การเพิ่มประสิทธิภาพเครือข่ายเซนเซอร์ไร้สายโดยใช้การเรียนรู้แบบเสริมกำลัง

อาณัฐ บุญเลิศ

คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง กรุงเทพฯ

Email: arnutboonlert@gmail.com

บทคัดย่อ

เครือข่ายเซนเซอร์ไร้สายคือกลุ่มของเซนเซอร์ไร้สายที่มีส่วนประกอบ คือ แหล่งพลังงานขนาดเล็ก (แบตเตอรี่) ตัวรับส่งข้อมูลและตัวประมวลผล ซึ่งเซนเซอร์หล่านี้จะถูกติดตั้งในพื้นที่เพื่อเก็บข้อมูลต่างๆ เช่น แสง อุณหภูมิ ความชื้น เป็นต้น เมื่อเก็บข้อมูลแล้วเซนเซอร์จะส่งข้อมูลเหล่านั้นกลับมาที่สถานีฐาน (Base station, Sink) แต่ด้วยระยะการสื่อสารที่จำกัด ทำให้เซนเซอร์บางตัวไม่สามารถส่งข้อมูลไปหาสถานีฐานได้โดยตรง เซนเซอร์เหล่านั้นจำเป็นต้องส่งข้อมูลโดยใช้การติดต่อสื่อแบบหลายทอด (Multi hop communication) ซึ่งต้องใช้กลไกการหาเส้นทาง (Routing) เพื่อทำให้ส่งข้อมูลจากต้นทางไปให้ถึงปลายทางได้สำเร็จ ด้วยการสื่อสารรูปแบบนี้ทำให้เซนเซอร์ตัวที่อยู่ติดกับสถานีฐานนั้นต้องใช้พลังงานมากเพื่อส่งต่อข้อมูลให้ตัวอื่นๆ ทำให้เซนเซอร์เหล่านี้จะมีพลังงานน้อยเมื่อเทียบกับตัวอื่นและทำให้พลังงานหมดไปก่อนเซนเซอร์ตัวอื่นๆ ด้วยข้อจำกัดนี้ทำให้เกิดการศึกษามากมายเพื่อปรับปรุงการใช้พลังงานในเครือข่ายเซนเซอร์ไร้สาย

โครงงานนี้มุ่งเน้นไปที่การศึกษาการใช้พลังงานอย่างมีประสิทธิภาพผ่านกลไก routing โดยการนำการเรียนรู้แบบเสริมกำลัง (Reinforcement learning, RL) เข้ามาช่วยเพื่อตัดสินใจเลือกเส้นทางที่เหมาะสมกับสภาพของเครือข่าย ทำให้เกิดการสมดุลพลังงานระหว่างเซนเซอร์เพื่อยืดอายุการใช้งานเครือข่ายให้นานขึ้น นอกนี้ยังได้นำ Software-Defined Networking (SDN) มาใช้ในเครือข่ายเซนเซอร์ไร้สายเพื่อแยกระหว่าง Control plane คือ ข้อมูลที่จำเป็นที่ใช้สำหรับอัลกอริทึม RL หรือแอพลิเคชันอื่นๆ และ Data plane คือ ข้อมูลที่เซนเซอร์เพื่อใช้สำหรับการวิเคราะห์หรือเป้าหมายอื่นๆ ในการเก็บข้อมูล

คำสำคัญ – เครือข่ายเซนเซอร์ไร้สาย; การเรียนรู้แบบเสริมกำลัง; กลไกการหาเส้นทาง

1. บทนำ

การใช้งานเครือข่ายเซนเซอร์ไร้สายเป็นที่นิยมมากขึ้นทุกวัน เนื่องจากการใช้งานที่มีประสิทธิภาพ เช่น การตรวจสอบด้านสิ่งแวดล้อม เมืองอัจฉริยะ ระบบอัตโนมัติทางอุตสาหกรรม และการดูแลสุขภาพ โดยไม่ต้องเชื่อมต่อสาย ซึ่งในเครือข่ายจะมีเซนเซอร์ไร้สายที่อาจจะอยู่กับที่หรือสามารถเคลื่อนที่ได้ ที่มีแหล่งพลังงานขนาดเล็ก (แบตเตอรี่) ตัวรับส่งข้อมูลและตัวประมวลผล เซนเซอร์เหล่านี้จะกระจายตัวอยู่ในพื้นที่ที่กำหนดเพื่อเก็บข้อมูลและส่งข้อมูลเหล่านั้นกลับมาที่สถานีฐานตามรูปที่ 1 แต่อย่างไรก็ตามการทำงานกับเครือข่ายเซนเซอร์ไร้สายก็ยังมีข้อจำกัดหลายอย่าง และเนื่องด้วยความนิยมที่เพิ่มขึ้น ทำให้เกิดการศึกษาเกี่ยวกับเครือข่ายเซนเซอร์ไร้สายในหลายๆ ด้าน หนึ่งในด้านที่เป็นประเด็นสำคัญคือการใช้พลังงาน ซึ่งจะส่งผลต่ออายุการใช้งานของเครือข่าย เนื่องจากเซนเซอร์ต้องทำงานในพื้นที่เพื่อเก็บข้อมูลเป็นเวลานานและการเข้าถึงเซนเซอร์เหล่านั้นเพื่อเปลี่ยนแบตเตอรี่สามารถทำได้ยากหรืออาจจะไม่คุ้มค่าเมื่อเทียบกับการทำให้เซนเซอร์สามารถใช้พลังงานได้อย่างมีประสิทธิภาพ

กลไกการหาเส้นทางเป็นส่วนสำคัญในการทำงานของเครือข่ายเซนเซอร์ไร้สาย เป้าหมายเพื่อส่งข้อมูลจากต้นทาง (เซนเซอร์ไร้สาย) ไปให้ถึงปลายทาง (สถานีฐาน) และเนื่องจากเครือข่ายเซนเซอร์ไร้สายนั้นเป็นเครือข่ายแบบเฉพาะกิจ (Ad hoc network) ซึ่งไม่มีอุปกรณ์เครือข่ายมาช่วยทำงานระหว่างกลาง ทำให้เซนเซอร์ไร้สายบางตัว ต้องหาเส้นทางเพื่อส่งข้อมูลไปให้ถึงปลายทางผ่านการส่งต่อข้อมูลให้เซนเซอร์ตัวอื่นๆ แบบหลายทอด ต้นเหตุเกิดจาดข้อจำกัดด้านการสื่อสารของเซนเซอร์ไร้สาย คือ ระยะการสื่อสารของเซนเซอร์นั้นมีจำกัด เซนเซอร์ตัวที่อยู่ติดกับสถานีฐานจะส่งข้อมูลไปหาสถานีฐานโดยตรงหรือเป็นการสื่อสารแบบทอดเดียว แต่เซนเซอร์ที่อยู่ไกลจากสถานีฐานออกไปจนระยะการสื่อสารไม่ถึง เซนเซอร์เหล่านี้จะต้องหาเซนเซอร์ข้างเคียงเพื่อส่งข้อมูลไปให้เซนเซอร์ตัวนั้นส่งต่อข้อมูลไปให้ถึงสถานีฐาน กระบวนการนี้จึงส่งผลอย่างมากต่อประสิทธิภาพการทำงานในเครือข่าย เนื่องจากเซนเซอร์ตัวที่อยู่ติดกับสถานีฐานต้องคอยรับภาระในการส่งข้อมูลต่อให้กับตัวอื่น ทำให้เซนเซอร์เหล่านี้ต้องใช้พลังงานมากกว่าและเหลือพลังงานน้อยกว่าเซนเซอร์ตัวอื่นๆ และทำให้พลังงานหมดเร็วกว่าเมื่อเทียบกับตัวอื่นในเครือข่าย

การเรียนรู้แบบเสริมกำลังคือรูปแบบหนึ่งของการเรียนรู้ของเครื่อง (Machine learning) ที่มีหลักการคือให้ ผู้กระทำ (Learner, Agent) คือตัวที่เราต้องการให้เรียนรู้ เรียนรู้จากการลองผิดลองถูกแล้วเก็บประสบการณ์ ซึ่ง Agent จะรับรู้สถานะ (State) ซึ่งเป็นสถานปัจจุบันของสภาพแวดล้อม (Environment) คือ สิ่งที่ Agent จะมีปฏิสัมพันธ์ด้วย แล้ว Agent จะตัดสินใจเลือกการกระทำ (Action) คือ สิ่งที่จะมีปฏิสัมพันธ์กับ Environment บนพื้นฐานของนโยบาย (Policy) คือ วิธีที่ Agent จะตัดสินใจเลือก Action เมื่อรับรู้ State ของ Environment แล้ว และเมื่อ Agent ทำ Action ที่เลือก Agent จะรับรู้ State ใหม่ของ Environment และจะได้รับผลลัพธ์ในรูปแบบของรางวัล (Reward) เพื่อใช้เก็บเป็นประสบการณ์และเพื่อตัดสินใจเลือก Action ในครั้งต่อไป ซึ่งค่า Q (Q-value) จะถูกคำนวณและกำหนดในกับทุกๆ Action ที่เป็นไปได้และ Agent อัพเดทค่า Q ด้วยค่า Reward ในทุกครั้งที่ทำ Action และ Agent จะตัดเลือก Action ที่เหมาะสมจากค่า Q อีกทีตามรูปที่ 2 การนำการเรียนรู้แบบเสริมกำลังในเครือข่ายเซนเซอร์ไร้สายนั้นช่วยให้เซนเซอร์สามารถตัดสินเลือกวิธีการทำงานให้เหมาะสมกับสถานะหรือสภาพของเครือข่ายในช่วงเวลานั้นได้ เช่น การเลือกกลุ่มของเซนเซอร์ (Cluster) การเลือกเส้นทางเพื่อส่งข้อมูลไปให้ถึงปลายทาง เป็นต้น

Software-Defined Networking (SDN) คือสถาปัตยกรรมระบบเครือข่าย ซึ่งช่วยทำให้สามารถควบคุมเครือข่ายได้มีประสิทธิภาพมากขึ้น เครือข่ายจะถูกควบคุมด้วยตัวควบคุมแบบศูนย์กลาง (Centralize controller, Controller) Controller สามารถมองเห็นภาพและควบคุมทั้งเครือข่ายได้ การนำ SDN มาใช้ร่วมกับเครือข่ายเซนเซอร์ไร้สายทำให้เซนเซอร์ไม่ต้องเป็นตัวตัดสินใจเลือกวิธีการทำงานด้วยตนเอง แต่จะส่งข้อมูลที่จำเป็นต่อการทำงานหรือการคำนวณในแอปพิลเคชันไปให้กับ Controller แทน ซึ่งจะทำให้เซนเซอร์ไม่จำเป็นต้องมีฟังก์ชันที่ซับซ้อน เพื่อใช้ทรัพยากรที่มีอยู่อย่างจำกัดได้อย่างมีประสิทธิภาพ

1.2 วัตถุประสงค์

โครงงานนี้จัดทำขึ้นเพื่อศึกษาการทำงานของเครือข่ายเซนเซอร์ไร้สาย ตลอดไปจนศึกษาข้อเสนอแนะของงานวิจัยในอดีต โดยมุ่งเน้นศึกษาไปในเรื่องการใช้พลังงานอย่างมีประสิทธิภาพในเครือข่าย ด้วยการนำการทำงานอื่นมาปรับใช้หรือพยายามทำให้การทำงานที่จำเป็นในเครือข่ายทำงานได้อย่างมีประสิทธิภาพมากขึ้น จากการศึกษาพบว่าด้วยข้อจำกัดในการสื่อสารของเซนเซอร์ไร้สายทำมีเซนเซอร์บางตัวต้องใช้พลังงานมากกว่าตัวอื่นๆ ส่งผลให้เซนเซอร์เหล่านั้นจะมีพลังงานที่น้อยกว่าและทำให้พลังงานหมดไปก่อนจะใช้เครือข่ายได้อย่างเต็มที่ จึงต้องพยายามศึกษาหาวิธีเพื่อมาอุดข้อจำกัดในส่วนการทำงานนี้ โครงงานนี้คือหนึ่งในการวิจัยที่ศึกษาเกี่ยวกับการใช้พลังงานในเครือข่ายเซนเซอร์ไร้สาย เพื่อเสนอวิธีที่ทำให้เครือข่ายมีอายุการใช้งานที่นานขึ้น ซึ่งเกิดขึ้นจากการสมดุลพลังงานระหว่างเซนเซอร์

1.3 ขอบเขตของโครงงาน

โครงงานนี้มุ่งเน้นในด้านการศึกการใช้พลังงานอย่างมีประสิทธิภาพเพื่อเพิ่มอายุการใช้งานของเครือข่ายเซนเซอร์ไร้สาย ซึ่งการทำงานที่เป็นส่วนสำคัญและส่งผลต่อประสิทธิภาพของเครือข่ายอย่างมากนั่นคือกลไกการหาเส้นทาง และเพื่อการใช้พลังงานอย่างมีประสิทธิภาพการนำการเรียนรู้แบบเสริมกำลังมาปรับใช้เพื่อช่วยตัดสินใจเลือกเส้นทางภายในเครือข่าย จะช่วยให้เซนเซอร์ไร้สายสามารถใช้เส้นทางที่เหมาะสมกับสถานะของเครือข่ายในช่วงเวลานั้นได้ รวมถึงนำเทคโนโลยี SDN มาใช้งานร่วมกับการเรียนรู้แบบเสริมกำลังและเครือข่ายเซนเซอร์ไร้สายเพื่อแยกการสื่อสารเป็นสองส่วนคือ Data plane และ Control plane ซึ่งข้อมูลที่มาจาก Control plane นั้น Controller จะนำไปประมวลผลเพื่อตัดสินใจเลือกเส้นทางโดยใช้การเรียนรู้แบบเสริมกำลังและจะส่งผลลัพธ์กลับไปให้เซนเซอร์ในเครือข่าย

1.4 วิธีการดำเนินงาน

วิธีการดำเนินการเพื่อเพิ่มประสิทธิภาพเครือข่ายเซนเซอร์ไร้สายโดยใช้การเรียนรู้แบบเสริมกำลังจะเกี่ยวข้องกับขั้นตอนต่อไปนี้

ศึกษางานวิจัยในอดีตเพื่อทำความเข้าใจการทำงานของเครือข่ายเซนเซอร์ไร้สายและดูแนวโน้มของข้อจำกัดหรือปัญหาเพื่อเลือกหัวข้อสำหรับการศึกษาของผู้ศึกษา รวมถึงต้องศึกษาเพื่อให้เข้าใจภาพกว้างๆ เกี่ยวกับ Machine learning

เมื่อได้กรอบในเรื่องที่ต้องการศึกษาแล้ว ผู้ศึกษาเลือกงานวิจัยมาจำนวนหนึ่งและเริ่มศึกษาแนวคิด วิธีการที่งานวิจัยนำเสนอ และผลการทดลองของงานวิจัยนั้นๆ

เริ่มสร้างการทดลองตามวิธีที่งานวิจัยได้เสนอมาทีละงานเพื่อศึกษาลงในรายละเอียดของอัลกอริทึมและเพื่อให้เข้าใจงานวิจัยนั้นๆ มากขึ้น รวมถึงเปรียบเทียบผลการทดลองของผู้ศึกษากับผลการทดลองของงานวิจัยว่าเหมือนหรือแตกต่างกันอย่างไร

ในขณะที่สร้างการทดลองตามงานวิจัย ผู้ศึกษาพยายามหาข้อจำกัดของอัลกอริทึมที่งานวิจัยนั้นๆ นำเสนอและหาแนวทางแก้ไขและต่อยอดแนวคิดเพื่อเป็นแนวทางสำหรับวิธีและอัลกอริทึมที่จะเป็นวิธีการของผู้ศึกษาเอง

1.5 ประโยชน์ที่คาดว่าจะได้รับ

การศึกษาการเพิ่มประสิทธิภาพเครือข่ายเซนเซอร์ไร้สายโดยใช้การเรียนรู้แบบเสริมกำลังทำให้ผู้ศึกษาได้เรียนรู้และเข้าใจหลักการทำงานของเครือข่ายเซนเซอร์ไร้สายและการเรียนรู้แบบเสริมกำลังรวมถึงการเรียนรู้รูปแบบอื่นๆ ของ Machine learning ด้วย นอกจากนี้ผู้ศึกษายังได้เรียนรู้การศึกษาจากการอ่านงานวิจัย การเลือกกรอบหัวข้อและวิธีศึกษาหาวิธีที่จะเป็นวิธีที่ใหม่ๆ ผู้ศึกษาจะนำเสนอ ทำให้เข้าใจวิธีในการศึกษาทำงานวิจัยมากขึ้นและได้ฝึกฝนการเขียนโปรแกรมให้ดีและมีประสิทธิภาพในภาษา Python รวมถึงฝึกทักษะด้านการสื่อสาร การสรุปและการนำเสนอเพื่อใช้ต่อยอดในการทำงานในอนาคต

2. การทบทวนวรรณกรรมที่เกี่ยวข้อง

YOUNUS และคณะ [3] นำ SDN มาใช้ร่วมกับเครือข่ายเซนเซอร์ไร้สาย ในช่วงแรก Controller และโหนดในเครือข่ายจะส่งข้อมูลคุยกันเพื่อนค้นหากันและกัน เมื่อ Controller ค้นพบโหนดครบแล้ว Controller จะรู้รูปร่างเครือข่าย (Topology) หลังจากนั้น Controller ก็สามารถหา All possible routing path ได้จาก STP และสิ่งที่แตกต่างกับ [2] คือสมการหาค่า Reward (Reward function) ในงานวิจัยนี้ค่าที่นำมาใช้หา Reward นั้นจะเป็นค่า

อัตราส่วน ประกอบไปด้วย อัตราส่วนของระยะทาง, อัตราส่วนของจำนวนทอดไปหา Controller, อัตราส่วนของพลังงาน และอัตราส่วนของการตอบกลับเมื่อได้รับแพ็คเก็ต (Packet acknowledgment) หลังจากได้ Reward แล้ว Controller จะเลือกเส้นทางที่มีค่า Q สูงที่สุดและสร้างเส้นทางนั้นไปให้กับโหนดในเครือข่าย ผู้เขียนได้วัดประสิทธิภาพสองด้าน 1. อายุการใช้งานเครือข่าย ซึ่งจะแยกย่อยเป็น อายุการใช้งานเมื่อโหนดแรกหมดพลังงาน, อายุการใช้เมื่อ X% โหนดหมดพลังงาน, อายุการใช้งานจนกว่าโหนดสุดท้ายจะหมดพลังงาน และ 2. อัตราการส่งแพ็คเก็ตสำเร็จ โดยเปรียบเทียบกับวิธีเก่าๆ คือ Q-Learning, RLBR, T_SDWSN, EASDN, QR-SDN, TIDE ซึ่งผลการทดลองแสดงให้เห็นว่าวิธีที่ผู้เขียนนำเสนอนั้นทำให้อายุการใช้งานมากที่สุดทั้งในพื้นที่ขนาด 100 x 150 เมตร และ 200 x 300 เมตร รวมถึงอัตราการส่งแพ็คเก็ตสำเร็จคือ 99.99% และ 99.12% ของทั้งสองพื้นที่ตามลำดับ ซึ่งมากที่สุดเมื่อเทียบกับวิธีอื่นๆ

3. วิธีการดำเนินการวิจัย

3.1 การทบทวนงานวิจัย

เริ่มการศึกษาจากการทบทวนงานวิจัยในอดีตเพื่อทำความเข้าใจพื้นฐานหลักการทำงานของเครือข่ายเซนเซอร์ไร้สายและ Machine learning รวมถึงการเรียนรู้แบบเสริมกำลังด้วย การศึกษาในส่วนนี้ทำให้ผู้ศึกษาได้เห็นภาพกว้างๆ ซึ่งผู้ศึกษาสรุปเนื้อหาได้ดังนี้

ผู้ศึกษามุ่งเน้นศึกษาไปในด้านการใช้พลังงานอย่างมีประสิทธิภาพในเครือข่ายเซนเซอร์ไร้สาย ซึ่งหมายถึงการพยายามทำให้เครือข่ายมีอายุการใช้งานมากขึ้นด้วยวิธีการต่างๆ เช่น การเลือกกลุ่มของเซนเซอร์ การออกแบบตำแหน่งของเซนเซอร์ในเครือข่าย การเลือกเส้นทางเพื่อส่งข้อมูลไปให้ถึงปลายทาง เป็นต้น

การนำการเรียนรู้แบบเสริมกำลังมาใช้ในเครือข่ายเซนเซอร์ไร้สายจะทำให้เครือข่ายสามารถตัดสินใจเลือกการกระทำให้เหมาะสมกับสถานะของเครือข่าย ซึ่งในกลไกการหาเส้นทาง การเรียนแบบเสริมกำลังจะเรียนรู้และตัดสินใจเลือกเส้นทางที่เหมาะสมกับสภาพเครือข่ายในช่วงเวลานั้นได้ ส่งผลให้เกิดการสมดุลพลังงานในเครือข่ายเซนเซอร์ไร้สายได้

3.2 เลือกงานวิจัยเพื่อศึกษาลงรายละเอียด

ผู้ศึกษาได้เลือกงานวิจัยสองงานเพื่อมาศึกษาลงในรายละเอียด เพราะต้องการความเข้าใจที่มากขึ้นเพื่อใช้ต่อยอดหาวิธีใหม่ๆ ของผู้ศึกษาเอง

ในงานวิจัย [3] ก็ได้เสนอการนำ SDN เข้ามาทำงานร่วมกับเครือข่ายเซนเซอร์ไร้สาย โดยที่ผู้ศึกษาได้ดำเนินการตามงานวิจัยนี้เป็นหลัก ในงานวิจัยนี้ผู้เขียนได้กำหนดให้ Controller คือ Sink โดยการทำงานเริ่มจาก Controller และโหนดในเครือข่ายจะส่งข้อมูลคุยกันเพื่อค้นหากันและกัน หลังจากนั้น Controller จะคำนวนหา All possible routing path ด้วยการใช้ STP เพื่อทำให้เหลือแค่เส้นทางที่ไม่ทำให้เกิด Loop ขึ้นภายเครือข่าย รวมถึงสร้างตารางค่า Q ขึ้นมาพร้อมกับใส่ค่าเริ่มต้นเข้าไปในตารางก่อนที่จะเริ่มใช้งาน โดยคำนวณจากกรณีที่แย่ที่สุด คือ ทุกๆ โหนดในเครือขายพลังงานหมดโดยที่ยังไม่ได้ส่งข้อมูล การทำงานในเครือข่ายจะนับการทำงานเป็นรอบๆ โดนหนึ่งรอบหมายถึงทุกๆ โหนดส่งข้อมูลครบทุกตัวแล้ว ซึ่งข้อมูลแหล่านั้นนั้นอาจจะไปถึงหรือไม่ถึง Controller ก็ได้ ในรอบแรก Controller จะสุ่มเลือกเส้นทางมาหนึ่งเส้นทางและออกอากาศ (Broadcast) แพ็คเก็ตเส้นทาง (Route packet) ไปให้กับโหนดในเครือข่าย เมื่อโหนดได้รับเส้นทางแล้ว โหนดจะตรวจสอบพลังงานของตนเองว่าเพียงพอต่อการรับส่งข้อมูลต่อไปหรือไม่ ถ้ามีพลังงานเพียงพอก็จะใช้เส้นทางที่ได้รับมาในการส่งข้อมูลไปหา Controller ซึ่งข้อมูลจะประกอบไปด้วยข้อมูลที่เก็บได้จากพื้นที่และข้อมูลควบคุมที่จำเป็นสามค่า 1. พลังงานที่เหลืออยู่ (Remaining energy) 2. จำนวนที่ได้รับ Acknowledgment หลังจากส่งแพ็คเก็ต 3. จำนวนครั้งที่ส่งแพ็คเก็ต

เมื่อจบรอบการทำงาน Controller จะคำนวณ Reward จากการใช้เส้นทางนั้นด้วยสมการ (1)

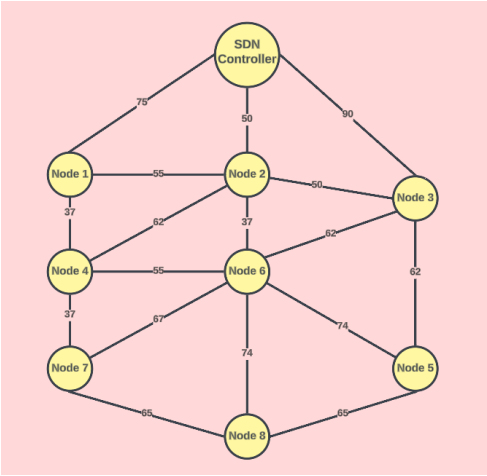
$$R_t = \sum_{i=0}^{N} (w_1 * d(s_i) + w_2 * H(s_i) + w_3 * E(s_i) + w_4 * p(s_i)) \quad (1)$$

หลังจากได้ Reward แล้ว Controller จะคำนวณค่า Q ของคู่ State-Action (s_t, a_t) และเก็บลงไปในตารางค่า Q (Q-table) โดย สามารถคำนวณค่า Q ได้ตามสมการ (2)

$$Q(s_t, a_t) = (1-\alpha)Q(s_t, a_t) + \alpha(R_t + \gamma max Q(s_t, a)) \quad (2)$$

3.3 สร้างการทดลองตามงานวิจัยและนำเสนอ

ผู้ศึกษาได้เขียนโปรแกรมภาษา Python เพื่อจำลองการทำงานทั้งหมดของ SDN ร่วมกับเครือข่ายเซนเซอร์ไร้สายตั้งแต่การหา All possible routing path, การรับส่งข้อมูลควบคุมและข้อมูลที่โหนดเก็บได้, การคำนวณ Reward function และการคำนวณพลังงานโดยใช้ Energy model



รูปที่ 1 Topology สำหรับทำการทดลอง [3]

ตารางที่ 1 การเปรียบเทียบความเหมือนและแตกต่างของ [3] กับงานวิจัยของผู้ศึกษาเอง

	งานวิจัย [3]	งานวิจัยของผู้ศึกษา
Topology	รูปที่ 1 (ระยะทางระหว่างโหนด ไม่ได้ระบุไว้ใน paper)	รูปที่ 1
Overhead ช่วง topology discovery	มี	ไม่มี
Reward function	ผลรวม Reward ของแต่ละโหนด (Sum)	ค่า Reward ของโหนดที่มี Reward น้อยที่สุด (Min)
State	RT (เส้นทาง, Routing Table)	(Emax – Emin), (Emin)
Action	RT (เส้นทาง, Routing Table)	RT (เส้นทาง, Routing Table)
Controller มี Energy model	Controller ไม่มี Energy model	Controller มี Energy model
วิธีการทดลอง	Raspberry pi	เขียนโปรแกรมภาษา Python

ตารางที่ 1 (ต่อ)

3.4 แนวทางการนำเสนอวิธีการของผู้ศึกษา

ผู้ศึกษาได้ตั้งสมมติฐานสำหรับการศึกษา [3] และงานวิจัยของผู้ศึกษาเอง ดังนี้

1. Controller มี Energy model ทำให้ Controller สามารถคำนวณพลังงานที่แต่ละโหนดจะใช้สำหรับส่งข้อมูล ซึ่งโหนดอาจใช้พลังงานต่างกันเมื่อใช้เส้นทางที่ต่างกัน สมมติฐานนี้ทำให้ Controller สามารถทดลองทำทุก Action ก่อนที่จะเลือก Action ได้ ไม่ต้องรอได้ข้อมูลจากโหนดก่อน

2. Controller และโหนดรับรู้ Topology อยู่แล้ว โดยไม่ต้องมีช่วง topology discovery ทำให้ไม่มี overhead ในช่วงนี้

3. ขนาดของ Control packet มีค่าเท่ากับ 0 bytes

4. กำหนดค่า Weight ของ $p(s_i)$ เป็น 0 เพื่อตัดความซับซ้อนในการทดลองช่วงแรกๆ จนถึงปัจจุบัน เนื่องจากการทำให้เครือข่ายเกิด Loss ขึ้นในการส่งข้อมูลจำเป็นต้องสุ่มการส่งแพ็คเก็ตที่ไม่สำเร็จ รวมถึงต้องมี overhead ในส่วนของการส่งซ้ำ

การนิยาม States, Action และ Reward Function ในงานวิจัยของผู้ศึกษาจะเป็นดังนี้

States: พลังงานของโหนดในเครือข่าย ซึ่งผู้ศึกษาได้นำเสนอออกมาสองแบบ ตามสมการ (3) และ สมการ (4)

Actions: All possible routing paths

Reward function: ค่า Reward ของโหนดที่มี Reward น้อยที่สุด ซึ่งสามาถคำนวณได้จากสมการ (5)

$$S = E_{Max} - E_{Min} \qquad (3)$$

S คือ *State,* E_{Max} คือ พลังงานของโหนดที่มีพลังงานมากที่สุด E_{Min} คือ พลังงานของโหนดที่มีพลังงานน้อยที่สุด

$$S = E_{Min} \qquad (4)$$

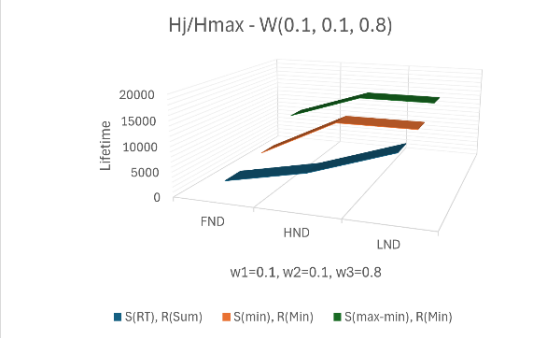
$$R_t = Min[w_1 * d(s_i) + w_2 * H(s_i) + w_3 * E(s_i) + w_4 * p(s_i)] \qquad (5)$$

4. ผลการทดลองเบื้องต้นหรือระบบต้นแบบ

การประเมินประสิทธิภาพในงานวิจัยนี้ คือ อายุการใช้งานเครือข่าย (Network Lifetime) ซึ่งนิยามได้ 3 แบบ คือ 1. เมื่อมีโหนดใดโหนดหนึ่งในเครือข่ายพลังงานหมดไป (First node die, FND) 2. เมื่อมีโหนดครึ่งหนึ่งพลังงานหมดไป (Half node die, HND) 3. เมื่อโหนดทั้งหมดพลังงานหมดไป (Last node die, LND)

โดยที่นิยามทั้ง 3 ในการทดลองปัจจุบันจะหมายถึง node 1, node 2, node 3 ตามรูปที่ 1 เท่านั้น และทดลองในขนาดพื้น 100 เมตร x 150 เมตร ซึ่งจะเปรียบกับผลการทดลองของงานวิจัย [3] และใช้วิธีกำหนด State S[(Emax-Emin), S(Emin)], คำนวณ Reward [R(Sum), R(Min)] และ การให้ค่า Weight ที่แตกต่างกัน

4.2 ผลการทดลอง

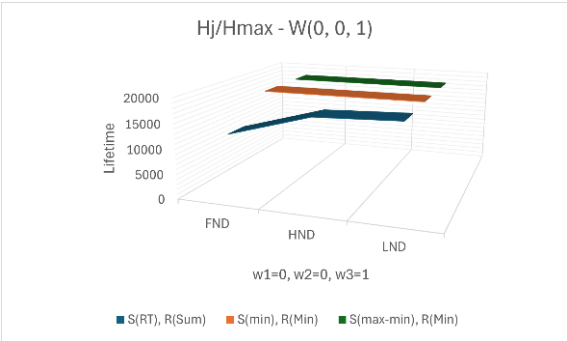


รูปที่ 2 อายุการใช้งานเครือข่ายเปรียบเทียบระหว่าง [S(RT), R(Sum)], [S(Min), R(Min)], [S(Max-Min), R(Min)] โดยใช้ค่าน้ำหนัก w_1*=0.1,* w_1*=0.1,* w_3*=0.8,* w_4*=0*

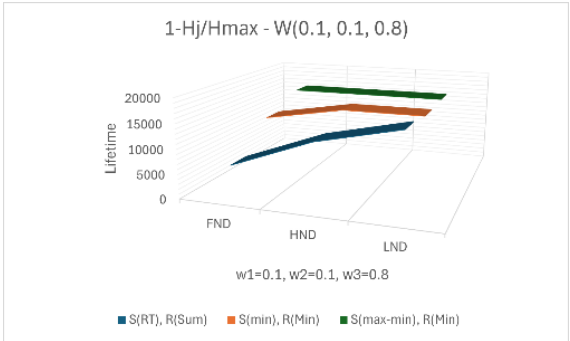
การทดลองในสถานการณ์นี้คือการกำหนดค่า Weight ที่ใช้ในการคำนวณสมการ (1) และ (5) ดังนี้ w_1**=0.1**, w_1**=0.1**, w_3**=0.8**, w_4**=0** ซึ่งจะใช้วิธีการระบุ State ที่และใช้ Reward function ที่ต่างกัน 3 แบบ

1. S(RT), R(Sum) (วิธีของงานวิจัย [3], สีน้ำเงิน)
2. S(Min), R(Min) (วิธีของงานผู้ศึกษา, สีส้ม)
3. S(Max-Min), R(Min) (วิธีของงานผู้ศึกษา, สีเขียว)

ผลการทดลองแสดงให้เห็นว่าวิธีสีส้มกับสีเขียวมีอายุการใช้งานทั้ง FND ไปถึง HND ที่ดีกว่าสีน้ำเงิน และ HND ไปถึง LND สีส้มกับสีเขียวมีอายุการใช้งานใกล้เคียงกันคือสมดุลพลังงานได้ดีกว่าสีน้ำเงิน



รูปที่ 3 อายุการใช้งานเครือข่ายเปรียบเทียบระหว่าง [S(RT), R(Sum)], [S(Min), R(Min)], [S(Max-Min), R(Min)] โดยใช้ค่าน้ำหนัก w_1*=0,* w_1*=0,* w_3*=1,* w_4*=0*



รูปที่ 4 อายุการใช้งานเครือข่ายเปรียบเทียบระหว่าง [S(RT), R(Sum)], [S(Min), R(Min)], [S(Max-Min), R(Min)] โดยใช้ค่าน้ำหนัก w_1*=0.1,* w_1*=0.1,* w_3*=0.8,* w_4*=0* หลังจากปรับสมการ reward ของ hop count

การทดลองในสถานการณ์นี้คือการกำหนดค่า Weight ที่ใช้ในการคำนวณสมการ (1) และ (5) ดังนี้ w_1**=0**, w_1**=0**, w_3**=1**, w_4**=0** การกำหนดค่า Weight แบบนี้ทำเพื่อโฟกัสแค่พลังงาน สำหรับทดสอบว่าการเรียนรู้แบบเสริมกำลังจะสามารถตัดสินใจเลือกเส้นทางที่เหมาะสมกับค่าพลังงานปัจจุบันของโหนดแบบตรงไปตรงมาได้อย่างไรบ้าง ซึ่งผลแสดงให้เห็นว่าทั้ง 3 แบบมี FND, HND และ LND ที่ดีขึ้น รวมถึงแบบสีส้มกับสีเขียวสามารถสมดุลพลังงานตั้งแต่ FND ไปถึง LND ได้ดีขึ้น

เมื่อเปรียบเทียบกับรูปที่ 2 ที่ให้ w_1**=0.1**, w_1**=0.1**, w_3**=0.8**, w_4**=0** พบว่าการให้น้ำหนักแบบw_1**=0**, w_1**=0**, w_3**=1**, w_4**=0** ได้ผลลัพธ์ที่ดีกว่า ทำให้ผู้ศึกษาได้ตั้งคำถามว่าการหา reward ของระยะทางและ hop count แบบ [3] เหมาะสมแล้วหรือไม่เมื่อนำมาปรับใช้ในสมการ (5) ซึ่งเป็นวิธีของผู้ศึกษา ดังนั้นผู้ศึกษาจึงกลับไปทบทวนงานวิจัย [1] ที่ออกแบบ reward function ซึ่งตรงข้ามกับ reward function ของ [3] ผู้ศึกษาจึงได้ทดลองนำแนวคิดของ reward function ของ [1] มาปรับใช้ โดยมีสมการ reward ของ hop count สมการ (6)

$$H(s_i) = 1 - \frac{Hop\_Nbr\_to\_Sink(s_i)}{Hop\_to\_Sink_{max}(s_i)} \qquad (6)$$

การทดลองในสถานการณ์นี้คือการกำหนดค่า Weight ที่ใช้ในการคำนวณสมการ (1) และ (5) ดังนี้ w_1**=0.1**, w_1**=0.1**, w_3**=0.8**, w_4**=0** โดยคำนวณ reward ของ hop count ตามสมการที่ (6) พบว่าทำให้ทั้ง 3 แบบ มีอายุการใช้งาน FND ที่มากขึ้น และสีน้ำเงินมี HND มากขึ้นเมื่อเปรียบเทียบกับรูปที่ 2

5. บทสรุป

5.1 สรุปผลการศึกษา

กลไกการหาเส้นทางเป็นส่วนการทำงานสำคัญในเครือข่ายเซนเซอร์ไร้สายที่ส่งผลกระทบต่อการเพิ่มประสิทธิภาพการใช้พลังงาน การนำ SDN มาใช้งานร่วมกับ RL ทำให้เซนเซอร์ไม่ต้องใช้พลังงานไปกับการทำงานที่ซับซ้อน และทำให้ Controller สามารถเรียนและตัดสินใจเลือก Action ได้เหมาะสมให้กับเครือข่ายได้ ซึ่งก็คือเส้นทางที่เหมาะสมกับสภาพปัจจุบันของเครือข่าย ทำให้เกิดการสมดุลพลังงานระหว่างเซนเซอร์ Controller จะหาเส้นทางที่เป็นไปได้ทั้งหมด (All possible routing path) จากการใช้ STP เพื่อไม่ให้มีเส้นทางที่ทำให้เกิด Loop ในเครือข่าย

การเปรียบเทียบประสิทธิภาพ Reward function ระหว่าง [3] กับงานวิจัยของผู้ศึกษานั้นพบว่าทั้ง 3 แบบยังมี LND ที่ใกล้เคียงกันการใช้ แต่ [S(Min), R(Min)] และ [S(Max-Min), R(Min)] ช่วยให้อายุการใช้งานเครือข่ายเพิ่มขึ้นและสมดุลพลังงานได้ดีกว่าเมื่อเทียบกับแบบ [S(RT), R(Sum)] ของ [3] ทั้ง FND และ HND รวมถึงการ

ปรับสมการตาม (6) ก็ช่วยเพิ่มประสิทธิการทำงานของการรียนรู้แบบเสริมกำลังได้เทียบจากรูปที่ 2 กับรูปที่ 4

5.2 การต่อยอดในอนาคต

ในอนาคตผู้ศึกษาวางแผนเอาไว้ว่าจะปรับให้ Controller ไม่มี Energy model, เพิ่ม Topology discovery และเพิ่ม Packet loss เข้ามาในเครือข่าย ซึ่งจะทำให้สามารถคิดค่า $\mathbf{p(s_i)}$ ในสมการ (1) และ (5) ได้ นอกจากนี้ผู้ศึกษาจะทำให้การทดลองมีความใกล้เคียงการทำงานในเครือข่ายเซนเซอร์ไร้สายในชีวิตจริงมากขึ้น โดยการสุ่มจำนวนโหนด, สุ่มตำแหน่งของโหนด รวมถึงการลดจำนวนของเส้นทางที่เป็นไปได้ทั้งในเครือข่ายลงเพื่อไม่ให้มี action มากจนเกินไปทำให้การนำการเรียนรู้แบบเสริมกำลังมาใช้สามารถปรับขนาดได้ดีมากขึ้นเมื่อเครือข่ายมีขนาดใหญ่มากขึ้น

เอกสารอ้างอิง

[1] W. Guo, C. Yan, and T. Lu, “Optimizing the lifetime of wireless sensor networks via reinforcement-learning-based routing,” International Journal of Distributed Sensor Networks, vol. 15, no. 2, pp. 1-20, Feb. 2019.

[2] M. U. Younus, M. K. Khan, M. R. Anjum, S. Afridi, Z. A. Arain and A. A. Jamali, "Optimizing the lifetime of software defined wireless sensor network via reinforcement learning", IEEE Access, vol. 9, pp. 259-272, 2021.

[3] M. U. Younus, M. K. Khan and A. R. Bhatti, "Improving the software-defined wireless sensor networks routing performance using reinforcement learning", IEEE Internet Of Things J., vol. 9, no. 5, pp. 3495-3508, Mar. 2022.

ระบบตรวจจับการโจมตีไซเบอร์บนสถาปัตยกรรมเอสดีเอ็น

ยุทธศิลป์ บุตรโยจันโท[1]

[1]คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าคุณทหารลาดกระบัง กรุงเทพฯ

Emails: 63070144@it.kmitl.ac.th

บทคัดย่อ

ระบบตรวจจับการโจมตีไซเบอร์บนสถาปัตยกรรมเอสดีเอ็น ถูกพัฒนาขึ้นเพื่อเพื่อตรวจจับการโจมตีทางไซเบอร์และจำแนกประเภทการติดต่อสื่อสารโดยแบ่งเป็น การติดต่อสื่อสารปกติ การขัดขวางการให้บริการ ทั้งแบบดอสและดีดอส และการสแกนพอร์ต โดยระบบจะทำงานอยู่บนเครือข่ายที่ออกแบบตามสถาปัตยกรรมเอสดีเอ็น ที่มีความโดดเด่นด้านการจัดการและความยืดหยุ่นของเครือข่าย และใช้การเรียนรู้ของเครื่องเป็นวิธีจำแนกประเภทการติดต่อสื่อสาร มีการเปรียบเทียบประสิทธิภาพโมเดลการจำแนกและคัดเลือก 3 โมเดลที่มีประสิทธิภาพดีที่สุด ได้แก่ Random Forest, Gradient Boosting, และ CatBoost ทั้ง 3 การพัฒนาโมเดลการจำแนกจะดำเนินการโดยใช้ PyCaret และชุดข้อมูลที่ใช้ในการสร้างโมเดลได้มาจากการจำลองการติดต่อสื่อสารบนเครือข่ายที่สร้างโดยซอฟต์แวร์ Mininet การทดสอบประสิทธิภาพการตรวจจับการโจมตีและจำแนกประเภทการติดต่อสื่อสารด้วยสถานการณ์จำลอง 3 สถานการณ์ที่มีการติดต่อสื่อสารปกติกับการโจมตีในอัตราส่วนต่าง ๆ แสดงให้เห็นว่าระบบตรวจจับการโจมตีเมื่อใช้แต่ละมีความถูกต้องอยู่ในช่วง 95.70 เปอร์เซ็นต์ ถึง 97.70 เปอร์เซ็นต์

คำสำคัญ – เอสดีเอ็น; ปัญญาประดิษฐ์; การโจมตีทางไซเบอร์

1. บทนำ

แม้ว่าพัฒนาการของคอมพิวเตอร์ จะส่งผลให้เกิดความก้าวหน้าทางเทคโนโลยีที่เป็นประโยชน์ต่อการพัฒนาอุตสาหกรรม เศรษฐกิจ และการดำรงชีวิตของผู้คน แต่ในทำนองเดียวกันนี้ พัฒนาการของคอมพิวเตอร์ยังส่งผลให้การโจมตีทางไซเบอร์ (Cyber Attack) มีพัฒนาการด้วยเช่นกัน หนึ่งในการโจมตีทางไซเบอร์ที่สร้างความเสียหายได้มากในปัจจุบัน คือ การขัดขวางการให้บริการ (Denial of Service) จากข้อมูลสถิติของการขัดขวางการให้บริการปี 2023 [2] แสดงให้เห็นว่าการโจมตีทางไซเบอร์ไม่ได้มีสัญญาณที่จะลดลง ในทางกลับกันมีการคาดการณ์ว่าการโจมตีจะเพิ่มขึ้นมากกว่า 3 เท่าภายในปี 2023 ด้วยเหตุนี้เอง การตรวจจับและป้องกันการโจมตีทางไซเบอร์จึงมีความสำคัญ เพราะช่วยให้สามารถยับยั้ง หรืออย่างน้อยลดความเสียหายที่เกิดจากการถูกโจมตีได้ และจากการศึกษาและวิเคราะห์กายวิภาคของการโจมตีทางไซเบอร์ (Anatomy of Cyber Attack) ด้วย Cyber Kill Chain ซึ่งเป็นกรอบการทำงานที่ใช้ในการทำความเข้าใจการโจมตีไซเบอร์ต่าง ๆ พบว่าในทุกการโจมตีทางไซเบอร์จะต้องผ่านช่วง Reconnaissance หรือช่วงรวบรวมข้อมูลของเป้าหมายการโจมตี ซึ่งเทคนิคที่มักใช้ในการรวบรวมข้อมูลคือการสแกนพอร์ต (Port Scanning)

ในการรับมือกับการโจมตีทางไซเบอร์ ได้มีการนำระบบตรวจจับการบุกรุก (Intrusion Detection System) หรือ IDS และระบบป้องกันการบุกรุก (Intrusion Prevention System) หรือ IPS มาใช้ ซึ่งระบบตรวจจับและป้องกันการบุกรุกถูกใช้งานมาอย่างยาวนานและมีการพัฒนาวิธีตรวจจับการโจมตีอยู่อย่างต่อเนื่อง

อีกทั้งผลจากความก้าวหน้าของเทคโนโลยีและประสิทธิภาพของคอมพิวเตอร์สูงขึ้น ยังทำให้เทคโนโลยีปัญญาประดิษฐ์ได้รับความนิยมและเริ่มมีบทบาทอย่างมากในหลายอุตสาหกรรม ปัญญาประดิษฐ์มีวิธีการเรียนรู้ที่เรียกว่า การเรียนรู้ของเครื่อง (Machine Learning) ซึ่งเป็นแขนงหนึ่งของปัญญาประดิษฐ์ที่มีความสามารถในการเรียนรู้ด้วยตนเอง สามารถเรียนรู้จากข้อมูลที่ได้รับและทำการคาดการณ์หรือตัดสินใจเองได้

นอกจากนี้ ในการจัดการเครือข่ายที่มีขนาดใหญ่ การออกแบบเครือข่ายตามสถาปัตยกรรมเอสดี์เอ็น ที่มีการแบ่งเครือข่ายออกเป็นส่วนควบคุม (Control Plane) และส่วนรับส่งข้อมูล (Data Plane) แล้วออกแบบการทำงานของส่วนควบคุมให้

เป็นแบบรวมศูนย์ (Centralized) ช่วยให้การจัดการเครือข่ายสามารถดำเนินการได้อย่างมีประสิทธิภาพ

ด้วยเหตุเหล่านี้ “ระบบตรวจจับการโจมตีไซเบอร์บนสถาปัตยกรรมเอสดีเอ็น” จึงถูกพัฒนาขึ้น ซึ่งระบบจะทำงานบนเครือข่ายที่ออกแบบตามสถาปัตยกรรมเอสดีเอ็นและใช้ปัญญาประดิษฐ์ในการตรวจจับการโจมตี โดยที่มุ่งเน้นการตรวจจับไปที่การสแกนพอร์ต ดอส และดีดอส

2. ทบทวนวรรณกรรมที่เกี่ยวข้อง

2.1 การโจมตีทางไซเบอร์

ความหมายของการโจมตีทางไซเบอร์ยังถูกนิยามโดยสถาบันมาตรฐานและเทคโนโลยีแห่งชาติ (NIST) ซึ่งสามารถสรุปใจความได้ว่า การโจมตีซึ่งอยู่บนโลกไซเบอร์ โดยมีเป้าหมายเป็นการใช้งานโลกไซเบอร์ขององค์กรหนึ่ง ๆ เพื่อที่จะขัดขวาง ทำให้ใช้งานไม่ได้ ทำลาย ควบคุมคอมพิวเตอร์หรือโครงสร้างพื้นฐานในทางที่ไม่ดี หรือ ทำลายความถูกต้องของข้อมูลหรือขโมยข้อมูล [6]

แม้ว่าการโจมตีทางไซเบอร์จะถูกนิยามไว้อย่างหลากหลาย แต่ใจความสำคัญที่นิยามต่าง ๆ กล่าวถึงเหมือนกันคือ การโจมตีทางไซเบอร์คือการกระทำที่มีเป้าหมายเพื่อทำให้ทรัพย์สินซึ่งเป็นได้ทั้ง ข้อมูล คอมพิวเตอร์ ผู้คน ตกอยู่ในสถานะที่ไม่ปกติ [7]

2.1.1 กายวิภาคของการโจมตีทางไซเบอร์

Cyber Kill Chain [8] เป็นกรอบการทำงาน (Framework) ที่ใช้ในการทำความเข้าใจการโจมตีไซเบอร์ต่าง ๆ โดยแบ่งการโจมตีออกเป็น 7 ขั้นที่เรียงต่อกัน ดังรูปที่ 2.1 การโจมตีเริ่มจากช่วง Reconnaissance คือช่วงรวบรวมข้อมูลของเป้าหมายที่มีโอกาสโจมตี หลังจากได้ข้อมูลของเป้าหมายแล้วผู้โจมตีจะเริ่มออกแบบวิธีที่ใช้เจาะระบบ และเครื่องมือที่ใช้ เรียกช่วงนี้ว่า Weaponize จากนั้นจะเข้าสู่ช่วง Delivery คือช่วงนำเครื่องมือที่จัดเตรียมไว้ส่งไปยังคอมพิวเตอร์เป้าหมายเพื่อทำการติดตั้ง ก่อนการติดตั้งจะเป็นช่วง Exploitation คือการใช้ประโยชน์จากช่องโหว่ระบบของเป้าหมาย เพื่อไม่ให้ระบบรักษาความปลอดภัยตรวจพบ จากนั้นระบบก็จะทำการติดตั้งเครื่องมือและโปรแกรมต่าง ๆ เรียกช่วงนี้ว่า Installation เมื่อติดตั้งสำเร็จเรียบร้อย จากนั้นจะเป็นช่วง Command and Control คือช่วงที่โปรแกรมหรือเครื่องมือที่อยู่บนคอมพิวเตอร์เป้าหมายติดต่อสื่อสารกับผู้โจมตีเพื่อสั่งการทำงานต่าง ๆ และท้ายที่สุด ช่วง Act on Objective คือช่วงที่ผู้โจมตีดำเนินการโจมตี ผลกระทบที่เกิดขึ้นจะขึ้นอยู่กับวัตถุประสงค์ของผู้โจมตี

2.1.2 การสแกนพอร์ต

การสแกนพอร์ตเป็นหนึ่งในการเตรียมการก่อนการโจมตีในช่วง Reconnaissance การสแกนพอร์ตจะแบ่งออกเป็น 2 รูปแบบ ได้แก่ การสแกนแนวตั้ง (Vertical Scanning) และการสแกนแนวนอน (Horizontal Scanning) ซึ่งการสแกนแนวตั้ง คือการสแกนหมายเลขพอร์ตต่าง ๆ บนเครื่องคอมพิวเตอร์เครื่องหนึ่ง เพื่อให้ทราบว่าคอมพิวเตอร์เครื่องนั้นให้บริการใดหรือเปิดใช้งานหมายเลขพอร์ตใดอยู่บ้าง ส่วนการสแกนแนวนอน คือการสแกนหมายเลขพอร์ตใดหมายเลขพอร์ตหนึ่งบนคอมพิวเตอร์หลายเครื่อง เพื่อให้ทราบว่ามีคอมพิวเตอร์เครื่องใดบ้างที่เปิดใช้งานหมายเลขพอร์ตหรือบริการที่สนใจ

2.1.3 การขัดขวางการให้บริการ

การขัดขวางการให้บริการ หรือ ดอส (DoS) คือการที่ผู้โจมตีดำเนินการส่งข้อมูลปริมาณมาก หรือ ข้อมูลไม่พึงประสงค์ ที่เมื่อคอมพิวเตอร์เป้าหมายได้รับแล้วจะเกิดการทำงานที่ผิดพลาด หรือใช้ทรัพยากรของระบบมากจนไม่สามารถให้บริการตามปกติได้ และเป็นเหตุให้ผู้ใช้งานไม่สามารถเข้าถึงบริการได้ ปัจจุบันดอสมักสร้างการโจมตีโดยใช้คอมพิวเตอร์หลายเครื่องที่เรียกว่าบอทเน็ต (Botnet) ซึ่งเป็นกลุ่มของอุปกรณ์คอมพิวเตอร์ที่ถูกติดตั้งมัลแวร์ ซึ่งเป็นโปรแกรมสำหรับอนุญาตให้ผู้ใช้ที่ไม่ได้รับอนุญาตสามารถควบคุมคอมพิวเตอร์เครื่องนั้นจากระยะไกลได้ เมื่อดอสมีที่มาของการโจมตีจากแหล่งหลายแหล่งพร้อม ๆ กัน จะเรียกการโจมตีแบบนี้ว่า ดีดอส (Distributed Denial of Service ตัวย่อภาษาอังกฤษ DDoS) ซึ่งมีประสิทธิภาพในการโจมตีมากกว่าดอสหลายเท่า [5][7][11]

2.2 ระบบตรวจจับการโจมตีทางไซเบอร์

ระบบตรวจจับการโจมตีทางไซเบอร์ [12][13] เป็นเทคโนโลยีที่ถูกพัฒนาขึ้นเพื่อใช้ในการดูแลสอดส่องการติดต่อสื่อสารบนเครือข่าย เพื่อตรวจจับความผิดปกติ หรือ การกระทำต่าง ๆ ที่ไม่ได้รับอนุญาต เมื่อตรวจพบความผิดปกติหรือการกระทำที่ไม่ได้รับอนุญาต ระบบจะมีการดำเนินการป้องกันตามมาตรการที่กำหนดไว้ในระบบ เช่น แจ้งผู้ดูแลระบบ หรือ ปิดกั้นการติดต่อสื่อสารดังกล่าว เป็นต้น

ประเภทระบบตรวจจับการโจมตีแบ่งออกได้เป็น 2 ประเภทหลัก ได้แก่ HIDS (Host Intrusion Detection System) ระบบตรวจจับการโจมตีที่ดูแลเจาะจงไปที่คอมพิวเตอร์ (Host) เครื่องใดเครื่องหนึ่ง โดยระบบตรวจจับการโจมตีประเภทนี้จะรวบรวมข้อมูลที่ใช้ในการวิเคราะห์การโจมตีจากคอมพิวเตอร์เครื่องนั้น และ NIDS (Network Intrusion Detection System) ระบบ

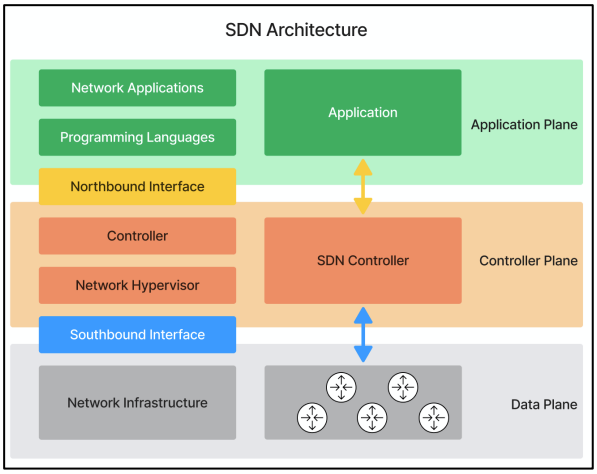
ตรวจจับการโจมตีที่ดูแลครอบคลุมระดับเครือข่าย วิเคราะห์แพ็คเก็ตที่เดินทางผ่านช่องทางเข้าออกต่าง ๆ ของเครือข่าย โดยข้อมูลที่ใช้ในการวิเคราะห์การโจมตีมักจะเป็นข้อมูลที่อยู่ใน ชั้น Transport Layer และ Internet Layer เป็นหลัก

วิธีการตรวจจับการโจมตีแบ่งออกได้เป็น 3 วิธี ได้แก่ Signature-based Detection อาศัยข้อมูลช่องโหว่ที่เคยถูกใช้ในการโจมตีหรือรูปแบบการโจมตีที่เคยเกิดขึ้น และบันทึกเก็บไว้ในฐานข้อมูล วิธีนี้มักให้ความสำคัญในการตรวจจับการโจมตีที่มีรูปแบบเหมือนกับการโจมตีที่เคยเกิดขึ้นและใช้ช่องโหว่เดิมที่เคยถูกโจมตี, Anomaly-based Detection เป็นวิธีการตรวจจับการโจมตีโดยสังเกตรูปแบบหรือพฤติกรรมที่ผิดแปลกไปจากความปกติที่ระบบดูแลอยู่ วิธีตรวจจับการโจมตีแบบ Anomaly-based Detection จึงมีความสามารถในการตรวจจับการโจมตีที่ยังไม่เคยเกิดขึ้นมาก่อน ไม่มีอยู่ในฐานข้อมูลการโจมตี แต่สามารถเกิดความผิดพลาดในการตรวจจับการโจมตีได้มากเช่นกัน, และ Specification-based Detection เป็นวิธีการตรวจจับการโจมตีที่ทำงานคล้ายคลึงกับวิธีตรวจจับการโจมตีแบบ Anomaly-based Detection แต่จะมีการ ดึงเอาข้อจำกัดของการตรวจจับการโจมตีออกมาโดยผู้เชี่ยวชาญ เพื่อเพิ่มประสิทธิภาพในการตรวจจับการโจมตีของระบบ

2.3 เอสดีเอ็น

เอสดีเอ็น (SDN) [15] เป็นแนวทางในการจัดการเครือข่าย ถูกพัฒนามาเพื่อช่วยให้การจัดการและการออกแบบเครือข่ายทำได้ง่ายขึ้น โดยแยกส่วนควบคุม (Control plane) ออกจากส่วนรับส่งข้อมูล (Data plane) ที่อยู่ในอุปกรณ์เครือข่าย (Network Device) แล้วรวมส่วนควบคุมที่แยกออกมาไว้ด้วยกัน อีกทั้งยังมีส่วนแอปพลิเคชัน (Application Plane) ไว้สำหรับแอปพลิเคชันเครือข่ายที่ทำงานร่วมกับส่วนควบคุมได้อีกด้วย

ภายในส่วนควบคุมจะมีตัวควบคุม (Controller) ซึ่งตัวควบคุมและอุปกรณ์เครือข่ายจะติดต่อสื่อสารกันโดยอาศัย Southbound Interface ที่มักจะเป็นโปรโตคอล OpenFlow ส่วนการติดต่อสื่อสารระหว่างตัวควบคุมและแอปพลิเคชันเครือข่ายจะอาศัย Northbound Interface อย่างเช่น REST API เป็นต้น ทำให้ผู้ดูแลระบบสามารถกำหนดการทำงานต่าง ๆ ของอุปกรณ์เครือข่ายได้อย่างยืดหยุ่น



รูปที่ 2.1 สถาปัตยกรรมและลำดับชั้นของเอสดีเอ็น

2.4 การเรียนรู้ของเครื่อง

Jafar et al. [20] ได้อธิบายว่าการเรียนรู้ของเครื่องนั้นจำเป็นในการทำให้คอมพิวเตอร์สามารถทำงานได้โดยไม่ต้องมีการสอนจากมนุษย์ และสามารถเรียนรู้จากประสบการณ์ใหม่ ๆ เพื่อให้สามารถทำงานที่มีความซับซ้อนและต้องการความสามารถในการปรับตัว ซึ่งการเรียนรู้ของเครื่องถูกพิสูจน์แล้วว่าสามารถแก้ปัญหาต่าง ๆ ด้านวิทยการข้อมูล (Data Science) ได้ โดยปัญหาด้านวิทยาการข้อมูลสามารถแบ่งออกได้เป็น 5 กลุ่ม ได้แก่ Classification Problem, Anomaly Detection Problem, Regression Problem, Clustering Problem, และ Reinforcement Problem

การเรียนรู้ของเครื่องประกอบไปด้วย 6 องค์ประกอบ ได้แก่ การรวบรวมและจัดเตรียมข้อมูล, การเลือกฟีเจอร์ (Features), เลือกอัลกอริทึม, เลือกโมเดลและค่าพารามิเตอร์ (Parameters), การสอน, และ การประเมินประสิทธิภาพ การทำงานของแต่ละองค์ประกอบจะเรียงต่อกันตั้งแต่การรวบรวมข้อมูลไปจนถึงการประเมินประสิทธิภาพ

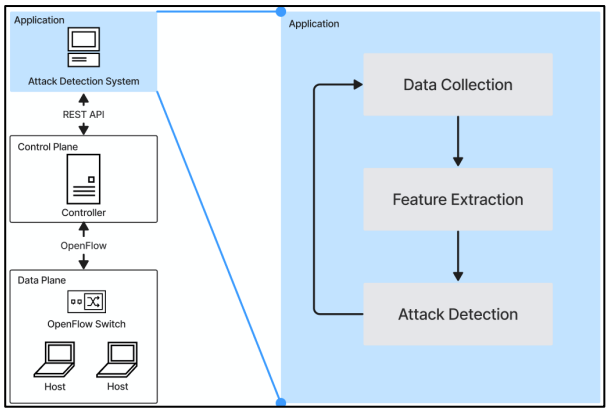
จากงานวิจัยของ Jafar et al. [20] ที่แบ่งประเภทวิธีการเรียนรู้ของเครื่องออกเป็น 10 ประเภท มี 3 ประเภทที่น่าสนใจ ได้แก่ Supervised Learning ซึ่งเป็นการเรียนรู้ที่ชุดข้อมูลที่ใช้ในการฝึกฝนอัลกอริทึมมีการกำหนดผลลัพธ์ที่ถูกต้องไว้แล้ว เหมาะใช้ในการแก้ปัญหาจำพวก Classification และ Regression, Unsupervised Learning เป็นการเรียนรู้ที่ชุดข้อมูลที่ใช้ในการฝึกฝนอัลกอริทึมไม่ได้กำหนดผลลัพธ์ที่ถูกต้องไว้ ซึ่งส่วนมากมักถูกนำไปใช้ในการหารูปแบบหรือโครงสร้างที่ซ่อนอยู่ในข้อมูล, และ Ensemble Learning เป็นโมเดลการเรียนรู้ของเครื่องที่หลายโมเดลถูกฝึกฝนเพื่อแก้ปัญหาร่วมกัน

3. วิธีดำเนินการวิจัย

ระบบตรวจจับการโจมตีทางไซเบอร์ในโครงงานนี้ มุ่งเน้นไปที่การตรวจจับการขัดขวางการให้บริการทั้ง ดอส ดีดอส และการสแกนพอร์ตแบบ Vertical Scanning ที่เน้นการสแกนพอร์ตหลายพอร์ตของคอมพิวเตอร์เป้าหมายเครื่องเดียว

3.1 สถาปัตยกรรมของระบบ

ระบบตรวจจับการโจมตีทำงานอยู่ในส่วนแอปพลิเคชันของเครือข่ายที่ออกแบบตามสถาปัตยกรรมเอสดีเอ็น องค์ประกอบของระบบมีแบ่งเป็น 3 องค์ประกอบ ได้แก่ ส่วนรวบรวมข้อมูล ส่วนคำนวณฟีเจอร์ และ ส่วนตรวจจับการโจมตี การทำงานในส่วนของเอสดีเอ็นนั้น ส่วนควบคุมจะมีตัวควบคุมที่คอยกำหนดการติดต่อสื่อสารของส่วนรับส่งข้อมูลผ่านทาง OpenFlow Switch โดยใช้โปรโตคอล OpenFlow ในการแลกเปลี่ยนข้อมูล การทำงานหลักในส่วนของเอสดีเอ็นนั้น OpenFlow Switch จะนำแพ็คเก็ตที่ได้รับเข้ามาเปรียบเทียบข้อมูล Header ของแพ็คเก็ตเหล่านั้นกับข้อมูล Match ของ Flow Entry ที่มีอยู่ หากไม่มีข้อมูล Match ที่ตรงกันกับข้อมูล Header ของแพ็คเก็ต OpenFlow Switch จะส่งต่อแพ็คเก็ตนั้นให้กับตัวควบคุมโดยใช้ Packet-In Message เพื่อให้ตัวควบคุมกำหนดวิธีการจัดการ ตัวควบคุมจะส่ง Packet-Out Message กลับไปยัง OpenFlow Switch โดยที่ภายใน Packet-Out Message จะมีวิธีจัดการกับแพ็คเก็ตดังกล่าว จากนั้นตัวควบคุมจะกำหนด Flow Entry ใหม่ให้กับ OpenFlow Switch โดยใช้ Modify State Message เพื่อให้ OpenFlow Switch สามารถดำเนินการกับแพ็คเก็ตที่มีข้อมูล Header เหมือนกันกับแพ็คเก็ตดังกล่าวในอนาคต ส่วนการติดต่อสื่อสารระหว่างระบบตรวจจับการโจมตีและตัวควบคุมจะใช้ REST API ซึ่งตัวควบคุมจะมี API ต่าง ๆ สำหรับให้ระบบตรวจจับการโจมตีส่งคำร้องขอข้อมูลหรือกำหนดการทำงานของเครือข่าย



รูปที่ 3.1 สถาปัตยกรรมของระบบ

3.2 การทำงานของระบบ

องค์ประกอบการทำงานของระบบตรวจจับการโจมตีแบ่งออกเป็น 3 องค์ประกอบ ได้แก่ ส่วนรวบรวมข้อมูล (Data Collection), ส่วนคำนวณฟีเจอร์ (Feature Extraction), และ ส่วนตรวจจับการโจมตี (Attack Detection) ทั้ง 3 องค์ประกอบนี้จะถูกเรียกใช้งานทุก ๆ 3 วินาที ซึ่งเป็นระยะเวลาที่เหมาะสม ดังที่มีการกล่าวในการศึกษาแนวทางที่มีอยู่ในปัจจุบัน [4][22] บวกกับการทดลองใช้งานแล้วในโครงงานนี้ โดยที่การทำงานจะเริ่มจากส่วนรวบรวมข้อมูล ร้องขอข้อมูลการติดต่อสื่อสาร ซึ่งก็คือ Flow ที่อยู่ใน Flow Table ของ OpenFlow Switch ผ่านทางตัวควบคุมโดยใช้ REST API จากนั้นนำข้อมูลที่ได้ส่งให้กับส่วนคำนวณฟีเจอร์ ส่วนคำนวณฟีเจอร์จะคัดแยกคุณสมบัติของแต่ละ Flow ที่ได้รับมาให้เหลือเพียงคุณสมบัติที่สำคัญ ได้แก่ จำนวนแพ็คเก็ต, หมายเลขไอพีผู้ส่ง, หมายเลขไอพีผู้รับ, และหมายเลขพอร์ตผู้รับ จากนั้นนำมาคำนวณคุณสมบัติเบื้องต้น 4 คุณสมบัติดังตารางที่ 3.1 แล้วจึงนำคุณสมบัติทั้ง 4 นี้ไปคำนวณเป็นฟีเจอร์ที่ใช้ในการตรวจจับการโจมตี 6 ฟีเจอร์ ดังตารางที่ 3.2 หลังจากได้รับฟีเจอร์ที่ใช้ตรวจจับการโจมตีจากส่วนคำนวณฟีเจอร์แล้ว ส่วนตรวจจับการโจมตีจะนำฟีเจอร์เหล่านี้ไปเปรียบเทียบกับโมเดลการจำแนกที่สร้างขึ้นด้วยเครื่องมือการเรียนรู้ของเครื่อง PyCaret [30] โดยใช้ชุดข้อมูลที่จัดทำขึ้นในโครงงานนี้ ผลลัพธ์การตรวจจับการโจมตีที่ได้จากส่วนตรวจจับการโจมตี คือประเภทของการติดต่อสื่อสาร ได้แก่ การติดต่อสื่อสารปกติ, ดอส, ดีดอส, และการสแกนพอร์ต

ตารางที่ 3.1 คุณสมบัติที่ผ่านการคำนวณ

สัญลักษณ์	คุณสมบัติผ่านการคำนวณ
PC	จำนวนแพ็คเก็ต (Packet Count)
USIP	จำนวนหมายเลขไอพีผู้ส่งที่ไม่ซ้ำกัน (Unique Source IP Address)
UDIP	จำนวนหมายเลขไอพีผู้รับที่ไม่ซ้ำกัน (Unique Destination IP Address)
UPT	จำนวนหมายเลขพอร์ตผู้รับที่ไม่ซ้ำกัน (Unique Destination Port)

ตารางที่ 3.2 ฟีเจอร์ที่ใช้ตรวจจับการโจมตี

สัญลักษณ์	การคำนวณ	ฟีเจอร์
A1	PC	จำนวนแพ็คเก็ต
A2	USIP	จำนวนหมายเลขไอพีผู้ส่งที่ไม่ซ้ำกัน
A3	USIP / UDIP	จำนวนหมายเลขไอพีผู้ส่งที่ไม่ซ้ำกันต่อหนึ่งหมายเลขไอพีผู้รับ
A4	PC / UDIP	จำนวนพ็คเก็ตต่อหนึ่งหมายเลขไอพีผู้รับ
A5	USIP / UPT	จำนวนหมายเลขไอพีผู้ส่งที่ไม่ซ้ำกันต่อหนึ่งพอร์ตผู้รับ
A6	PC / UPT	จำนวนแพ็คเก็ตต่อหนึ่งพอร์ตผู้รับ

3.3 การพัฒนาโมเดลจำแนกประเภทการติดต่อสื่อสารด้วยการเรียนรู้ของเครื่อง

ชุดข้อมูลสำหรับสอนและทดสอบโมเดลจะสร้างขึ้นจากเครือข่ายที่มีการติดต่อสื่อสารพื้นหลังเป็นการเข้าใช้บริการบนเซิร์ฟเวอร์จากผู้ใช้งานอย่างถูกต้อง แล้วจำลองการติดต่อสื่อสารประเภทต่าง ๆ ได้แก่ การดาวน์โหลดไฟล์ การสแกนพอร์ต การดอส และดีดอส ที่เวลาและด้วยระยะเวลาที่กำหนด ความสมดุลของข้อมูลการติดต่อสื่อสารแต่ละประเภทนั้นมีผลต่อการสอนและทดสอบโมเดล ทำให้ต้องคำนึงถึงปริมาณข้อมูลของแต่ละประเภทการติดต่อสื่อสาร โดยชุดข้อมูลสำหรับสอนและทดสอบโมเดลจะประกอบไปด้วยการติดต่อสื่อสาร 4 ประเภท ได้แก่ การติดต่อสื่อสารปกติ ซึ่งได้มาจากการเข้าใช้บริการเว็บไซต์และบริการควบคุมคอมพิวเตอร์จากระยะไกล รวมไปถึงการดาวน์โหลดไฟล์ด้วยความเร็ว 2 และ 4 เมกะไบต์ต่อวินาที สำหรับการสแกนพอร์ต ดอส และดีดอส จะได้มาจากการใช้เครื่องมือทดสอบเครือข่าย Hping3 สร้างและส่งแพ็คเก็ตด้วยอัตราต่าง ๆ ได้แก่ สูงมาก, สูง, กลาง, ต่ำ, และ ต่ำมาก การติดต่อสื่อสารปกติ สแกนพอร์ต ดอส และดีดอส มีจำนวนแถวข้อมูลอยู่ที่ 2324 1996 2464 และ 2456 ตามลำดับ รวมเป็น 9240 แถวข้อมูล

เมื่อนำชุดข้อมูลนี้มาใช้ในการสอนและทดสอบโมเดลที่มีให้ใช้งานใน PyCaret **รูปที่ 3.2** จะพบว่าโมเดลที่มี Accuracy หรือความถูกต้องสูงที่สุด 3 โมเดลแรกคือ CatBoost, Gradient Boosting, และ Random Forest โดยมีความถูกต้องอยู่ที่ 0.9797, 0.9796, และ 0.9794 ตามลำดับ ซึ่ง 3 โมเดลนี้จะนำไปใช้ในการทดลองกับระบบตรวจจับการโจมตีที่พัฒนา

	Model	Accuracy	AUC	Recall	Prec.	F1
catboost	CatBoost Classifier	0.9797	0.9984	0.9797	0.9798	0.9797
gbc	Gradient Boosting Classifier	0.9796	0.9987	0.9796	0.9797	0.9796
rf	Random Forest Classifier	0.9794	0.9980	0.9794	0.9795	0.9794
lightgbm	Light Gradient Boosting Machine	0.9790	0.9985	0.9790	0.9791	0.9790
xgboost	Extreme Gradient Boosting	0.9788	0.9984	0.9788	0.9789	0.9788
et	Extra Trees Classifier	0.9762	0.9970	0.9762	0.9763	0.9762
knn	K Neighbors Classifier	0.9743	0.9933	0.9743	0.9748	0.9743
mlp	MLP Classifier	0.9736	0.9978	0.9736	0.9739	0.9735
dt	Decision Tree Classifier	0.9702	0.9799	0.9702	0.9703	0.9702
gpc	Gaussian Process Classifier	0.9641	0.9961	0.9641	0.9642	0.9641
rbfsvm	SVM - Radial Kernel	0.9618	0.9961	0.9618	0.9619	0.9618
lr	Logistic Regression	0.9485	0.9947	0.9485	0.9487	0.9485
svm	SVM - Linear Kernel	0.9394	0.0000	0.9394	0.9417	0.9393
qda	Quadratic Discriminant Analysis	0.9372	0.9953	0.9372	0.9449	0.9369
lda	Linear Discriminant Analysis	0.8963	0.9858	0.8963	0.9020	0.8972
nb	Naive Bayes	0.8930	0.9946	0.8930	0.9162	0.8932
ridge	Ridge Classifier	0.8865	0.0000	0.8865	0.9025	0.8848
ada	Ada Boost Classifier	0.7497	0.9095	0.7497	0.7445	0.6655
dummy	Dummy Classifier	0.2667	0.5000	0.2667	0.0711	0.1123

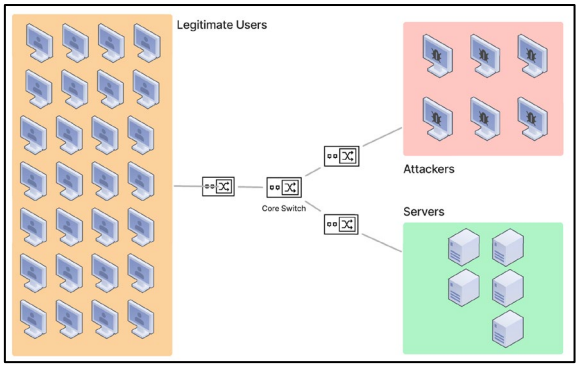
รูปที่ 3.2 ผลลัพธ์การสอนด้วยวิธีการจำแนกต่าง ๆ

4. ผลการทดลองเบื้องต้น

สภาพแวดล้อมของระบบที่ใช้ในการทดลองเป็นเครือข่ายที่ออกแบบตามสถาปัตยกรรมเอสดีเอ็น แบ่งเครือข่ายออกเป็น 2 ส่วน คือ ส่วนควบคุม และ ส่วนรับส่งข้อมูล ถูกจำลองขึ้นบน Virtual Machine 1 เครื่อง ที่ติดตั้งระบบปฏิบัติการ Ubuntu เวอร์ชัน 22.04

4.1 เครือข่ายจำลอง

เครือข่ายที่จำลองขึ้นจะทำงานอยู่ในส่วนรับส่งข้อมูล ถูกจำลองขึ้นโดย Mininet องค์ประกอบภายในเครือข่ายประกอบด้วย OpenFlow Switch จำนวน 4 เครื่อง, เซิร์ฟเวอร์ จำนวน 5 เครื่อง, ผู้ใช้งานอย่างถูกต้อง จำนวน 28 เครื่อง, และผู้โจมตี จำนวน 6 เครื่อง OpenFlow Switch จะทำหน้าที่เป็น Switch ของคอมพิวเตอร์แต่ละกลุ่ม การเชื่อมต่อระหว่าง OpenFlow Switch แต่ละตัวจะเป็น Star Topology และการเชื่อมต่อระหว่างคอมพิวเตอร์ในแต่ละกลุ่มกับ OpenFlow Switch เป็น Star Topology เช่นเดียวกัน ดังรูปที่ 4.1



รูปที่ 4.1 เครือข่ายจำลอง

การจำลองการติดต่อสื่อสารในส่วนรับส่งข้อมูลจะแบ่งออกเป็น การติดต่อสื่อสารแบบปกติ คือการใช้งานบริการของผู้ใช้งานอย่างถูกต้อง และ การโจมตี คือการโจมตีบริการของผู้โจมตี โดยในการติดต่อสื่อสารปกติ ผู้ใช้จะเข้าใช้งานบริการเว็บไซต์ด้วยการส่งคำสั่ง Curl ไปยังเซิร์ฟเวอร์เพื่อร้องขอหน้าเว็บไซต์ และเข้าใช้งานบริการควบคุมระยะไกลด้วยคำสั่ง SSH แต่ในการโจมตี ผู้โจมตีจะใช้งานเครื่องมือสร้างการโจมตีอย่าง Hping3 ในการสแกนหาเป้าหมายและบริการที่ให้บริการอยู่บนเซิร์ฟเวอร์ แล้วโจมตีบริการเหล่านั้นด้วยการสร้างแพ็คเก็ตที่ปลอมแปลงหมายเลขไอพีผู้ส่งจำนวนมากไปยังพอร์ตที่ให้บริการในอัตราต่าง ๆ

4.2 สถานการณ์จำลอง

สถานการณ์จำลองการติดต่อสื่อสารบนเครือข่ายที่สร้างโดย Mininet มีลักษณะการติดต่อสื่อสาร หรือ การโจมตีที่คล้ายคลึงกับการโจมตีจริง โดยมีการติดต่อสื่อสารพื้นหลังเป็นการเข้าใช้บริการเว็บและบริการควบคุมคอมพิวเตอร์จากระยะไกล และสุ่มการติดต่อสื่อสารที่เป็นการดาวน์โหลดไฟล์ของผู้ใช้และการโจมตีทั้งหมด นอกจากนี้ยังแบ่งสถานการณ์จำลองจึงแบ่งออกได้เป็น 3 สถานการณ์ตามความน่าจะเป็นของการเกิดการติดต่อสื่อสารแต่ละประเภท ดังตารางที่ 4.1

ตารางที่ 4.1 สถานการณ์จำลอง

สถานการณ์	โอกาสเกิดการติดต่อสื่อสาร				
	ปกติ		การโจมตี		
	ไม่มีดาวน์โหลด	ดาวน์โหลดไฟล์	สแกนพอร์ต	ดอส	ดีดอส
สถานการณ์ 1	28%	22%	17%	16%	17%
สถานการณ์ 2	40%	30%	10%	10%	10%
สถานการณ์ 3	52%	38%	3%	3%	4%

4.3 ตัวชี้วัดการประเมิน

Confusion Matrix สามารถใช้สรุปประสิทธิภาพของข้อมูลแยกแต่ละประเภท โดยทำ Confusion Matrix ของข้อมูลประเภทที่สนใจให้มี 2 ประเภทข้อมูล คือ Positive และ Negative สำหรับ Positive จะหมายความว่าเป็นข้อมูลประเภทที่สนใจ ส่วน Negative หมายความว่าไม่ใช่ข้อมูลประเภทที่สนใจ เมื่อแบ่งให้ออกเป็นเช่นนี้ ข้อมูลในตารางทั้ง 4 ช่องของ Confusion Matrix จึงถูกกำหนดให้เป็น True Positive (TP), False Negative (FN), False Positive (FP), และ True Negative (TN) โดยที่ True มีความหมายว่าการจำแนกที่เป็น Positive หรือ Negative นั้นถูกต้อง ส่วน False มีความหมายว่าการจำแนกเป็น Positive หรือ Negative นั้นไม่ถูกต้องโครงงานนี้เลือกใช้ตัวชี้วัดการประเมินที่มักถูกใช้ในการวัดประสิทธิภาพของระบบตรวจจับการโจมตีที่ใช้ปัญญาประดิษฐ์ [31][32] ได้แก่ Accuracy, Precision, Recall, F1-Score, Receiver Operating Characteristic Curve (ROC), และ Area under ROC (AUC)

- Accuracy หรือความถูกต้อง คือ ผลการจำแนกที่ถูกต้องทั้งเกิดขึ้นและไม่เกิดขึ้นจากผลการจำแนกทั้งหมด

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (4.1)$$

- Precision หรือความแม่นยำ คือ ผลการจำแนกว่าสิ่งนั้นเกิดขึ้นที่ถูกต้อง เทียบกับผลการจำแนกว่าสิ่งนั้นเกิดขึ้นทั้งถูกต้องและไม่ถูกต้องทั้งหมด

$$\text{Precision: } P = \frac{TP}{TP + FP} \quad (4.2)$$

- Recall คือ ผลการจำแนกว่าสิ่งนั้นเกิดขึ้นที่ถูกต้อง เทียบกับจำนวนตัวอย่างที่สิ่งนั้นเกิดขึ้นทั้งหมด

$$\text{Recall: } R = \frac{TP}{TP + FN} \quad (4.3)$$

- F-Score คือ ค่าเฉลี่ยของความแม่นยำ และ Recall ที่ถูกถ่วงน้ำหนักด้วย β แต่โดยปกติแล้ว F-Score จะหมายถึง F1-Score ซึ่งหมายถึงค่าเฉลี่ยของ Precision และ Recall และเมื่อ $\beta = 1$ F-Score จึงมีวิธีการคำนวณดังนี้

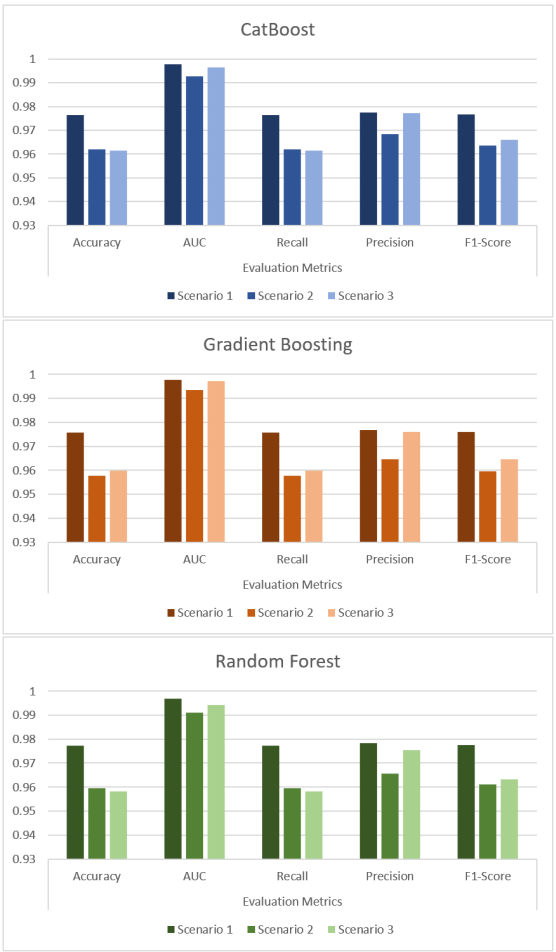
$$\text{F1-Score} = 2 * \frac{P * R}{P + R} \quad (4.4)$$

- Receiver Operating Characteristic Curve (ROC) เป็นกราฟเส้นโค้งที่บ่งบอกความถูกต้องของโมเดล ณ ค่าเกณฑ์จำแนกต่าง ๆ พื้นที่ใต้กราฟของเส้น ROC เรียกว่า Area under the ROC curve หรือ AUC

4.4 ผลการทดลอง

จากรูปที่ 4.16 ซึ่งแสดงกราฟเปรียบเทียบประสิทธิภาพของระบบตรวจจับการโจมตีในสถานการณ์ต่าง ๆ เมื่อใช้โมเดลการจำแนกแต่ละโมเดล แสดงให้เห็นว่าระบบที่ใช้โมเดล CatBoost มีความถูกต้องสูงถึง 0.9763 ในสถานการณ์ที่ 1 ที่มีอัตราส่วนของการโจมตี

มาก คิดเป็น 35 เปอร์เซ็นต์จากการติดต่อสื่อสารทั้งหมด แต่ความถูกต้องกลับลดลงมาให้สถานการณ์ที่ 2 และ 3 ที่มีอัตราส่วนของการโจมตีน้อย คิดเป็น 17 เปอร์เซ็นต์ และ 9 เปอร์เซ็นต์ ตามลำดับ ความสูงของกราฟลดลงอย่างเห็นได้ชัด โดยระบบมีความถูกต้องที่ประมาณ 0.9617 เมื่อสังเกตกราฟที่แสดงประสิทธิภาพของระบบเมื่อใช้โมเดล Gradient Boosting และ Random Forest จะพบว่าลักษณะของแท่งกราฟคล้ายคลึงกับกราฟของระบบเมื่อใช้โมเดล CatBoost โดยโมเดล Gradient Boosting มีความถูกต้อง 0.9757 ในสถานการณ์ที่มีอัตราส่วนของการโจมตีมาก และลดลงอย่างเห็นได้ชัดในสถานการณ์ที่มีอัตราส่วนของการโจมตีลดลง โดยมีความถูกต้องประมาณ 0.9585 เช่นเดียวกันกับโมเดล Random Forest ในสถานการณ์ที่ 1 ระบบมีความถูกต้อง 0.9774 และลดลงไปที่ประมาณ 0.9585 ในสถานการณ์ที่ 2 และ 3 หากสังเกตตาราง Confusion Matrix ของโมเดลทั้ง 3 จะพบว่าระบบมีการจำแนกที่เป็น False Positive และ False Negative มากในประเภทการติดต่อสื่อสารแบบดอสและการติดต่อสื่อสารปกติ เนื่องจากดอสที่มีอัตราแพ็ตเก็ตต่ำนั้นส่งผลต่อค่าฟีเจอร์ทั้ง 6 ค่าในทำนองเดียวกันกับการติดต่อสื่อสารปกติ



รูปที่ 4.2 กราฟประสิทธิภาพระบบตรวจจับการโจมตีแบ่งตามโมเดลการจำแนก

5. บทสรุป

ระบบตรวจจับการโจมตีไซเบอร์บนสถาปัตยกรรมเอสดีเอ็นนี้พัฒนาขึ้นเพื่อศึกษาแนวทางและพัฒนาระบบตรวจจับการโจมตีไซเบอร์ที่จะนำไปใช้กับเครือข่ายแบบเอสดีเอ็น โดยมีเทคโนโลยีปัญญาประดิษฐ์เป็นแนวทางที่สนใจ มีการโจมตีแบบการสแกนพอร์ตและการขัดขวางการให้บริการทั้งดอส และดีดอส เป็นการโจมตีที่ใช้ในการทดสอบระบบ ซึ่งระบบตรวจจับการโจมตีที่พัฒนาขึ้นใช้เครื่องมือด้านการเรียนรู้ของเครื่องที่ชื่อ PyCaret ในการพัฒนาโมเดลการจำแนก สำหรับนำมาใช้กับระบบตรวจจับการโจมตี โดยคัดเลือกเอา 3 โมเดลที่มีประสิทธิภาพสูงที่สุดในการสอนและทดสอบ และเมื่อนำโมเดลที่ได้เหล่านี้ไปทดสอบในสถานการณ์ที่จำลองขึ้น พบว่าโมเดลมีความถูกต้องในการตรวจจับการโจมตีอยู่ในช่วง 95.70 เปอร์เซ็นต์ ถึง 97.70 เปอร์เซ็นต์

เอกสารอ้างอิง

[1] Tripathi N., Mehtre B. **“DoS and DDoS Attacks: Impact, Analysis and Countermeasures.”** December 2013. pp. 1 - 6.

[2] Nivedita J. **“45 Global DDOS Attack Statistics 2023.”** [Online] Available: https://www.getastra.com December 2023.

[3] David J., Thomas C. “Efficient DDoS flood attack detection using dynamic thresholding on Flow-based network traffic”**Computers & Security**. vol. 82, no. 1, May 2019. pp. 284 - 295.

[4] Neu CV, Tatsch CG, Lunardi RC, Michelin RA, Orozco AMS, Zorzo A. **“Lightweight IPS for port scan in openFlow sdn networks”** IEEE/IFIP Network Operations and Management Symposium (NOMS). April 2018. pp. 1 - 6.

[5] Yuchong L., Qinghui L. “A comprehensive review study of cyber-attacks and cyber security; Emerging trends and recent developments” **Energy Reports**. vol. 7, August 2021. pp. 8176 - 8186.

[6] NIST. **“Special Publication 800-30 Revision 1.”** [Online]. Available: https://doi.org/10.6028/NIST.SP.800-30r1. September 2023.

[7] Bendovschi A. “Cyber-Attacks – Trends, Patterns and Security Countermeasures” **Procedia Economics and Finance**. vol. 28, December 2015. pp. 24 - 31.

[8] Yadav T., Rao A. “Technical Aspects of Cyber Kill Chain” **Springer International Publishing**. vol. 536, August 2015. pp. 438 - 452.

[9] Vivo M., Ke L., Isern G., Vivo G. “A review of port scanning techniques” **Computer Communication Review**, vol. 29, April 1999. pp. 41 - 48.

[10] Ono D., Guillen L., Izumi S., Abe T., Suganuma T. "A proposal of port scan detection method based on Packet-In Messages in OpenFlow networks and its evaluation" **International Journal of Network Management**, vol. 31, no. 6, September 2021. pp. 1055 – 7148.

[11] Tasnim K. **"A NEW PROPOSED STACKING GENERALIZATION MODEL FOR DETECTING DDOS ATTACKS IN SDN ENVIRONMENT."** M. Sc. Thesis The Department of Computer Engineering Institute of Graduate Programs,Karabük University. 2023.

[12] Hung-Jen L., Chun-Hung Richard L., Ying-Chih L., Kuang-Yuan T., "Intrusion detection system: A comprehensive review" **Journal of Network and Computer Applications**. vol. 36, no. 1, 2013. pp. 16 – 24.

[13] Shailendra S., Sanjay S. "A Survey of Cyber Attack Detection Systems" **IJCSNS International Journal of Computer Science and Network Security**. vol. 9, no. 5, May 2009.

[14] Bangui H., Ge M., Buhnova B. **"A hybrid machine learning model for intrusion detection in VANET"** Computing. vol. 104, no. 1, 2022.

[15] H. Abdelgader E., Bozed K. A., Younis H. "Software Defined Networking" **2019 19th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA), Sousse, Tunisia.** 2019. pp. 620 - 625.

[16] IBM. **"What is an API."** [Online]. Available: https://www.ibm.com/topics/api

[17] Open Networking Foundation. **"OpenFlow Switch Specification."** [Online]. Available: https://opennetworking.org

[18] Samuel A. L. "Some studies in machine learning using the game of checkers" IBM Journal of research and development. vol. 3, no. 3, July 1959. pp. 210 - 229.

[19] Mitchell T. "Machine learning" McGraw Hill. vol. 7, 1997. pp. 2 - 5

[20] Alzubi J., Nayyar A., Kumar A. "Machine Learning from Theory to Algorithms: An Overview" IOP Publishing. vol. 1142, 2018.

[21] Parvinder S. S., Sunny B., Sajal B. **"Detection of DDoS Attacks using Machine Learning Algorithms"** 2020 7th International Conference on Computing for Sustainable Global Development (INDIACom). 2020. pp. 16 - 21.

[22] Banitalebi D. A., Soltan A. M., Zamani B. F. "The DDoS attacks detection through machine learning and statistical methods in SDN" **The Journal of Supercomputing**. March 2021.

[23] Jose G., Elie F. Kfoury, Jorge C., Gautam S. "A survey on network simulators, emulators, and testbeds used for research and education" **Computer Networks**. vol. 237, no. 1, 2023.

[24] Python. **"What is Python? Executive Summary."** [Online]. Available: https://www.python.org/

[25] Salman O., Elhajj I., Kayssi A., Chehab A. "SDN controllers: A comparative study" **2016 18th Mediterranean Electrotechnical Conference (MELECON).** vol. 1, no. 1, 2016. pp. 1 - 6.

[26] OpenSystems Media. **"A survey of open networking standards, part 2: Open vSwitch."** [Online]. Available: https://embeddedcomputing.com

[27] Balaji, **"Hping3 – Network Scanning Tool – Packet Generator."** [Online]. Available: https://gbhackers.com/

[28] Manso P., Moura J., Serrão C. "SDN-Based Intrusion Detection System for Early Detection and Mitigation of DDoS Attacks" **Information**. vol. 10, no. 1, May 2019.

[29] You X., Feng Y., Sakurai K. **"Packet in Message Based DDoS Attack Detection in SDN Network Using OpenFlow."** Institute of Electrical and Electronics Engineers. vol. 1, no. 1, 2017. pp. 552 - 528.

[30] PyCaret. **"PyCaret 3.0."** [Online]. Available: https://pycaret.gitbook.io/docs.

[31] Dhanya K.A., Sulakshan V., Kartik S., Anjali T., T. Senthil Kumar, T. Gireesh K. **"Detection of Network Attacks using Machine Learning and Deep Learning Models"** Procedia Computer Science. vol. 218, 2023. pp. 57 - 66.

[32] Dalianis Hercules, Dalianis Hercules. **"Evaluation Metrics and Evaluation"** Clinical Text Mining**:** Secondary Use of Electronic Patient Records. May 2018. pp. 45 - 53.

[33] Ting Kai Ming. "Confusion Matrix" **Encyclopedia of Machine Learning and Data Mining.** April 2017. pp. 260 - 260.

[34] Vujovic Zeljko. "Classification Model Evaluation Metrics" **International Journal of Advanced Computer Science and Applications.** vol. 12, July 2021. pp. 599 - 606.

การพัฒนาระบบเพื่อช่วยบริหารจัดการค่าใช้จ่ายของระบบการศึกษาปฏิบัติทางด้านคลาวด์

ภาสกร คุณพิสุทธิ์[1] และ ยุทธนา พงษ์เผือก[2]

คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง กรุงเทพฯ

Emails: [1]63070136@kmitl.ac.th, [2]63070143@kmitl.ac.th

บทคัดย่อ

ในปัจจุบันการเรียนการสอนวิชาภาคปฏิบัติรายวิชา Cloud Computing ของคณะเทคโนโลยีสารสนเทศ สภาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังได้มีการใช้ระบบ Cloud Learning System มาประกอบกับสื่อการเรียนการสอนเป็นหลักและนำมาใช้ในกระบวนการทำโครงงานของนักศึกษาที่ในรายวิชาต่าง ๆ ดังนั้น Cloud Learning System จึงเป็นสิ่งที่ควรให้ความสำคัญ เนื่องจากระบบนี้มีความจำเป็นต่อกระบวนการเรียนรู้และการสอนต่าง ๆ อย่างไรก็ตาม การใช้บริการ Cloud Learning System ก็มีค่าใช้จ่ายที่เพิ่มขึ้นเรื่อย ๆ โดยทางคณาจารย์ได้ตระหนักถึงความสำคัญของการจัดการเกี่ยวกับค่าใช้จ่ายในระยะยาว จึงได้สนใจที่จะทำการวางแผนและคำนวณค่าใช้จ่ายของระบบ โดยใช้แนวคิดทางการเงินที่เรียกว่า FinOps เพื่อหาวิธีในการลดค่าใช้จ่ายโดยที่ประสิทธิภาพของ Cloud Learning System ยังคงทำงานได้ตามระดับการทำงานที่ตั้งไว้ ในกระบวนการศึกษาและพัฒนาระบบนี้ ได้เลือกใช้บริการจาก Amazon Web Service (AWS) เป็น Cloud Learning System หลัก ที่จะใช้ในการเรียนการสอน และได้ทำการศึกษาความเป็นไปได้ในการปรับปรุงระบบให้มีความเสถียรและตอบสนองความต้องการของนักศึกษาและอาจารย์ โดยใช้แนวคิด FinOps เพื่อให้การใช้งานระบบเป็นไปอย่างมีประสิทธิภาพและลดค่าใช้จ่ายให้อยู่ในระดับที่เหมาะสม

คำสำคัญ – FinOps; Cost Control; Cloud Learning System

1. บทนำ

โดยในปัจจุบันผู้ให้บริการระบบคลาวด์โดยส่วนมากมักคิดค่าบริการตาม Service ต่างๆซึ่งมีผลกระทบคือหากต้องการคุณภาพของระบบคลาวด์ที่สูงขึ้นจะต้องใช้ค่าใช้จ่ายที่มากขึ้นตามไปด้วยส่งผลต่องบประมาณที่ต้องใช้มากขึ้นซึ่งโดยในหลายองค์กรงบประมาณนั้นมีจำกัดทำให้ไม่สามารถควบคุมคุณภาพของระบบคลาวด์ได้อย่างต่อเนื่องดังนั้นจึงมีหลักการในการลดค่าใช้จ่ายของระบบคลาวด์และสามารถคงคุณภาพของระบบให้คงเดิมซึ่งก็คือ FinOps ซึ่งเป็นหลักการที่ทำให้ภายในทีมสามารถจัดการค่าใช้จ่ายที่เกิดขึ้นบนระบบคลาวด์ได้ ซึ่งเราจะสามารถเช็คได้ว่าภายในระบบ ณ ตอนนี้มีค่าใช้จ่ายเท่าไหร่บ้าง Serviceแต่ละจุดมีค่าใช้จ่ายเป็นอย่างไรซึ่งจะทำให้เราสามารถควบคุมและจำกัดค่าใช้จ่ายได้ตามที่ต้องการ เราสามารถนำข้อมูลค่าใช้จ่ายทั้งหมดเหล่านี้มาวางแผนในระบบคลาวด์เพื่อควบคุมคุณภาพได้อย่างต่อเนื่อง ซึ่งเราที่มาและความสำคัญจะนำมาใช้กับระบบ Private Cloud Learning System ในการต่อยอดการเรียนรู้ระบบคลาวด์

2. การทบทวนวรรณกรรมที่เกี่ยวข้อง

2.1 Cloud Computing

เป็นระบบคอมพิวเตอร์ หรือเครือข่ายของคอมพิวเตอร์ โดยจะเป็นระบบคอมพิวเตอร์สำหรับให้บริการสำหรับผู้ที่ต้องการเช่าเพื่อการประมวลผล หรือจัดเก็บข้อมูลผ่านเครือข่ายอินเตอร์เน็ตได้ความต้องการของผู้ใช้โดยที่ผู้ใช้ไม่จำเป็นต้องมีการติดตั้งอุปกรณ์ใดๆในการใช้งานคอมพิวเตอร์ และสามารถจัดสรรทรัพยากรของระบบคอมพิวเตอร์ได้ตามความต้องการหรือตามความต้องการใช้งาน

Cloud Computing นั้นจะมีหลักการทำงานอยู่ 2 ส่วน ส่วนของผู้ใช้งานหรือที่เรียกกันว่า Client ประกอบด้วยคอมพิวเตอร์หรือเครือข่ายคอมพิวเตอร์ของผู้ใช้รวมถึงแอปพลิเคชันที่จำเป็นในการเข้าถึงระบบคลาวด์ สำหรับส่วนหลังก็คือส่วนระบบ cloud computing ซึ่งในส่วนนี้จะมี Server Computer และระบบจัดเก็บข้อมูลต่าง ๆ รวมอยู่ ทั้งโปรแกรมคอมพิวเตอร์ระบบที่ใช้ในการประมวลผลข้อมูล ไปจนการเก็บข้อมูลในรูปแบบต่าง ๆ ซึ่งในแต่ละโปรแกรมหรือแอปพลิเคชันก็มักจะถูกเก็บอยู่ในเซิร์ฟเวอร์แต่ละเครื่องของตัวเองที่เรียกว่า Dedicate Server

2.2 Database

กลุ่มข้อมูลขนาดใหญ่ที่ถูกเก็บรวบรวมไว้ที่ใดที่หนึ่ง โดยเป็นข้อมูลที่มีความสัมพันธ์กัน ซึ่งถูกจัดเก็บอย่างเป็นระบบ โดยมีซอฟต์แวร์เข้ามาควบคุมกระบวนการใช้งาน การทำงาน หรือการประมวลผล ทำให้ผู้ใช้สามารถใช้ข้อมูลได้อย่างมีประสิทธิภาพ

2.3 Container

เป็นเทคโนโลยีที่ช่วยในการแยกการใช้งานและการพัฒนาโปรแกรมให้เป็นอิสระ มีประสิทธิภาพและยืดหยุ่น คอนเทนเนอร์ทำให้นักพัฒนาสามารถสร้างและทดสอบโปรแกรมในสภาพแวดล้อมที่เหมือนกันในละอุปกรณ์ได้ โดยไม่ต้องมีปัญหาเรื่องขึ้นอยู่กับซอฟต์แวร์และฮาร์ดแวร์ที่ต่างกัน

2.4 Amazon Web Service(AWS)

Amazon Web Services (AWS) เป็นแพลตฟอร์ม Cloud Computing ที่นำเสนอบริการจากศูนย์ข้อมูลต่างๆทั่วโลกซึ่งในโครงการนี้ระบบที่ผู้จัดทำจะพัฒนานั้นจะเน้นไปที่ Service พื้นฐานของ AWS เป็นหลักที่เป็นสิ่งที่ต้องใช้ในการเรียนการสอนภาคปฏิบัติด้าน Cloud Computing ของทางคณะเทคโนโลยีสารสนเทศ

2.4.1 AWS Elastic Compute Cloud (EC2)

เป็นบริการเว็บที่ให้พื้นที่การประมวลผลที่ปรับเปลี่ยนขนาดได้ในระบบคลาวด์ ซึ่งออกแบบมาเพื่อช่วยให้นักพัฒนาประมวลผลแบบ Web-Scale ได้ง่ายขึ้น อินเทอร์เฟซบริการเว็บของ Amazon EC2 ที่ใช้งานง่ายนี้ช่วยให้ได้รับประสิทธิภาพและกำหนดค่าความสามารถได้โดยมีข้อผิดพลาดน้อยที่สุด โดยจะให้คุณควบคุมทรัพยากรการประมวลผลได้อย่างสมบูรณ์ รวมทั้งเรียกใช้ด้วยการเชื่อมต่อสภาพแวดล้อมการประมวลผลที่ผ่านการรับรองของ Amazon ซึ่ง EC2 เป็นบริการ Infrastructure as a Service (IaaS) ซึ่ง AWS จะให้เราเข้าไปใช้ Server ได้ตามความต้องการ โดยเราสามารถเลือกขนาดของ Instance ได้ตาม Catalog ที่ AWS จัดไว้ให้ ซึ่งบริการ EC2 เป็นที่นิยมมาก เนื่องจากผู้ใช้งานสามารถนำไปใช้ประโยชน์ได้หลากหลายรูปแบบ ไม่ว่าจะเป็นการทำ Website, Database, Application และอื่น ๆ

2.4.2 AWS Simple Storage Service (S3)

เป็น Cloud Storage หรือ Object store ตัวนึง ที่ถูกสร้างขึ้นมาเพื่อ จัดเก็บวัตถุ หรือ ข้อมูลใด ๆ สามารถนำข้อมูลมาวิเคราะห์ และ สามารถเข้าถึงข้อมูลนี้ได้จากทุกที่ ไม่ว่าจะเก็บ website, mobile app หรือพวก data ต่าง ๆ ที่ต้องการ นอกจากเรื่อง storage แล้ว มันก็ยังสามารถ integrate กับบรรดา third-party ต่าง ๆ ได้ อีกทั้งยังสามารถ integrate กับ service อื่น ๆ ที่อยู่ใน AWS ได้

2.4.3 AWS Elastic Load Balancer (ELB)

คือบริการ Load Balancing ที่มีการจัดการอย่างเต็มรูปแบบ ซึ่งจะกระจายปริมาณการใช้งานแอปพลิเคชันขาเข้าโดยอัตโนมัติไปยังเป้าหมายหลายรายการและอุปกรณ์เสมือนทั่วทั้ง AWS และทรัพยากรในองค์กร คุณสามารถใช้เพื่อปรับขนาดแอปพลิเคชันที่ทันสมัยโดยไม่ต้องกำหนดค่าที่ซับซ้อนหรือ API Gateway

2.4.4 AWS Relational Database Service (RDS)

เป็นบริการที่ได้รับการจัดการที่ช่วยให้ผู้ใช้สามารถตั้งค่าใช้งาน และปรับขนาดฐานข้อมูลแบบเชิงสัมพันธ์ในระบบคลาวด์ได้อย่างง่ายดาย บริการนี้จะมอบความจุที่คุ้มค่าและปรับขนาดได้ พร้อมกับจัดการงานการดูแลระบบฐานข้อมูลที่ใช้

เวลานาน Amazon RDS จะช่วยให้คุณเข้าถึงความสามารถของฐานข้อมูล PostgreSQL MySQL MariaDB Oracle SQL Server ซึ่งใน RDS การจัดการฐานข้อมูลดังกล่าวได้รับการจัดการด้วย Managed service ของ RDS ดังนั้นภาระงานของผู้ใช้งานข้อมูลจึงลดลง นอกจากนี้ยังมีฟังก์ชันช่วยลดภาระงานอีกมาก ไม่ว่าจะเป็นการตั้งค่า Snapshot สำรองข้อมูลอัตโนมัติ การตรวจสอบการทำงานด้วย CloudWatch เป็นต้น และRDSยังสนับสนุนการทำ Fail over system ด้วยการทำ Multi Availability Zone เพื่อให้ DB instance พร้อมใช้งานเสมอ

2.5 Cost Optimization in AWS Managed Service

สำหรับ AWS นั้นจะมีอยู่ 3 ขั้นตอนหลักในการ Optimize ค่าใช้จ่ายใน AWS ซึ่งก็คือ ระบุทางการปรับปรุงต้นทุนในสภาพแวดล้อมที่คุณจัดการ, นำเสนอแผนการปรับปรุงต้นทุน และ ช่วยเหลือในการบรรลุการปรับปรุงต้นทุนที่สามารถวัดได้ ซึ่งทั้ง 3 ขั้นตอนจะช่วยในด้านของค่าใช้จ่ายบนคลาวด์ได้เป็นอย่างดี

3.วิธีการดำเนินการวิจัย

3.1 การศึกษาแนวทางในการทำ Cost Optimization ในแต่ละ Service บน AWS

เนื่องจากในระบบที่เราจะพัฒนานี้จะต้องใช้งานร่วมกับ Private Cloud Learning โครงงงานนี้จะมุ่งเป้าไปที่ Service ของ AWS ทีมีการใช้ในการเรียนการสอนภาคปฏิบัติในวิชาเป็นหลัก สำหรับในโครงการนี้ผู้จัดทำได้มีการนำ Service ของ Amazon Web Service มาใช้ในการพัฒนาระบบโดย Service ที่เราจะนำมาใช้มีดังต่อไปนี้

3.1.1 AWS Elastic Compute Cloud (EC2)

โดยจะนำวิธีการใช้งานของ EC2 ทั้งหมดมาใช้ในการคำนวณว่าเราควรจะใช้วิธีการใช้งานแบบไหนถึงจะเหมาะสมต่อรูปแบบงานของแต่ละองค์กรโดยจะมีตัวแปรอื่นๆร่วมด้วยเช่น S3 Storage Class/S3 Glacier, RDS เป็นต้น

3.1.2 AWS Simple Storage Service (S3)

โดยจะนำ Storage Class ทั้ง 7 รูปแบบ มาใช้ร่วมกับตัวแปรอื่นๆในโครงการเพื่อหาวิธีการใช้งานที่เหมาะสมที่สุดโดยที่ทุก Storage Class

3.1.3 AWS Elastic Load Balancer (ELB)

สำหรับ ELB จะมีตัวแปรในการคำนวณค่าใช้จ่ายเป็นการใช้งานรายชั่วโมงและอิงจาก Region ที่ใช้, ปริมาณการใช้งานและ Availability Zone ของผู้ใช้งาน

3.1.4 AWS Relational Database Service (RDS)

สำหรับ RDS จะมีตัวแปรในการคำนวณค่าใช้จ่ายเป็น Database Engine ของ RDS ที่เราต้องการใช้งาน (Aurora, MySQL, PostgreSQL, mariaDB, Oracle, Microsoft SQL Server)

3.2 องค์ประกอบของระบบที่จะพัฒนา

ผู้จัดทำได้ดำเนินการออกแบบระบบในการพัฒนาโดยแบ่งส่วนประกอบออกเป็น 5 ส่วนดังนี้

3.2.1. AWS

เป็นส่วนที่นำ Service ของ AWS มาทำการควบคุมการทำงานในระบบทั้งหมดและปรับเปลี่ยนตัวแปรในการคำนวณค่าใช้จ่าย ซึ่งจะมี Service หลักที่ใช้ในการทำงานคือ AWS EC2, AWS S3, AWS ELB และ AWS RDS ซึ่งเป็นที่ๆไว้สำหรับการติดตั้งระบบเว็บไซต์และเป็นตัวแปรหลักในการคำนวณหาค่าใช้จ่าย

3.2.2. ฐานข้อมูล

เป็นส่วนที่จัดเก็บข้อมูลที่เกี่ยวข้องกับระบบทั้งหมดโดยทำงานผ่านเอนจิน Databaseคือ mariaDB และเชื่อมต่อผ่านระบบเว็บไซต์

3.2.3. Dashboard

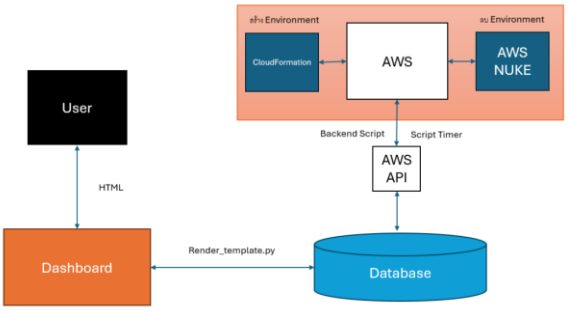
เป็นส่วนที่แสดงผลที่เกี่ยวกับค่าใช้จ่ายทั้งหมดภายในระบบโดยจะอิงตาม User ว่าควรจะมองเห็นได้แค่ไหนและจุดที่สำคัญที่ User ในแต่ละ Case ต้องรู้คืออะไร โดยDashboard นั้นจะเชื่อมต่อผ่านระบบเว็บไซต์โดยผ่าน AWS เพื่อรวบรวมข้อมูลค่าใช้จ่ายมาทำเป็น Dashboard และเปิดเผยอิงจาก Priority ของ User

3.2.4. CloudFormation

เป็นส่วนที่เอาไว้สร้าง Environment ของ Cloud ใน AWS โดยจะใช้ Template อิงจาก AWS Academy ในการสร้างขึ้นมา

3.2.5.AWS Nuke

เป็นส่วนที่เอาไว้ทำลาย Environment บน Cloud



รูปที่ 1 แสดงถึงองค์ประกอบภายในระบบที่จะพัฒนา

3.3 การทำงานของ FinOps ที่ใช้งานในโครงงานนี้

•ใช้เพื่อการคำนวณค่าใช้จ่ายของโครงการให้มีประสิทธิภาพโดยจะต้องอิงควบคู่ไปกับคุณภาพของงานที่เกิดขึ้นด้วยเพื่อให้สามารถควบคุมงบประมาณได้อย่างรัดกุม

•ใช้เพื่อการวางแผนล่วงหน้าว่าควรจะปรับเปลี่ยนรูปแบบของโครงการอย่างไรบ้าง เช่น รูปแบบการเรียนการสอน, รูปแบบ Instance ต่างๆ

•ใช้เพื่อการร่วมมือกันระหว่างผู้ใช้งานและฝั่งงบประมาณให้สามารถควบคุมงานให้ไปในทิศทางเดียวกันและชัดเจน

3.4 หลักการทำงาน

3.4.1 AWS Resource (แสดงด้วยกรอบสีเทา)

ใช้ AWS API เป็นตัวติดต่อกับ CloudFormation เพื่อสร้าง Lab ใน Resource โดยใช้ CloudFormation เป็นเครื่องมือหลักในการจัดการสร้างEnvironment และมีการใช้ AWS Nuke เป็นเครื่องมือในการลบ Resource ที่เราต้องการซึ่งเป็นวิธีที่มีประสิทธิภาพในการลบทรัพยากรที่ไม่ต้องการออกจากระบบ และใช้ Script Python เพื่อ Query ข้อมูลของ Lab เพื่อให้สามารถดึงข้อมูลที่เกี่ยวข้องกับ Environment และประมวลผลข้อมูลอย่างมีประสิทธิภาพ การใช้ Python ในการ Query ข้อมูลช่วยให้เราสามารถจัดการและวิเคราะห์ข้อมูลของ Lab ได้อย่างมีประสิทธิภาพ ซึ่งเราใช้ Protocol AWS API เพื่อติดต่อกับ CloudFormation เพื่อสร้าง Lab ใน AWS Resource โดยใช้ CloudFormation เป็นเครื่องมือหลักในการจัดการสร้าง Environment โดยในกระบวนการนี้ เราใช้ AWS Nuke เพื่อลบ Resource ที่ไม่ต้องการออกจากระบบเพื่อเสริมประสิทธิภาพของการจัดการทรัพยากร เรายังใช้ Python Script เพื่อ Query ข้อมูลของ Lab เพื่อดึงข้อมูลที่เกี่ยวข้องกับ Environment และประมวลผลข้อมูลอย่างมีประสิทธิภาพ ด้วยการใช้ Python เราสามารถจัดการและวิเคราะห์ข้อมูลของ Lab ได้อย่างมีประสิทธิภาพที่สูงขึ้น เพื่อให้การดำเนินการทั้งหมดเป็นไปอย่างมีประสิทธิภาพและมีประสิทธิผลที่ดีที่สุดในการจัดการสภาพแวดล้อมของ AWS Resource

3.4.2 Script Python (แสดงด้วยกรอบสีส้ม)

ใช้เพื่อเพิ่มข้อมูลลงในฐานข้อมูลและเชื่อมต่อกับ MySQL Database Server โดยใช้ 2 Protocol เป็นตัวติดต่อ คือ TCP/IP เพื่อเชื่อมต่อกับ IP ของ MySQL Database และ MySQL Protocol เพื่อการสื่อสารกับ MySQL Database ตาม Script ที่กำหนด เราสามารถใช้ Python เพื่อจัดการการเชื่อมต่อและการสื่อสารกับ MySQL Database ได้อย่างมีประสิทธิภาพและสะดวกสบาย การใช้ 2 Protocol เหล่านี้ให้ความสามารถในการดำเนินการและการสื่อสารอย่างมีประสิทธิภาพระหว่าง

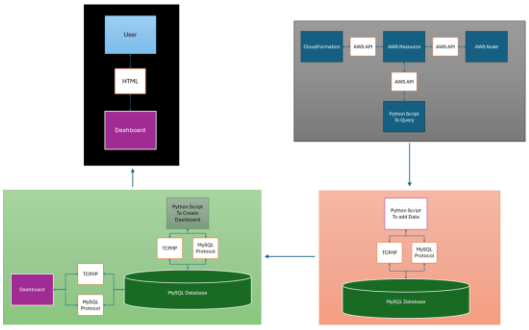
Script Python กับ MySQL Database Server อย่างมีประสิทธิภาพ อย่างไรก็ตาม หากเกิดปัญหาใดๆ กับ Protocol หนึ่ง เช่น TCP/IP หรือ MySQL Protocol จะส่งผลกระทบต่อการใช้งานของ Protocol อีกตัว และจะต้องมีการดำเนินการแก้ไขปัญหาเพื่อให้การทำงานของ Script ไม่เสียหาย ส่วนของ Script Python ที่ใช้เพิ่มข้อมูลลงในฐานข้อมูล MySQL เป็นส่วนที่สร้างความยืดหยุ่นและความสามารถในการจัดการข้อมูลฐานข้อมูลอย่างมีประสิทธิภาพ เราใช้ Python MySQL Connector เพื่อเชื่อมต่อกับ MySQL Database Server โดยใช้ TCP/IP Protocol สำหรับการเชื่อมต่อทางเครือข่าย และใช้ MySQL Protocol สำหรับการสื่อสารกับ MySQL Database Server เพื่อการดำเนินการเกี่ยวกับข้อมูล เช่น เพิ่มข้อมูลลงในตาราง อัปเดตข้อมูล ลบข้อมูล และอื่นๆ โดย Python เป็นภาษาที่มีความยืดหยุ่นและมีชุดคำสั่งที่สะดวกสบายในการจัดการฐานข้อมูล MySQL ทำให้สามารถพัฒนาและปรับปรุง Script ได้อย่างรวดเร็วและง่ายดาย เพื่อให้สามารถปรับแก้ปัญหาหรือปรับปรุงความสามารถของ Script ตามความต้องการของระบบ

3.4.3 MySQL Database และ Dashboard (แสดงด้วยกรอบสีเขียว)

มีการทำงานผ่าน Script Python เพื่อสร้าง Dashboard โดยฐานข้อมูลจะทำงานผ่านการใช้ 2 Protocol ร่วมกัน คือ TCP/IP เพื่อเชื่อมต่อกับ MySQL Database Server และ MySQL Protocol เพื่อติดต่อกับฐานข้อมูลโดยตรง เพื่อทำการ Query ตามคำสั่งของ Script ส่วนด้านของ Database ไปยัง Dashboard ยังคงใช้ Protocol เดียวกันอย่างเดิม ซึ่ง TCP/IP จะใช้ในการเชื่อมต่อกับ MySQL Database และ MySQL Protocol จะใช้ในการดึงข้อมูลเข้าสู่ Dashboard การใช้ Protocol 2 ตัวนี้ช่วยให้สามารถประสานงานระหว่างฐานข้อมูลและ Dashboard อย่างมีประสิทธิภาพและเสถียรได้โดยมีประสิทธิภาพ

3.4.4 User และ Dashboard (แสดงด้วยกรอบสีดำ)

มีการสื่อสารกันผ่าน HTML เพื่อให้ผู้ใช้สามารถเข้าถึงและโต้ตอบกับ Dashboard ได้อย่างมีประสิทธิภาพและสะดวกสบาย การใช้ HTML เป็นช่องทางในการสื่อสารช่วยให้สามารถสร้างอินเตอร์เฟซที่สวยงามและใช้งานได้ง่าย โดยผู้ใช้สามารถเข้าถึง Dashboard ผ่านทางเว็บบราวเซอร์หรือแอปพลิเคชันที่รองรับ HTML ซึ่งทำให้ง่ายต่อการเข้าถึงและใช้งานข้อมูลและฟังก์ชันต่าง ๆ บน Dashboard ในรูปแบบที่สะดวกสบายแก่ผู้ใช้การใช้ HTML เป็นช่องทางในการสื่อสารระหว่าง User และ Dashboard เป็นวิธีที่มีประสิทธิภาพและสะดวกสบายในการสร้างอินเตอร์เฟซที่สวยงามและใช้งานได้ง่าย โดยที่การสื่อสารนี้สามารถทำได้ผ่านการสร้างและปรับแต่ง HTML elements ต่าง ๆ เพื่อให้ผู้ใช้สามารถเข้าถึงและโต้ตอบกับ Dashboard ได้อย่างมีประสิทธิภาพ ซึ่ง HTML elements เหล่านี้สามารถแสดงผลข้อมูลต่าง ๆ และรองรับการโต้ตอบจากผู้ใช้ได้อย่างสมบูรณ์แบบ



รูปที่ 2 แสดงถึงหลักการทำงานทั้งหมดของโครงงาน

3.5 ประเภทของผู้ใช้งานของระบบ

ผู้จัดทำได้ทำการวางระบบโดยได้วางตามความสัมพันธ์ดังนี้

3.5.1. คณะ (Faculty)

โดยคณะจะมีหน้าที่ในฐานะ ADMIN ของระบบคอยสนับสนุนทางด้านการเงินเพื่อให้ทางอาจารย์เป็นคนปรับปรุงระบบและร่วมมือกับทางอาจารย์ในการปรับปรุงเนื้อหาการเรียนการสอนในภาคปฏิบัติและควบคุมงบประมาณตามแผนงานที่ต้องการตาม Dashboard

3.5.2. อาจารย์ (Teacher)

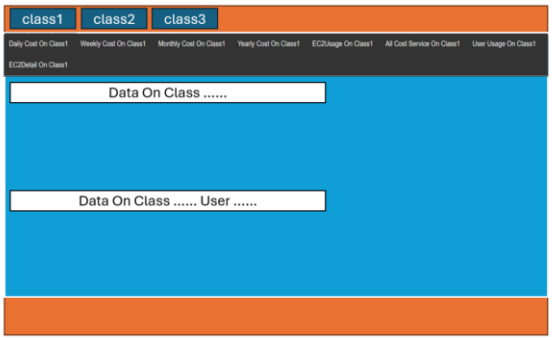
มีหน้าที่เป็นผู้ดูแลออกแบบการใช้งานของระบบทั้งหมดวาง Setting ของระบบและสร้าง Dashboard เพื่อนำมาวิเคราะห์ข้อมูลตามที่ต้องการ ทั้งนี้รวมไปถึงร่วมมือกับทางคณะในการปรับปรุงเนื้อหาการเรียนการสอนในภาคปฏิบัติและควบคุมงบประมาณตามแผนงานที่ต้องการตาม Dashboard

3.5.3. นักศึกษา (Student)

มีหน้าที่เป็น User ซึ่งระบบจะจำกัดการใช้งานและให้ใช้งานระบบตามที่ผู้ดูแลได้กำหนดไว้เท่านั้น ซึ่งสามารถดูข้อมูลค่าใช้จ่ายที่เกี่ยวข้องกับนักศึกษาเองตาม Dashboard

3.6 การออกแบบ Dashboard ของระบบ

โดยที่เราจะจัดอยู่ในรูปแบบนี้กล่าวคือเราจะมี 3 Class ให้กดเพื่อแสดง Class ต่างๆ และมี 8 ปุ่มจาก Class ลงไปในการกดปุ่มเพื่อแสดงข้อมูลต่างๆที่เราต้องการ เมื่อเราต้องการจะดูข้อมูลตรงจุดไหนก็สามารถกดไปยังปุ่มที่ต้องการได้

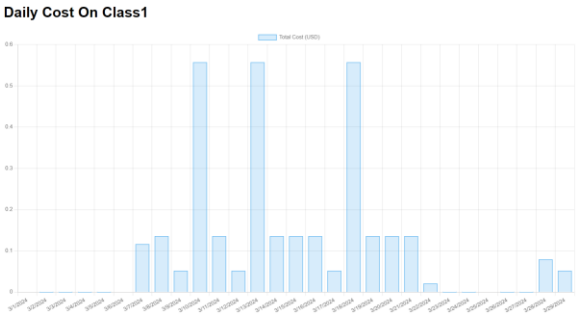


รูปที่ 3 แสดงถึงการออกแบบ Dashboard ของระบบ

ซึ่งเราจะมีการออกแบบให้มี Dashboard ทั้งหมด 8 หน้าซึ่งจะประกอบไปด้วย

3.6.1 Daily Cost

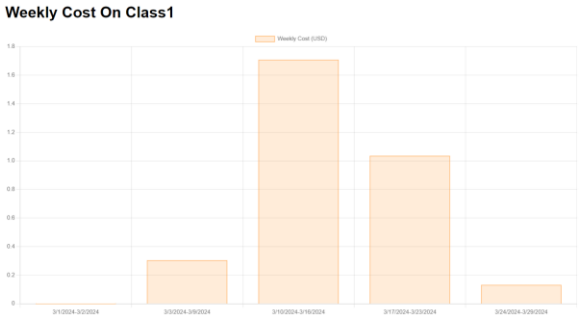
เป็นรายละเอียดของ Cost ในแต่ละวันโดยจะขึ้นเป็นกราฟแสดงให้เห็นถึงค่าใช้จ่ายต่างๆของแต่ละวันว่ามีค่าใช้จ่ายเป็นอย่างไร



รูปที่ 4 Daily Cost

3.6.2 Weekly Cost

เป็นรายละเอียดของ Cost ในแต่ละสัปดาห์โดยจะขึ้นเป็นกราฟแสดงให้เห็นถึงค่าใช้จ่ายต่างๆของแต่ละสัปดาห์ว่ามีค่าใช้จ่ายเป็นอย่างไร



รูปที่ 5 Weekly Cost

3.6.3 Monthly Cost

เป็นรายละเอียดของ Cost ในแต่ละเดือนโดยจะขึ้นเป็นกราฟแสดงให้เห็นถึงค่าใช้จ่ายต่างๆของแต่ละเดือนว่ามีค่าใช้จ่ายเป็นอย่างไร

รูปที่ 6 Monthly Cost

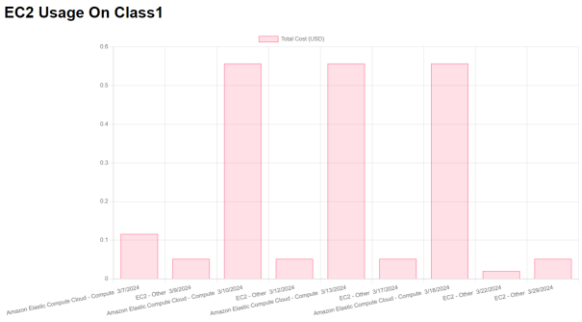
3.6.4 Yearly Cost

เป็นรายละเอียดของ Cost ในแต่ละปีโดยจะขึ้นเป็นกราฟแสดงให้เห็นถึงค่าใช้จ่ายต่างๆของแต่ละปีว่ามีค่าใช้จ่ายเป็นอย่างไร

รูปที่ 7 Yearly Cost

3.6.5 EC2Usage

รายละเอียดของค่าใช้จ่ายของ EC2 และ EC2-Other ของแต่ละวันโดยจะขึ้นเป็นกราฟแสดงให้เห็นถึงค่าใช้จ่ายของ EC2 และ EC2-Other แต่ละวัน



รูปที่ 8 EC2Usage

3.6.6. AllService

รายละเอียดค่าใช้จ่ายของ Service ของแต่ละวันมารวมกันโดยจะแสดงเป็นตารางโดยขึ้นเป็น Service ต่างๆ และยอดรวมของ Service นั้นๆ

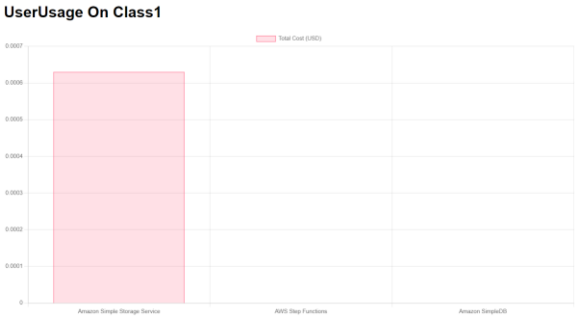
All Service On Class1

Service	Total Cost (USD)
AWS Key Management Service	0.00033
Amazon Simple Queue Service	0.0
AWS Secrets Manager	0.00014
Amazon Simple Storage Service	0.00063
AWS Step Functions	0.0
Amazon Simple Notification Service	0.0
Amazon SimpleDB	0.0
Amazon Elastic Compute Cloud - Compute	1.7864
AmazonCloudWatch	1.16287
EC2 - Other	0.22688

รูปที่ 9 AllService

3.6.7 UserUsage

แสดงถึงข้อมูลค่าใช้จ่ายที่เกิดขึ้นโดยผู้ใช้เองโดยจะแสดงเป็นกราฟที่ขึ้นอยู่กับ Service ที่ผู้ใช้สร้างขึ้นมา



รูปที่ 10 UserUsage

3.6.8 EC2Detail

แสดงถึงข้อมูลรายละเอียด Instance ของ EC2 ว่ามีอะไรบ้างโดยจะแสดงเป็นตารางแสดงถึงข้อมูล EC2 ต่างๆ

EC2Detail On Class1

Instance ID	Instance Type	Launch Time	Cost Per Hour (USD)
i-0fbdeec70e3239bb4	t2.micro	2024-03-07 19:26:53	0.0808
i-00d9228180d034a5c	t2.micro	2024-03-07 19:26:53	0.0808

รูปที่ 11 EC2Detail

3.7 Database Design and Import Data to Database

สำหรับฐานข้อมูลนั้นเราได้ทำการเก็บข้อมูลเป็นตาราง ซึ่งมีรายละเอียดดังนี้

3.7.1. all_cost_data

เป็นตารางสำหรับเก็บข้อมูลของค่าใช้จ่ายของ Service ที่เก็บจากการดึงข้อมูลจากๆไฟล์ .json

3.7.2. ec2_other_instance_data

เป็นตารางสำหรับเก็บข้อมูลของ Service EC2_Other

3.7.3. ec2_instance_data

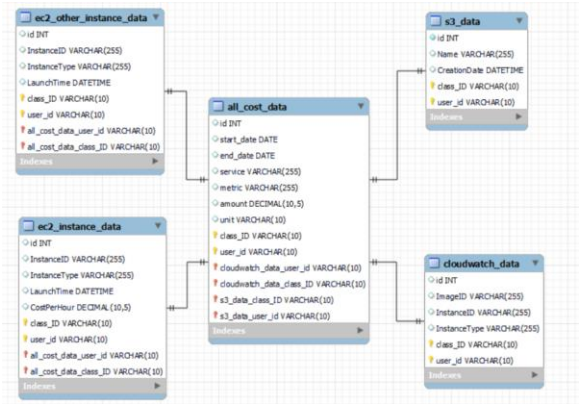
เป็นตารางสำหรับเก็บข้อมูลของ Service EC2

3.7.4. s3_data

เป็นตารางสำหรับเก็บข้อมูลของ Service s3

3.7.5. cloudwatch_data

เป็นตารางสำหรับเก็บข้อมูลของ Service CloudWatch



รูปที่ 12 Diagram แสดงที่จัดเก็บข้อมูล

4. ผลการทดลองเบื้องต้นหรือระบบต้นแบบ

4.1. Deploy

ผู้จัดทำได้ทำการติดตั้งระบบผ่านเครื่องคอมพิวเตอร์ของผู้จัดทำและในระบบ AWS Learning Lab โดยใช้โค้ดที่ได้รับจากงานวิจัยที่แล้วมาทำการติดตั้งซึ่งเราทำการติดตั้งผ่านทาง Docker Compose

สำหรับปัญหาแรกที่เจอคือการติดตั้งระบบเว็บไซต์ซึ่งสิ่งที่ผู้จัดทำเจอก็คือ

• หลังจากติดตั้งไปในช่วงแรกผู้จัดทำได้สังเกตว่ามี ERROR ที่เกิดขึ้นจากการที่ไม่พบไฟล์สกุล env ภายในไฟล์ ซึ่งทางผู้จัดทำได้เจอวิธีการแก้ไขคือการสร้างไฟล์สกุล .env เพื่อใช้เป็นตัวแปรในการติดตั้งระบบ

• หลังจากผ่านการสร้างไฟล์สกุล env มาแล้วก็ยังติดปัญหาในเรื่องของ Port ทีใช้ในระบบซึ่ง ณ ตอนนี้ก็ยังหาวิธีในการติดตั้งให้สมบูรณ์อยู่

4.2. Checking Service

ผู้จัดทำได้สำรวจ Service ของ AWS เพื่อที่จะใช้ในการพัฒนาในระบบของผู้จัดทำ โดยจะทำการศึกษา Service ทั้งหมดแล้วหา Service ที่ต้องการใช้และพัฒนาในระบบของผู้จัดทำซึ่ง Service ที่เราจะใช้จะมีดังต่อไปนี้

• AWS EC2 สร้าง Instance หลักของระบบ

• AWS S3 สร้าง Storage ของระบบ

• AWS RDS สร้าง Database Instance ของระบบ

4.3. Install Tool

ผู้จัดทำได้ติดตั้ง Tools ที่จะใช้ในการ Deploy ระบบ ซึ่งจะเกี่ยวเนื่องกันกับการ Deploy ในระบบซึ่งสำหรับ Tool ที่จะใช้จะมีดังต่อไปนี้

• Docker เอาไว้สำหรับการติดตั้งระบบของเว็บไซต์

• Database สำหรับการใช้งานควบคู่ไปกับ Docker โดยจะเป็นฐานข้อมูลของระบบเว็บไซต์

4.4. Environment Setup For Script

4.4.1. Script Python (ALLSERVICE.PY)

มีการใช้คำสั่ง ce get_cost_and_usage ของ aws cli มาใช้โดยผ่าน script ของภาษา python โดยเราจะมีการดึงค่าใช้จ่ายแบบ Unblended Cost แยกออกเป็นแต่ละวันโดยที่ให้หาว่าในวันดังกล่าวมี Service อะไรบ้างที่มีค่าใช้จ่ายและตอนนี้ค่าใช้จ่ายเป็นเท่าไหร่ โดยจะมี Array ชื่อ service_costs เป็นที่จัดเก็บข้อมูล จากนั้นทำการสร้าง LOOP เพื่อจัดเก็บข้อมูลตามที่เราต้องการนั่นคือเก็บข้อมูล Service ที่มีค่าใช้จ่าย Unblended Cost และเซฟข้อมูลจาก array service_costs เพื่อนำออกมาเป็นไฟล์ .json

```
"2024-03-01": {
    "AWS Key Management Service": {
        "Cost": "0",
        "Currency": "USD"
    },
    "AWS Secrets Manager": {
        "Cost": "0.000005",
        "Currency": "USD"
    },
    "Amazon Simple Notification Service": {
        "Cost": "0",
        "Currency": "USD"
    },
    "Amazon Simple Queue Service": {
        "Cost": "0.0000004",
        "Currency": "USD"
    },
    "Amazon Simple Storage Service": {
        "Cost": "0.0000586513",
        "Currency": "USD"
    },
    "Tax": {
        "Cost": "0.62",
        "Currency": "USD"
    }
}
```


รูปที่ 13 AllService.py Result

4.4.2 Task Scheduler

หลังจากที่เราสร้าง script allservice.py ที่เราต้องการแล้ว เราก็ต้องมาทำการตั้งเวลาเพื่อให้ Script สามารถทำงานตามเวลาที่เรากำหนดได้

รูปที่ 14 Task Scheduler Complete

4.5 Script Python to add data in database and .json file

เป็น Script ในการนำข้อมูลเข้าสู่ Database โดยทางผู้จัดทำได้ใช้ MySQL เป็นฐานข้อมูลในการจัดเก็บข้อมูล โดยในทุก Script จะมีการเชื่อมต่อผ่าน Database สำหรับ Script Python ที่จะใช้มีดังนี้

4.5.1. all_cost.py

สร้างตาราง all_cost_data จากข้อมูลที่ได้ผ่าน blendedcost.json และ unblendedcost.json จากนั้นนำข้อมูลที่ได้จากไฟล์ .json เพิ่มเข้า database โดยอิงตามตารางที่สร้างขึ้นมาจากไฟล์ .json

4.5.2. ec2_detail.py

เป็นการ query ผ่าน aws cli และเซฟเป็นไฟล์ .json โดยใช้ไฟล์ ec2_detail.py ซึ่งจะทำการดึงข้อมูลรายละเอียด instance

4.5.3 ec2_instance.py

ทำการนำข้อมูลของ ec2 ที่เราได้เพิ่มลงไปใน Database จากนั้นทำการสร้างตาราง ec2_instance_data ใน MySQL Database โดยถ้าหากไม่มีตารางใน Schema หรือ Database ที่เราจะลง Script จะสร้างตาราง ec2_instance_data โดยอัตโนมัติ จากนั้นทำการนำข้อมูลใส่ตาม Column ทั้งหมด โดยอิงจากไฟล์ .json

4.5.4 render_template.py

สำหรับ Script Python อันนี้จะใช้ในการดึงข้อมูลของ Database จาก MySQL มาใช้ในการสร้างตาราง Dashboard ผ่านหน้าเว็บโดยเราจะใช้คำสั่ง query ของ MySQL มาเป็นตัวกำหนดข้อมูลที่เราต้องการ

4.6 CloudFormation

สำหรับ CloudFormation ที่เราจะใช้จะมีทั้งหมด 3 Template ซึ่งแต่ละ Template จะอิงตาม AWS Academy โดยแต่ละ Template จะเป็นตัวแทนของ Class นั้นๆ

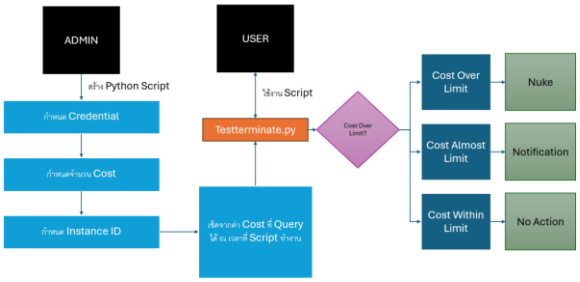
รูปที่ 15 CloudFormation Class1

4.7 AWS Nuke

เป็นชุดโปรแกรมเอาไว้สำหรับการลบข้อมูลหรือ Resource บน Cloud ซึ่งจะใช้ตัวชุดโปรแกรมจากทาง AWS หลักการการทำงานของ AWS-Nuke คือ จะมีการสร้างไฟล์ .yaml เป็นตัวกำหนดการลบ Resource และใช้คำสั่งบน Command Prompt ในการดำเนินการให้ลบ Resource บน Cloud

4.8 Lab Notification

เป็นการสร้างระบบการแจ้งเตือนให้กับผู้ใช้งนหากในกรณีที่ผู้ใช้งานใช้งาน Cloud จนเกินกำหนด Cost ที่เราได้กำหนดเอาไว้ใน Script โดยที่จะส่งข้อความแจ้งเตือนไปยังผู้ใช้ขณะที่กำลังRun Script นี้อยู่ โดยหากมีการใช้งานใกล้เกิน Cost หรือ เกิน Cost ไปแล้วจะมีการแจ้งเป็นข้อความ



รูปที่ 16 Diagram แสดง Process ของ Lab Notification

5. บทสรุป

จากการดำเนินงานตาม Diagram ที่เราได้วางแผนไว้ เราจึงได้ออกมาเป็นดังนี้

• มี code python ที่มีการตั้งเวลาการทำงานผ่าน Task Scheduler ให้ดึงข้อมูลและนำข้อมูลที่ได้ไปใส่ database ใน MySQL อัตโนมัติตามเวลาที่เราตั้งไว้

• dashboard ที่สามารถดูในเรื่องของค่าใช้จ่ายต่างๆ ทั้งของตัว class และ ตัวของ user แยกเป็นรายบุคคล และมี database รวมที่ไว้เก็บข้อมูลทั้งหมด

• มี Lab notification ที่ไว้เช็คว่าค่า cost นั้นเกินกว่าที่กำหนดหรือไม่และมีการนำ nuke มาลบ resource เมื่อค่า cost เกินกำหนด

เอกสารอ้างอิง

[1] Addy Osmani (2023). Yarn-Installation. 10 August 2023, https://classic.yarnpkg.com/lang/en/docs/install/#windows-stable

[2] Amazon Web Services, Inc (2023). AWS Service and Cost. 14 August 2023, https://aws.amazon.com/th/

[3] Amazon Web Services, Inc. (2023). CUDOS-Dashboard. 4 October 2023, https://aws.amazon.com/blogs/awsmarketplace/using-cudos-dashboard-visualizations-aws-marketplace-spend-visibility-optimization/

[4] Christian Schenk (2023). Miktex-Installation. 10 August 2023,

https://miktex.org/download

[5] Docker Inc. (2013-2023). File(.env) error problem. 18 August 2023, https://docs.docker.com/compose/environment-variables/

[6] FinOps Foundation (2023). FinOps Introduction. 19 October 2023, https://www.finops.org/introduction/what-is-finops/

[7] Francis Lavoie (2023). Web server using automatically HTTPS. 11 August 2023, https://caddyserver.com/

[8] IBM (2023). Containers. 18 August 2023,

https://www.ibm.com/topics/containers

[9] J. R. Storment, Mike Fuller. (2019). AWS-FinOps. 16 August 2023, https://learning.oreilly.com/library/view/aws-finops-simplified/9781803247236/

[10] Peter Chung. (2022). Cloud-FinOps. 17 September 2023, https://learning.oreilly.com/library/view/cloud-finops/9781492054610/

[11] Vercel (2023). React Framework for the Web. 12 August 2023,

https://nextjs.org/

[12] VMware (2007-2023). Message Broker. 11 August 2023,

https://www.rabbitmq.c

[13] Harminder Virk (2021). Adonisjs fully functional web app. 10 August 2023, https://adonisjs.com/

[14] AWS(2024). AWS CLI. 24 March 2024, https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html

เว็บไซต์เพื่อการศึกษาสำหรับการออกแบบระบบเครือข่าย

ธีรภพ โพธิ์เกิด

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

Emails: 63070086@kmitl..ac.th

บทคัดย่อ

คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ได้จัดสอนวิชาการออกแบบระบบเครือข่ายเบื้องต้น สำหรับนักศึกษากลุ่มแขนงวิชาเทคโนโลยีเครือข่ายและระบบ โดยใช้โปรแกรม GNS เพื่อจำลองการออกแบบระบบเครือข่าย เพื่อความสะดวกในการจัดส่งงานที่อาจารย์มอบหมาย โดยมีการจัดรูปแบบงานให้อยู่ในคอร์สเพื่อสามารถบริหารจัดการโจทย์ของงานได้ โครงงานนี้เป็นเว็บไซต์สำหรับการศึกษาในการออกแบบระบบเครือข่าย เพื่อพัฒนาเว็บแอปพลิเคชั่นสำหรับตรวจสอบผลลัพธ์ของงานที่อาจารย์มอบหมายให้นักศึกษา และพัฒนาระบบจัดเก็บงานให้อยู่ในรูปของคอร์สที่อาจารย์สามารถบริหารจัดการงานและดูผลลัพธ์งานของนักศึกษาได้ นอกจากนี้ยังสามารถตรวจสอบกิจกรรมการทำงานการส่งงานของนักศึกษาได้

โครงงานนี้ช่วยเพิ่มความสะดวกสบายให้กับอาจารย์และนักศึกษาในการทำงานและการเรียนการสอนได้อย่างมีประสิทธิภาพมากขึ้น

คำสำคัญ – เน็ตเวิค; การเรียนรู้; เว็บไซต์;

1. บทนำ

โครงงานนี้มุ่งเน้นการพัฒนาเว็บแอปพลิเคชั่นสำหรับการศึกษาและการจัดการงานในการออกแบบระบบเครือข่าย โดยเป้าหมายหลักคือการสร้างแพลตฟอร์มที่ช่วยให้อาจารย์และนักศึกษาสามารถสื่อสารและดำเนินงานได้อย่างมีประสิทธิภาพมากยิ่งขึ้น ดังนั้น วัตถุประสงค์ของโครงงานมุ่งเน้นการพัฒนาเว็บไซต์เพื่อการศึกษาและฝึกออกแบบระบบเครือข่าย รวมถึงการจัดการการออกโจทย์และตรวจสอบผลลัพธ์ของงานจากโปรแกรม GNS โดยการพัฒนาเว็บไซต์นี้จะช่วยเพิ่มประสิทธิภาพในการเรียนรู้และการสื่อสารระหว่างอาจารย์และนักศึกษา นอกจากนี้ โครงงานยังมีประโยชน์ต่อการจัดการงานและการตรวจสอบผลลัพธ์ของงานที่มอบหมายให้นักศึกษาด้วย โดยสรุปผลของการศึกษาและพัฒนานี้จะช่วยเพิ่มประสิทธิภาพในการทำงานและการศึกษาในระดับมหาวิทยาลัยในสาขาเทคโนโลยีสารสนเทศแขนงเทคโนโลยีเครือข่ายและระบบได้อย่างมีประสิทธิภาพและเห็นผลดีในระยะยาวและสั้นได้เช่นกัน

2. การทบทวนวรรณกรรมที่เกี่ยวข้อง

วรรณกรรมที่เกี่ยวข้องกับการออกแบบส่วนติดต่อผู้ใช้ (User Interface Design) และการพัฒนาซอฟต์แวร์ด้วย Software Development Lifecycle (SDLC) รวมถึงเทคโนโลยีที่ใช้ในการพัฒนาโปรเจคนี้ด้วย

การทบทวนวรรณกรรมเริ่มต้นด้วยการอธิบายหลักการของการออกแบบส่วนติดต่อผู้ใช้ที่สำคัญ ซึ่งรวม Information Design, Interaction Design, Information Architecture, Visual Design, และ Human Computer Interaction เน้นการจัดรูปแบบข้อมูลให้เข้าใจง่าย การออกแบบประสบการณ์การใช้งานที่เรียบง่ายและมีประสิทธิภาพ การจัดโครงสร้างข้อมูลให้มีระบบ การนำเสนอภาพให้สวยงามและเข้าใจง่าย และการออกแบบให้มีประสิทธิภาพและสะดวกสบายในการใช้งานระหว่างมนุษย์กับคอมพิวเตอร์

หลังจากนั้นจะมีการอธิบายเกี่ยวกับ Software Development Lifecycle (SDLC) ซึ่งเป็นกระบวนการที่มุ่งเน้นการวางแผนและการดำเนินงานในการสร้างซอฟต์แวร์ โดยแบ่งเป็นขั้นตอนต่างๆ เช่น การวางแผน การออกแบบ การดำเนินการ การทดสอบ การติดตั้งใช้จริง และการรักษา โดย SDLC จะช่วยลดความเสี่ยงและประหยัดเวลาในการพัฒนาซอฟต์แวร์ และช่วยให้การส่งมอบซอฟต์แวร์เป็นไปตามเป้าหมายและความต้องการของลูกค้าได้

เมื่อทราบถึง SDLC แล้ว จะได้อธิบายเกี่ยวกับเทคโนโลยีที่ใช้ในการพัฒนาโปรเจคนี้ โดยมี Vue.js เป็น framework สำหรับสร้าง user interface ซึ่งเหมาะสำหรับการทำ Single-Page Applications (SPA) และเป็นเครื่องมือที่ดีในการจัดการกับส่วนแสดงผลของเว็บไซต์ และยังมี Tailwind CSS เป็น CSS Utility Framework ที่ช่วยให้นักพัฒนาสร้าง UI ได้ด้วยตัวเองอย่างรวดเร็วและสามารถปรับแต่งได้ตามต้องการ นอกจากนี้ยังมี Prisma เป็น GraphQL database proxy ที่ช่วยเปลี่ยนฐานข้อมูลให้เป็น GraphQL API และมี PostgreSQL เป็นระบบจัดการฐานข้อมูลเชิงวัตถุ (Object Relational Database Management) ที่เหมาะสำหรับการใช้งานในโปรเจคนี้

ทั้งนี้ยังมี GNS3 Web Client และ GNS3 Server API เป็นเทคโนโลยีที่ใช้ในการจำลองและจัดการข้อมูลการออกแบบระบบเครือข่าย ซึ่งเป็นส่วนสำคัญในโปรเจคนี้ด้วย

3. การวิเคราะห์และการออกแบบ

การวิเคราะห์และออกแบบระบบ เริ่มต้นด้วยการเก็บรวบรวมและวิเคราะห์ข้อมูล จากนั้นทำการพัฒนาและจัดทำเว็บไซต์สำหรับการศึกษาเกี่ยวกับการออกแบบระบบเครือข่าย และเว็บไซต์สำหรับฝึกออกแบบระบบเครือข่ายผ่านการทำโจทย์ตามที่อาจารย์กำหนด สุดท้ายทำการ Deploy เว็บไซต์เพื่อให้ผู้ใช้งานสามารถใช้งานระบบได้จริง

แผนภาพแสดงสถาปัตยกรรมเว็บแอปพลิเคชั่นและโครงสร้างฐานข้อมูลในระบบ ถูกประกาศเพื่อให้เห็นภาพรวมของระบบที่ออกแบบและวางแผนไว้

นอกจากนี้ยังมีคำอธิบายของเว็บไซต์ที่ถูกสร้างขึ้นสำหรับการศึกษาในการออกแบบระบบเครือข่าย ทั้งสำหรับนักศึกษาและอาจารย์ ซึ่งมีฟังก์ชันการทำงานต่างๆ เช่น การแสดงคอร์สการเรียนรู้ การจัดการงานที่มอบหมาย และการตรวจสอบผลลัพธ์ของงานในส่วนของกระบวนการทำงานของเว็บไซต์ เรียงลำดับการทำงานของระบบตั้งแต่การแสดงชุดการเรียนรู้ การแสดงงานที่มอบหมาย และการตรวจสอบผลลัพธ์ของงาน โดยทำการดึงข้อมูลและประมวลผลผ่าน API Server และการเข้าถึงฐานข้อมูล Postgresql โดยเน้นที่การตรวจสอบผลลัพธ์ของงานที่ถูกส่งเข้ามาโดยผู้ใช้งาน โดยใช้การเชื่อมต่อและการส่งคำสั่งไปยังอุปกรณ์ในเครือข่ายเพื่อตรวจสอบคุณสมบัติที่ต้องการ และแสดงผลลัพธ์ที่ได้ให้กับผู้ใช้งานเพื่อการวิเคราะห์และให้คะแนนผลลัพธ์นั้นๆ

4. การพัฒนาระบบ

การพัฒนาระบบต้นแบบของเว็บไซต์โดยให้ผู้ใช้สามารถทำงานตามการมอบหมายต่างๆ ได้โดยมีขั้นตอนการเลือกหน้าที่ต้องการทำงาน และการแสดงผลลัพธ์ของงานที่ผู้ใช้ทำไว้

ในส่วนของระบบต้นแบบเว็บไซต์ มีการแสดงหน้ารวมชุดการเรียนรู้ที่ผู้ใช้มีสิทธิ์เข้าถึง และการเลือกงานที่มอบหมายเพื่อจำลองการออกแบบระบบเครือข่าย นอกจากนี้ยังมีการส่งผลลัพธ์และแสดงอุปกรณ์ในระบบเครือข่ายที่จำลองพร้อมผลลัพธ์การตรวจสอบ

ส่วนการจัดการข้อมูลและการใช้งาน มีหน้าจัดการชุดการเรียนรู้และงานที่มอบหมายซึ่งผู้ใช้สามารถดูและแก้ไขข้อมูลได้ตามต้องการ

ระบบต้นแบบนี้จะช่วยให้ผู้ใช้สามารถทำงานตามการมอบหมายต่างๆ และจัดการข้อมูลในระบบได้อย่างมีประสิทธิภาพ

5. การสรุปผล

สรุปผลโครงงาน 2 เน้นการพัฒนาเว็บไซต์เพื่อการศึกษาสำหรับการออกแบบระบบเครือข่าย เพื่อให้อาจารย์สามารถจัดการเรียนการสอนให้สะดวกยิ่งขึ้น นักศึกษาสามารถฝึกการออกแบบระบบเครือข่ายได้อย่างมีประสิทธิภาพ และผู้ใช้สามารถทดสอบและส่งงานเพื่อแสดงผลลัพธ์ได้

ในส่วนของการพัฒนา โครงงานได้สำเร็จการพัฒนาให้ผู้ใช้สามารถทำงานตามมอบหมายต่างๆ และมีส่วนของการจัดการข้อมูลให้มีประสิทธิภาพ

ทั้งนี้ยังพบปัญหาเกี่ยวกับการใช้ API ของโปรแกรม GNS และความยากลำบากในการเรียนรู้การใช้งาน API นี้ อีกทั้งยังมีปัญหาเกี่ยวกับการกำหนดขนาดของโครงงานที่ขยายได้เพิ่มเรื่อยๆ

ในแผนการดำเนินงาน มีการกำหนดให้โครงงานเสร็จสิ้นภายในวันที่ 15 เมษายน 2567 โดยมีแผนการดำเนินงาน

เกี่ยวกับการจัดการระบบการสร้างโจทย์และการ Deploy Project ลงบน Cloud Server เพื่อให้โครงงานสามารถใช้งานได้จริง ๆ ส่วนนี้จะช่วยให้โครงงานมีความสมบูรณ์และสามารถนำไปใช้งานได้จริงในอนาคต

เอกสารอ้างอิง

[1] BY ICONEXT WRITER “UX/UI Design คืออะไร” [Online].
Avialble : https://iconext.co.th/th/2021/07/19/ux-ui-design-คืออะไร 2021

[2] BY AWS “SDLC คืออะไร” [Online].
Avialble : https://aws.amazon.com/th/what-is/sdlc/ 2023

[3] BY Developer “Vue.Js 101 ฉบับมาเร็ว ไปเร็ววว” [Online].
Avialble : https://www.borntodev.com/c/webdeveloper/vuejs-101-ฉบับมาเร็ว-ไปเร็ววว-5fa8f3c902f5a 2020

[4] BY Product Development “Tailwind CSS คืออะไร?” [Online].
Avialble : https://morphos.is/th/blog/tailwind-css-a-framework-that-makes-dev-work-easier 2021

[5] BY Kimiiz Pongnares “Prisma คืออะไรกันนะ?” [Online].
Avialble : https://medium.com/@kimiiz/prisma-101-สร้าง-graphql-sever-แบบขี้เกียจด้วย-prisma-6fea3cd98932 2018

[6] BY ICONEXT WRITER “PostgreSQL การติดตั้งและใช้งานเบื้องต้น [Part 1]” [Online].
Avialble : https://toomtamdn.medium.com/postgresql-การติดตั้งและใช้งานเบื้องต้น-part-1-418666b6a9fb 2019