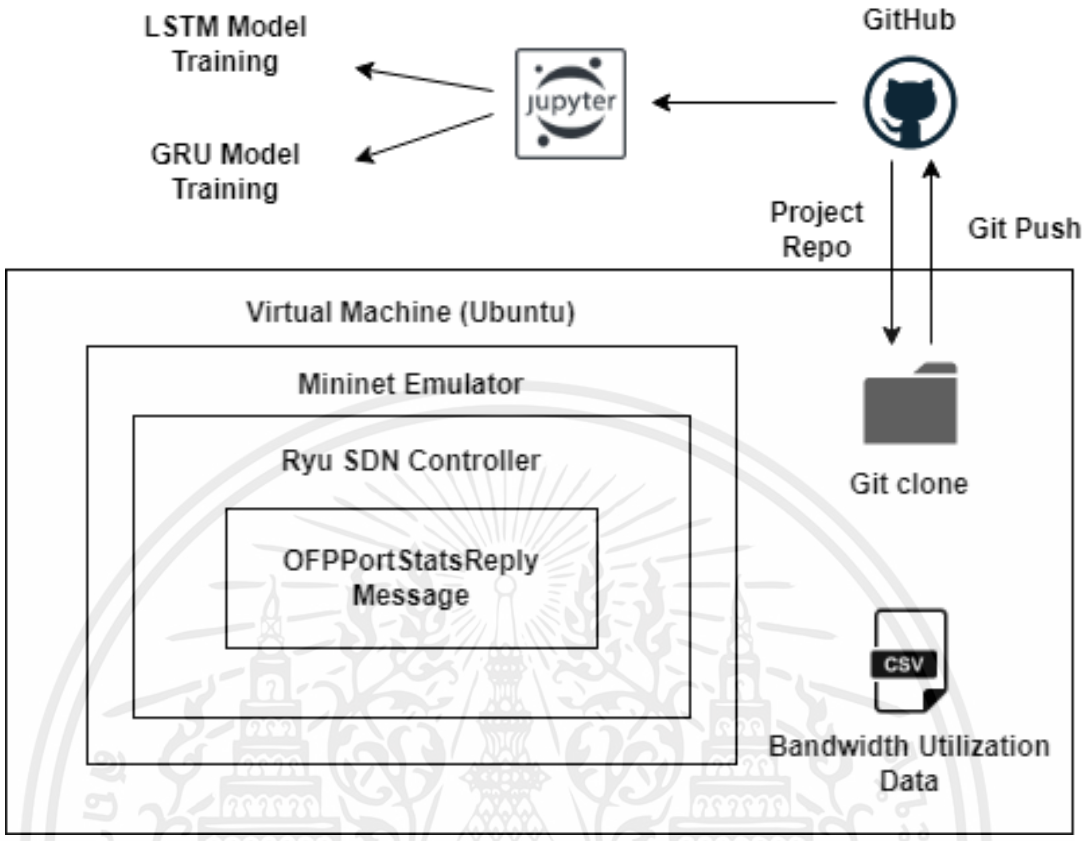
3.1.4 ขั้นตอนการสร้างแอปพลิเคชันในโครงงาน

ก่อนที่เรานำโมเดล Bi-LSTM และ GRU ไปทดสอบในโครงงาน เราได้ทำการเขียนโมเดลการเรียนรู้เชิงลึกประเภท Bidirectional Long Short Term Memory (Bi-LSTM) และ Gated Recurrent Units (GRU) โดยใช้แพลตฟอร์มจูปิเตอร์ (Jupyter Notebook) และทดสอบบนเอนไวรอนเมนต์ของอนาคอนด้า (Anaconda) บนระบบปฏิบัติการวินโดว์ (Windows OS)

3.1.4.1 การรวบรวมข้อมูล

ข้อมูลที่นำมาเทรนในโมเดล Bi-LSTM และ GRU มาจากโมดูลสังเกตการณ์การส่งข้อมูลภายในเครือข่าย (Traffic Monitor Module) และเราได้นำข้อมูลนั้นออกมาสร้างเป็นไฟล์ชนิดซีเอสวี (CSV) เก็บเอาไว้ในริวเอสดีเอ็นคอนโทรลเลอร์ ซึ่งเราได้ใช้วิธีส่งไฟล์จากเครื่องเสมือน (Virtual Machine) ไปยังระบบปฏิบัติการวินโดว์ (Windows OS) โดยใช้ Git เพราะเราไม่สามารถส่งจากเครื่องเสมือนไปยังระบบปฏิบัติการวินโดว์ได้โดยตรงดังแสดงในรูป 3.12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 การรวบรวมข้อมูลจากเอสดีเอ็นไปยังแพลตฟอร์มจูปิเตอร์

3.1.4.2 กระบวนการเตรียมข้อมูล

หลังจากที่รวบรวมชุดข้อมูลจากริวคอนโทรลเลอร์ได้แล้ว ก่อนที่เราจะทำการเทรนโมเดล เราควรต้องทำการเตรียมข้อมูลก่อน เพื่อที่จะจัดข้อมูลอยู่ในรูปแบบที่เหมาะสมกับการเทรน และเพื่อที่จะให้การเรียนรู้เชิงลึกอย่างมีประสิทธิภาพ กระบวนการต่อไปที่จำเป็นต่อการเทรนหลายๆ โมเดล คือ การทำ Normalization กล่าวคือเป็นวิธีการแปลงข้อมูลให้ชุดตัวเลขมีค่าระหว่าง 0 ถึง 1 ซึ่งจะช่วยให้โมเดลการเรียนรู้เชิงลึกประเภท Bi-LSTM และ GRU เรียนรู้ได้แม่นยำ

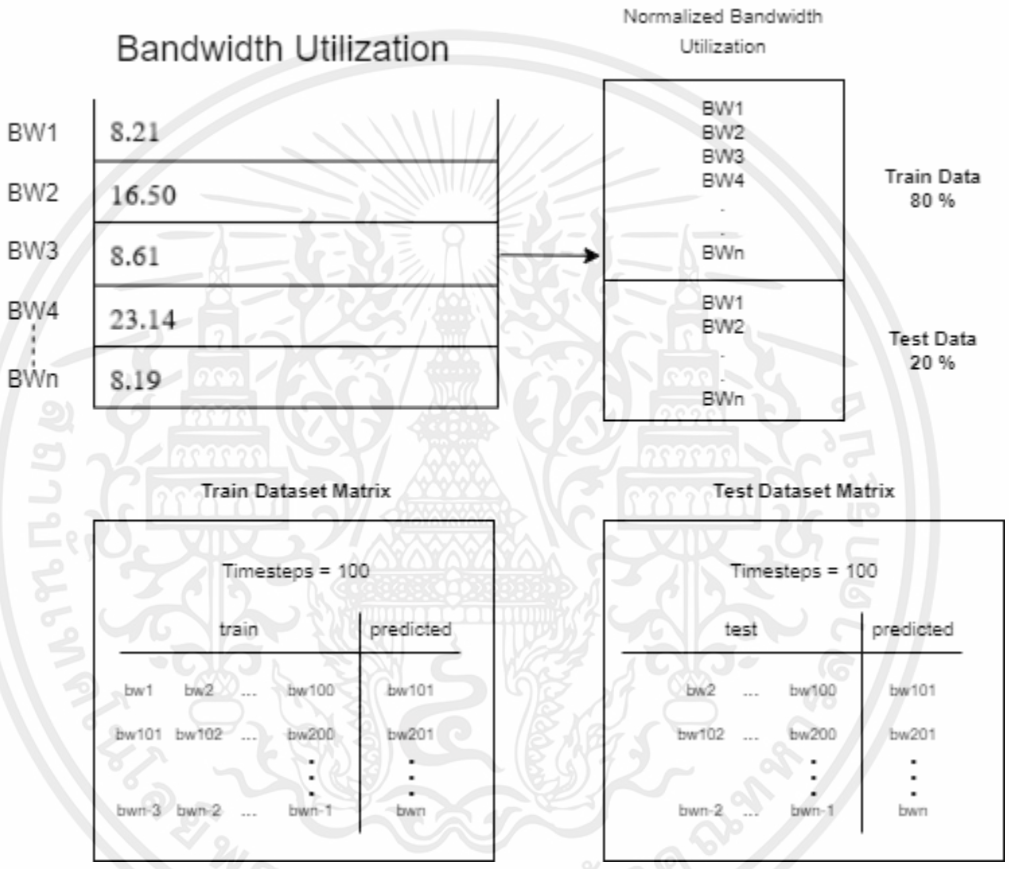
3.1.4.3 ชุดข้อมูลทดลองและชุดข้อมูลทดสอบ

เมื่อทำการเตรียมข้อมูลเสร็จแล้ว เราจะทำการแบ่งข้อมูลออกเป็น 2 ส่วน ได้แก่ ชุดข้อมูลทดลอง (Train Data) และ ชุดข้อมูลทดสอบ (Test Data) โดยแบ่งออกเป็นสัดส่วน ชุดทดลอง 80 % และ ชุดทดสอบ 20 % หลังจากนั้นจะเริ่มทำการทดสอบโมเดลว่าโมเดลมีความแม่นยำสูงเพียงใดโดยทดสอบกับข้อมูลชุดทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.4.4 ชุดข้อมูลในเชิงเมทริกซ์

กล่าวคือรูปแบบอินพุตสำหรับในการเทรนข้อมูลในโมเดลการเรียนรู้เชิงลึกประเภท Bi-LSTM และ GRU ชุดข้อมูลที่ถูกแบ่งออกเป็น Train Data และ Test Data จะถูกแบ่งออกเป็น Train Dataset Matrix และ Test Dataset Matrix ดังแสดงในรูปที่ 3.13



รูปที่ 3.11 กระบวนการเตรียมข้อมูลก่อนทำการเทรนโมเดล Bi-LSTM และ GRU

Timesteps เป็นตัวแปรที่กำหนดจำนวนอินพุตของ (Node) โครงข่าย Bi-LSTM และความกว้างของ Train Data หรือที่เรียกกันว่า Sliding Window กล่าวคือ Train Data จะมีขนาดเป็น 100 คอลัมน์ ส่วน Predicted จะมีขนาดเป็น 1 คอลัมน์เสมอ ซึ่งเป็นสาเหตุที่ทำให้ชั้นอินพุต (Input Layer) ของโมเดล Bi-LSTM มีจำนวน 100 โหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.4.5 โครงสร้างของโมเดล Bi-LSTM และ GRU

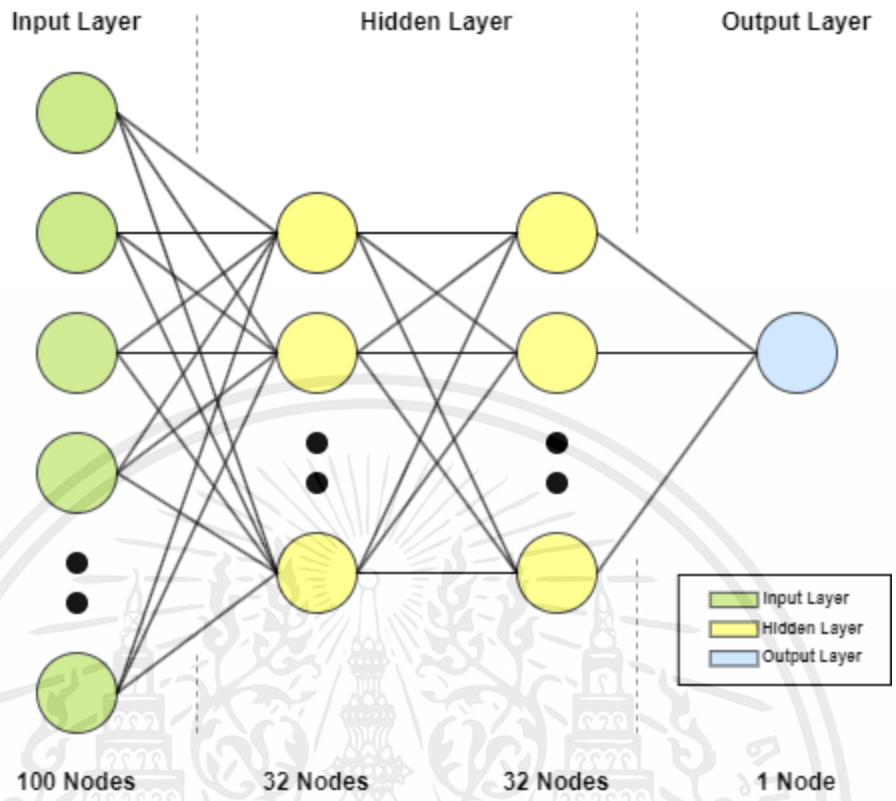
หลังจากที่ผ่านกระบวนการเตรียมข้อมูลและแบ่งข้อมูลเป็น 2 ชุดแล้ว ขั้นตอนต่อไปคือการสร้างโมเดลการเรียนรู้เชิงลึกประเภท Bi-LSTM และ GRU หลังจากนั้นเราได้ทำการปรับจูนพารามิเตอร์ต่างๆ ในการทดลองโมเดล รวมไปถึงทำการทดลองปรับพารามิเตอร์ตามเอกสารงานวิจัยที่อ้างอิงในบทที่ 2 เพื่อทดสอบว่าโมเดลที่เรานำเสนอนั้นมีความแม่นยำมากน้อยเพียงใดเมื่อเทียบกับงานวิจัยที่เกี่ยวข้อง และให้เกิดความแม่นยำมากที่สุด

ทั้งสองโมเดลจะประกอบไปด้วยชั้นอินพุต (Input Layer) ที่มีจำนวน 100 โหนด เชื่อมต่อด้วยชั้นซ่อนตัว (Hidden Layer) 2 ชั้น ชั้นละ 32 โหนด และเชื่อมต่อไปยังชั้นเอาท์พุต (Output Layer) ที่มีจำนวน 1 โหนดดังแสดงในรูปที่ 3.14

พารามิเตอร์ (Parameter)	ค่าของพารามิเตอร์ (Value)
Epochs	5
Optimizer	RMS Prop
Loss	Mean Absolute Error (MAE)
Batch Size	256
Learning Rate	0.001

ตารางที่ 3.2 ค่าพารามิเตอร์ที่ใช้ในการทดลองของโมเดล Bi-LSTM และ GRU

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12 โครงสร้างของโครงข่ายประสาทเทียมชนิด Bi-LSTM และ GRU ของโครงงาน

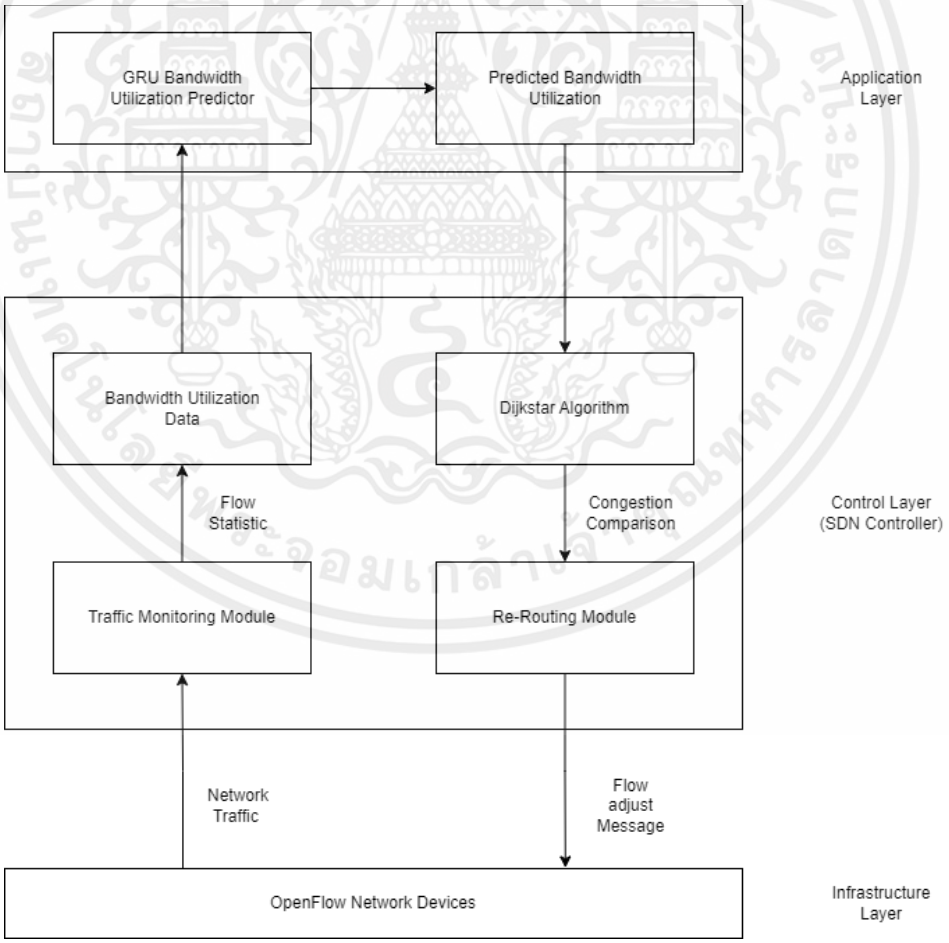
ในชั้นเอาท์พุตจะเป็นชั้นที่บ่งบอกว่าเลขที่คาดการณ์ออกมาเป็นอย่างไร และเลขที่ถูกคาดการณ์ออกมาจากชั้นนี้ ก็จะถูกนำไปเปรียบเทียบกับค่าการใช้ประโยชน์แบนด์วิดท์ในชุดข้อมูลจริง เพื่อทดสอบและวัดประสิทธิภาพโมเดล ว่าโมเดลนี้สามารถคาดการณ์ค่าในอนาคตที่จะทำนายออกมาได้แม่นยำมากน้อยเพียงใด

หลังจากที่โมเดล Bi-LSTM และ โมเดล GRU ถูกทดสอบมาหลายครั้ง มีการปรับจูนพารามิเตอร์ให้ทำงานได้อย่างมีประสิทธิภาพ และคาดการณ์แม่นยำแล้ว เราจึงนำโมเดลนี้ไปทำงานบนริวคอนโทรลเลอร์ เพื่อคาดการณ์ค่าการใช้ประโยชน์แบนด์วิดท์แบบเรียลไทม์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

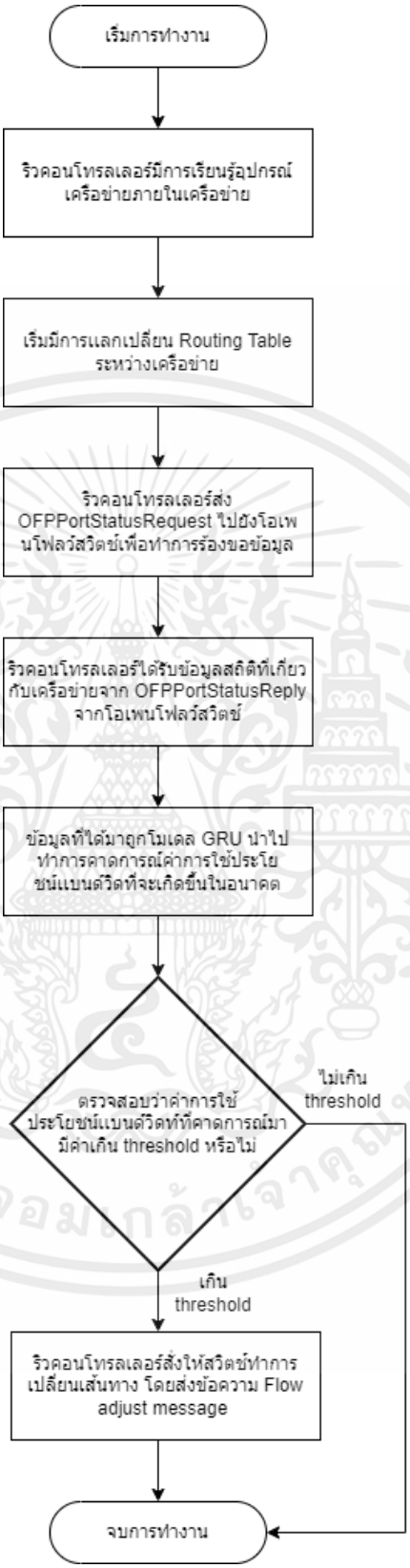
3.2 ขั้นตอนการทำงานของระบบทั้งหมด

กระบวนการทำงานเริ่มต้นจากมินิเน็ต มินิเน็ตจะเป็นซอฟต์แวร์ที่จำลองเครือข่าย (Network Topology) ขึ้นมา และริวคอนโทรลเลอร์จะเป็นตัวควบคุมอุปกรณ์ที่อยู่ในชั้นโครงสร้างพื้นฐานทั้งหมด (Infrastructure Layer) เมื่อมีการส่งแพ็กเก็ตข้อมูลภายในเครือข่าย ริวเอสคอนโทรลเลอร์จะทำการเก็บข้อมูลในเครือข่ายไว้อย่างต่อเนื่องและทำการส่งข้อมูลไปยังชั้นแอปพลิเคชัน ที่ประกอบไปด้วยโมเดลการเรียนรู้เชิงลึกประเภท GRU เซึ่งทำหน้าที่คาดการณ์ค่าการใช้ประโยชน์แบนด์วิดท์ (Bandwidth Utilization) จากนั้นโมเดลจะทำการส่งข้อมูลที่คาดการณ์มาแล้วกลับไปยังริวคอนโทรลเลอร์เพื่อเช็คระดับความคับคั่งของเครือข่ายในแต่ละลิงค์ (Link) หากไม่เกินเกณฑ์ที่ตั้งไว้ก็จะทำการส่งข้อมูลปกติ แต่ถ้าหากเกินเกณฑ์ที่ตั้งไว้ก็จะทำการเปลี่ยนเส้นทางในการส่งข้อมูล (Re-Routing) และส่งข้อความ Flow adjust Message ไปยังสวิตช์เพื่ออัพเดทการส่งข้อมูลใหม่ดังแสดงในรูปที่ 3.15



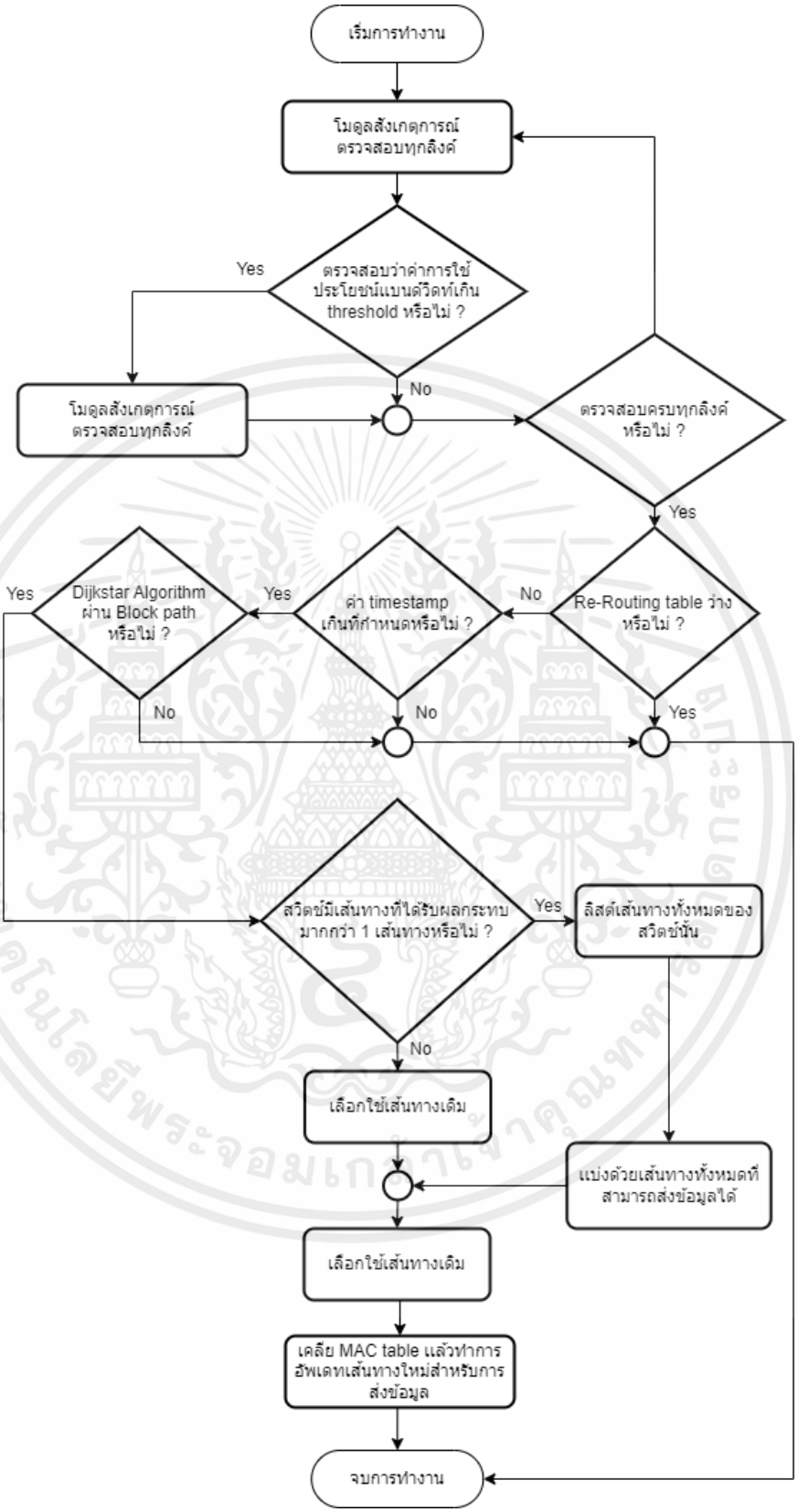
รูปที่ 3.13 ภาพรวมขั้นตอนทำการทำงานของระบบทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.14 โฟลว์ชาร์ตการทำงานของระบบอย่างละเอียด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.15 โฟลว์ชาร์ตการทำงานของโมดูลการจัดเส้นทางใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลอง

ในบทนี้ เราจะทำการกล่าวถึงความก้าวหน้าของโครงงานและสิ่งที่ได้ปฏิบัติไปแล้วในภาคเรียนนี้ หรือ วิชาโครงงาน 1 (Project I) โดยจะประกอบไปด้วย การติดตั้งโปรแกรม (Environment Setup), การออกแบบโครงสร้างเครือข่าย (Topology Design), การรวบรวมข้อมูลจากการจำลอง (Data Gathering) และนำข้อมูลมาเรียนรู้ในโมเดลการเรียนรู้เชิงลึก (Deep Learning Training) เพื่อทำการเปรียบเทียบระหว่างโมเดล Bi-LSTM และ LSTM ในการคาดการณ์ค่าการใช้ประโยชน์แบนด์วิดท์ (Bandwidth Utilization) ในเครือข่ายที่ถูกควบคุมโดยซอฟต์แวร์ เพื่อนำไปสู่การยกระดับประสิทธิภาพของเครือข่าย โดยจะแบ่งเป็นหัวข้อต่าง ๆ ดังนี้

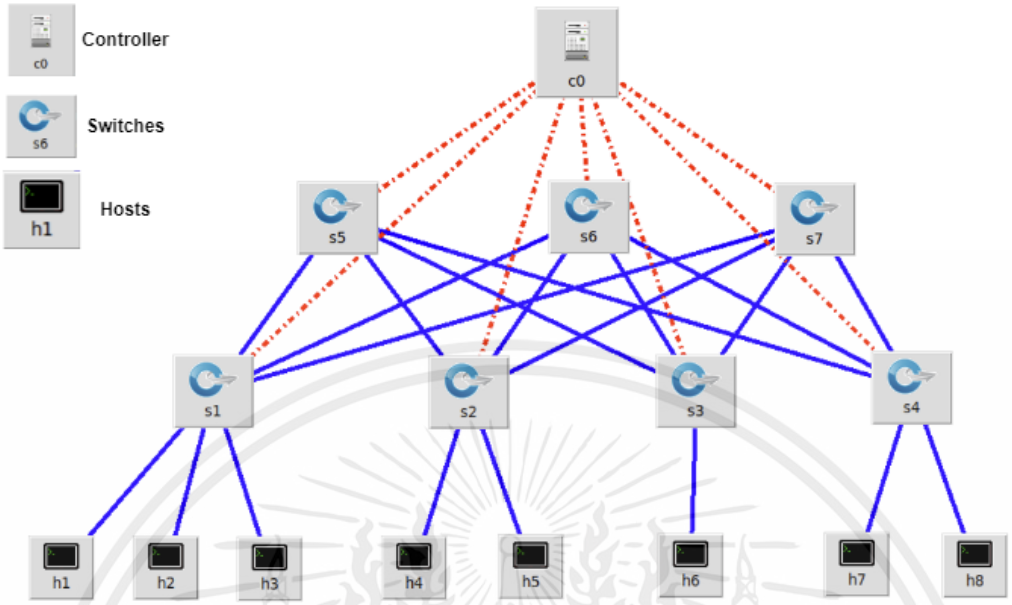
4.1 การติดตั้งโปรแกรม

ในโครงงานนี้ เราได้ทำการทดลองทั้งหมดภายใน Virtual Machine โดยใช้งานผ่านระบบปฏิบัติการอุบุนตู (Ubuntu OS) ซึ่งเหมาะกับการใช้งานกับงานประเภทเครือข่าย โดยเราได้ใช้มินิเน็ตอิมูเลเตอร์ (Mininet Emulator) เพื่อทำการจำลองเครือข่ายและได้เลือกใช้ริวคอนโทรลเลอร์ (Ryu Controller) ในการควบคุมอุปกรณ์เครือข่าย ในการจำลองการทำงานของเครือข่ายเราได้ใช้โปรแกรม D-ITG (Distributed Internet Traffic Generator) ในการสร้างโฟลว์ภายในเครือข่าย เพื่อจำลองข้อมูลที่จะส่งจากโฮสท์หนึ่งไปยังอีกโฮสท์หนึ่ง

4.1.1 การออกแบบโครงสร้างเครือข่าย

ในการออกแบบเครือข่าย (Topology Design) เราได้นำเครือข่ายจากเอกสารงานวิจัย “SDPredictNet-A Topology based SDN Neural Routing Framework with Traffic Prediction Analysis” มาทำการทดลอง เนื่องจากเราจะประเมินประสิทธิภาพของโมเดลการเรียนรู้เชิงลึกระหว่าง Bi-LSTM และ LSTM โดยเราจะใช้พารามิเตอร์จากเอกสารงานวิจัยในการวัดประเมินประสิทธิภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



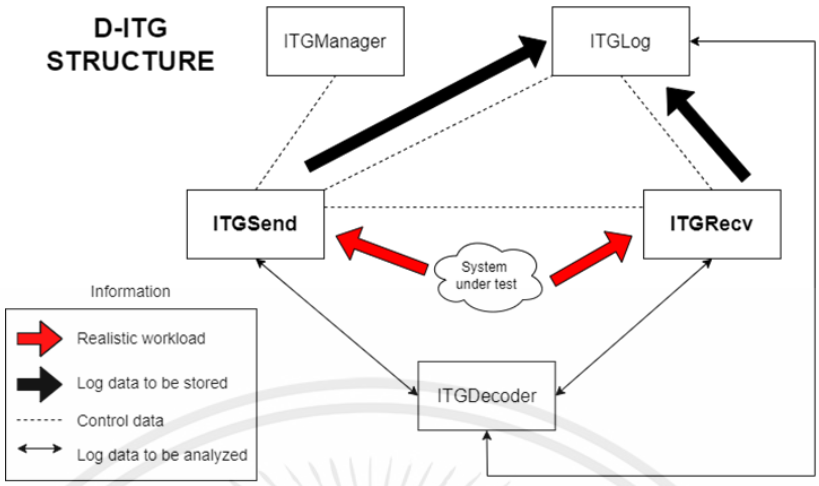
รูปที่ 4.1 การออกแบบเครือข่ายโดยใช้ซอฟต์แวร์มินิเน็ต

นอกจากนี้ เราได้ทำการกำหนดให้แต่ละลิงค์มีแบนด์วิดท์ที่สูงที่สุดที่มีขนาด 10 Mbps เพื่อที่จะทำการกำหนดขอบเขตในการส่งข้อมูล รวมไปถึงการตรวจจับความคับคั่ง (Congestion Detection)

4.1.2 D-ITG (Distributed Internet Traffic Generator)

เนื่องจากโครงงานของเราไม่ได้ใช้อุปกรณ์เครือข่ายโดยตรง ซึ่งเป็นการทดลองภายใน Virtual Machine แทน เพราะค่าใช้จ่ายค่อนข้างสูง เราจึงทำการนำโปรแกรม D-ITG มาใช้งานเพื่อจำลองการส่งข้อมูลภายในเครือข่าย โดยจะให้โฮสท์หนึ่งทำหน้าที่เป็นตัวส่งข้อมูล (Source Host) และ โฮสท์อีกตัวหนึ่งทำหน้าที่เป็นตัวรับข้อมูล (Destination Host) โดยแสดงได้ดังรูปที่ 4.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 โครงสร้างของซอฟต์แวร์ D-ITG

เราสามารถตรวจสอบข้อมูลทางสถิติของเครือข่ายระหว่างโฮสท์ถึงโฮสท์ (End to End) ได้ เช่น Delay, Packet loss, Throughput ฯลฯ โดยตรวจสอบผ่าน Decoder ของซอฟต์แวร์ (ITGDecoder) ซึ่งสามารถเก็บอยู่ในประเภท Logfile ได้ด้วย

```
Flow number: 1
From 10.0.0.1:36390
To   10.0.0.2:8999
---------------------------------------------------
Total time                =     14.945078 s
Total packets             =           150
Minimum delay             =      0.000040 s
Maximum delay             =      0.000569 s
Average delay             =      0.000065 s
Average jitter            =      0.000013 s
Delay standard deviation  =      0.000044 s
Bytes received            =         15000
Average bitrate           =      8.029399 Kbit/s
Average packet rate       =     10.036749 pkt/s
Packets dropped           =             0 (0.00 %)
Average loss-burst size   =      0.000000 pkt
---------------------------------------------------
```

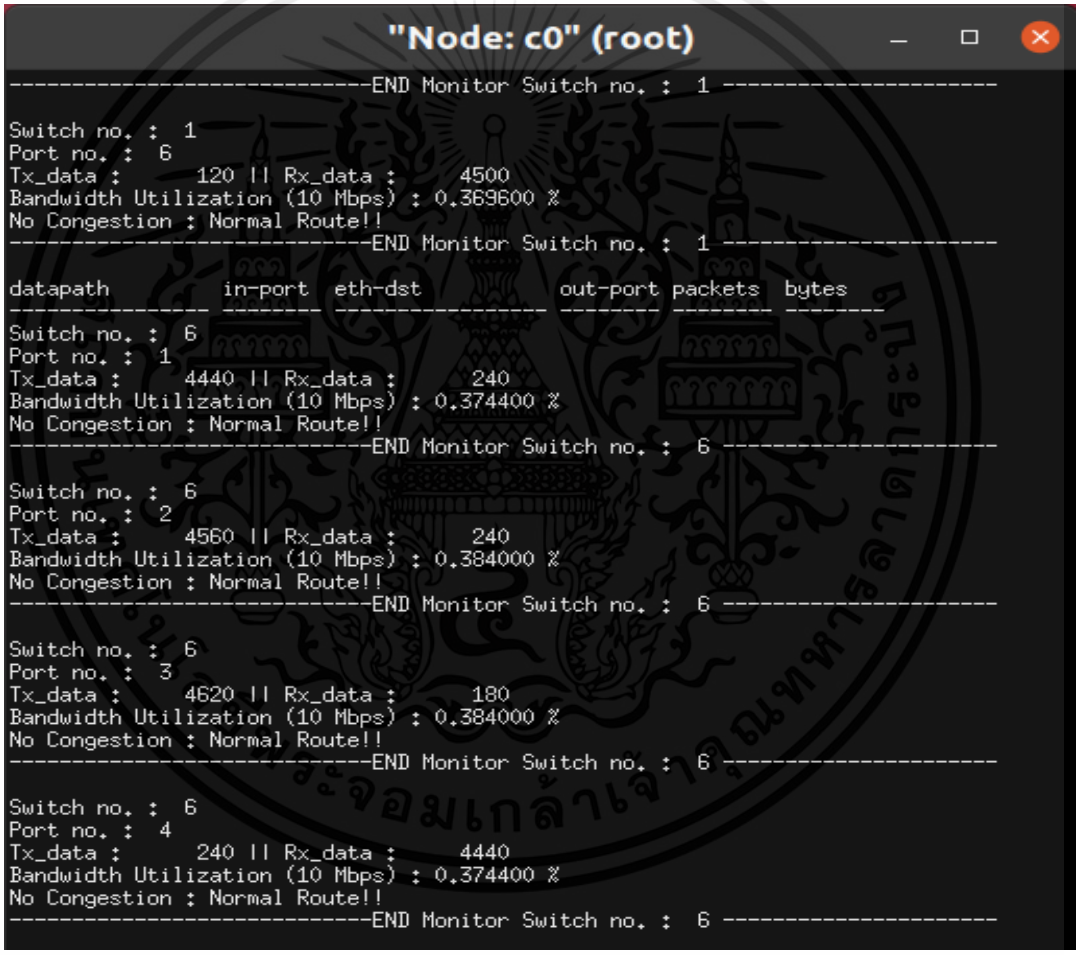
รูปที่ 4.3 ตัวอย่างข้อมูลทางสถิติจากซอฟต์แวร์ D-ITG

จากรูปที่ 4.3 เราได้ทำการทดลองให้มีการจำลองการส่งข้อมูลจาก Host 1 (IP : 10.0.0.1) ไปยัง Host 2 (IP : 10.0.0.2) และตรวจสอบข้อมูลทางสถิติผ่าน Decoder ของซอฟต์แวร์ D-ITG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.3 ริวคอนโทรลเลอร์

ริวคอนโทรลเลอร์ (Ryu Controller) เป็นเฟรมเวิร์ค (Framework) ชนิดหนึ่งที่มีการควบคุมโดยใช้ภาษาไพทอน (Python) ทำหน้าที่ควบคุมเอสดีเอ็นและการทำงานต่างๆ โดยควบคุมอุปกรณ์เครือข่ายผ่านโพรโทรคอลโอเพนโฟลว์ (OpenFlow) ซึ่งเราได้ออกแบบโมดูลของการทำงานหลัก ๆ 3 โมดูล ได้แก่ โมดูลสังเกตการณ์เครือข่าย (Traffic Monitor Module), อัลกอริทึมไดก์สตรา (Dijkstra Algorithm) และ โมดูลการจัดเปลี่ยนเส้นทาง (Re-Routing Module)



รูปที่ 4.4 โมดูลสังเกตการณ์เครือข่าย (Traffic Monitor Module)

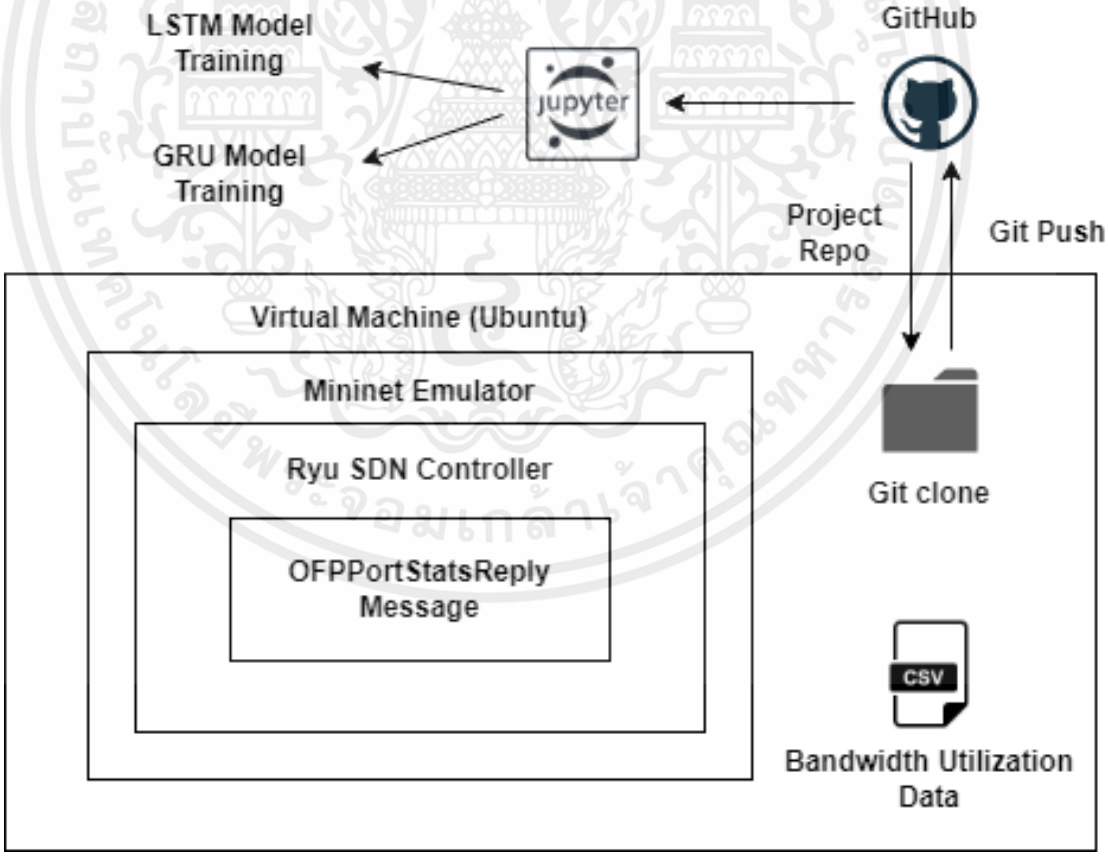
โมดูลสังเกตการณ์เครือข่าย จะทำหน้าที่ตรวจสอบความคับคั่งของเครือข่ายแบบเรียลไทม์ (Realtime Monitor) โดยจะแสดงข้อมูลดังนี้ หมายเลขของสวิตช์ (Switch No.), หมายเลขพอร์ต (Port

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

No.) จำนวนไบต์ที่รับ (Rx_Byte), จำนวนไบต์ที่ส่ง (Tx_Byte) และ ค่าการใช้ประโยชน์แบนด์วิดท์ (Bandwidth Utilization) ซึ่งข้อมูลเหล่านี้จะนำมาจากแพ็กเก็ตของโอเพนโฟลว์โพรโทคอลประเภท OFPPortStatusReply จากริวคอนโทรลเลอร์ เพื่อแสดงรายละเอียด ณ ลิงค์นั้น

4.2 การรวบรวมข้อมูล

หลังจากที่เราได้มีการออกแบบเครือข่าย การใช้โปรแกรม D-ITG ในการจำลองการส่งข้อมูล และใช้ริวคอนโทรลเลอร์ทำโมดูลสังเกตการณ์เครือข่าย เราได้ใช้ริวคอนโทรลเลอร์ทำการเก็บข้อมูลสถิติทางเครือข่ายไว้ และแปลงให้อยู่ในไฟล์ประเภทซีเอสวี (CSV file) และทำการส่งข้อมูลผ่าน Git สู่ GitHub เพื่อนำไปทำการเรียนรู้ในการเรียนรู้เชิงลึก (Deep Learning Training) บนระบบปฏิบัติการวินโดว์ (Window OS) ดังรูปที่ 4.5



รูปที่ 4.5 ขั้นตอนการนำข้อมูลจาก Virtual Machine สู่ระบบปฏิบัติการวินโดว์ผ่าน Git

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.1 การเทรนข้อมูล

โมเดลการเรียนรู้เชิงลึกที่เราได้นำมาประยุกต์ใช้ในโครงงานได้แก่ Bidirectional LSTM ซึ่งเป็นโมเดลที่ถูกพัฒนามาจาก LSTM เราได้นำข้อมูล Bandwidth Utilization Data มาเรียนรู้ภายในโมเดลทั้งสอง เพื่อทำการเปรียบเทียบความแม่นยำในการคาดการณ์ข้อมูลที่จะเกิดขึ้นในอนาคต โดยพารามิเตอร์ที่เราได้ทำการตั้งค่า เราได้ตั้งค่าให้เหมือนกับเอกสารงานวิจัยที่เราได้นำมาประยุกต์ใช้งานกับโครงงานดังตารางที่ 4.1 และผลลัพธ์การเรียนรู้ทั้งสองโมเดลดังรูปที่ 4.6

พารามิเตอร์ (Parameter)	ค่าของพารามิเตอร์ (Value)
Epochs	5
Optimizer	RMS Prop
Loss	Mean Absolute Error (MAE)
Batch Size	256
Learning Rate	0.001

ตารางที่ 4.1 พารามิเตอร์ที่ได้นำมาใช้จากเอกสารงานวิจัย

โครงงานชิ้นนี้ เราได้เลือกใช้อัลกอริทึมการเรียนรู้เชิงลึกที่เป็นการทำงานแบบอนุกรมเวลา (Time Series) หลากหลายประเภท เพื่อที่จะนำมาใช้งานในการคาดการณ์ค่าการใช้ประโยชน์แบนด์วิดท์ และ เพื่อทำการวิเคราะห์ว่าในลิงค์แต่ละลิงค์มีความคับคั่งหรือไม่

โมเดลที่ได้นำมาใช้ในการทดลองภายในโครงงานนี้ได้แก่ Long-Short Term Memory, Bi-Directional Long-Short Term Memory และ Gated Recurrent Units และได้เลือกใช้ Mean Absolute Error (MAE) เป็นตัวแปรที่ใช้ในการเปรียบเทียบระดับความแม่นยำของการคาดการณ์ของแต่ละโมเดล

4.2.1.1 Mean Absolute Error

Mean Absolute Error (MAE) เป็นตัวแปรที่นิยมใช้ในการประเมินความแม่นยำหรือความถูกต้องของโมเดลที่ทำนายค่าออกมาเป็นตัวเลข (Numerical Value) โดยสามารถคำนวณได้มาจากสูตรดังนี้

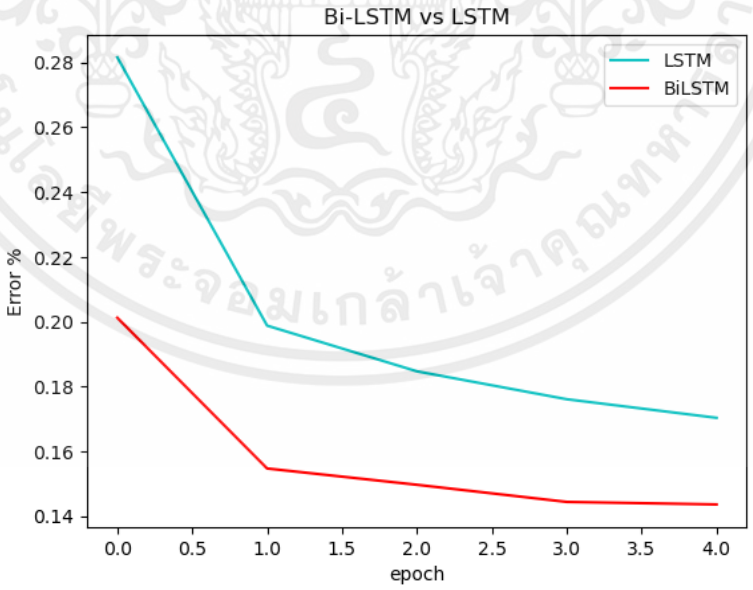
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$MAE = \sum_{y=1}^{n} \frac{|y_i - \hat{y}|}{n} \quad (1)$$

จากสมการดังกล่าว y_i หมายถึงค่าจริง (Actual Value) และ $\hat{y}$ หมายถึงค่าที่โมเดลทำการคาดการณ์ออกมา (Predicted Value) $y_i - \hat{y}$ แสดงถึงการคำนวณค่าความผิดพลาดของการคาดการณ์จากโมเดล และ n แสดงถึงจำนวณความผิดพลาดที่เกิดขึ้น โดยค่า MAE ของแต่ละโมเดลวัดได้ดังแสดงในตาราง 4.2 (ค่าความผิดพลาดในตารางเป็นค่าความผิดพลาดที่ดีที่สุดที่ได้มาจากการทดลองทั้งหมด)

โมเดลการเรียนรู้เชิงลึก	Mean Absolute Error
LSTM	14.4 %
Bi-LSTM	14.9 %
GRU	14.1 %

ตารางที่ 4.2 แสดงค่าความผิดพลาดจากการทำนายของแต่ละโมเดล

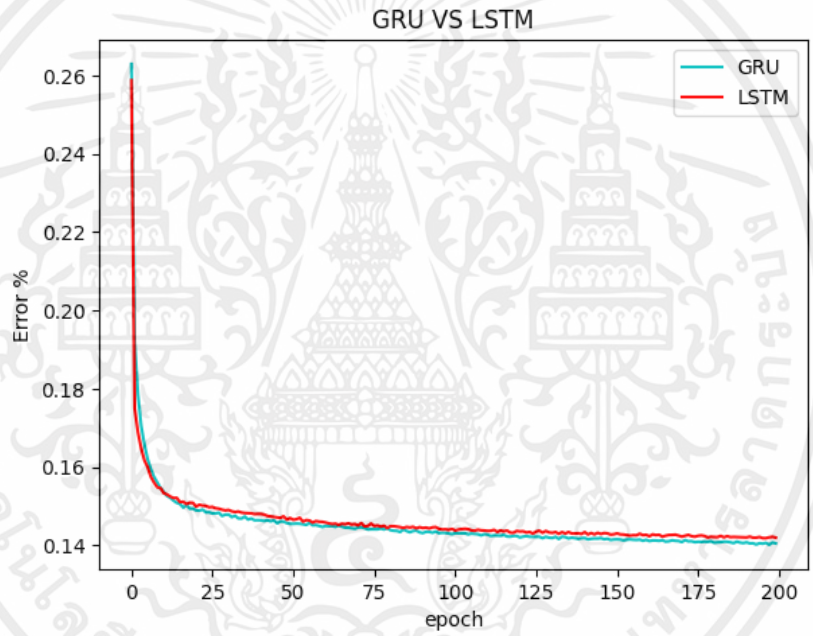


รูปที่ 4.6 ผลลัพธ์ของการเรียนรู้ระหว่างโมเดล LSTM และ Bi-LSTM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากกราฟ ค่าความแม่นยำของโมเดล Bi-LSTM มีค่าที่สูงกว่าโมเดล LSTM อย่างชัดเจนอยู่ที่ 3 % แต่เวลาที่ใช้ในการคาดการณ์ (Computation Time) ที่ใช้ในการทำนายนั้น โมเดล LSTM และ โมเดล Bi-LSTM ใช้เวลาในการทำนายค่อนข้างนานและใกล้เคียงกันระหว่างสองโมเดลนี้

หลังจากที่ได้ทดลองโมเดล Bi-LSTM และ LSTM ในการคาดการณ์ค่าการใช้ประโยชน์แบนด์วิดท์แล้ว เราได้ทำการศึกษาโมเดลที่นอกจากสองโมเดลนี้ได้แก่โมเดล GRU (Gated Recurrent Units) การทำงานของโมเดล GRU ทำหน้าที่คล้ายคลึงกับโมเดล LSTM เพียงแต่โมเดล GRU มีหลักการทำงานที่น้อยกว่าโมเดล LSTM ซึ่งให้มีการทำนายที่เร็วขึ้นกว่าเดิม



รูปที่ 4.7 การเปรียบเทียบการเรียนรู้ระหว่างโมเดล LSTM และ GRU

จากกราฟ ค่าความแม่นยำของโมเดล LSTM และ GRU มีค่าที่ใกล้เคียงกันอย่างชัดเจน แต่สิ่งที่โมเดล GRU ทำได้ดีกว่าคือเวลาที่ใช้ในการคาดการณ์ (Computation Time) ที่มีค่าลดลงดังรูปที่ 4.8 และรูปที่ 4.9 ตามลำดับ จากรูปเราสามารถสังเกตได้ว่าโมเดล GRU สามารถทำนายได้เร็วกว่าโมเดล LSTM อยู่ที่ 0.037 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
107725 function calls (103004 primitive calls) in 0.478 seconds
```

รูปที่ 4.8 เวลาที่ใช้ในการคาดการณ์ของโมเดล GRU

```
107697 function calls (103033 primitive calls) in 0.515 seconds
```

รูปที่ 4.9 เวลาที่ใช้ในการคาดการณ์ของโมเดล LSTM

4.3 ผลการทดลอง

หลังจากที่เราได้นำข้อมูลไปทดลองกับโมเดลทั้งสามแล้ว ซึ่งได้แก่ Long-Short Term Memory, Bi-Directional Long-Short Term Memory และ GRU เราได้นำโมเดลเหล่านี้ไปประยุกต์ใช้กับระบบเอสดีเอ็น (SDN) ซึ่งมีหลักการที่จะให้สวิตช์ทำการทำนายค่าการใช้ประโยชน์แบนด์วิดท์ที่จะเกิดขึ้นในแต่ละลิงค์ ค่าเมทริกซ์หรือตัวแปรที่เราได้นำมาใช้ในการประเมินประสิทธิภาพของเครือข่ายได้แก่ Delay, Throughput และ Packet loss ดังรูปที่ 4.10, รูปที่ 4.11 และ รูปที่ 4.12 ตามลำดับ

4.3.1 พารามิเตอร์ของการทดลอง

ในการทดลองนี้ ผู้จัดทำได้ทำการจำลองเครือข่ายสำหรับการส่งข้อมูลภายในเครื่องเสมือน (Virtual Machine) โดยทำการส่งข้อมูลตามพารามิเตอร์การทดลองดังที่แสดงในตารางที่ 4.3

ซอฟต์แวร์จำลองเครือข่าย	มินิเน็ต (Mininet)
จำนวนสวิตช์	7 เครื่อง
จำนวนโฮสท์ (PC)	8 เครื่อง
แบนด์วิดท์ของลิงค์	10 Mbps
เวอร์ชันโอเพนโฟลว์	Version 1.3
โพรโทคอลที่ใช้ในการทดลอง	Transmission Control Protocol (TCP)
เอสดีเอ็นคอนโทรลเลอร์	ริวเอสดีเอ็นคอนโทรลเลอร์ (RYU)

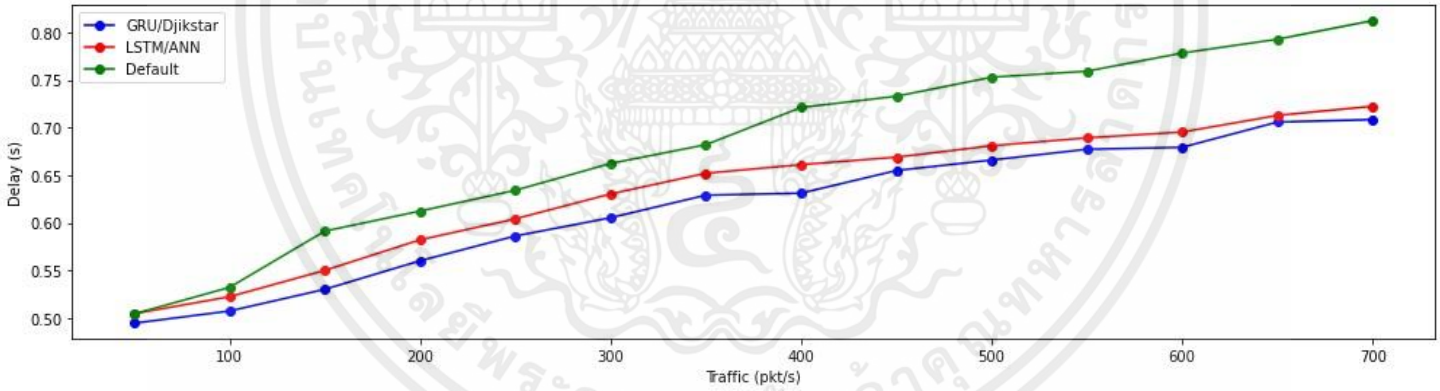
ตารางที่ 4.3 พารามิเตอร์ที่ใช้ในการจำลองเครือข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซอฟต์แวร์ที่เราได้ใช้ในการจำลองส่งข้อมูลภายในเครือข่ายจำลองได้แก่ ซอฟต์แวร์ D-ITG (Distributed Internet Traffic Generator) ซึ่งในการทดลองครั้งนี้ได้ทำการส่งข้อมูลจากโฮสท์ต้นทาง จำนวน 3 เครื่อง ได้แก่ Host1, Host2 และ Host3 รวมไปถึงโอสท์ปลายทางได้แก่ Host4, Host5, Host7 และ Host8 เราได้ทำการกำหนดพารามิเตอร์ในการส่งข้อมูลด้วยซอฟต์แวร์ D-ITG ดังแสดงในตารางที่ 4.4

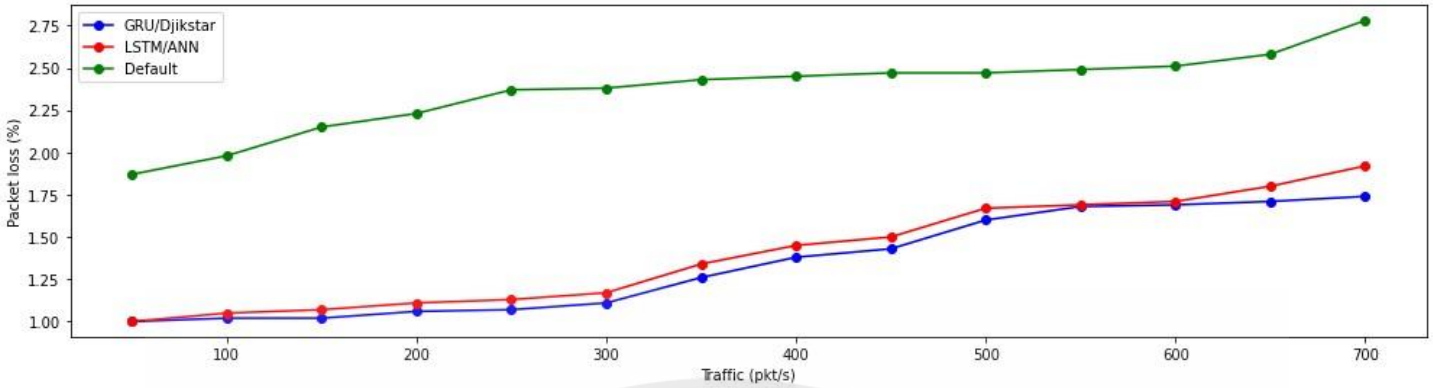
โฮสต์ต้นทาง	โฮสต์ปลายทาง	การแจกแจงแพ็กเก็ตข้อมูลแบบปัวส์ซอง	ขนาดของแพ็กเก็ต	โพรโทคอล	ระยะเวลาในการส่งข้อมูล
Host 1	Host 4,6	-O 200	-c 500	TCP	1 นาที
Host 2	Host 5,7	-O 200	-c 500	TCP	1 นาที
Host 3	Host 7,8	-O 200	-c 500	TCP	1 นาที

ตารางที่ 4.4 พารามิเตอร์ที่ใช้ในการทดลองส่งข้อมูลด้วยซอฟต์แวร์ D-ITG

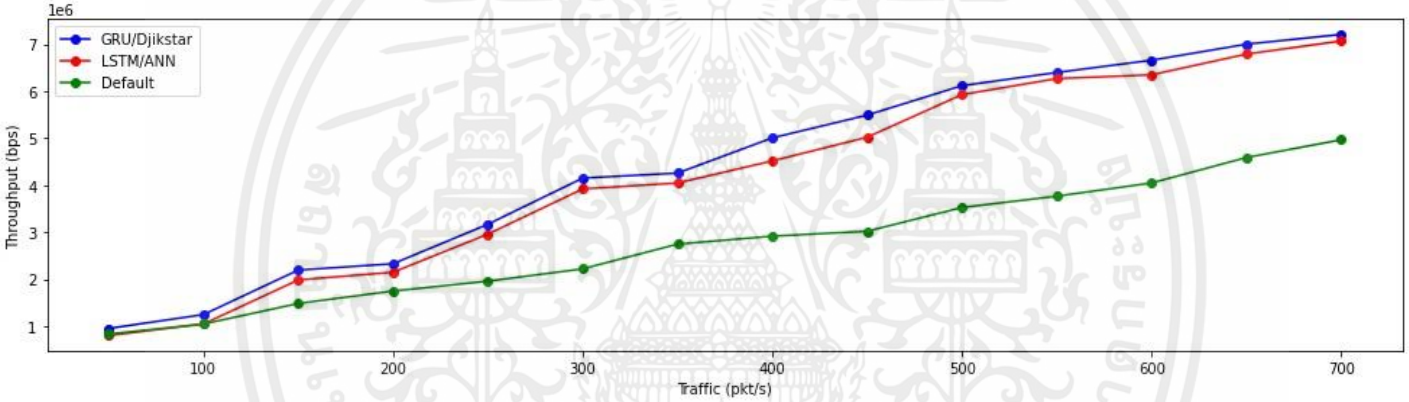


รูปที่ 4.10 การประเมินประสิทธิภาพเครือข่ายที่วัดจากดีเลย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.11 การประเมินประสิทธิภาพเครือข่ายที่วัดจากอัตราสูญหายของแพ็กเก็ต



รูปที่ 4.12 การประเมินประสิทธิภาพเครือข่ายที่วัดจากทรูพุต

จากกราฟทั้งสามรูปด้านบน เกิดจากการทดลองของการส่งข้อมูลแบบ 3 ประเภท ได้แก่ ประเภทที่หนึ่ง คือการส่งข้อมูลด้วยซอฟต์แวร์ D-ITG ทั่วไป กล่าวคือไม่มีการประยุกต์ใช้โมเดลการเรียนรู้เชิงลึกร่วมกับสวิตช์ในโทโพโลยี ประเภทที่สอง คือการส่งข้อมูลและมีการใช้อัลกอริทึม LSTM ในการคาดการณ์ค่าการใช้ประโยชน์แบนด์วิดท์และมีการจัดเปลี่ยนเส้นทางใหม่ด้วยโครงข่ายประสาทเทียม ประเภทสุดท้าย คือการส่งข้อมูลทั่วไปและมีการใช้อัลกอริทึม GRU ในคาดการณ์ค่าการใช้ประโยชน์แบนด์วิดท์แทน LSTM รวมไปถึงมีการใช้อัลกอริทึมไดก์สตราในการจัดเปลี่ยนเส้นทางใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กราฟอันดับแรก เป็นกราฟที่แสดงประสิทธิภาพที่ถูกประเมินในรูปแบบดีเลย์ (Delay) ส่วนกราฟอันดับที่สอง เป็นกราฟที่แสดงประสิทธิภาพที่ถูกประเมินในรูปแบบอัตราการสูญหายของแพ็กเก็ต (Packet Loss) และ กราฟสุดท้ายเป็นกราฟแสดงทรูพุต (Throughput) จะสังเกตได้ว่าการนำโมเดล GRU และอัลกอริทึมไดก์สตรามาใช้นั้นให้ประสิทธิภาพที่ดีที่สุด เนื่องจากดีเลย์ลดลงและค่าทรูพุตมีค่าที่เพิ่มขึ้น รวมไปถึงการลดระดับของอัตราการสูญหายของแพ็กเก็ต การคาดการณ์ค่าการใช้ประโยชน์แบนด์วิดท์สามารถช่วยให้สวิตช์ตัดสินใจว่าควรจะส่งแพ็กเก็ตหรือไม่ มากไปกว่านั้น วิธีการเช่นนี้สามารถช่วยลดระดับความคับคั่งในเครือข่าย และ ยับยั้งไม่ก่อให้เกิดการสูญหายของแพ็กเก็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผล

ในบทนี้ เราได้อธิบายเกี่ยวกับการสรุปผลของการทดลอง วิธีการเพิ่มประสิทธิภาพให้กับเครือข่ายที่กำหนดด้วยซอฟต์แวร์ด้วยการจัดการเส้นทางที่มีการคาดการณ์แบนด์วิดท์ล่วงหน้า ชี้แจงถึงปัญหาที่ได้เกิดขึ้นตลอดการทำโครงงานด้านวิจัย รวมไปถึงการเสนอแนะแนวทางในการพัฒนาระบบเอสดีเอ็นให้ดียิ่งขึ้นต่อไปในอนาคต

5.1 บทสรุป

โครงานนนี้ได้ทำการจำลองเครือข่ายด้วยเทคโนโลยีเครือข่ายที่กำหนดด้วยซอฟต์แวร์หรือเอสดีเอ็น และ วิธีการเพิ่มประสิทธิภาพของเครือข่ายโดยนำอัลกอริทึมการเรียนรู้เชิงลึก Gated Recurrent Unit (GRU) และ Long-Short Term Memory (LSTM) มาใช้คาดการณ์ค่าการใช้ประโยชน์แบนด์วิดท์ของลิงค์ที่จะเกิดขึ้นภายในอนาคต หากค่าที่คาดการณ์มานั้นมีค่ามากกว่าเทรชโฮลด์ (Threshold) ที่เรากำหนดไว้ ริวคอนโทรลเลอร์ (RYU Controller) จะสั่งให้สวิตช์ทำการเปลี่ยนเส้นทางในการส่งข้อมูลเพื่อหลีกเลี่ยงความคับคั่งของเครือข่าย (Network Congestion)

ในการทดลองของโมเดลเรียนรู้เชิงลึกระหว่างโมเดล LSTM และ GRU โมเดลทั้งสองโมเดลมีความสามารถในการทำนายข้อมูลได้แม่นยำใกล้เคียงกัน แต่โมเดล GRU ใช้เวลาในการคาดการณ์ได้เร็วกว่าโมเดล LSTM และยกระดับประสิทธิภาพของเครือข่ายด้วยการเพิ่มทรูพุต (Throughput) การลดดีเลย์ (Delay) รวมไปถึงการลดอัตราการสูญหายของแพ็กเก็ต (Packet Loss) ซึ่งเราสันนิษฐานว่าการลดเวลาในการคาดการณ์ข้อมูลที่จะเกิดขึ้นภายในอนาคต สามารถช่วยให้เครือข่ายสามารถจัดเส้นทางได้อย่างเหมาะสม และ หลีกเลี่ยงการเกิดความคับคั่งในที่สุด

โครงงานของเรามีการทำงานหลักอยู่ 3 ส่วน ได้แก่ การออกแบบเครือข่าย (Infrastructure Layer) ส่วนของริวคอนโทรลเลอร์ที่ประกอบไปด้วย 3 โมดูล (Control Layer) และ ส่วนของแอปพลิเคชั่นคาดการณ์ค่าการใช้ประโยชน์แบนด์วิดท์ (Application Layer)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.1 การออกแบบเครือข่าย (Infrastructure Layer)

1) มีการทดลองออกแบบครือข่ายในหลายรูปแบบ เช่น Linear Topology, Tree Topology, Star Topology โดยใช้ซอฟต์แวร์มินิเน็ต

2) การออกแบบเครือข่าย ต้องประกอบไปด้วยสวิตช์ที่มีเส้นทาง (Path) ในการส่งข้อมูลมากกว่า 1 เส้นทาง ถึงจะอยู่ในขอบเขตของโครงงาน

5.1.2 ริวคอนโทรลเลอร์ (Control Layer)

1) ได้มีการเก็บข้อมูลจาก OFPPortStatusReply เพื่อนำข้อมูลจากการจำลองไปทดสอบกับโมเดล Long-Short Term Memory (LSTM) โมเดล Bidirectional Long-Short Term Memory (Bi-LSTM) และ โมเดล Gated Recurrent Unit (GRU)

2) ประกอบไปด้วยโมดูลสังเกตการณ์เครือข่าย (Traffic Monitor Module) เพื่อตรวจสอบความคับคั่งภายในเครือข่ายแบบเรียลไทม์

5.1.3 โมเดลการเรียนรู้เชิงลึก (Deep Learning Training)

1) ข้อมูลที่ถูกนำมาเทรนระหว่างโมเดล Bi-LSTM, LSTM และ GRU แสดงให้เห็นว่า การคาดการณ์ของโมเดล Bi-LSTM แม่นยำกว่าการคาดการณ์ของโมเดล LSTM และ GRU ส่วนโมเดล GRU ใช้เวลาในการคาดการณ์เร็วที่สุดระหว่าง 3 โมเดล

2) หลังจากที่เรานำโมเดลการเรียนรู้เชิงลึกไปประกอบกับระบบเอสดีเอ็น การใช้โมเดล GRU ส่งผลให้ประสิทธิภาพของเครือข่ายเพิ่มขึ้นได้ดีที่สุดระหว่าง 3 โมเดล

5.2 ปัญหาและอุปสรรคที่พบ

5.2.1 ข้อจำกัดของเครื่องเสมือน

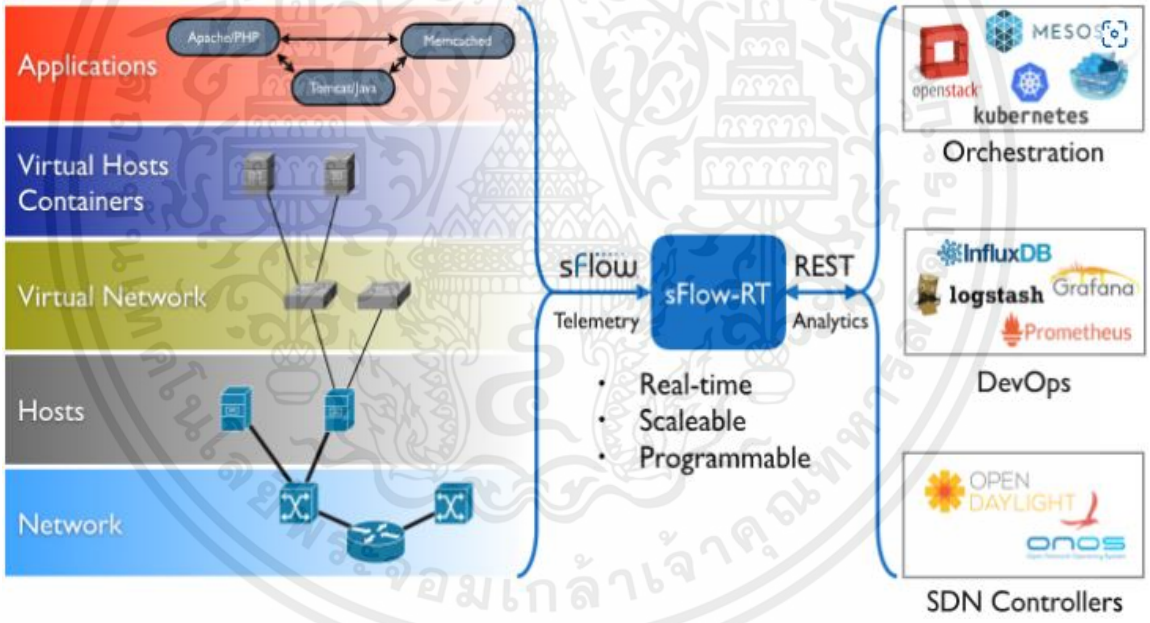
เราได้ทดลองใช้เครื่องเสมือน (Virtual Machine) สองโปรแกรม ได้แก่ Oracle VM VirtualBox และ VMware Workstation ซึ่งโปรแกรมทั้งสองนั้นมีข้อดีข้อเสียที่แตกต่างกัน แต่โปรแกรมที่เราได้ใช้สำหรับโครงงานนี้คือ Oracle VM VirtualBox เนื่องจาก VMware Workstation ไม่สามารถแสดงโมดูลสังเกตการณ์เครือข่ายแบบเรียลไทม์ได้ (Traffic Monitoring Module) แต่ในทางกลับกันโปรแกรม VMware WorkStation มีความเสถียรมากกว่าโปรแกรม Oracle VM VirtualBox ในหลายๆด้าน เช่น การติดตั้งแพ็กเก็จและการติดตั้งซอฟต์แวร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.2 ปัญหาของซอฟต์แวร์

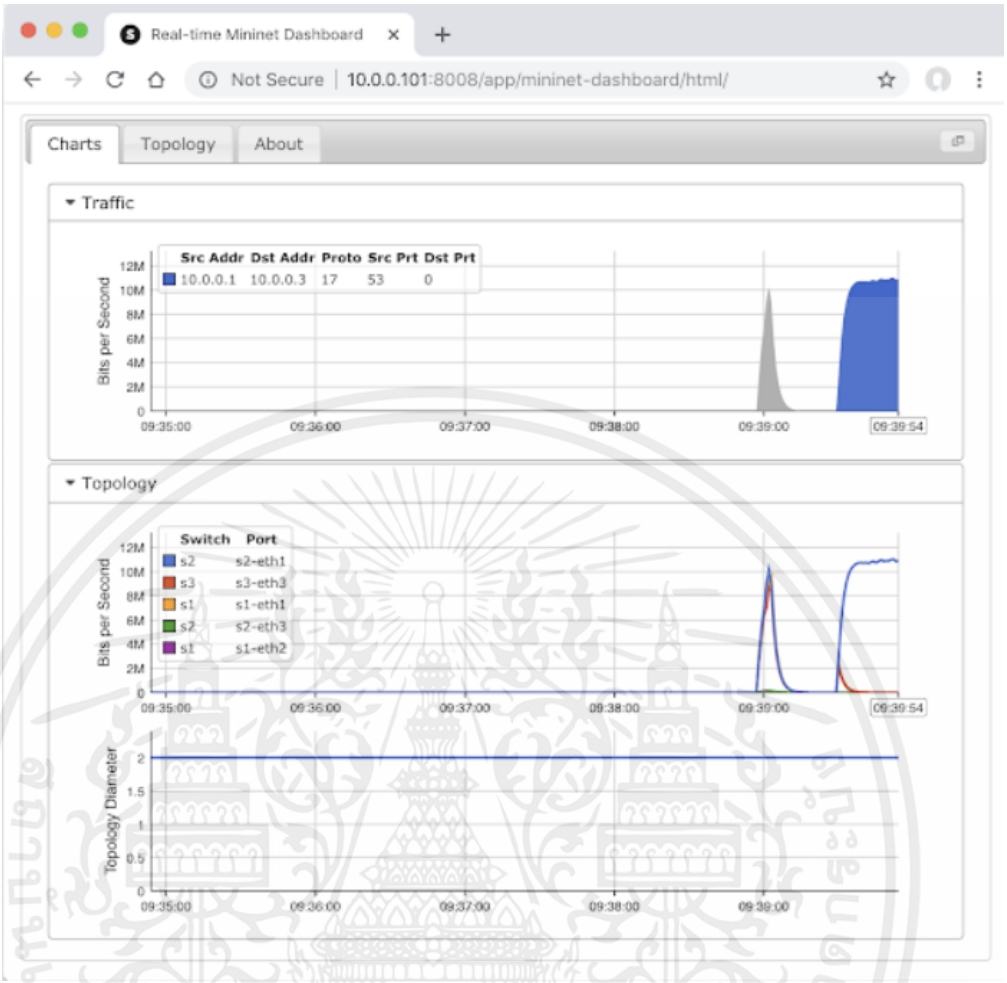
โมดูลสังเกตการณ์เครือข่ายในปัจจุบัน เป็นโมดูลอ่านค่ามาจากโพรโทคอลโอเพนโฟลว์ที่สวิตช์ได้สื่อสารกับริวคอนโทรลเลอร์ เมื่อสวิตช์ส่งข้อมูลให้กับคอนโทรลเลอร์ เราก็จะสามารถนำข้อมูลนั้นมาแสดงได้โดยแสดงผ่านเทอร์มินัล (Terminal)

Sflow-rt เป็นซอฟต์แวร์ที่มีความสามารถในการติดตามโฟลว์ (Tracking Flow) ตรวจสอบข้อมูลแต่ละลิงค์หรือพอร์ทได้อย่างละเอียด ข้อดีของ Sflow-rt ที่เหนือกว่านั้นได้แก่ การสร้างกราฟเพื่อแสดงสถิติได้แบบเรียลไทม์ ก็เปรียบเสมือนว่าเราสามารถตรวจสอบระบบเครือข่ายทั้งหมดแบบเรียลไทม์ได้และง่ายต่อการใช้งาน ปัญหาที่เกิดขึ้นคือ Sflow-rt ไม่สามารถนำมาใช้งานร่วมกับระบบได้



รูปที่ 5.1 องค์ประกอบของซอฟต์แวร์ sFlow-RT (blog.sflow.com)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.2 ตัวอย่างการตรวจสอบโฟลว์แบบเรียลไทม์ของซอฟต์แวร์ sFlow-RT (blog.sflow.com)

ซอฟต์แวร์ D-ITG (Distributed-Internet Traffic Generator) ที่เราได้นำมาใช้จำลองในการส่งขอมูลภายในเครือข่ายเอสดีเอ็น ซึ่งซอฟต์แวร์ตัวนี้ไม่สามารถส่งข้อมูลได้จำนวนมหาศาล ซึ่งสอดคล้องกับเทคโนโลยีฟาสท์อีเทอร์เน็ต (Fast Ethernet) แต่ในทางกลับกัน ถ้าหากมีการส่งข้อมูลที่เยอะขึ้น ก็อาจจะส่งผลให้คอมพิวเตอร์มีการทำงานหนักจนเกินความจำเป็น (CPU Utilization)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 แนวทางการพัฒนาต่อ

จากการทำโครงงานด้านวิจัยที่เกี่ยวกับกลไกการเพิ่มประสิทธิภาพในเครือข่ายที่กำหนดด้วยซอฟต์แวร์ เราได้เรียนรู้ข้อผิดพลาดและวางแผนถึงการพัฒนาต่อไปนี้

1) ในระหว่างการทำโครงงาน ช่วงของการค้นหาโมเดลสำหรับการนำมาทดลองเปรียบเทียบ เราได้พบโมเดล Fast Fourier Transform Long Short Term Memory (FFT/LSTM) ซึ่งจากที่เราทำการทดลองภายในโครงงานนี้ การนำโมเดล FFT มาใช้ควบคู่กับโมเดล LSTM ซึ่งอาจจะทำให้ความแม่นยำในการทำนาย รวมไปถึงความเร็วในการประมวลผลสูงขึ้น
2) นำอัลกอริทึมของการเรียนรู้เชิงลึกไปใช้กับอุปกรณ์เครือข่ายได้จริง ที่ไม่ได้เป็นเพียงการจำลองเครือข่าย
3) ภายในอนาคตอาจจะมีตัวแปรสำคัญที่ทำให้เครือข่ายมีประสิทธิภาพที่ดีขึ้นมากกว่าการใช้เวลาในการคาดการณ์ รวมไปถึงอาจจะมีโมเดลการเรียนรู้เชิงลึกที่ใช้แสดงประสิทธิภาพได้ดีกว่าโมเดล Gated Recurrent Units (GRU)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] Omran M. A. Alssaheli, Z. Zainal Abidin, N. A. Zakaria and Z. Abal Abas “Implementation of Network Traffic Monitoring Using Software Defined Networking Ryu Controller”

[2] Wucherl Yoo and Alex Sim “Time-series Forecast Modeling on High-Bandwidth Wide Area Network Measurements”

[3] Sima Siami-Namini, Neda Tavakoli and Akbar Siami Namin “The Performance of LSTM and BiLSTM in Forecasting Time Series”, IEEE, 2019

[4] Sowmya Sanagavarapu and Sashank Sridhar. “SDPredictNet-A Topology based SDN Neural Routing Framework with Traffic Prediction Analysis”, 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)

[5] S. Avallone, S. Guadagno, D. Emma and A. Pescape “D-ITG Distributed Internet Traffic Generator”

[6] Md. Tariqul Islam, Nazrul Islam and Md. AI Refat “Node to Node Performance Evaluation through RYU SDN Controller

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[7] Rui Fu, Zuo Zhang and Li Li “Using LSTM and GRU Neural Network Methods for Traffic Flow Prediction”, IEEE, 2016

[8] รู้จักกับ Software-defined Network (SDN) https://www.blognone.com/node/56144

[9] Deep Learning คืออะไร เข้าใจง่ายๆ https://blog.pttexpresso.com/get-to-know-deep-learning/

[10] Bandwidth คืออะไร มาทำความรู้จักแบนด์วิดท์กัน ฉบับเข้าใจง่าย https://personet.co.th/what-is-bandwidth/

[11] RNN vs GRU vs LSTM https://medium.com/analytics-vidhya/rnn-vs-gru-vs-lstm-863b0b7b1573

[12] [Tips&Tricks for TOI] EP.1: Dijkstra Shortest Path https://medium.com/@weeklycpproblems/tips-tricks-for-toi-ep-1-dijkstra-shortest-path-de30ed55c9a1

[13] The basics of SDN and the OpenFlow Network Architecture https://noviflow.com/the-basics-of-sdn-and-the-openflow-network-architecture/

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้