กรอบงานการจัดลำดับความสำคัญในการเรียนรู้แบบสหพันธ์สำหรับข้อมูลคลื่นไฟฟ้าสมอง

1.บทนำ

ในปัจจุบันที่ข้อมูลมีบทบาทสำคัญในการพัฒนาโมเดลปัญญาประดิษฐ์ (Artificial Intelligence - AI) การเรียนรู้ของเครื่อง (Machine Learning - ML) ได้กลายเป็นเครื่องมือสำคัญในการฝึกฝนโมเดลด้วยข้อมูล อย่างไรก็ตาม การรวบรวมข้อมูลมหาศาลจากแหล่งต่าง ๆ มายังส่วนกลางเพื่อฝึกฝนโมเดล AI ก่อให้เกิดความท้าทายหลายประการ โดยเฉพาะอย่างยิ่งในด้านความเป็นส่วนตัวของข้อมูล (Data Privacy) โดยการเก็บข้อมูลที่มีความเป็นส่วนตัวนั้นทำให้ผิดกฎหมายต่าง ๆ เช่น PDPA GDPR และการฝึกฝนโมเดลที่ศูนย์กลางโดยใช้ข้อมูลจำนวนมากในคราวเดียวเป็นการกินทรัพยากรส่วนกลางอย่างมาก ดังนั้น เพื่อตอบสนองต่อปัญหาเหล่านี้ การเรียนรู้แบบสหพันธ์ (Federated Learning - FL) จึงถูกพัฒนาขึ้นมา เพื่อเป็นแนวทางที่ช่วยให้สามารถฝึกฝนโมเดลปัญญาประดิษฐ์ได้จากข้อมูลที่กระจายอยู่บนอุปกรณ์ปลายทางต่าง ๆ เป็นการแบ่งปัญหาใหญ่ออกเป็นปัญหาย่อย ทำให้สามารถลดภาระทรัพยากรการคำนวณของส่วนกลางลงไปได้ อีกทั้งยังได้ความเป็นส่วนตัวของข้อมูลจากการที่ไม่ต้องส่งข้อมูลไปยังส่วนกลาง [3]

แม้การเรียนรู้แบบสหพันธ์จะทำให้แก้ปัญหาดังกล่าวได้ แต่ด้วยความซับซ้อนของมันทำให้มีปัญหาต่าง ๆ ตามมา โดยเฉพาะอย่างยิ่งปัญหา ความแตกต่างกันของระบบ (System Heterogeneity) และความแตกต่างกันของข้อมูล (Data Heterogeneity) [3] ปัญหาความแตกต่างกันของระบบเกิดจากความหลากหลายของอุปกรณ์ปลายทางที่มีประสิทธิภาพเช่น การประมวลผล หน่วยความจำ และเสถียรภาพทางเครือข่ายที่ต่างกัน ทำให้เกิดภาวะคอขวดที่อุปกรณ์ประสิทธิภาพต่ำถ่วงกระบวนการฝึกโดยรวม ในขณะที่ปัญหาความแตกต่างกันของข้อมูลเกิดขากการที่ข้อมูลบนอุปกรณ์ปลายทางแต่ละเครื่องมีการกระจายตัวที่ไม่เหมือนกันและไม่เป็นอิสระต่อกัน (Non-IID) ซึ่งส่งผลกระทบอย่ามากต่อความแม่นยำและประสิทธิภาพของโมเดลส่วนกลางที่ได้จากการฝึกฝนด้วยขั้นตอนวิธีพื้นฐาน [1] , [8]

การประยุกต์ใช้การเรียนรู้แบบสหพันธ์กับข้อมูลทางการแพทย์อย่างคลื่นไฟฟ้าสมอง (Electroencephalography - EEG) มีศักยภาพสูงในการพัฒนาการวินิจฉัยและการรักษาโรคทางระบบประสาท เนื่องจากข้อมูลคลื่นไฟฟ้าสมองมีความละเอียดอ่อนและถือเป็นข้อมูลส่วนบุคคล การใช้การเรียนรู้แบบสหพันธ์ จึงเป็นแนวทางที่เหมาะสมในการฝึกฝนโมเดลปัญญาประดิษฐ์โดยไม่ต้องเปิดเผยข้อมูลผู้ป่วย ซึ่งสอดคล้องกฎหมายที่เข้มงวด และ การจัดการกับความแตกต่างของระบบและข้อมูลในบริบทของข้อมูลคลื่นไฟฟ้าสมองที่มีความซับซ้อนและหลากหลายเป็นพิเศษนั้นเป็นสิ่งจำเป็นอย่างยิ่ง

จากความท้าทายข้างต้น การพัฒนากลไกการจัดลำดับความสำคัญ (Scheduling) ที่มีประสิทธิภาพในสภาพแวดล้อม การเรียนรู้แบบสหพันธ์ จึงมีความสำคัญอย่างมาก โดยเฉพาะอย่างยิ่งเมื่อต้องจัดการกับงานหลายประเภทที่เป็นอิสระต่อกัน (Federated Multi-Job Learning) การจัดลำดับความสำคัญที่ดีจะช่วยให้ระบบสามารถใช้ทรัพยากรได้อย่างเหมาะสม ลดเวลาในการลู่เข้าของโมเดล และไม่ลดความแม่นยำโดยรวมของโมเดล โดยไม่เลือกวิธีการที่ทำให้เกิดปัญหาความสำคัญกับข้อมูลจากอุปกรณ์ปลายทางทั้งหมดไม่เท่าเทียมกันอย่างขั้นตอนวิธีแบบละโมบ (Greedy Algorithm) [7]

ดังนั้น โครงงานนี้จึงมุ่งเน้นการพัฒนากรอบการทำงานสำหรับการจัดลำดับความสำคัญ (Scheduling Framework) การพัฒนากรอบการทำงานนี้จะช่วยเพิ่มประสิทธิภาพและความน่าเชื่อถือของระบบการเรียนรู้แบบสหพันธ์ในการวิเคราะห์ข้อมูลคลื่นไฟฟ้าสมองซึ่งจะนำไปสู่การพัฒนาแอปพลิเคชั่นปัญญาประดิษฐ์ทางการแพทย์ที่มีประโยชน์และปลอดภัยมากยิ่งขึ้นในอนาคต โดยจะใช้ประโยชน์จากเครื่องมือและแนวคิดที่ทันสมัย เช่น คูเบอร์นีตีส (Kubernetes - k8s) ในการจำลองและจัดการสภาพแวดล้อมที่มีความแตกต่างกันของระบบและข้อมูล [2] , [5]

2.ทบทวนวรรณกรรม

2.1 นิยามของการเรียนรู้แบบสหพันธ์ (Federated Learning - FL)

การเรียนรู้แบบสหพันธ์ คือ แนวทางการฝึกฝนโมเดลปัญญาประดิษฐ์ที่ช่วยให้เราสามารถเรียนรู้จากข้อมูลที่กระจัดกระจายอยู่บนอุปกรณ์ต่าง ๆ โดยไม่ต้องรวบรวมข้อมูลเหล่านั้นมาที่ส่วนกลาง โดยมีขั้นตอนหลัก ๆ ดังนี้

1) เซิร์ฟเวอร์ส่งโมเดลปัญญาประดิษฐ์ที่ยังไม่ได้รับการฝึกฝน หรือ ได้รับการฝึกฝนขั้นต้นแล้ว ไปยังอุปกรณ์ปลายทาง

2) อุปกรณ์ปลายทางแต่ละเครื่องจะนำโมเดลที่ได้รับมา ซึ่งต่อไปเราจะขอเรียกว่า โมเดลปลายทาง (local model) เราจะนำโมเดลปลายทางไปฝึกฝนต่อด้วยข้อมูลที่แต่ละอุปกรณ์ปลายทางมีอยู่

3) หลังจากอุปกรณ์ปลายทางมีการฝึกโมเดลปลายทางเสร็จสิ้น จะส่งเฉพาะผลลัพธ์ของการปรับปรุงโมเดล เช่น ค่าพารามิเตอร์ที่เปลี่ยนไปจากโมเดลเดิมที่ได้รับมา กลับมายังเซิร์ฟเวอร์ จากนั้นเซิร์ฟเวอร์จะนำผลลัพธ์จากแต่ละอุปกรณ์ปลายทางมารวมกันเพื่อสร้างโมเดลส่วนกลาง (global model) ที่ได้รับการปรับปรุงให้ดีขึ้น จากนั้นจะส่งโมเดลส่วนกลางที่ปรับปรุงนี้กลับไปยังอุปกรณ์ปลายทางเพื่อแทนโมเดลปลายทางเดิม [3]

2.2 ปัญหาความแตกต่างกันของระบบ (System Heterogeneity)

ปัญหาความแตกต่างกันของระบบ เกิดจากการนำอุปกรณ์ปลายทางที่มีประสิทธิภาพ เช่น การประมวลผล ทรัพยากรหน่วยความจำ เสถียรภาพทางเครือข่าย แตกต่างกันอย่างมากมารวมกลุ่มกันเพื่อใช้ในการฝึกฝนโมเดลปัญญาประดิษฐ์ในการเรียนรู้แบบสหพันธ์ [3]

ความแตกต่างเหล่านี้ทำให้กระบวนการฝึกฝนโมเดลโดยรวมไม่มีประสิทธิภาพ เนื่องจากอุปกรณ์ปลายทางที่มีข้อจำกัดจะกลายเป็นคอขวดที่ทำให้การพัฒนาโมเดลปลายทางเป็นไปได้อย่างล่าช้า

2.2.1 การแก้ไขปัญหาความแตกต่างกันของระบบ

2.2.1.1 การสื่อสารแบบไม่พร้อมกัน (Asynchronous Communication)

ในสภาพแวดล้อมที่มีปัญหาความแตกต่างกันของระบบ การสื่อสารแบบพร้อมกัน (Synchronous Communication) จะทำให้เกิดการหยุดชะงักได้ง่าย เนื่องจากต้องรอให้อุปกรณ์ปลายทางทุกเครื่องส่งโมเดลกลับมายังเซิร์ฟเวอร์ แล้วรอเซิร์ฟเวอร์ส่งโมเดลส่วนกลางที่ปรับปรุงแล้วกลับมา จึงจะสามารถทำงานต่อได้ แต่จากปัญหาดังกล่าว ทำให้อุปกรณ์ปลายทางที่มีประสิทธิภาพสูงเกิดคอขวดจากอุปกรณ์ปลายทางที่มีประสิทธิภาพต่ำได้

การนำการสื่อสารแบบไม่พร้อมกันมาใช้แทน จะช่วยขจัดปัญหาคอขวดนี้ได้อย่างมีประสิทธิภาพ เนื่องจากอุปกรณ์ปลายทางแต่ละเครื่องสามารถส่งการโมเดลกลับมายังเซิร์ฟเวอร์ได้ทันทีที่พร้อม โดยไม่ต้องรอให้อุปกรณ์อื่น ๆ ทำงานเสร็จ ทำให้กระบวนการฝึกฝนโดยรวมดำเนินไปได้อย่างราบรื่นและรวดเร็วยิ่งขึ้น แม้ว่าจะมีอุปกรณ์บางเครื่องที่ทำงานช้ากว่าก็ตาม

แม้ว่าการสื่อสารแบบไม่พร้อมกัน จะสามารถกำจัดปัญหาคอขวดที่อุปกรณ์ปลายทางที่มีประสิทธิภาพสูงต้องรออุปกรณ์ที่มีประสิทธิภาพต่ำไปได้ แต่ทำให้เกิดปัญหาใหม่ขึ้นมาจากความล่าช้าในการรับส่งข้อมูลจากอุปกรณ์ที่มีประสิทธิภาพด้านเสถียรภาพทางเครือข่ายต่ำ เนื่องจากอุปกรณ์ที่มีประสิทธิภาพด้านนี้ต่ำจะส่งข้อมูลไปยังเซิร์ฟเวอร์ช้ากว่าอุปกรณ์ที่มีประสิทธิภาพสูง ซึ่งถ้ามองในมุมมองของเซิร์ฟเวอร์ เซิร์ฟเวอร์จะได้ข้อมูลที่ล้าสมัยในการปรับปรุงโมเดล ซึ่งอาจทำให้โมเดลมีประสิทธิภาพแย่ลง และถ้ามองในมุมมองของ

อุปกรณ์ อุปกรณ์จะได้โมเดลส่วนกลางที่มีความล้าสมัย เมื่อเทียบกับโมเดลส่วนกลางปัจจุบันของเซิร์ฟเวอร์ ดังนั้น ความล่าช้าในการรับส่งข้อมูลจึงเป็นความท้าทายหลักของการนำสื่อสารแบบไม่พร้อมกันมาใช้งาน [3]

โดย [10] ได้มีการนำเสนอการคัดเลือกและส่งเฉพาะพารามิเตอร์ของโมเดลปัญญาประดิษฐ์ที่มีการปรับปรุงค่ามากที่สุด แทนที่จะส่งไปทั้งหมด เป็นการลดต้นทุนด้านการสื่อสาร ทำให้สามารถรับ-ส่งข้อมูลได้รวดเร็วขึ้น ซึ่งใช้ได้ดีกับโมเดลปัญญาประดิษฐ์ที่มีขนาดใหญ่เนื่องจากมักมีการปรับปรุงค่าพารามิเตอร์จำนวนเล็กน้อยเป็นจำนวนมาก ทำให้สามารถตัดพารามิเตอร์เหล่านี้ทิ้งไปได้โดยไม่กระทบต่อความแม่นยำสุดท้ายของโมเดลส่วนกลางมากนัก

2.2.1.2 การสุ่มเลือกอุปกรณ์ (Sampling)

ในการเรียนรู้แบบสหพันธ์ ไม่จำเป็นต้องให้อุปกรณ์ปลายทางทุกเครื่องเข้าร่วมกระบวนการฝึกฝนในทุกรอบของการเรียนรู้สัมพันธ์ (FL Round) วิธีการสุ่มเลือกอุปกรณ์จึงถูกนำมาใช้เพื่อแก้ปัญหาความแตกต่างกันของระบบ จากการลดจำนวนอุปกรณ์ปลายทางที่ต้องรอในการฝึกฝนข้อมูล โดยจะแบ่งออกเป็น 2 ประเภท 1) อุปกรณ์จะถูกเลือกให้เข้าร่วมการฝึกฝนโดยเซิร์ฟเวอร์ 2) อุปกรณ์เลือกที่จะเข้าร่วมการฝึกฝนเอง [3]

2.2.1.3 กลไกทนทานต่อความผิดพลาด (Fault-tolerant Mechanism)

ในสภาพแวดล้อมที่มีปัญหาความแตกต่างกันของระบบ อาจมีอุปกรณ์ปลายทางที่มีเสถียรภาพทางเครือข่ายต่ำ หรือ เซิร์ฟเวอร์มีเสถียรภาพทางเครือข่ายต่ำเอง ดังนั้น กลไกทนทานต่อความผิดพลาดจึงมีความสำคัญอย่างมากกับการเรียนรู้แบบสหพันธ์ในการป้องกันไม่ให้ระบบล่มจากเสถียรภาพทางเครือข่ายที่ต่ำ เนื่องจากการเรียนรู้แบบสหพันธ์นั้นเป็นระบบกระจายที่อุปกรณ์ปลายทางหลายเครื่องทำงานร่วมกัน หากมีอุปกรณ์เครื่องใดเครื่องหนึ่งเกิดความผิดพลาด อาจส่งผลกระทบให้อุปกรณ์อื่น ๆ หรือทำให้ระบบหยุดชะงักได้ [3]

2.2.1.4 การจัดสรรทรัพยากรด้วยการเรียนรู้แบบเสริมกำลังเชิงลึก (Deep Reinforcement Learning - DRL)

ใน [9] ได้ระบุถึงปัญหาการนำการเรียนรู้แบบสหพันธ์มาใช้ในระบบบล็อกเชน (Blockchain) ซึ่งแม้บล็อกเชนจะช่วยให้ฝึกฝนโมเดลได้ในสภาวะแบบกระจายศูนย์ แต่ยังมีปัญหาที่อุปกรณ์ปลายทางแต่ละตัวที่เข้าร่วมมีทรัพยากรที่แตกต่างกัน ซึ่งการจัดสรรทรัพยากรที่ไม่เหมาะสมอาจส่งผลต่อประสิทธิภาพโดยรวมของระบบ จึงมีแนวทางแก้ไขคือจัดสรรทรัพยากรด้วย การเรียนรู้แบบเสริมกำลังเชิงลึกโดยใช้ขั้นตอนวิธี Actor-Critic ซึ่งให้ผลลัพธ์โดยรวมดีกว่าการใช้ขั้นตอนวิธีพื้นฐานอย่างขั้นตอนวิธีแบบละโมบ (Greedy Algorithm)

2.2.2 การจำลองสภาพแวดล้อมที่มีความแตกต่างกันของระบบ

เอฟแอลสเคไลซ์ (FLScalize) ใช้ประโยชน์จาก คูเบอร์นีตีส (Kubernetes - k8s) เพื่อสร้างสภาพแวดล้อมที่จำลอง ความแตกต่างกันของระบบ [2] ด้วยคุณสมบัติของคูเบอร์นีตีส ทำให้เราสามารถปรับแต่งประสิทธิภาพ เช่น กำหนดขีดจำกัดของหน่วยความจำ พลังการประมวลผล รวมถึงการตั้งค่าเฉพาะอื่น ๆ ให้กับแต่ละ พ็อด (pod) ซึ่งเป็นหน่วยการทำงานพื้นฐานในระบบ เพื่อให้พ็อดเหล่านั้นมีประสิทธิภาพที่แตกต่างกันไป เสมือนกับอุปกรณ์จริงที่มีประสิทธิภาพต่างกันในเครือข่ายการเรียนรู้แบบสหพันธ์

โดยเฟรมเวิร์กอื่น ๆ เช่น คูบเฟท (KubeFATE) ก็ได้ใช้คูเบอร์นีตีสเป็นแกนหลักในการพัฒนาการเรียนรู้แบบสหพันธ์ เช่นเดียวกัน [5]

2.3 ปัญหาความแตกต่างกันของข้อมูล (Data Heterogeneity)

ปัญหาความแตกต่างกันของข้อมูล เกิดจากการที่อุปกรณ์ปลายทางแต่ละอุปกรณ์ มีข้อมูลที่แตกต่างกัน ซึ่งข้อมูลคือรากฐานสำคัญในการฝึกฝนโมเดลปัญญาประดิษฐ์ [3]

ในการเก็บข้อมูลโดยให้อุปกรณ์ปลายทางเป็นมือถือ ผู้ใช้มือถือแต่ละเครื่องมีการใช้งานมือถือที่แตกต่างกัน ทำให้การฝึกฝนโมเดลปลายทางให้ผลลัพธ์ที่ต่างกัน

ซึ่งเราจะเรียกข้อมูลเหล่านี้ว่า ข้อมูลที่มีการกระจายของที่ไม่เหมือนกันและไม่เป็นอิสระต่อกัน (Non-Independent and Identically Distributed - Non-IID) [1]

เมื่อมีการรวบรวมข้อมูลที่มีการกระจายตัวไม่สม่ำเสมอจากอุปกรณ์ปลายทางหลายฝ่ายเพื่อฝึกโมเดลด้วยขั้นตอนวิธีพื้นฐานอย่าง FedAvg จะส่งผลกระทบต่อความแม่นยำของโมเดลส่วนกลางอย่างมาก ซึ่งสาเหตุที่ความแม่นยำลดลงนี้มาจากความคลาดเคลื่อนของน้ำหนัก (Weight Divergence - WD) ซึ่งเป็นปรากฏการณ์ที่ค่าน้ำหนักหรือพารามิเตอร์ของโมเดลที่ฝึกบนอุปกรณ์ปลายทางแต่ละเครื่องเบี่ยงเบนจากทิศทางของโมเดลส่วนกลางที่ดีที่สุด เนื่องจากอุปกรณ์ปลายทางแต่ละตัวเห็นข้อมูลที่เอนเอียงไปคนละทิศคนละทาง [8]

2.3.1 วิธีการแก้ไขปัญหาความแตกต่างกันของข้อมูล

2.3.1.1 การเรียนรู้แบบสหพันธ์หลายงาน (Federated Multi-Task Learning - FMT)

สร้างโมเดลปัญญาประดิษฐ์ที่แตกต่างกันสำหรับแต่ละงาน (task) โดยเฉพาะ ทำให้เป็นวิธีการที่เหมาะสมในการแก้ปัญหาความแตกต่างกันของข้อมูล [3]

เฟรมเวิร์ก MOCHA [1] ที่ใช้วิธีการปรับปรุงค่าของเมทริกซ์ความสัมพันธ์ และ น้ำหนักของโมเดล โดยเป็นการทำวนซ้ำ 2 ขั้นตอน

1) อุปกรณ์ปลายทางจะทำการฝึกฝนโมเดลโดยการปรับปรุงค่าน้ำหนักของโมเดลปลายทางของตัวเองโดยใช้เมทริกซ์ความสัมพันธ์เป็นตัวชี้นำในการปรับปรุงค่า หลังจากฝึกฝนเสร็จสิ้น ส่งโมเดลกลับไปให้เซิร์ฟเวอร์

2) เมื่อเซิร์ฟเวอร์ได้รับโมเดลที่ได้รับการฝึกฝนแล้ว จะนำน้ำหนักของโมเดลที่ได้จากโมเดลปลายทางมาเพื่อปรับปรุงค่าของเมทริกซ์ความสัมพันธ์ จากนั้นส่งเมทริกซ์ความสัมพันธ์ที่ได้ปรับปรุงแล้วไปให้อุปกรณ์ปลายทาง และวนซ้ำขั้นตอนที่ 1) จนกว่าโมเดลจะมีการลู่เข้าหรือได้ผลลัพธ์ตามต้องการ

2.3.1.2 การเรียนรู้แบบสหพันธ์หลายงานโดยแต่ละงานเป็นอิสระต่อกัน (Federated Multi-Job Learning - FMJ)

FMJ ต่างจาก FMT โดย FMT จะมีเมทริกซ์ความสัมพันธ์ที่แสดงให้เห็นถึงความสัมพันธ์ระหว่างแต่ละงาน ในขณะที่ FMJ จะมองแต่ละงานแยกออกจากกันโดยสิ้นเชิง โดยมีความท้าทายคือ แต่ละอุปกรณ์ปลายทางต้องฝึกทุกโมเดลปัญญาประดิษฐ์ โดยจำนวนของโมเดลเท่ากับงาน ทำให้ต้องใช้ทรัพยากรมหาศาล ดังนั้นจึงต้องมีแนวทางการจัดลำดับความสำคัญ (scheduling) เข้ามาเกี่ยวข้อง โดยจาก [7] ได้มีการนำการเรียนรู้แบบเสริมกำลัง (Reinforcement Learning - RL) เป็นแนวทางสำหรับงานที่ซับซ้อน และ การปรับค่าให้เหมาะสมแบบเบย์เซียน (Bayesian Optimization) เป็นแนวทางสำหรับงานที่ซับซ้อนน้อยกว่า เมื่อทดสอบโดยใช้โมเดล Support Vector Machine และ Convolutional Neural Network เพื่อวัดความแม่นยำของข้อมูลบนชุดทดสอบ พบว่ามีประสิทธิภาพกว่าการใช้วิธีพื้นฐานอย่างขั้นตอนวิธีแบบละโมบ (Greedy algorithm) อย่างมีนัยสำคัญ เนื่องจากการเลือกแต่อุปกรณ์ที่ประสิทธิภาพดีที่สุด ไม่ได้ให้ความสำคัญกับข้อมูลจากอุปกรณ์ปลายทางทั้งหมดอย่างเท่าเทียมกัน ส่งผลให้ความแม่นยำ (accuracy) สุดท้ายลดลงเมื่อเทียบกับ 2 ดังกล่าว

2.3.1.3 กลยุทธ์แบ่งปันข้อมูลที่ส่วนกลาง (Data Sharing Strategy)

มุ่งเน้นในการแก้ปัญหาข้อมูลเบ้สุดขั้ว (highly skewed) เช่นแต่ละอุปกรณ์ปลายทางมีข้อมูลของเพียงคลาสเดียว โดยจะสร้างข้อมูลส่วนกลางขนาดเล็กซึ่งมีการกระจายตัวของทุกคลาส อย่างสม่ำเสมอ (uniform distributed) โดยก่อนจะเริ่มกระบวนการการเรียนรู้แบบสหพันธ์ จะส่งส่วนหนึ่งของข้อมูลนี้ (ประมาณ 5%) ไปยังอุปกรณ์ปลายทางแต่ละเครื่อง ซึ่งอุปกรณ์ปลายทางก็จะใช้ข้อมูลที่ได้นี้ รวมกับข้อมูลส่วนตัวของตนในการฝึกฝนโมเดล ผลลัพธ์ที่ได้คือความแม่นยำ (accuracy) เพิ่มขึ้นจาก 44% เป็น 74% [8]

2.3.2 การจำลองสภาพแวดล้อมความแตกต่างกันของข้อมูล

เอฟแอลสเคไลซ์ (FLScalize) ทำได้โดยใช้เทคนิคการแบ่งข้อมูล (Data Partitioning) โดยมีแนวคิดหลักคือการกำหนดรหัสระบุตัวตนที่ไม่ซ้ำกันให้แต่ละไคลเอนต์ (client) ซึ่งทำให้สามารถจำลองสถานการณ์ที่แต่ละไคลเอนต์มีข้อมูลที่มีการกระจายของข้อมูลที่ไม่เหมือนกันและไม่เป็นอิสระต่อกัน โดยการที่ไคลเอนต์มีรหัสระบุตัวตนนี้ทำให้เราสามารถจงใจทำให้ไคลเอนต์บางตัวมีข้อมูลของบางคลาสมากผิดปกติได้ ทำให้สามารถจำลองสภาพแวดล้อมที่มีปัญหานี้ได้ [2]

2.4 การจัดลำดับความสำคัญ (scheduling)

มีการประเมินประสิทธิภาพของการใช้นโยบายการจัดลำดับความสำคัญ (scheduling policy) ต่าง ๆ โดยการประเมินจากอัตราการลู่เข้าสู่คำตอบ (convergence rate) โดยวัดจากจำนวนรอบการสื่อสาร ยิ่งใช้น้อยรอบในการไปถึงคำตอบลู่เข้า ยิ่งแปลว่านโยบายที่ใช้มีประสิทธิภาพสูง [6]

2.5 การนำการเรียนรู้แบบสหพันธ์มาใช้กับข้อมูลทางการแพทย์

ต้องหาวิธีในการแก้ปัญหาการทำวิศวกรรมย้อนกลับ (reverse engineering) การฝึกโมเดลด้วยการเรียนรู้สหพันธ์แบบทั่วไป อาศัยการส่งค่าเกรเดียนต์ (gradient) จากอุปกรณ์ปลายทางไปยังเซิร์ฟเวอร์ ซึ่งข้อมูลเหล่านี้ยังคงถูกใช้เพื่ออนุมานข้อมูลดิบที่เป็นความลับของผู้ใช้ได้ จึงได้เสนอแนวทางการใช้ ขั้นตอนวิธีเชิงวิวัฒนาการ (Evolutionary Algorithm) เป็นเทคนิคการปรับค่าให้เหมาะสม (optimization) โดยไม่ต้องใช้ค่าเกรเดียนต์เลย ซึ่งผลลัพธ์ที่ได้แม้จะมีความแม่นยำ (accuracy) ที่ต่ำกว่าการใช้หลักการพื้นฐานอย่างการแพร่กระจายย้อนกลับ (Backpropagation) เล็กน้อย แต่ก็แสดงให้เห็นว่าการฝึกโมเดลพื้นฐานโดยไม่ใช้เกรเดียนต์นั้นเป็นไปได้จริง [4]

ซึ่งมีคนจำนวนไม่น้อยที่กังวลกับการทำวิศวกรรมย้อนกลับนี้ แต่คนส่วนมากเลือกใช้การเข้ารหัสแบบโฮโมมอร์ฟิก (Homomorphic Encryption) กัน ซึ่งเป็นวิธีที่ทำให้สามารถปรับปรุงค่าของโมเดลส่วนกลางได้โดยไม่ต้องถอดรหัส ซึ่ง คูบเฟท (KubeFATE) ใช้วิธีนี้ด้วยเช่นกัน และการเติมสัญญาณรบกวน (noise) ลงไปในพารามิเตอร์ เพื่อไม่ให้โมเดลส่วนกลางมีการเปลี่ยนแปลงพารามิเตอร์จากการใช้ข้อมูลของอุปกรณ์ปลายทางเดียวอย่างมีนัยสำคัญ [3] , [5]

เอกสารอ้างอิง

[1] Smith, V., Chiang, C.-K., Sanjabi, M., & Talwalkar, A. (2017). Federated Multi-Task Learning. arXiv. https://arxiv.org/abs/1705.10467

[2] Yang, S., Moon, J., Kim, J., Lee, K., & Lee, K. (2023). FLScalize: Federated Learning Lifecycle Management Platform. IEEE Access, 11, 48118–48131. https://doi.org/10.1109/ACCESS.2023.3275439

[3] Zhang, C., Xie, Y., Bai, H., Yu, B., Li, W., & Gao, Y. (2021). A survey on federated learning. Knowledge-Based Systems, 216, 106775. https://doi.org/10.1016/j.knosys.2021.106775

[4] Szegedi, G., Kiss, P., & Horváth, T. (2019). Evolutionary federated learning on EEG-data. In ITAT 2019. Retrieved from http://star.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-2473/paper14.pdf

[5] Federated AI. (2025). KubeFATE. [GitHub repository]. Retrieved from https://github.com/FederatedAI/KubeFATE

[6] Yang, H. H., Liu, Z., Quek, T. Q. S., & Poor, H. V. (2019). Scheduling Policies for Federated Learning in Wireless Networks. arXiv. https://arxiv.org/abs/1908.06287

[7] Zhou, C., Liu, J., Jia, J., Zhou, J., Zhou, Y., Dai, H., & Dou, D. (2021). Efficient Device Scheduling with Multi-Job Federated Learning. arXiv. https://arxiv.org/abs/2112.05928

[8] Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., & Chandra, V. (2018). Federated Learning with Non-IID Data. arXiv. https://doi.org/10.48550/arXiv.1806.00582

[9] Liu, H., Zhou, H., Chen, H., Yan, Y., Huang, J., Xiong, A., Yang, S., Chen, J., & Guo, S. (2023). A Federated Learning Multi-Task Scheduling Mechanism Based on Trusted Computing Sandbox. Sensors, 23(4), 2093. https://doi.org/10.3390/s23042093

[10] Thonglek, K., Takahashi, K., Ichikawa, K., Nakasan, C., Leelaprute, P., & Iida, H. (2022). Sparse Communication for Federated Learning. In 2022 IEEE International Conference on Federated Learning in Intelligent Systems and Applications (ICFEC). https://doi.org/10.1109/ICFEC54809.2022.00008