**บทที่ 3**

**วิธีการดำเนินการวิจัย**

**3.1. การศึกษาแนวการคิดการดำเนินงานวิจัยเก่าเพื่อการต่อยอดงานวิจัยใหม่**

**3.1.1. การนำกฎของไฟร์วอลล์มาใช้อ้างอิงในการสร้างชุดข้อมูลฝึกสอน**

ไฟร์วอลล์โดยทั่วไปจะมีกฎไฟร์วอลล์เพื่อใช้สำหรับคัดกรองข้อมูลแพ็คเกต ซึ่งจะเป็นเรื่องที่ง่ายกว่าถ้าหากนำกฎของไฟร์วอลล์มาเป็นตัวอ้างอิงในการสร้างชุดข้อมูลฝึกสอนซึ่งมีมาตรฐานและมีความน่าเชื่อถือมากกว่าการนำข้อมูลการโจมตีที่เคยเกิดขึ้นมาก่อน โครงงานเดิมนั้นได้มีการนำกฎไฟร์วอลล์แต่ละกฎมาสร้างชุดข้อมูลฝึกสอนโดยการอ่านและประมวลผลทีละกฎไฟร์วอลล์เพื่อสร้างชุดข้อมูลฝึกอบรมที่มาจากกฎนั้นๆ โดยจำนวนชุดข้อมูลที่สร้างขึ้นจะแบ่งออกเป็น จำนวนชุดข้อมูลฝึกสอนต่อ 1 กฎไฟร์วอลล์ และจำนวนชุดข้อมูลฝึกสอนทั้งหมดของกฎไฟร์วอลล์ทั้งหมดรวมกัน

**3.1.2. ขั้นตอนการดำเนินงานวิจัยเก่าโดยรวม**

ชุดข้อมูลฝึกสอนโมเดลที่สร้างขึ้นที่มีการอ้างอิงจากกฎของไฟร์วอลล์จะถูกสร้างขึ้นมาหลายชุดด้วยกันเพื่อนำไปฝึกฝนโมเดล โดยข้อมูลฝึกสอนแต่ละชุดนั้นสำหรับโมเดลจะมีจำนวนแพ็คเกตฝึกสอนต่อ 1 กฎไฟร์วอลล์ (Sample per Rule) ที่แตกต่างกัน หลังจากได้โมเดลที่ผ่านการเรียนรู้แล้วให้นำโมเดลมาเปรียบเทียบประสิทธิภาพในด้านต่างๆด้วยชุดข้อมูลทดสอบชุดเดียวกันเพื่อหาและสรุปผลว่าโมเดลที่เรียนรู้จากชุดข้อมูลฝึกสอนที่จำนวนใดสามารถให้ค่าที่ดีที่สุด

Diagram

Description automatically generated

ระบบการทำงานหรือวัฎจักรการพัฒนาโมเดลของงานวิจัยเก่า

**3.1.3. การออกแบบขอบเขตของแพ็คเกตและโครงสร้างที่ใช้**

โครงสร้างของข้อมูลฝึกสอนที่ใช้ทุกชุดจะประกอบไปด้วย 2 ส่วน ได้แก่ ส่วนของการตัดสินใจ (Decision Field หรือ Action Field) เป็นส่วนที่ทำให้โมเดลทราบได้ว่าแพ็คเกตถูกยอมปล่อยให้ผ่านหรือไม่สามารถปล่อยให้ผ่านไปได้ โดยตัดสินจากส่วนของข้อมูลดิบ (Data Field) ที่มีการจำลองอยู่ในรูปแบบโครงสร้างของ Packet Header แปลงให้อยู่ในรูปแบบของเลขฐานสอง โมเดลจะมีการคำนวณและแปลงข้อมูลดิบให้อยู่ในรูปแบบของค่าน้ำหนัก เพื่อหาความสัมพันธ์ระหว่างกันเพื่อหาคำตอบที่ตรงกับ Action Field มากที่สุด

ในส่วนของ Decision Field หรือ ส่วนของการตัดสินใจของไฟร์วอลล์ เป็นดังนี้

* Action (1) – Allow or Deny

ในส่วนของ Data Field หรือแพ็คเกตได้มีการนำ Field จาก Packet Header มาใช้ดังนี้

* Direction (1)
* Interface (3)
* Source IP Address (32)
* Subnet Mask (32)
* Source Port (16)
* Destination IP Address (32)
* Subnet Mask (32)
* Destination Port (16)
* Protocol (8)

จาก field ของ Packet Header ที่ใช้ทำให้รวมกันทั้งหมดใน 1 แพ็คเกตหรือชุดข้อมูล Train set 1 ชุด ประกอบไปด้วย Data Field 9 ส่วน และมีเลข binary ทั้งหมด 172 ตัวเลข ซึ่งค่าเหล่านี้จะต้องนำไปอ้างอิงในหลักการเลือกใช้โมเดลโครงข่ายประสาทเทียมในขั้นตอนต่อไปเพื่อหาโมเดลที่เหมาะสมสำหรับข้อมูล

และเมื่อได้ทำการออกแบบโครงสร้างของแต่ละแพ็คเกตแล้วก็จะเป็นการกำหนดขอบเขตของแพ็คเกต ซึ่งก็คือการกำหนดกฎของไฟร์วอลล์เนื่องจากแพ็คเกตที่ถูกสร้างขึ้นนั้นจะมีการอ้างอิงมาจากกฎของไฟร์วอลล์ ในที่นี้ถ้าหากเรากำหนดขอบเขตหรือไฟร์วอลล์ให้มีขนาดความกว้างมากหรือไม่ครอบคลุมอาจจะทำให้เราต้องการชุดข้อมูลฝึกสอนที่มากขึ้น และอาจทำให้วิเคราะห์ผลลัพธ์ได้ยาก ยกตัวอย่างเช่น Default Rule ที่แพ็คเกตสามารถเป็นอะไรก็ได้ในระบบเครือข่ายถ้าไม่ได้ตั้งค่ากฎอื่นเอาไว้ โดยในงานวิจัยได้กำหนดกฎของไฟร์วอลล์ไว้ดังนี้

Table

Description automatically generated  
การออกแบบกฎไฟร์วอลล์ของงานวิจัยเดิม

**3.1.4. อัลกอริทึ่มสำหรับการสร้างชุดข้อมูลฝึกสอน**

สำหรับการสร้างชุดข้อมูลฝึกสอนจะดำเนินการโดยโปรแกรม Packet Generator ที่สร้างขึ้นเองเพื่อให้สามารถอ้างอิงและปรับเปลี่ยนให้เข้ากับกฎของไฟร์วอลล์ที่กำหนดได้ รูปแบบ Format ของข้อมูลฝึกสอนมีการถอดแบบเสมือน Packet Header ที่ดึงค่าที่ต้องการไว้และจับแยกให้อยู่ในแต่ละ Data Field ตามหมวดหมู่

ชุดข้อมูลฝึกสอนสร้างโดยการสุ่มจากกฎของไฟร์วอลล์แต่ละกฎ และนำมารวมกันในภายหลัง โดยจำนวนที่ต้องการสร้างในแต่ละกฎจะมีการตั้งค่าไว้แล้วในตัวของโปรแกรมโดยอัลกอริทึ่มที่ใช้จะอยู่ในรูปแผนผังการทำงาน ดังนี้

Text

Description automatically generated  
ตัวอย่างอัลกอริทึ่มการสร้างชุดข้อมูลฝึกสอน (Sample per Rule)

Diagram

Description automatically generated

อัลกอริทึ่มการสร้างชุดข้อมูลฝึกสอนจากไฟร์วอลล์ที่มี 3 กฎ (Sample per Rule)

และเพื่อให้สามารถนำไปใช้ต่อได้ จะเป็นการบันทึกข้อมูลให้อยู่ในรูปแบบไฟล์เอกสารให้โมเดลที่สร้างไว้สามารถประมวลผลต่อได้

**3.1.5. หลักการออกแบบโครงสร้างของโมเดล**

การเลือกโครงสร้างและชนิดของโมเดลสำหรับทำปัญญาประดิษฐ์ประสาทเทียมเชิงลึกเป็นอีกส่วนหนึ่งที่สำคัญในงานวิจัย การเลือกโมเดลที่เหมาะสมขึ้นอยู่กับรูปแบบ จุดประสงค์และโครงสร้างของข้อมูลฝึกอบรม ซึ่งโครงข่ายประสาทเทียมหรือโมเดลมีหลายแบบให้เลือกด้วยกัน ในงานวิจัยนี้เน้นความเรียบง่าย และเพื่อให้เป็นจุดเริ่มต้นสำหรับการศึกษาจึงเลือกโมเดลที่นิยมใช้แพร่หลาย ทำให้โมเดลโครงข่ายประสาทเทียมเป็นแบบจำลอง Multi-Layer Perception (MLP) ประกอบไปด้วย 1 Input Layer, 3 Hidden Layers และ 1 Output Layer แต่ละ Hidden Layer ประกอบไปด้วย 250 โหนด Output Layer มี 2 โหนดประกอบไปด้วย Allow และ Deny

Diagram

Description automatically generated

**3.1.6. การออกแบบชุดข้อมูลทดสอบ**

การออกแบบชุดข้อมูลทดสอบถูกแบ่งออกเป็นสามชุด เพื่อวิเคราะห์หาประเด็นความถูกต้องของการทำงานว่าเกี่ยวข้องกับข้อมูลของกฎของไฟร์วอลล์ หรือไม่ อย่างไร โดยข้อมูล 3 ชุดนี้จะประกอบไปด้วย

* ข้อมูลภายใต้กฎไฟร์วอลล์ เช่น IP Address, Port เข้าเงื่อนไขในกฎไฟร์วอลล์นั้นๆ
* ข้อมูลที่ไม่ได้อยู่ในเงื่อนไขกฎไฟร์วอลล์ ข้อมูลที่อยู่นอกกฎไฟร์วอลล์ ข้อมูลชุดนี้ให้ใช้วิธีการสร้างแบบเดียวกันกับ Default Rule
* ข้อมูลที่อยู่ในกฎไฟร์วอลล์และนอกกฎไฟร์วอลล์หรือแบบผสม

หลังจากนั้นให้แยกนอกกฎและในกฎอีกทีหนึ่งเมื่อได้ค่าวัดผลประสิทธิภาพจากโมเดลแล้ว

**3.2. ปํญหาที่เกิดขึ้นจากงานวิจัยเดิมและวิธีการแก้ไขตามสมมติฐาน**

จะเห็นได้ว่างานวิจัยเดิมมีการออกแบบกฎของไฟร์วอลล์ค่อนข้างรัดกุมมาก หรือก็คือมีการกำหนดแทบทุก Field ว่าต้องการให้มีค่าเป็นอย่างไร และตัวแปรที่ยืดหยุ่นได้เช่น IP Address มีการใช้ Subnet เดียวกันหมด ทำให้กฎของไฟร์วอลล์ทุกเงื่อนไขที่ตั้งมานั้นมีขนาด Sample Space ของแพ็คเกตที่สามารถสร้างขึ้นได้ที่เท่ากันหมด ส่งผลให้เป็นข้อคิดได้ว่า ถ้าหากมีกฎใดกฎหนึ่งสามารถสร้างแพ็คเกตได้มีความหลากหลายมากกว่า มีขนาด Sample Space ที่เยอะกว่าอีกกฏที่อยู่ในไฟร์วอลล์เดียวกัน เช่น การทำให้ IP Address อยู่ในวงของ Subnet /16 เป็นต้น การแบ่งจำนวนชุดข้อมูลฝึกสอนในแต่ละกฎไฟร์วอลล์ด้วยจำนวน N Sample per Rule ยังสามารถใช้ได้อยู่อีกหรือไม่ อย่างไร

โดยเราได้ตั้งสมมติฐานว่าจากปํญหาข้างต้น หากเกิดความไม่เท่าเทียมกันจากการแบ่งชุดข้อมูลฝึกสอนโดยอ้างอิงจากขนาดของกฎของไฟร์วอลล์ หรือ Sample Space ที่ไม่เท่ากันแล้ว ให้คำนวณโดยใช้อัตราส่วนความเป็นไปได้ทั้งหมดที่สามารถเกิดขึ้นในกฎไฟร์วอลล์นั้นให้เท่ากันแทน หรือก็คือ ถ้ากฎไฟร์วอลลนั้นมีขนาดใหญ่ก็จะได้จำนวนชุดข้อมูลฝึกสอนที่มากกว่ากฎที่มีขนาดเล็กกว่า โดยให้มีจำนวนที่แบ่งให้อิงตามโอกาสความเป็นไปได้ทั้งหมด

Diagram

Description automatically generatedอัลกอริทึ่มการแบ่งชุดข้อมูลฝึกสอนแบบใหม่ (อัตราส่วน Ratio)

**3.3. วัฏจักรการพัฒนางานวิจัยใหม่หลังได้รับการแก้ไข**

ในการพัฒนาวิจัยนี้จะเป็นทั้งการต่อยอดและทำใหม่เพื่อแก้ไขในแง่ของแนวคิดและปัญหาที่เราได้พบให้ดีขึ้น โดยชี้ไปที่ปัญหาการแบ่งชุดข้อมูลฝึกสอนจากอัลกอริทึ่มเพื่อให้ได้โมเดล Deep Neural Network ใหม่ที่ดีกว่าเดิม รูปแบบการดำเนินงานส่วนใหญ่จะมีความคล้ายคลึงไปกับงานวิจัยเดิม แต่ให้มีการปรับเปลี่ยนอัลกอริทึ่มที่ใช้ในการสร้างชุดข้อมูลฝึกสอนเพื่อให้สามารถตอบโจทย์สมมติฐานที่สร้างขึ้นและมีความครอบคลุมมากขึ้น ดังนั้นจึงต้องมีการปรับเปลี่ยนวิธีการดำเนินงานใหม่ตั้งแต่วิธีการออกแบบกฎไฟร์วอลล์รวมไปถึงส่วนของโปรแกรมสร้างชุดข้อมูลและโมเดล โดยแผนผังการดำเนินงานใหม่ของงานวิจัยที่ทำขึ้นใหม่ มีดังนี้

Diagram

Description automatically generated  
แผนผังการดำเนินงานโดยรวม

เพื่อพิสูจน์ว่าการแบ่งแบบอัตราส่วนหรือแบบใหม่ สามารถให้ผลลัพธ์ที่ดีกว่าการแบ่งแบบ N Sample หรือ Sample per Rule ที่มีการแบ่งชุดข้อมูลฝึกสอนเท่ากันทั้งหมดโดยไม่สนขนาดของกฎไฟร์วอลล์ดังที่ได้เห็นในหัวข้อก่อนหน้า จึงต้องมีการทำการทดลองทั้ง 2 อัลกอริทึ่มแบบเก่าและแบบใหม่ เพื่อให้สามารถชี้ให้เห็นถึงความแตกต่าง จุดดี จุดด้อย ของแต่ละอัลกอริทึ่มได้

จากรูปภาพ Block Diagram ข้างต้น สามารถแบ่งกระบวนการทำงานออกเป็นขั้นตอนได้ 6 ขั้นตอน ดังนี้

* + การกำหนดขอบเขตของข้อมูล Data Field ที่จะพิจารณา และการกำหนดกฎของไฟร์วอลล์
  + การสร้างชุดข้อมูลสำหรับการฝึกสอนโมเดล
  + การนำโมเดลไปผ่านการเรียนรู้ด้วยชุดข้อมูลสำหรับฝึกสอน
  + การนำโมเดลไปประมวลผล ทำนายผลลัพธ์จากชุดข้อมูลสำหรับทดสอบ

**3.3.1. การกำหนดขอบเขตของ Data Field ที่จะพิจารณา และการกำหนดกฎไฟร์วอลล์**

**Diagram

Description automatically generated**

เป็นขั้นตอนที่สำคัญสุดของงานวิจัย เป็นการชี้ประเด็นที่จะศึกษาและแนวทางของผลลัพธ์ที่จะเป็น โดยเริ่มจากการทำการทดลองอิงจากงานวิจัยเก่า ทดลองตั้งสมมติฐาน นำไปต่อยอดและสรุปเป็นประเด็นใหม่ที่สามารถพิสูจน์ได้

เงื่อนไขหลักของการวิจัยคือการสร้างชุดข้อมูลฝึกสอนจากกฎของไฟร์วอลล์ เพื่อให้ได้ระบบการทำงานคัดกรองข้อมูล Packet ที่ได้มาตรฐานและเรียนรู้ได้เองอย่างมีประสิทธิภาพ มีความแม่นยำสูง สิ่งที่ต้องทำในส่วนแรกคือการกำหนดขอบเขตความเป็นไปได้ที่ข้อมูลจะสามารถเกิดขึ้นในเครือข่าย และการกำหนดกฎของไฟร์วอลล์เพื่อให้สามารถสร้างชุดข้อมูล Packet ที่จะนำไปฝึกสอนให้กับโมเดล สร้างชุดข้อมูลทดสอบโมเดลที่สามารถเปรียบเทียบความถูกต้องของผลลัพธ์ที่ได้จากโมเดลหลังผ่านการเรียนรู้แล้ว

**3.3.1.1. การกำหนด Default Pool และ Data Field ที่จะใช้พิจารณา**

การกำหนดขอบเขตของ Packet ที่สามารถเกิดขึ้นหรือการกำหนด Default เองเป็นอีกหนึ่งขั้นตอนที่สำคัญ เพื่อลดปัญหาในการใช้ Workload และลดเวลาที่ใช้ในการทดลองของคอมพิวเตอร์ที่มากเกินจำเป็นในการคำนวณหา Sample Space เพราะ Packet ที่เกิดขึ้นจริงมีจำนวนมหาศาล แม้มีข้อมูลภายใน Field เพียงชุดเดียวที่แตกต่างกัน ชุดข้อมูลนั้นจะถูกสรุปเหมือนเป็นชุดข้อมูลใหม่ แต่ถึงกระนั้นการลดจำนวน Default จะต้องไม่น้อยเกินไปและยังสามารถสร้างกฎไฟร์วอลล์ที่ใช้ในการทดลองได้

Data Field รวมกันทั้งหมด 125 bits ประกอบไปด้วยดังนี้

* Version (4)
* Inner Header Length (4)
* Direction (1)
* Interface ID (4)
* Differentiated Services (6)
* Explicit Congestion Notification (2)
* Source IP Address (32)
* Source Port (16)
* Destination IP Address (32)
* Destination Port (16)
* Protocol (8)

**3.3.1.2. การกำหนดกฎไฟร์วอลล์สำหรับใช้สร้างชุดข้อมูล**

ขั้นตอนต่อมาคือการสร้างกฎของไฟร์วอลล์ ในขั้นตอนนี้จะเป็นการกำหนดกระบวนการทำ Packet Filtering ที่จะเป็นการตัดสินว่า ข้อมูล Packet ชุดดังกล่าวจะสามารถถูกตัดสินให้ผ่านหรือไม่ ซึ่ง Packet ทุกชุดจะถูกตรวจสอบในทุกกฎของไฟร์วอลล์โดยมี 2 คำสั่งหลัก ได้แก่ “Allow” ปล่อยผ่านให้ข้อมูลชุดนั้นเข้าสู่ระบบหรือ “Deny” ไม่ปล่อยให้ข้อมูลชุดนั้นผ่านเข้าสู่ระบบ ค่าในตารางจะเป็น Parameter ที่จำเป็นในการสร้างชุดข้อมูลใน Packet Generator ในขั้นตอนต่อไป

**Table

Description automatically generated**  
**ตารางที่ 3.2** ตัวอย่างการสร้างเงื่อนไขภายในชุดกฎของไฟร์วอลล์

**3.3.2. ขั้นตอนที่ 2 การสร้างชุดข้อมูลสำหรับการฝึกสอนโมเดล**

**Diagram

Description automatically generated**

ชุดข้อมูลฝึกสอนชุดหนึ่งจะประกอบไปด้วยตัวอย่างข้อมูล Packet ที่ตรงตามเงื่อนไขในแต่ละกฎไฟร์วอลล์ มีวิธีการแบ่งจำนวนตามสมมติฐานที่วางเอาไว้ และจะเพิ่มจำนวนขึ้นไปเรื่อยๆตามการทดลอง

เพื่อให้ชุดข้อมูลฝึกสอนอยู่ในรูปแบบที่โมเดลสามารถใช้งานได้และอยู่ในขอบเขตของงานวิจัย จึงตัดสินใจสร้างชุดข้อมูลฝึกสอนโดยใช้โปรแกรม Packet Generator ที่สร้างขึ้นเอง ชุดข้อมูลฝึกสอนที่ถูกสร้างขึ้นจะถูกจัดระเบียบอยู่ใน Cell ของไฟล์นามสกุล CSV ทำให้ง่ายแก่การดึงข้อมูลกลับมาใช้ต่อในขั้นตอนถัดไป

แต่ก่อนที่จะสร้างชุดข้อมูล Packet นั้นจะต้องทราบความต้องการและจุดประสงค์ของโมเดล ว่าโมเดลดังกล่าวต้องการชุดข้อมูลที่มีความสัมพันธ์และมีจำนวน Input และ Output อย่างไร การสร้างชุดข้อมูล Packet จะเป็นการสุ่มเลือกจากความเป็นไปได้ทั้งหมดของชุดข้อมูล Packet ทั้งหมด และหลังจากนั้นจะเป็นการเพิ่ม Decision Field เข้าไปในชุดข้อมูล Packet แต่ละชุด เพื่อให้โมเดลนำไปเข้ากระบวนการเรียนรู้ และเปรียบเทียบผลลัพธ์ในขั้นตอนหลังการทดสอบ (Evaluate) ตัดสินจากความแม่นยำในการทำนาย Decision Field ซึ่งจะถูกสร้างอ้างอิงกับกฎของไฟร์วอลล์ในขั้นตอนแรก

**3.3.2.1. หลักการในการออกแบบชุดข้อมูลฝึกสอน**

ชุดข้อมูลเราได้ทำการจำลองมาจาก Packet Header และเพื่อแปลงข้อมูลให้อยู่ในรูปแบบที่เหมาะแก่การนำมาประมวลผลได้ จึงมีการเปลี่ยนแปลงรูปแบบและแทนค่าข้อมูลดังกล่าว ดังนี้

* การแทนค่าเป็นเลขฐานสองใน Decision Field
* Allow แทนค่า เป็น 1
* Deny แทนค่า เป็น 0
* ข้อมูลอื่นใน Packet Header จะถูกแปลงเป็นเลขฐานสองตามขนาดของ Label นั้นๆ

ชุดข้อมูล Packet ที่สร้างขึ้นเป็นการประยุกต์ใช้วิธีเรียนรู้แบบ Supervised Learning หรือ การจับกลุ่มเรียนรู้จากข้อมูลที่มีโครงสร้าง ดังนั้นเพื่อให้ชุดข้อมูลฝึกสอนสามารถใช้งานได้เต็มประสิทธิภาพ ชุดข้อมูลฝึกสอนจะต้องออกแบบให้มีความครอบคลุม ไม่เกิดปัญหา Underfitting หรือ Overfitting

**Underfitting** คือ การที่โมเดลของเราไม่สามารถทำงานได้ จากการที่ไม่สามารถจัดแนวโน้มของข้อมูลได้ อันเนื่องมากจากโมเดลเราไม่เหมาะสมหรือข้อมูลมีจำนวนน้อยไป กรณีนี้โมเดลมีค่าความเอนเอียงสูง (high bias) ยกตัวอย่างเช่น หากเรานำช้อมูลที่ Train มาลองแล้วได้ความแม่นยำต่ำ เมื่อนำชุดข้อมูลทดสอบมาลองก็จะได้ความแม่นยำต่ำเช่นกัน

**Overfitting** คือ การที่โมเดลตอบสนองต่อการรบกวน (noise) จำนวนมาก จนเริ่มเรียนจากการรบกวนและรายละเอียดของข้อมูลที่ไม่ถูกต้อง แล้วโมเดลของเราจะไม่เหมาะสมสำหรับการสามารถทำนายข้อมูล เช่น ทำนายข้อมูลที่ไม่เคยมีอย่างผิดพลาดกว่าที่คาดจะเป็นมาก (ล้มเหลวที่จะทำนายข้อมูลได้ถูกต้อง) เพราะมีรายละเอียดและการรบกวนมากเกินไป กรณีนี้โมเดลมีค่าค่าความแปรปรวนของข้อมูลสูง (high variance) ยกตัวอย่างเช่น โมเดลที่พัฒนาขึ้นมีความแม่นยำจากชุดข้อมูลทดสอบมากถึง 99% แต่เมื่อนำชุดข้อมูลทดสอบซึ่งไม่เคยปรากฏเลยในชุดข้อมูลฝึกสอนมาทดสอบ ทำให้ความแม่นยำเหลืออยู่เพียง 40% ปัญหานี้คือ Overfitting

**3.3.2.2.** **การพิจารณา Default Rule เพื่อใช้สร้างชุดข้อมูลฝึกสอน**

นอกจากกฎไฟร์วอลล์ที่กำหนดขึ้นทั่วไป ยังมีกฎของ Default Rule ซึ่งจำเป็นต้องพิจารณาแยกเป็นกรณีพิเศษ เนื่องจากจำนวนความเป็นไปได้ของข้อมูลของกฏไฟร์วอลล์ที่มีการกำหนดมีขนาดที่ต่างกับ Default Rule มาก จึงทำให้การทดสอบต้องแบ่งออกเป็น 3 แบบ ได้แก่ With Default Rule, Without Default Rule และแบบผสม ซึ่งเราได้ตั้ง Default Rule เป็น Deny any หรือ Deny ทุกข้อมูลที่นอกเหนือจากไฟร์วอลล์ที่เรากำหนดไว้

**3.3.2.3. กลไกการแบ่งชุดข้อมูลฝึกสอน**

เป็นวิธีการในการออกแบบการแบ่งชุดข้อมูลฝึกสอนที่ใช้ในงานวิจัยนี้โดยเฉพาะ มีเป้าหมายเพื่อพิสูจน์ว่าอัลกอริทึ่มที่สร้างขึ้นจากสมมติฐานแบบใดจะสามารถให้ประสิทธิภาพในการฝึกสอนได้ดีกว่า โดยอัลกอริทึ่มที่จะนำมาใช้พิจารณา ประกอบไปด้วยดังนี้

* **การแบ่งชุดข้อมูลฝึกสอนแบบ N Sample**

การแบ่งชุดข้อมูลแบบ N Sample หรือการแบ่งชุดข้อมูลฝึกสอนให้มีจำนวนเท่ากันทั้งหมดในแต่ละกฎไฟร์วอลล์แม้ความเป็นไปได้ของชุดข้อมูลฝึกสอนในแต่ละกฎไฟร์วอลล์จะมีขนาดไม่เท่ากันก็ตาม อัลกอริทึ่มนี้สามารถสร้างขึ้นโดยเริ่มจากกฎละ 1 ข้อมูลฝึกสอนได้ แต่เพื่อให้เห็นผลกราฟในระยะยาวที่มีจำนวนชุดข้อมูลฝึกสอนมากๆและลดเวลาที่ใช้ในการทดสอบ ทำให้เราเลือกใช้วิธีการเพิ่มชุดข้อมูลแบบก้าวกระโดด

Graphical user interface, text, application

Description automatically generated

* **การแบ่งชุดข้อมูลฝึกสอนแบบอัตราส่วน Ratio**

การแบ่งชุดข้อมูลฝึกสอนแบบอัตราส่วน Ratio เป็นสมมติฐานที่ตั้งขึ้นในงานวิจัยจากการคาดเดาว่ากฎไฟร์วอลล์ที่มีจำนวนข้อมูลเข้าเงื่อนไขสูงมากๆ โมเดลจำเป็นต้องมีการเรียนรู้จากข้อมูลฝึกสอนจากกฎดังกล่าวที่มากกว่า เนื่องจากข้อมูลมีขนาดกว้างทำให้ต้องใช้ข้อมูลฝึกสอนมากขึ้น โดยเราได้ใช้วิธีการแบ่งให้แต่ละกฎไฟร์วอลล์ได้รับจำนวนชุดข้อมูลฝึกสอนแบบอัตราส่วน หมายความว่ากฎที่มีจำนวนข้อมูลเข้าเงื่อนไขมากจะได้จำนวนโควต้าในชุดข้อมูลฝึกสอนมาก กฎไฟร์วอลล์ที่มีจำนวนน้อยกว่าจะได้รับจำนวนชุดข้อมูลฝึกสอนที่น้อยกว่า ซึ่งทั้งหมดจะต้องได้รับตามอัตราส่วนจากความเป็นไปได้ทั้งหมดภายในกฎไฟร์วอลล์นั้นเท่าๆกัน

Text

Description automatically generated

**3.3.3. การนำโมเดลไปผ่านการเรียนรู้ด้วยชุดข้อมูลสำหรับฝึกสอน**

**Diagram

Description automatically generated**

เป็นขั้นตอนการนำชุดข้อมูลฝึกสอนที่สร้างขึ้นไปประมวลผลผ่านโมเดลให้เกิดการเรียนรู้ โดยขั้นตอนการฝึกโมเดลจะต้องมีการกำหนดค่าพารามิเตอร์และปรับปรุงแก้ไขการประมวลผลหาคำตอบที่ขึ้นอยู่กับขอบเขตของงานหรือข้อมูลที่จะพิจารณา ซึ่งในส่วนนี้เราสามารถหาหลักการได้ จากคำแนะนำของผู้พัฒนาโมเดล หรืองานวิจัยที่มีการใช้งานใกล้เคียงกัน โดยมีจุดประสงค์เพื่อ พัฒนาให้โมเดลสามารถเรียนรู้ผ่านชุดข้อมูลฝึกสอนได้อย่างมีประสิทธิภาพขึ้นได้

**3.3.3.1. หลักการออกแบบโครงสร้างของโมเดล**

สำหรับการออกแบบโครงสร้างของโมเดลใหม่ เราใช้การต่อยอดและงานวิจัยเก่าและแหล่งอ้างอิงที่น่าเชื่อถือ โครงสร้างของโมเดลจะถูกกำหนดขึ้นอยู่กับรูปแบบความสัมพันธ์ของข้อมูล รูปแบบของคำตอบที่ต้องการ ซึ่งค่าเหล่านี้จะเป็นการกำหนดรูปแบบอัลกอริทึ่มเชิงลึกคำนวณค่าน้ำหนักด้วยไลบราลีของเครื่องมือที่เลือกใช้ โดยประเด็นหัวข้อสำคัญที่เราเลือกใช้ มีดังนี้

* Binary Logistic Regression

การวิเคราะห์สมการถดถอยโลจิสติกเป็นเทคนิคการวิเคราะห์สถิติเชิงคุณภาพ แตกต่างไปจากการวิเคราะห์เชิงปริมาณ ซึ่งตัวแปรเชิงคุณภาพเรียกอีกอย่างหนึ่งว่าเชิงกลุ่ม โดยเทคนิคการวิเคราะห์นี้ใช้กับตัวแปรที่ถูกแบ่งออกเป็น 2 กลุ่มย่อย (dichotomous variable) คือ 0 และ 1 ซึ่งในงานวิจัยของเราถูกแบ่ง Output ที่วิเคราะห์ออกมาเป็น 2 กลุ่มเช่นเดียวกันคือ Allow และ Deny

Chart, line chart, scatter chart

Description automatically generated

* ฟังก์ชั่นประมวลผล Sigmoid

ฟังก์ชั่นประมวลผล คือ ฟังก์ชั่นที่รับผลการประมวลผลทั้งหมดจากทุก Input ภายใน 1 นิวรอนแล้วพิจารณาว่าจะส่งต่อ Output ออกมาเป็นอย่างไร โดยฟังก์ชั่นประมวลผล Sigmoid จะถูกวางไว้ใน Layer สุดท้าย มักเป็นที่นิยมใช้ใน Deep Learning เพราะมีการถูกใช้ใน Artificial Neural Network ที่มีการต้องหาค่าความน่าจะเป็น ซึ่งจะเป็นค่า Output 0 ถึง 1 เท่านั้น

หนึ่งในฟังก์ชั่น Sigmoid ที่พบบ่อยที่สุดคือฟังก์ชั่น Sigmoid Logistic สิ่งนี้มักถูกใช้กับการเรียนรู้โครงข่ายประสาทเทียม โดยฟังก์ชั่นจะถูกกำหนดไว้ ดังนี้

Chart

Description automatically generated

Table

Description automatically generated

* จำนวน Hidden Layer

จำนวน Hidden Layer ขึ้นอยู่กับความซับซ้อนของโมเดล ซึ่งในงานวิจัยของเราคือการสร้างโมเดลโครงข่ายประสาทเทียมเชิงลึกหรือ Deep Neural Network ซึ่งต้องการมากกว่า 2 Hidden Layer ขึ้นไป

Table, timeline

Description automatically generated

* จำนวน Node หรือนิวรอนในแต่ละ Layer

การกำหนดจำนวนนิวรอนในแต่ละชั้น Layer เป็นส่วนสำคัญในสถาปัตยกรรมโครงสร้างโครงข่ายประสาทเทียม แม้ว่าจำนวนชั้น Layer จะไม่ค่อยส่งผลกระทบมาก แต่จำนวนนิวรอนมีอิทธิผลอย่างมากต่อผลลัพธ์สุดท้าย ดังนั้นการเลือกใช้จำนวนนิวรอนจะต้องมีการพิจารณาอย่างรอบคอบ

การใช้เซลล์ประสาทน้อยเกินไปใน Hidden Layer จะทำให้เกิดสิ่งที่เรียกว่า Underfitting ปํญหาที่จำนวนนิวรอนมีจำนวนน้อยเกินไปจนทำให้ระบบไม่สามารถหาความสัมพันธ์ได้อย่างเพียงพอต่อข้อมูลที่มีมากเกิน หรือถ้าหากตั้งจำนวนนิวรอนมีมากเกินไปก็ส่งผลไปหลายประการเช่นกัน ประการแรกคือระบบมีความสามารถในการประมวลผลมากเกินไปทำให้ไม่เพียงพอต่อข้อมูลที่มีอยู่เพียงจำกัด ประการที่สองคือ ระยะเวลาในการฝึกสอนโมเดลสามารถเพิ่มขึ้นจนถึงจุดที่โมเดลไม่สามารถฝึกได้อย่างเพียงพอ

* Input Layer = 125 Nodes ตามจำนวน Data Field
* Output Layer = 2 Nodes ได้แก่ Allow และ Deny

และจำนวน Node ในแต่ละ Hidden Layer มีสูตรคำนวณ ดังนี้

A picture containing diagram

Description automatically generated

NiNi = จำนวนนิวรอนของ Input Layer  
NoNo = จำนวนนิวรอนของ Output Layer.  
NsNs = จำนวนของชุดฝึกสอนโมเดลที่ใช้ในโมเดล  
αα = ค่าตัวแปรสำหรับปรับขนาด มักใช้เลขระหว่าง 2-10

**3.3.4. การนำโมเดลไปประมวลผล ทำนายผลลัพธ์จากชุดข้อมูลสำหรับทดสอบ**

**Diagram

Description automatically generated**

เป็นขั้นตอนทดสอบ (Evaluate) เพื่อทำนายความแม่นยำของโมเดลที่ผ่านการเรียนรู้แล้วโดยใช้ข้อมูลทดสอบอีกชุดหนึ่ง ในส่วนนี้จะใช้โปรแกรม Compare Engine ที่เขียนขึ้นเอง เริ่มจากการนำเข้าโมเดลที่ผ่านการเรียนรู้แล้วจากขั้นตอนที่ 3 นำไปคาดเดาชุดข้อมูลทดสอบจากขั้นตอนที่ 4 ตัวโปรแกรมจะทำการแยกส่วนชุดข้อมูล CSV เป็นส่วนของข้อมูลและผลลัพธ์เช่นเดียวกันกับตอนฝึกโมเดล ด้วยฟังก์ชั่น model.predict ของ Keras จะสามารถทำนายผลด้วยโมเดลได้ทันทีว่าจากชุดข้อมูล Packet ทดสอบนั้น ให้ผลลัพธ์ Allow หรือ Deny ซึ่งผลลัพธ์สุดท้ายจะเป็นสรุปในการหาความแม่นยำของโมเดลนั้นตาม Reference Variant Set ดังนี้

Table

Description automatically generated

Reference Variant Set เป็น Matrix ที่ใช้ในการอ้างอิงในการหาข้อสรุปของโมเดลว่ามีความแม่นยำหรือไม่ อย่างไร ซึ่งมักถูกใช้กับโมเดลที่มีการเรียนรู้และแก้ปัญหาในการแบ่งกลุ่ม โดยผลลัพธ์ที่ได้จะประกอบไปทั้งหมด 4 รูปแบบ ได้แก่

* True Positive

โมเดลอนุญาติให้ข้อมูลผ่านตรงตามกฏของไฟร์วอลล์ หรือให้ Allow ถูกต้อง

* True Negative

โมเดลไม่อนุญาติให้ข้อมูลผ่านตรงตามกฏของไฟร์วอลล์ หรือให้ Deny ถูกต้อง

* False Positive

โมเดลอนุญาติให้ข้อมูลผ่านไม่ตรงตามกฏของไฟร์วอลล์ หรือให้ Allow ผิดพลาด

* False Negative

โมเดลไม่อนุญาติให้ข้อมูลผ่านไม่ตรงตามกฏของไฟร์วอลล์ หรือให้ Deny ผิดพลาด

**3.4. การศึกษาค้นคว้าเทคโนโลยีและเครื่องมือที่ใช้ในการพัฒนาโมเดล**

ในการดำเนินการวิจัย เราเลือกใช้ Python เป็นภาษาหลักในการพัฒนาโปรแกรมสร้างชุดข้อมูลฝึกสอนและโมเดล DNN ดังนั้นเพื่อให้การทำงานและการใช้งานเป็นไปตามที่งานวิจัยต้องการ จึงจำเป็นต้องศึกษาความเข้ากันได้ของเครื่องมือและไลบรารีที่เกี่ยวข้องในการพัฒนา

* Anaconda3 โปรแกรมจัดการแพ็คเกจและสร้าง Environment ที่จำเป็นในการเขียนซอฟต์แวร์ภาษา Python เหมาะแก่งาน Data Visualization, Machine Learning, Neural Network และยังสามารถใช้งานร่วมกันกับ IDE ได้หลากหลาย

Version: Anaconda 3.8 64-Bit

* Spyder โปรแกรมพัฒนาซอฟต์แวร์ด้วยภาษา Python สามารถตรวจสอบตัวแปรได้ง่าย

Version: Spyder 4.1.4

* TensorFlow ไลบราลีพื้นฐานในการพัฒนา Neural Network Model

Version: TensorFlow 2.3.0 สามารถใช้ได้กับ Python 64-Bit เท่านั้น

* Sklearn เป็นเครื่องมือสำคัญในการทำ Model Selection และ Data Preprocessing ทำงานโดยพื้นฐานของ Numpy

Version: Scikit-learn 0.23.2

* Keras เป็น Deep Learning Framework ที่สำคัญ สามารถประมวลผลได้ทั้ง CPU และ GPU

Version: Keras 2.4.3

* Pandas เป็นไลบรารีช่วยในการจัดกลุ่ม แยกประเภทข้อมูลกลุ่มโครงสร้าง เช่น ไฟล์นามสกุล CSV

Version: Pandas 1.1.2

* Pip เครื่องมือที่ช่วยในการติดตั้งแพ็คเกจในภาษา Python

Version: pip 20.2.3

* Tkinter ไลบรารีพัฒนาการสร้าง GUI ด้วยภาษา Python

Version: Tk 8.6.10

* NVIDIA CUDA เครื่องมือช่วยให้คอมพิวเตอร์สามารถประมวลผลผ่าน GPU ได้

Version: CUDA 11.1.0

* NVIDIA cuDNN เครื่องมือช่วยในการประมวลผล DNN ผ่าน GPU

Version: cuDNN 8.0