

การศึกษาการประยุกต์เทคโนโลยีการเรียนรู้เชิงลึกสำหรับการ จรรยาบรรณเครือข่ายเอสดีเอ็น

บุญญพัฒน์ แปลงพระเนตร¹ และ ศรัณญ์ รื่นรวย²

¹ คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง กรุงเทพฯ

² คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง กรุงเทพฯ

Emails: 61070122@it.kmitl.ac.th, 61070215@it.kmitl.ac.th

บทคัดย่อ

เอกสารนี้รวมระบบเครือข่ายในปัจจุบันมีการขยายตัวจนมีขนาดใหญ่ และมีความซับซ้อนมากขึ้นส่งผลให้การบริหารจัดการเป็นไปด้วยความยุ่งยาก และหนึ่งในปัญหาสำคัญคือความแออัดในระบบเครือข่ายการขยายตัวของผู้ใช้งานที่เพิ่มขึ้น ซึ่งระบบเครือข่ายแบบดั้งเดิมไม่สามารถแก้ไขปัญหาได้ งานวิจัยนี้จึงได้เสนอวิธีการทำนายปริมาณการใช้งานของลิงก์ภายในระบบเครือข่ายโดยใช้โมเดลการเรียนรู้เชิงลึก ผู้วิจัยได้ทำการประเมินผลและเปรียบเทียบโมเดลการเรียนรู้เชิงลึกทั้งหมดสามโมเดล ได้แก่ LSTM (Long Short-Term Memory), BiLSTM (Bidirectional Long Short-Term Memory) และ GRU (Gated Recurrent Unit) เพื่อที่จะทำนายปริมาณการใช้งานของลิงก์ จากการผลการทดลองชี้ให้เห็นว่า BiLSTM ให้ผลการทดลองได้ดีที่สุด โดยมีค่า RMSE เพียง 2.335% และเมื่อนำไปใช้กับระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์สามารถลดความแออัดได้ถึง 23.4%

คำสำคัญ – ระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์; การเรียนรู้เชิงลึก; การกระจายการจรรยาบรรณเครือข่าย;

1. บทนำ

ระบบเครือข่ายในปัจจุบันมีขนาดใหญ่และประกอบไปด้วยอุปกรณ์ที่มีความหลากหลายส่งผลให้ระบบเครือข่ายมีความซับซ้อนมากขึ้น หนึ่งในปัญหาสำคัญคืออุปกรณ์จะเลือกเส้นทางที่มีประสิทธิภาพภายใต้เครือข่ายที่มีความซับซ้อนได้อย่างไร การเลือกเส้นทางที่ไม่มีประสิทธิภาพประกอบกับการขยายตัวของผู้ใช้งานที่เพิ่มขึ้น อาจส่งผลให้เกิดความแออัดในเครือข่าย ซึ่งระบบเครือข่ายแบบดั้งเดิมไม่สามารถแก้ไขปัญหาได้ ปัจจุบันได้มีรูปแบบการจัดการระบบเครือข่ายที่มีประสิทธิภาพในการจัดการระบบเครือข่ายที่มีชื่อว่าระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์

เครือข่ายที่กำหนดโดยซอฟต์แวร์ เป็นการแยกการทำงานออกเป็น 2 ส่วน คือส่วนของการควบคุมและส่วนของข้อมูลซึ่งส่วนของการควบคุมจะอยู่ในส่วนที่เรียกว่าคอนโทรลเลอร์ โดยเครือข่ายที่กำหนดโดยซอฟต์แวร์สามารถแก้ปัญหาดังกล่าวได้ด้วยการกระจาย

การจรรยาบรรณจากการเลือกเส้นทางเพิ่มเติมที่เหมาะสม โดยใช้ความสามารถของตัวคอนโทรลเลอร์ที่สามารถมองภาพรวมทั้งหมดของเครือข่ายได้ แต่อย่างไรก็ตามวิธีการนี้จะแก้ไขได้ก็ต่อเมื่อคอนโทรลเลอร์ตรวจพบเท่านั้น การทำนายปริมาณการใช้งานของลิงก์จะช่วยให้แก้ไขปัญหานี้ในเชิงรุกได้

หนึ่งในตัวชี้วัดประสิทธิภาพของระบบเครือข่ายคือปริมาณการใช้งานของลิงก์ ถ้าหากมีปริมาณการใช้งานของลิงก์ที่สูงจะทำให้การสูญหายของแพ็คเก็ตและเวลาแฝงที่เพิ่มขึ้น

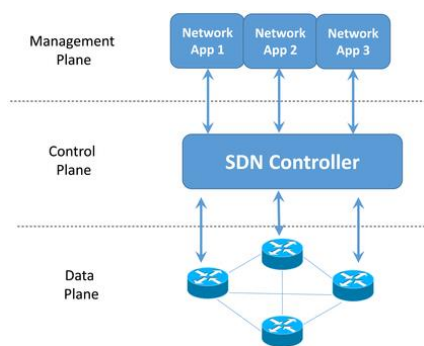
ในงานวิจัยนี้ผู้วิจัยได้เสนอการนำการเรียนรู้เชิงลึกมาใช้ในการทำนายปริมาณการใช้งานของลิงก์ในอนาคต โดยสร้างระบบเครือข่ายจำลองด้วยโปรแกรม Mininet และเก็บรวบรวมสถิติการใช้งานของลิงก์โดยคอนโทรลเลอร์เพื่อป้อนข้อมูลให้โมเดลการเรียนรู้เชิงลึก ซึ่งผู้วิจัยได้เลือกสามโมเดล ได้แก่ LSTM, BiLSTM, GRU เพื่อทดสอบและทำนายปริมาณการใช้งานของลิงก์

2. ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1. ระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์

(Software Defined Networks : SDN)

ระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์ [1] เป็นระบบเครือข่ายที่มีออกแบบมาเพื่อควบคุม และสั่งการจากส่วนกลางเพียงส่วนเดียว โดยแนวคิดของระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์จะทำการแยกส่วนของการควบคุมและส่วนของการส่งข้อมูลออกจากกันดังรูปที่ 1 ซึ่งส่วนของการควบคุมจะอยู่ในส่วนที่เรียกว่าคอนโทรลเลอร์ (Controller) และส่วนของการส่งข้อมูลจะอยู่ที่สวิตช์ (Switch) โดยที่ SDN นั้นสามารถแก้ปัญหาต่าง ๆ ของสถาปัตยกรรมเครือข่ายแบบเดิมได้ เพราะตัวคอนโทรลเลอร์มีซอฟต์แวร์ที่สามารถจัดการเครือข่ายทั้งหมดมารวมอยู่ในที่เดียว โดยคอนโทรลเลอร์จะสื่อสารกับสวิตช์โดยใช้โปรโตคอลมาตรฐานที่มีชื่อว่า OpenFlow



รูปที่ 1. สถาปัตยกรรมเครือข่ายที่กำหนดโดยซอฟต์แวร์

2.2. โอเพนวิสวิตช์

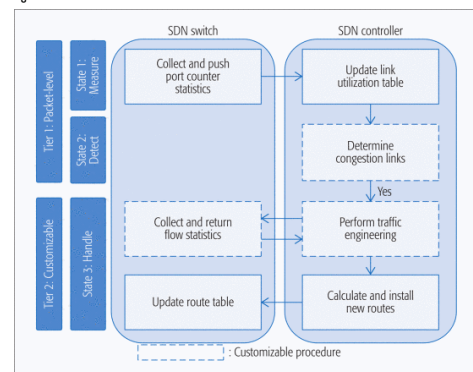
โอเพนวิสวิตช์ (Open vSwitch) เป็นเทคโนโลยีที่จำลองการทำงานของสวิตช์แบบมัลติเลเยอร์ (Multilayer Switch) โดยถูกออกแบบมาให้สามารถจัดการเครือข่ายแบบอัตโนมัติผ่านการเขียนโปรแกรม และยังคงซัพพอร์ตโปรโตคอลมาตรฐาน อื่น ๆ เช่น NetFlow, sFlow, LACP

2.3. การจัดการจราจรบนเครือข่าย

การจัดการจราจรบนเครือข่ายหมายถึงการคงไว้ซึ่งความพร้อมใช้งานของระบบเครือข่าย แนวคิดวิศวกรรมจราจรบนเครือข่ายกำหนดด้วยซอฟต์แวร์ [2] เป็นแนวคิดที่ออกแบบ และควบคุมการจราจรของข้อมูลในเครือข่าย

เพื่อให้การจราจรในเครือข่ายนั้นสามารถกระจายการทำงานได้อย่างสมดุล และมีประสิทธิภาพ โดยส่วนใหญ่จะเกี่ยวข้องกับการกระจายการส่งข้อมูลผ่านเส้นทางหลายเส้นทางด้วยการใช้อัลกอริทึมต่าง ๆ เพื่อแบ่งปริมาณการจราจรอย่างเหมาะสม

แนวคิด Two-tier mechanism [3] เป็นแนวคิดการออกแบบระบบจัดการจราจรรูปแบบหนึ่งที่จะแบ่งการทำงานเองเอสดีเอ็นคอนโทรลเลอร์ออกเป็นสองส่วน โดยมีส่วนที่ตรวจสอบสถานะการจราจร และส่วนที่เป็นกลไกการวัดและส่วนของการจัดการเครือข่าย ดังแสดงในรูปที่ 2



รูปที่ 2. Two-Tier Mechanism

2.4. การเรียนรู้เชิงลึก (Deep Learning)

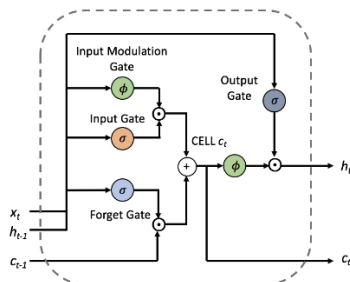
การเรียนรู้เชิงลึก เป็นวิธีหนึ่งของการเรียนรู้ของเครื่องซึ่งพัฒนามาจากโครงข่ายประสาทเทียม (Artificial Neural Network : ANN) เป็นการจำลองรูปแบบการประมวลผลของสมองมนุษย์ โดยมีโครงสร้างแบบลำดับชั้นเชื่อมต่อชั้นต่าง ๆ ผลลัพธ์ของชั้นก่อนหน้าจะกลายเป็นข้อมูลนำเข้าของชั้นถัดไป จากโครงสร้างสถาปัตยกรรมข้อมูลดังรูปที่ 2.3 ซึ่งจะประกอบไปด้วยโครงข่ายย่อย ๆ หลายชั้นสามารถใช้งานได้หลากหลายรูปแบบ เช่น การประมวลผลภาพ การรู้จำใบหน้า หรือการประมวลผลภาษาธรรมชาติ

2.5. โครงข่ายความจำระยะสั้นขนาดยาว (Long-Short Term Memory : LSTM)

โครงข่ายความจำระยะสั้นขนาดยาวเป็นส่วนหนึ่งของการเรียนรู้เชิงลึกที่สร้างขึ้นเพื่อแก้ปัญหาการลืมข้อมูลของ RNN ในกรณีที่มีข้อมูลนำเข้ายาวมาก ทำให้การส่งค่าผลลัพธ์ฟิเจอร์ไม่สามารถที่จะเก็บรักษาข้อมูลในช่วงเวลาที่ผ่านมาได้ทั้งหมด เอกลักษณะสำคัญ ของ

โครงข่ายชนิดนี้ คือ เซลล์ความจำ (Cell หรือ Memory cell) มีหน้าที่เก็บข้อมูลขาเข้าที่สำคัญเอาไว้ ไม่ให้เลือนหายไปในช่วงระยะเวลาหนึ่ง เป็นการแก้ปัญหาการพึ่งพาระยะยาว ซึ่งเหมาะสำหรับการพยากรณ์ปริมาณการใช้งานเครือข่ายเอสดีเอ็นทีที่มีชุดข้อมูลปริมาณมาก หรือพยากรณ์ปริมาณการใช้แบนด์วิธในเครือข่าย [4]

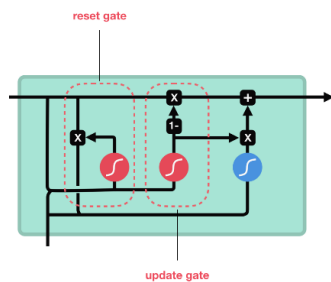
LSTM ประกอบด้วยหน่วยความจำ (Cell), ประตูสัญญาณรับเข้า (Input Gate), ประตูสัญญาณสำหรับกันลืม (forget gate), ประตูสัญญาณสำหรับผลลัพธ์ (Output Gate) และผลคูณแบบฮาดามาร์ด (Hadamard Product) ดังแสดงในรูปที่ 3



รูปที่ 3. โครงสร้างของหน่วยความจำระยะยาว

2.6. หน่วยเวียนกลับแบบมีประตู (Gated Recurrent Unit: GRU)

ถูกพัฒนาต่อยอดมา จาก LSTM ซึ่งพัฒนาในส่วนของการลดความซับซ้อนในการทำงานของโครงข่ายประสาทแบบ LSTM เนื่องจากจำนวนหน่วยย่อยใน Cell จำนวนมาก ซึ่งมีผลต่อประสิทธิภาพในการวิเคราะห์และทำนายผล โดย GRU ได้ทำการลดความซับซ้อนในการทำงานของโครงข่ายดังรูปที่ 4 ประสาทแบบ LSTM โดยการลดหน่วยย่อยใน Cell เหลือเพียง 2 ส่วน ได้แก่ Update Gate และ Reset Gate

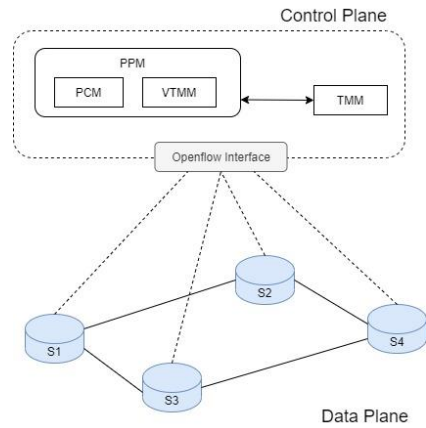


รูปที่ 4. โครงสร้างของหน่วยเวียนกลับแบบมีประตู

3. วิธีการดำเนินการวิจัย

3.1. การปรับปรุงองค์ประกอบภายในคอนโทรลเลอร์

การปรับปรุงองค์ประกอบภายในของคอนโทรลเลอร์ จะนำองค์ประกอบมาตรฐานภายในคอนโทรลเลอร์ของ RYU Controller มาปรับปรุงและเพิ่มเติม โดยที่องค์ประกอบมาตรฐานภายในคอนโทรลเลอร์ประกอบไปด้วยส่วนต่าง ๆ ดังรูปที่ 5



รูปที่ 5. องค์ประกอบภายในคอนโทรลเลอร์

3.1.1. Path Provision Module (PPM)

ทำหน้าที่จัดการข้อความจากโพรโทคอลทั้งหมดจากองค์ประกอบภายในคอนโทรลเลอร์ โดยภายในจะมีองค์ประกอบย่อยคือ Path Computation Module (PCM) เมื่อ PPM ได้รับแพ็คเกจมาจากสวิตช์ PPM จะทำการส่งต่อข้อความไปยัง PCM เพื่อคำนวณหาเส้นทางไปยังปลายทางต่อไป

3.1.2. Path Computation Module (PCM)

ทำหน้าที่ในการหาเส้นทางที่มีระยะที่สั้นที่สุดและมีแบนด์วิธมากที่สุดในการส่งข้อมูล โดยใช้ข้อมูลโทโพโลยีที่ได้จาก Virtual Topology Management Module (VTMM) จากนั้นจะส่งผลการคำนวณหาเส้นทางกลับไปยัง PPM และทำการสร้าง FlowMod Message เพื่อใช้ในการติดตั้งกฎการส่งออก จากนั้นจะทำการติดตั้งกฎการส่งออกไปยังทุกสวิตช์ที่อยู่ในเส้นทางที่ได้คำนวณไว้

3.1.3. Virtual Topology Management (VTMM)

ทำหน้าที่ในการหาเส้นทางที่มีระยะที่สั้นที่สุดและมีแบนด์วิธมากที่สุดในการส่งข้อมูล โดยใช้ข้อมูลโทโพโลยีที่ได้จาก Virtual Topology Management

3.1.4. Flow Monitoring Module (FMM)

ทำหน้าที่ในการทำนายปริมาณการใช้งานของลิงก์ในทุก ๆ ช่วงเวลาและใช้เป็นเงื่อนไขในการลบสวิตช์ที่เกิดความแออัด เมื่อโมเดลทำนายว่ามีปริมาณการใช้งานของลิงก์เกินกว่า 70 เปอร์เซ็นต์

3.2. อัลกอริทึมในการหาเส้นทางสำหรับการย้ายโฟลว์บนระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์

สำหรับอัลกอริทึมในโครงงานนี้ จะใช้อัลกอริทึมในการหาเส้นทางสำหรับการย้ายโฟลว์โดยพิจารณาจากขนาดของโฟลว์และความพร้อมใช้ของแบนด์วิธที่คงเหลือต่ำสุดของเส้นทางนั้น โดยอาศัยข้อมูลหลายอย่างในการคำนวณ ในรูปแบบ $P_{A \rightarrow B}$ คือเซตของเส้นทางทั้งหมดจาก A ไป B, e_i คือลิงก์ของเส้นทางนี้, c_i คือ ค่าใช้จ่ายของลิงก์นั้น ๆ, b_i คือ ค่าปริมาณการใช้งานของลิงก์ และ a_i คือ ค่าความจุคงเหลือที่พร้อมใช้ของลิงก์นั้น ๆ อันดับแรกต้องทำการคำนวณหาปริมาณการใช้งานของลิงก์ในแต่ละลิงก์จากสมการที่ 1 โดยที่ n_i คือ ค่าปริมาณการส่งผ่าน และ T คือ เวลาในการเรียกตรวจสอบในแต่ละครั้ง

$$b_i = \frac{n_i(t) - n_i(t-T)}{T} \quad (1)$$

เมื่อได้ค่าปริมาณการใช้งานของลิงก์แล้ว ระบบนำข้อมูลเข้าสู่โมเดลเพื่อทำนายปริมาณการใช้งานในอนาคตและนำค่าไปปรับปรุงโทโพโลยีจำลองและ PCM จะนำไปหาเส้นทางสำหรับการย้ายโฟลว์โดยพิจารณาตามลำดับขนาดของโฟลว์ จากนั้นจะต้องนำไปตรวจสอบว่าเส้นทางที่คำนวณออกมาจะสามารถย้ายโฟลว์ไปที่เส้นทางนั้นได้หรือไม่โดยไม่ก่อให้เกิดความแออัดซ้ำซ้อนซึ่งต้องคำนวณหาปริมาณแบนด์วิธที่คงเหลือต่ำสุดที่พร้อมใช้งานของเส้นทางนั้น ในขั้นแรกต้องทำการหาเส้นทางที่ดีที่สุดสำหรับโฟลว์นั้นก่อนซึ่งสามารถหาได้จากสมการที่ 2 โดยที่ BP คือ เส้นทางที่ดีที่สุดสำหรับโฟลว์นั้น ๆ

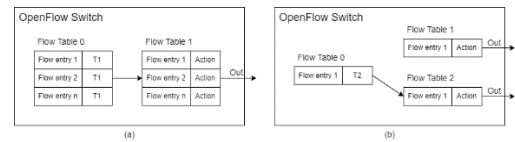
$$BP = \min_{P \in P_{A \rightarrow B}} \sum_{e_i \in P} c_i \quad (2)$$

เมื่อได้เส้นทางที่ดีที่สุดแล้ว จากนั้นจะหาปริมาณแบนด์วิธที่คงเหลือที่ต่ำที่สุดจากทุกลิงก์ของเส้นทางนั้น [6] จากสมการที่ 3 โดยที่ ABW คือ ค่าแบนด์วิธต่ำสุดที่พร้อมใช้งานของเส้นทาง

$$ABW = \min_{e_i \in BP} a_i \quad (3)$$

เมื่อได้ค่าแบนด์วิธต่ำสุดที่พร้อมใช้งานแล้ว จากนั้นจะนำมาเปรียบเทียบกับขนาดของโฟลว์ที่จะทำการย้ายการเลือกเส้นทางที่เหมาะสมที่สุดโดยพิจารณาจากค่าแบนด์วิธต่ำสุดที่พร้อมใช้งานและขนาดของโฟลว์

โดยปกติสวิตช์จะสามารถส่งข้อมูลไปได้โดยอาศัยข้อมูลจาก Flow Table กล่าวคือเมื่อมีโฟลว์หรือแพ็คเกจถูกส่งมาที่สวิตช์ สวิตช์จะนำมาเปรียบเทียบกับข้อมูลใน Flow Table ถ้าหากว่ามีข้อมูลที่ตรงกันก็จะส่งออกตามพอร์ตที่กำหนด แต่ถ้าหากไม่มีข้อมูลที่ตรงกันสวิตช์จะร้องขอข้อมูลเส้นทางจากคอนโทรลเลอร์ สถานะปกติสวิตช์จะ Flow Table ทั้งหมด 2 ตารางดังรูปที่ 6 (a) โดยตารางที่ 0 จะส่งโฟลว์ไปตรวจสอบกับตารางที่ 1 เพื่อหาว่าข้อมูลสำหรับการส่งออก แต่ถ้าหากมีการย้ายเส้นทางเกิดขึ้นคอนโทรลเลอร์จะสร้างตารางที่ 2 ที่มีข้อมูลเส้นทางใหม่ให้กับสวิตช์ที่อยู่ในเส้นทางใหม่และเปลี่ยนให้ตารางที่ 0 ส่งโฟลว์มาที่ตารางที่ 2 ดังรูปที่ 6 (b) ถ้าหากโฟลว์ที่ถูกย้ายเส้นทางส่งข้อมูลครบแล้วคอนโทรลเลอร์จะเปลี่ยนให้ตารางที่ 0 ส่งโฟลว์ให้กับตารางที่ 1 เพื่อกลับมาใช้เส้นทางเดิม



รูปที่ 6. Flow Table ภายในสวิตช์

3.3. ข้อมูลที่ใช้ในงานวิจัย

ข้อมูลที่ใช้ที่ใช้ในการวิจัยจะเป็นข้อมูลประเภทอนุกรมเวลาที่ได้มาจากการเก็บค่าปริมาณการใช้งานของแต่ละลิงก์ในทุกช่วงเวลา โดยจะเก็บข้อมูลจากการสร้างการจราจรในเครือข่ายที่มีการผสมระหว่างรูปแบบการจราจรที่มีการบังคับให้เกิดความแออัดและรูปแบบการจราจรแบบสุ่ม ซึ่งจัดเก็บในรูปแบบไฟล์นามสกุล csv

3.4. การจัดการข้อมูล

ข้อมูลที่ได้มามีจำนวนไม่มากนัก แต่มีความผิดพลาดอยู่ จึงจำเป็นต้องทำการตรวจสอบ แก้ไข หรือลบเพื่อให้รายการข้อมูลที่ไม่ถูกต้องออกไปจากชุดข้อมูลจึงจะนำไปใช้ประโยชน์ได้อย่างไม่เกิดความผิดพลาดความสับสน

หลังจากทำความสะอาดข้อมูลเสร็จแล้ว จากนั้นจะต้องทำการปรับขอบเขตของข้อมูล (Feature

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (4)$$

Diagram illustrating the sliding window mechanism for feature extraction. The input features are processed sequentially, with a sliding window of size 5 applied to extract features and generate labels. The labels are 1, 1, and 1.

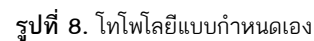
3.5. การทดสอบโมเดลการเรียนรู้เชิงลึก

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

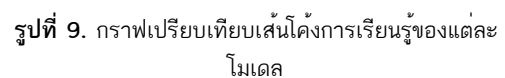
Hyperparameter	LSTM	BiLSTM	GRU
Activation Function	ReLu	ReLu	ReLu
Optimizer	Adam	Adam	Adam
Learning rate	0.001	0.001	0.001
Dropout rate	0.2	0.2	0.2
Batch size	16	16	16
Number of neurons	100	75	100
Epoch	36	27	30

4. การทดลองและผลการทดลอง

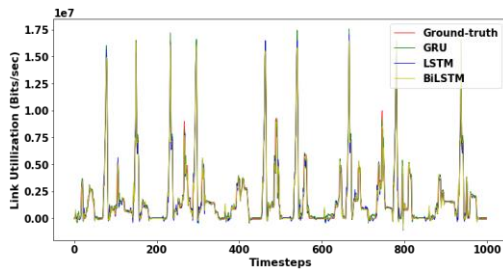
การทดลองนี้นำคอนโทรลเลอร์ที่ได้ออกแบบไว้ในบทที่ 3 มาทดลองและวัดประสิทธิภาพโดยสร้างระบบเครือข่ายเสมือนจริงจากโปรแกรม Mininet ประกอบไปด้วย Open vSwitch จำนวน 5 ตัว และ PC จำนวน 4 ตัว ดังรูปที่ 8 แต่ละเครื่องจะสุมสลับการรับ-ส่งข้อมูลโดยใช้โปรแกรม iPerf [5] สร้างแพ็คเก็ตแบบ UDP ส่งเข้าไปในเครือข่ายเพื่อสร้างการจราจรแบบสุมในระบบเป็นเวลาหนึ่งชั่วโมง



จากการทดลองสามารถแสดงผลได้ดังรูปที่ 9 โดยจะเห็นได้ว่าโมเดลสามารถเรียนรู้ได้ดีโดยไม่เกิดปัญหา Overfitting หรือ Underfitting



ในส่วนของการเปรียบเทียบค่าจริงและค่าทำนายสามารถแสดงผลเป็นกราฟได้ดังรูปที่ 10 โดยเลือกใช้ข้อมูลของสวิตช์ที่ 1 พอร์ตที่ 2 ในการเปรียบเทียบ



รูปที่ 10 กราฟเปรียบเทียบค่าจริงและค่าทำนายของแต่ละโมเดล

เมื่อได้ผลการทำนายแล้วจากนั้นจะต้องนำมาประเมินประสิทธิภาพของโมเดล โดยผู้วิจัยได้เลือก 4 ตัวชี้วัดพื้นฐานสำหรับโมเดลพยากรณ์อนุกรมเวลา ได้แก่ Mean Absolute Error (MAE), Mean Squared Error (MSE) และ Root Mean Squared Error (RMSE) เพื่อเปรียบเทียบความแม่นยำของแต่ละโมเดลดังแสดงในตารางที่ 2 จากผลลัพธ์แสดงให้เห็นว่า BiLSTM มีประสิทธิภาพมากที่สุด

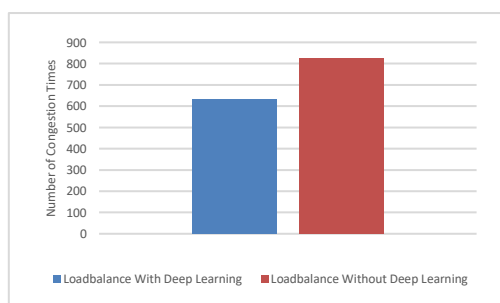
ตารางที่ 2. ตารางเปรียบเทียบความคลาดเคลื่อนของแต่ละโมเดล

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

4.3. ผลการทดลองการทำงานของระบบ

ในการวัดประสิทธิภาพการทำงานของระบบที่พัฒนาในโครงงานนี้บนโทโพโลยีที่กำหนดจะใช้ตัวชี้วัดประสิทธิภาพการทำงานคือ จำนวนครั้งในการเกิดความแออัด

จำนวนครั้งในการเกิดความแออัดในเครือข่ายที่กำหนดด้วยซอฟต์แวร์โดยใช้คอนโทรลเลอร์ที่พัฒนาขึ้นสามารถแสดงได้ดังภาพที่ 11



รูปที่ 11 กราฟเปรียบเทียบแสดงจำนวนการเกิดความแออัดในระยะเวลาหนึ่งชั่วโมง

หลังจากได้นำโมเดลการเรียนรู้เชิงลึก จำนวนครั้งในการเกิดความแออัดมีจำนวนที่ลดลง ซึ่งสอดคล้องกับรูปที่ 11 โดยหมายความว่าคอนโทรลเลอร์สามารถกระจายการจราจรก่อนจะเกิดความแออัดในเครือข่ายได้อย่างมีประสิทธิภาพ

5. สรุปผลการทดลอง

5.1 สรุปผล

ในปัจจุบันระบบเครือข่ายในปัจจุบันได้มีการขยายตัวจนมีขนาดใหญ่ การจัดการปัญหาความแออัดแบบเดิมตามกลไกของระบบเครือข่ายนั้นยังไม่สามารถจัดการได้มีประสิทธิภาพดีพอ ดังนั้นโครงงานนี้จึงได้พัฒนาคอนโทรลเลอร์ของระบบเครือข่ายที่กำหนดโดยซอฟต์แวร์ที่สามารถกระจายการจราจรในเครือข่ายก่อนเกิดความแออัด โดยใช้การเรียนรู้เชิงลึกและอัลกอริทึมการหาเส้นทางสำหรับการย้ายโฟลว์ ซึ่งเส้นทางที่สารลใช้งานได้นั้น จะต้องสอดคล้องกับเกณฑ์ที่กำหนดไว้ โดยผลลัพธ์ที่ได้ทำการทดลองแสดงให้เห็นว่าคอนโทรลเลอร์สามารถกระจายการจราจรก่อนที่เกิดความแออัดได้อย่างมีประสิทธิภาพบนโทโพโลยีที่กำหนด โดยสามารถลดได้ถึง 23.04 เปอร์เซ็นต์

5.2 แนวทางในการพัฒนาต่อ

คอนโทรลเลอร์ที่พัฒนาในโครงงานนี้สามารถทำนายปริมาณการใช้งานของลิงก์ภายใต้โทโพโลยีและรูปแบบการจราจรที่กำหนดไว้แล้วเท่านั้น การพัฒนาโครงงานในอนาคตจะพัฒนาเทคโนโลยีการเรียนรู้เชิงลึกให้สามารถพยากรณ์ปริมาณการใช้งานของลิงก์ให้ใช้ได้กับโทโพโลยีและรูปแบบการจราจรที่หลากหลาย

เอกสารอ้างอิง

- [1] Xenofon Foukas, Mahesh K. Marina, Kimon Kontovasilis, "Software Defined Networking Concepts," The University of Edinburgh & NCSR "Demokritos.
- [2] I Zhaogang Shu, Jiafu Wan, Jiaxiang Lin, Shiyong Wang, Di Li, Seungmin Rho, Changcai Yang, "Traffic Engineering in Software-defined Networking: Measurement and Management," IEEE Access, vol. 4, pp.3242-3256 (2016).

- [3] Ming-Hung Chen, Yen-Chen Tien, Yuan-Ting Huang, "A Low-Latency Two-Tier Measurement and Control Platform for Commodity SDN," IEEE Communications Magazine, pp. 0163-6804 September (2016).
- [4] Maxime Labonne; Charalampos Chatzinakis; Alexis Olivereau, "Predicting Bandwidth Utilization on Network Links Using Machine Learning," 2020 European Conference on Networks and Communications (EuCNC), September (2020).
- [5] "iPerf - iPerf3 and iPerf2 user documentation." [Online]. Available from: <https://iperf.fr>
- [6] Péter Megyesi, Alessio Botta, Giuseppe Aceto, Antonio Pescapè, Sándor Molnár, "Available Bandwidth Measurement in Software Defined Network," 31st ACM/SIGAPP Symposium on Applied Computing, April (2016)