**บทที่ 3**

**วิธีการดำเนินการวิจัย**

การดำเนินการวิจัยเรื่อง การศึกษาวิจัยเพื่อพัฒนาสร้างชุดข้อมูลในการฝึกสอนไฟร์วอลล์ปัญญาประดิษฐ์ด้วยเทคโนโลยีโครงข่ายประสาทเทียมจากกฏของไฟร์วอลล์ (Researching for developing training set with artificial neural network technology based on firewall rules) เป็นการวิจัยเพื่อหาวิธีการสร้างชุดข้อมูลฝึกสอนโมเดลที่เหมาะสม มุ่งเน้นไปที่การหาจำนวนข้อมูลที่เหมาะสมในการนำไปฝึกสอนโมเดล ถ้าหากมีจำนวนข้อมูลที่มากเกินไปก็จะใช้เวลาในการฝึกมาก แต่ถ้าหากมีจำนวนน้อยเกินไปก็อาจจะทำให้โมเดลที่ได้มีความแม่นยำต่ำมีประสิทธิภาพไม่เพียงพอต่อการใช้งานจริง โดยงานวิจัยนี้จะทดลองเพื่อหาข้อเท็จจริงพร้อมกับต่อยอดงานวิจัยก่อนหน้าด้วย

## 3.1. การศึกษาการนำมาใช้งานของ Packet Filtering AI Firewall

**การทำงานของ Packet Filtering AI Firewall**

AI Firewall เป็นไฟร์วอลล์ที่มีความสามารถเสมือนกับ Packet Filtering Firewall คือ มีหน้าที่ในการตรวจสอบและกรั่นกรองข้อมูลแพ็ตที่นำเข้ามาในเครือข่าย โดยพิจารณาความน่าเชื่อถือของข้อมูลจากส่วน Packet Header ซึ่ง Firewall ทั่วไปจะใช้วิธีเปรียบเทียบกฎไฟร์วอลล์จากตาราง Access Control List (ACL) ที่กำหนดแต่สำหรับ AI Firewall หรือไฟร์วอลล์ที่เราจะพัฒนาขึ้นในงานวิจัยนั้น จะใช้วิธีคัดกรองจากการเรียนรู้อัตโนมัติด้วยการเลียนแบบโครงข่ายประสาทเทียมของมนุษย์ที่ซ้อนกันหลายชั้นทำให้สามารถตรวจจับรูปแบบและจัดกลุ่มข้อมูลได้ แต่ AI Firewall จะยังไม่สามารถกรั่นกรองแพ็คเกตหรือทำนายชุดข้อมูลได้จนกว่าจะได้รับการเรียนรู้จากชุดฝึกสอน ซึ่งความแม่นยำของโมเดลที่ประมวลผลจะขึ้นอยู่กับชุดข้อมูลตัวอย่างที่เราใช้

การทำ Packet Filtering ของ AI Firewall จำเป็นต้องมีการเรียนรู้แบบมีผู้สอนหรือ Supervised Learning โดยชุดฝึกสอนที่ทำการสอนจะต้องมีการกำหนดคำตอบที่เป็น Target Attribute ชัดเจน ต่างกับ Unsupervised Learning จะเป็นการแบ่งกลุ่มตามคุณลักษณะเท่านั้น โดยเป้าหมายของคำตอบที่เราต้องการคือการตัดสินใจให้แพ็คเกต Allow หรือ Deny นั่นเอง ซึ่งเราจะต้องเป็นผู้กำหนดว่าแพ็คเกตที่มีรูปแบบดังกล่าวจะถูกอนุญาติให้ผ่านหรือไม่อนุญาติให้ผ่าน

Diagram

Description automatically generated  
รูปที่ 3.1 กระบวนการพัฒนาและการทำงาน Packet Filtering AI Firewall

รูปนี้แสดงให้เห็นถึงการพัฒนาและการนำมาใช้งาน AI Firewall โดยเริ่มจากกรอบสีแดงแสดงถึงการสร้างและฝึกโมเดลที่เราต้องการ เพื่อให้โมเดลสามารถนำไปใช้งานได้จะมี 2 ปัจจัยหลักคือ ข้อมูลฝึกสอนโมเดล และการเลือกอัลกอริทึ่มในการประมวลผลของโมเดล อย่างแรกเราจะต้องมีข้อมูลฝึกสอนที่ถูกกำหนดให้มีรูปแบบเป็นไปตามมาตรฐานกฎไฟร์วอลล์ที่ผู้ควบคุมระบบหรือ Supervisor เป็นผู้ตั้งขึ้นมา และต่อมาคืออัลกอริทึ่มที่เลือกใช้ ในส่วนนี้จะเป็นเลือกอัลกอริทึ่เหมาะสมสำหรับความต้องการของระบบว่าข้อมูล Input มีโครงสร้างเป็นอย่างไรและต้องการให้โมเดลมีการทำนายผลลัพธ์ออกมาเป็นอย่างไร วิธีการออกแบบ Neural Network จะมีวิธีการเลือกเป็นไปตามที่เครื่องมือที่ใช้พัฒนาประดิษฐ์เป็นผู้กำหนดไว้อยู่แล้ว

กรอบสีเขียวจะแสดงให้เห็นถึงการทำงานของ Packet Filtering AI Firewall หลังจากโมเดล ซึ่งจะเห็นได้ว่าการทำงานทั้งระบบจะต้องผ่านอัลกอริทึ่มที่เราเป็นผู้กำหนดก่อน เป็นเหตุผลที่ว่าขั้นตอนการฝึกสอนและสร้างโมเดลที่เราต้องการมีความสำคัญยิ่ง โดยเฉพาะการออกแบบ Train Sample ที่เป็นจุดสำคัญในการบ่งบอกความสามารถของโมเดลตามมาตรฐานของเราเอง ต่างกับอัลกอริทึ่มที่ผู้พัฒนาเครื่องมือเป็นผู้กำหนดซึ่งหาได้ทั่วไป จึงเป็นสาเหตุว่าทำไมเราจึงมุ่งเน้นไปที่การศึกษาวิธีการสร้างชุดฝึกสอนให้ดีมีประสิทธิภาพ

## 3.2. การดำเนินงานของงานวิจัยเก่าและประเด็นปัญหาที่เกิดขึ้น

จากแหล่งอ้างอิงที่เราได้ทำการศึกษาค้นคว้า ได้เคยมีงานวิจัยเกี่ยวกับการพัฒนาหาวิธีการสร้างชุดข้อมูลฝึกสอนโมเดลให้มีประสิทธิภาพ โดยในงานวิจัยเดิมได้มีการกล่าวถึงการนำArtificial Neural Network (ANN) มาประยุกต์ใช้กับไฟร์วอลล์แบบ Intrusion Detection System (IDS) โดยเราได้วิเคราะห์ปัญหาที่เกิดและหาวิธีในการต่อยอด ซึ่งได้ใจความว่าดังนี้

**3.2.1. การสร้างชุดฝึกสอนที่อ้างอิงจากกฎของไฟร์วอลล์**

ผู้วิจัยมีความคิดว่าไฟร์วอลล์โดยทั่วไปมีการควบคุมความปลอดภัยคัดกรองข้อมูลด้วยกฎไฟร์วอลล์ ดังนั้นจะเป็นการดีกว่าถ้าหากนำกฎไฟร์วอลล์มาใช้อ้างอิงเพื่อสร้างชุดฝึกสอน วิธีนี้จะช่วยให้โมเดลสามารถเรียนรู้กับชุดข้อมูลที่มีการออกแบบตามมาตรฐานที่กำหนดไว้ได้ ซึ่งแต่เดิมจะใช้ข้อมูลการโจมตีที่เคยเกิดขึ้นมาก่อนมาเป็นชุดฝึกสอนซึ่งไม่ได้มาตรฐานแทนในการฝึกสอน

**3.2.2.** วิธี**การดำเนินงานและการพัฒนาเครื่องมือที่ใช้ในงานวิจัยเดิม**

รูปแบบการดำเนินงานจะเป็นไปตามวัฏจักรการพัฒนาโมเดลทั่วไป โดยแบ่งออกเป็น 3 ขั้นตอนหลัก ได้แก่ การสร้างชุดข้อมูลสำหรับฝึกสอนโมเดล การพัฒนาโมเดลกับการนำชุดข้อมูลไปฝึกอบรมโมเดล และการประมวลผลโมเดลเพื่อให้ได้มาซึ่งผลลัพธ์ประสิทธิภาพผ่านชุดข้อมูลทดสอบ ซึ่งผู้วิจัยได้ออกแบบชุดทดสอบ 3 ชุดด้วยกัน ได้แก่ ข้อมูลที่อยู่ภายใต้เงื่อนไขกฎไฟร์วอลล์ ข้อมูลที่ไม่ได้อยู่ภายใต้กฎไฟร์วอลล์ และแบบผสม เพื่อให้สามารถพิสูจน์ได้ว่าโมเดลที่มีการฝึกอบรมภายใต้ชุดข้อมูลฝึกสอนที่สร้างจากกฎไฟร์วอลล์จะสามารถทำงานและประมวลผลตามกฎที่ออกแบบไว้ได้อย่างถูกต้องหรือไม่ อย่างไร แผนภาพขั้นตอนการดำเนินงาน มีดังนี้

Diagram

Description automatically generated  
รูปที่ 3.2 ระบบการทำงานหรือวัฎจักรการพัฒนาโมเดลของงานวิจัยเก่า

**3.2.3. วิธีการสร้างชุดข้อมูลแพ็คเกตฝึกสอนจากกฎไฟร์วอลล์**

ชุดข้อมูลฝึกสอนและชุดข้อมูลทดสอบโมเดลจะมีการจำลองรูปแบบโครงสร้างและขนาดมาจาก Packet Header และมีการถูกนำมาใช้เป็นตัวแปรคัดกรองของกฎไฟร์วอลล์ที่ทำการออกแบบ ชุดฝึกสอนจะถูกสุ่มสร้างขึ้นโดยการอ่านและประมวลผลทีละกฎไฟร์วอลล์เพื่อสร้างชุดข้อมูลฝึกอบรมที่มาจากกฎนั้นๆ โดยจำนวนชุดข้อมูลที่สร้างขึ้นจะมีหน่วยเป็น จำนวนชุดข้อมูลฝึกสอนต่อ 1 กฎไฟร์วอลล์ หรือหมายถึงชุดข้อมูลฝึกสอนที่ได้ในแต่ละกฎจะมีจำนวนที่เท่ากันหมด

Packet Header ที่ใช้และกฎไฟร์วอลล์ที่งานวิจัยเดิมได้ทำการออกแบบ มีดังนี้

* 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)

Table

Description automatically generated  
รูปที่ 3.2 การออกแบบกฎไฟร์วอลล์ของงานวิจัยเดิม

**3.2.4.** **ผลการดำเนินการจากงานวิจัยเดิม**

เมื่อทำการแบ่งสร้างชุดข้อมูลฝึกสอนเป็นที่เรียบร้อยแล้วและนำไปฝึกอบรมให้กับโมเดลที่พัฒนาขึ้นผ่านชุดข้อมูลทดสอบ โดยชุดทดสอบประกอบไปด้วยชุดข้อมูลภายใต้กฎไฟร์วอลล์ 1,106 แพ็คเกต ชุดข้อมูลนอกกฎไฟร์วอลล์ 8,896 แพ็คเกต รวมเป็น 10,000 แพ็คเกต สรุปผลภายใต้กราฟแบ่งออกเป็น ชุดกฎไฟร์วอลล์ที่ทดลอง และชนิดของชุดข้อมูลที่ใช้ประมวลผลทดสอบ ได้ผลออกมา ดังนี้

Chart, line chart

Description automatically generated

Chart, line chart

Description automatically generated

รูปที่ 3.3 ผลการดำเนินการของโมเดลที่ได้จากงานวิจัยเดิม

จากกราฟผลการทดลองทั้ง 2 รูป แสดงถึงความแม่นยำของผลการดำเนินการของโมเดล ซึ่งแต่ละการทดลองทำขึ้นเพื่อศึกษาผลกระทบของขอบเขตภายใต้เงื่อนไขของกฎ และเพื่อหาจำนวนที่เหมาะสมต่อกฎไฟร์วอลล์ที่ดีที่สุด

สำหรับการทดลองทั้ง 2 รูปจะเห็นได้ว่าผลการทำนายชุดข้อมูลทดสอบ InScope มีความแม่นยำสูงมากหรือใกล้เคียง 100 เปอร์เซนต์ ซึ่งเป็นขอบเขตเงื่อนไขเดียวกันกับตัวอย่างชุดข้อมูลที่สุ่มสร้างจากกฎไฟร์วอลล์ที่ออกแบบ แต่สำหรับ OutScope ที่มีการให้ทำนายชุดข้อมูลนอกเหนือจากขอบเขตกฎไฟร์วอลล์ที่ออกแบบกลับให้ค่าความแม่นยำที่ต่ำ ดังนั้น MixScope จึงต้องอยู่ระหว่าง InScope และ OutScope ทำให้สรุปได้ว่าเพื่อให้ความแม่นยำของโมเดลนั้นสูงขึ้นจำเป็นต้องได้รับการฝึกฝนด้วยจำนวนชุดฝึกสอนที่มากขึ้นและมีความหลากหลายของข้อมูล ซึ่งดูได้จากรูปกราฟแรกแม้เพิ่มจำนวนก็ไม่สามารถเพิ่มความแม่นยำได้เนื่องจากไม่มีตัวอย่างของ Default Rule และกราฟที่สองที่ยิ่งเพิ่มจำนวนในการฝึกมากขึ้นความแม่นยำก็จะเพิ่มขึ้น

ต่อมาคืออัตราการเรียนรู้ของโมเดลโดยจะเห็นได้ว่าเมื่อเพิ่มจำนวนชุดฝึกสอนต่อกฎไฟร์วอลล์ขึ้น ค่าความแม่นยำของโมเดลจะเพิ่มขึ้นด้วยอัตราน้อยลงมากจนทำให้ค่าความแม่นยำเข้าใกล้ค่าคงที่ในจุดหนึ่ง ซึ่งหมายความว่าแม้จะเพิ่มจำนวนชุดฝึกสอนเพื่อหวังให้โมเดลสามารถทำนายได้แม่นยำขึ้นเมื่อถึงจุดอิ่มตัวก็ค่าความแม่นยำก็จะเพิ่มขึ้นน้อยหรือไม่เพิ่มขึ้นอีกต่อไปนั่นเอง

นอกจากนี้ยังมีอีกสาเหตุหนึ่งที่สำคัญคือการออกแบบกฎไฟร์วอลล์ให้มีความครอบคลุมทั้ง 2 คำตอบ Allow และ Deny ซึ่งจะเห็นได้จากตัวอย่างของชุดกฎไฟร์วอลล์ที่ 2 จะเห็นได้ว่ากฏมีการออกแบบให้มีแต่ Allow เท่านั้น ซึ่งเมื่อทดสอบโดยการใช้ชุดทดสอบที่มีเพียงกฎ Default Rule ที่ผลลัพธ์ที่ถูกต้องนั่นแพ็คเกตควรจะต้องให้ Deny ทั้งหมด โมเดลไม่สามารถทำนายได้อย่างถูกต้องเลยหรือมีการให้ Allow กับชุดทดสอบทั้งหมด นั่นหมายความว่าการสร้างชุดฝึกสอนจะต้องมีข้อมูลแพ็คเกตให้มีทั้ง Allow และ Deny เพื่อให้โมเดลสามารถทำนายผลลัพธ์ทั้ง 2 รูปแบบได้

จากข้อสังเกตข้างต้นทั้งหมดที่ผ่านมา ผู้วิจัยได้สรุปข้อเท็จจริงในการเพิ่มประสิทธิภาพให้กับโมเดล 4 ข้อด้วยกัน ได้แก่ (1) เพิ่มจำนวนชุดฝึกสอนให้มากเท่าที่เป็นไปได้ (2) ชุดฝึกสอนจำเป็นต้องมีผลทำนายที่เป็นทั้ง Allow และ Deny (3) เพิ่ม Default Rule ในชุดฝึกสอน (4) กำหนดขอบเขตของ Packet Header เพื่อลดเวลาฝึกโมเดล โดยจำนวนที่เหมาะสมที่สุดที่ใช้ในงานวิจัยคือ 300 แพ็คเกตฝึกสอนต่อ 1 กฎไฟร์วอลล์

**3.2.5. ปัญหาที่เกิดขึ้นจากการวิเคราะห์ผลการดำเนินการ**

* **วิธีการแบ่งชุดฝึกสอนในแต่ละกฎไฟร์วอลล์ไม่เหมาะสมที่จะนำมาใช้**

การแบ่งชุดฝึกสอนในแต่ละกฎไฟร์วอลล์มีความเกี่ยวข้องโดยตรงกับความซับซ้อนของแต่ละกฏที่ทำการออกแบบ ซึ่งถ้าไฟร์วอลล์มีรูปแบบที่สามารถสร้างออกมาให้แตกต่างกันได้มากก็ยิ่งจำเป็นต้องใช้ชุดข้อมูลฝึกสอนเยอะ นั่นหมายความว่าถ้ากฎมีขนาดความเป็นไปได้ไม่เท่ากฎก็ต้องย่อมใช้จำนวนชุดฝึกสอนที่ไม่เท่ากันอย่างแน่นอน ในงานวิจัยเดิมได้ใช้วิธีการแบ่งชุดฝึกสอนในแต่ละกฎด้วยวิธีแบ่งด้วยจำนวนที่เท่าๆกันทั้งหมด ถ้าเรามองย้อนกลับไปถึงปัญหาข้อแรกจะเห็นได้ว่ากฎไม่ได้ถูกออกแบบมาให้มีความหลากหลาย ทำให้งานวิจัยเดิมยังไม่สามารถตอบโจทย์ในตรงนี้ได้

อีกหนึ่งข้อสังเกตที่สำคัญจากผลการดำเนินการของงานวิจัยเดิม คือการทดสอบโดยเพิ่มจำนวนชุดฝึกสอนเข้าไปเรื่อยๆ โดยเราจะเห็นได้ว่าในการทดสอบเพื่อประมวลผลชุดทดสอบ InScope ไม่จำเป็นต้องใช้จำนวนชุดฝึกสอนจำนวนมากก็สามารถให้ความแม่นยำสูงใกล้เคียง 100 เปอร์เซนต์ แม้ไม่ถึงจำนวนที่เหมาะสมก็ยังสามารถทำนายได้แม่นยำ ต่างกับการประมวลผลกับชุดทดสอบ MixScope และ OutScope จากโมเดลที่มีการนำแพ็คเกตฝึกสอนจากกฎ Default Rule มาใช้เพิ่มเป็นชุดฝึกสอนด้วย แม้เราใช้จำนวนชุดฝึกที่เท่ากัน กลับไม่ได้ความแม่นยำตามการประมวลผลชุดทดสอบ InScope แต่ได้เพียงประมาณ 60 เปอร์เซนต์เท่านั้น ซึ่งเป็นความแม่นยำที่ไม่เพียงพอต่อการนำมาใช้งาน สรุปคือชุดฝึกสอนที่แต่ละกฎจะได้รับควรได้ตามจำนวนที่เหมาะสม กฎใดที่สามารถคำนวณได้แล้วก็ไม่ควรที่จะได้รับชุดฝึกสอนเพิ่มขึ้นอีก เนื่องจากจะทำให้เสียเวลาในการฝึกสอนโมเดลโดยไม่จำเป็น และกฎใดที่ยังไม่สามารถทำนายผลได้ดีก็ควรเพิ่มจำนวนชุดฝึกสอนให้เหมาะสมจนได้รับความแม่นยำตามที่กำหนด

* **กฎไฟร์วอลล์ที่ใช้สร้างชุดฝึกสอนมีความหลากหลายไม่มากพอ**

ถ้ามองในแง่สำหรับการออกแบบกฎของไฟร์วอลล์แล้ว แต่ละกฎก็จะมีความซับซ้อนและมีขนาดเป็นของตัวมันเอง ซึ่งจะแตกต่างหรือไม่ขึ้นอยู่กับการกำหนดกฎ เมื่อเราย้อนกลับไปดูการออกแบบกฎไฟร์วอลล์ของงานวิจัยเดิม จะเห็นได้ว่างานวิจัยเดิมมีการออกแบบกฎของไฟร์วอลล์ค่อนข้างรัดกุมมาก หรือก็คือมีการกำหนดแทบทุก Data Field ว่าต้องการให้มีค่าเป็นอย่างไร และตัวแปรที่ยืดหยุ่นได้เช่น IP Address มีการใช้ Subnet /16 เหมือนกันหมด ทำให้กฎของไฟร์วอลล์ทุกเงื่อนไขที่ตั้งมานั้นมีขนาดความเป็นไปได้ของแพ็คเกตที่สามารถสร้างขึ้นได้ที่เท่ากันหมด เมื่อนำกฎจากงานวิจัยเดิมมาแจกแจงจะได้ผลตามตาราง ดังนี้

**ตารางแจกแจงความเป็นไปได้ของกฎไฟร์วอลล์จากรูป 3.2**

|  |  |  |
| --- | --- | --- |
| Packet Header | จำนวนความเป็นไปได้ | หมายเหตุ |
| Direction | 1 | ทุกกฎใช้เป็น in |
| Interface | 1 | ทุกกฎใช้เป็น eth0 |
| Source IP | 65,534 | จำนวน host ของ subnet 16 |
| Subnet Mask | 1 | ปลายทางที่เดียว ใช้เป็น /16 |
| Source Port | 1 | ระบุว่าเป็น 22 ไม่ก็ 80 (ไม่ใช่ any) |
| Destination IP | 1 | 161.246.34.11 ปลายทางที่เดียว |
| Destination Mask | 1 | ปลายทางที่เดียว ใช้เป็น /24 |
| Destination Port | 1 | ระบุว่าเป็น 22 ไม่ก็ 80 (ไม่ใช่ any) |
| Protocol | 1 | ระบุว่าเป็น TCP ไม่ก็ UDP (ไม่ใช่ any) |

ตารางที่ 3.1 จำนวนความเป็นไปได้ของแพ็คเกตในแต่ละกฎของงานวิจัยเดิม

จากตารางข้างต้น ความเป็นไปได้ของชุดข้อมูลที่แตกต่างกันหรือความซับซ้อนของกฎไฟร์วอลล์ทุกกฎที่งานวิจัยเดิมออกแบบ มีความเป็นไปได้เท่ากันหมดที่ 65,534 รูปแบบความเป็นไปได้ โดยตัวเลขความเป็นไปได้นี้คำนวณโดยการนำผลคูณจากความเป็นไปได้ของแต่ละ Data Field ใน Packet Header เหมือนกับการหา Sample Space นั่นเอง

เมื่อเราศึกษาหลักการทำงานและการเรียนรู้ของปัญญาประดิษฐ์ เราจะทราบดีว่าความยากและความซับซ้อนของข้อมูลย่อมมีผลขนาดความเป็นไปได้ของข้อมูลที่เกิดขึ้นและออกมาแตกต่างกัน และขนาดความเป็นไปได้ของความแตกต่างของข้อมูลที่กว้างมากย่อมต้องใช้ชุดข้อมูลฝึกสอนตัวอย่างที่มากขึ้นตามไปด้วยเช่นกันเพื่อให้โมเดลสามารถทำงานและตีความได้ ดังนั้นจำนวนที่เหมาะสมที่สุดที่งานวิจัยเก่าได้สรุปไปที่ 300 แพ็คเกตต่อ 1 กฎไฟร์วอลล์ อาจจะใช้ไม่ได้กับทุกกฎไฟร์วอลล์ที่ออกแบบ สิ่งนี้จึงเป็นปัญหาข้อแรกที่เราจะนำมาพิจารณา

## 3.3. การต่อยอดพัฒนางานวิจัยใหม่

ในการพัฒนาวิจัยนี้จะเป็นทั้งการต่อยอดและทำใหม่เพื่อแก้ไขในแง่ของแนวคิดและปัญหาที่เราได้พบให้ดีขึ้น โดยชี้ไปที่ปัญหาการแบ่งชุดข้อมูลฝึกสอนจากอัลกอริทึ่มเพื่อให้ได้โมเดล Deep Neural Network ใหม่ที่ดีกว่าเดิม รูปแบบการดำเนินงานส่วนใหญ่จะมีความคล้ายคลึงไปกับงานวิจัยเดิม แต่ให้มีการปรับเปลี่ยนอัลกอริทึ่มที่ใช้ในการสร้างชุดข้อมูลฝึกสอนเพื่อให้สามารถตอบโจทย์สมมติฐานที่สร้างขึ้นและมีความครอบคลุมมากขึ้น ดังนั้นจึงต้องมีการปรับเปลี่ยนวิธีการดำเนินงานใหม่ตั้งแต่วิธีการออกแบบกฎไฟร์วอลล์รวมไปถึงส่วนของโปรแกรมสร้างชุดข้อมูลและโมเดล

ในงานวิจัยเดิมเป็นการนำเสนอการนำกฎของไฟร์วอลล์มาใช้สร้างชุดฝึกสอนโครงข่ายประสาทเทียมเชิงลึกหรือโมเดลปัญญาประดิษฐ์ ซึ่งทำให้ได้หลักการและข้อสรุปถึงวิธีการเพิ่มค่าความแม่นยำโมเดล เราจะยังคงหลักการเรื่องการนำกฎไฟร์วอลล์มาใช้อ้างอิงสร้างชุดข้อมูลฝึกสอนและวิธีการออกแบบชุดข้อมูลฝึกสอนและทดสอบโมเดลบางข้อของงานวิจัยเก่ามาใช้ในการต่อยอดงานวิจัยนี้ และแก้ไขจุดที่งานวิจัยเก่านั้นทำมาไม่ละเอียดชัดเจน

**ข้อแนะนำที่ได้จากการสร้างชุดฝึกสอนโมเดลของงานวิจัยเดิม**

* การเพิ่มจำนวนชุดฝึกสอนสามารถช่วยเพิ่มค่าความแม่นยำขึ้นได้ และค่าความแม่นยำจะเข้าสู่ค่าคงที่เมื่อมีจำนวนชุดฝึกสอนมากถึงจำนวนหนึ่ง
* ชุดฝึกสอนควรออกแบบให้มีผลทำนายทั้ง 2 รูปแบบ คือ Allow และ Deny เพื่อให้สามารถประมวลผลข้อมูลได้ทั้ง 2 รูปแบบ
* ควรเพิ่ม Default Rule พิจารณาเพิ่มเป็นกฎที่ใช้สร้างชุดฝึกสอนด้วย
* ขนาดของข้อมูล ขอบเขตของ Packet Header ควรมีการกำหนดให้ชัดเจน เพื่อลดเวลาการฝึกอบรม

**ปัญหาจากงานวิจัยเดิมที่จะทำการศึกษาเพิ่มและต่อยอด**

* จะมีวิธีการแบ่งชุดฝึกสอนอย่างไรให้สามารถใช้ได้กับทุกกฎไฟร์วอลล์ที่มีการออกแบบ โดยจำนวนที่แบ่งได้ในแต่ละกฎไฟร์วอลล์จะต้องเป็นจำนวนที่เหมาะสมที่ไม่ทำให้ค่าความแม่นยำของโมเดลต่ำจนไม่สามารถนำมาใช้จริงได้และใช้เวลาฝึกโมเดลไม่นานจนเกินไป

จากปัญหาที่เกิดขึ้น งานวิจัยนี้จึงมีเป้าหมายไปที่การนำเสนออัลกอริทึ่มการแบ่งชุดฝึกสอนใหม่ จำเป็นต้องมีการเปรียบเทียบทั้งอัลกอริทึ่มแบบเก่าและแบบใหม่ เพื่อเป็นการพิสูจน์ว่าการแบ่งแบบ N Sample หรือ Sample per Rule ที่มีการแบ่งชุดข้อมูลฝึกสอนเท่ากันทั้งหมดโดยไม่สนขนาดของกฎไฟร์วอลล์ดังที่ได้เห็นในงานวิจัยก่อนหน้า เป็นอัลกอริทึ่มที่เหมาะสมหรือไม่และอย่างไร และแบบใหม่จะสามารถแก้ไขปัญหาได้จริงหรือไม่ จึงต้องมีการทำการทดลองทั้ง 2 อัลกอริทึ่มแบบเก่าและแบบใหม่ เพื่อให้สามารถชี้ให้เห็นถึงความแตกต่าง จุดดี จุดด้อย ของแต่ละอัลกอริทึ่มได้

ในการเลือกอัลกอริทึ่มที่เหมาะสมจริงๆนั้น จะต้องสามารถตอบโจทย์การเลือกใช้ได้ โดยปัจจัยหลักที่อาจต้องนำมาศึกษาเปรียบเทียบเพิ่ม เช่น เวลาที่ใช้ในการสร้างชุดข้อมูลฝึกสอน เวลาที่ใช้ในการฝึกโมเดล เวลาประมวลผลของโมเดล ความแม่นยำในการทำนายผลของโมเดลในด้านต่าง รวมไปถึงความสะดวกของผู้ควบคุมที่จะนำอัลกอริทึ่มมาใช้ ซึ่งบทความในส่วนนี้จะถูกกล่าวถึงในหัวข้อถัดไป สำหรับการต่อยอดพัฒนางานวิจัยใหม่จะมีวัฏจักรการดำเนินงาน ดังนี้

Diagram

Description automatically generated

รูปที่ 3.4 ระบบการทำงานหรือวัฎจักรการพัฒนาโมเดลของงานวิจัยใหม่

งานวิจัยนี้จะมุ่งเน้นไปที่การหาวิธีการแบ่งชุดฝึกสอนที่เหมาะสมในการฝึกสอนโมเดล ซึ่งจะอยู่ในกรอบสีแดงจากรูปข้างต้น และเพื่อให้สามารถพิสูจน์หาข้อเท็จจริงจึงจำเป็นต้องมีการดำเนินการพัฒนาโมเดลให้ครบวงจร จึงแบ่งขั้นตอนการทำงานเป็น 4 ขั้นตอนหลัก ดังนี้

1. แบ่งจำนวนชุดฝึกสอนตามอัลกอริทึ่ม (กรอบสีแดง)
2. สุ่มสร้างชุดแพ็คเกตฝึกสอนด้วยโปรแกรม Packet Generator (กรอบสีส้ม)
3. นำชุดฝึกสอนฝึกโมเดล Deep Neural Network ที่ออกแบบ (กรอบสีน้ำเงิน)
4. ประมวลผลโมเดลด้วยชุดทดสอบ และ บันทึกผลการทดลอง (กรอบสีเขียว)

**3.3.1. การแบ่งจำนวนชุดฝึกสอนตามอัลกอริทึ่ม**

การแบ่งชุดฝึกสอนโมเดลที่จำนวนที่เหมาะสมเป็นเป้าหมายหลักที่จะศึกษาในงานวิจัยนี้ โดยวิธีการแบ่งชุดฝึกสอนจะต้องสามารถนำไปใช้ได้กับกฎไฟร์วอลล์ที่ออกแบบมาได้ทั้งหมด ซึ่งอัลกอริทึ่มที่จะนำเสนอในงานวิจัยนี้เป็นอัลกอริทึ่มที่คิดขึ้นมาใหม่โดยใช้สมมติฐานที่ได้จากการสังเกตความสัมพันธ์ต่างของข้อมูลและแนวทางการแก้ไขจากอัลกอริทึ่มก่อนหน้า การที่จะพิสูจน์ได้ว่าอัลกอริทึ่มใหม่จะสามารถตอบโจทย์ประสิทธิภาพการทำงานของโมเดลได้นั้นจะต้องได้รับการทดลอง จนถึงการทำนายผลลัพธ์ของโมเดลหลังจากฝึกสอนเสร็จ และวิเคราะห์สรุปผลถึงข้อดี ข้อเสีย ของการนำอัลกอริทึ่มดังกล่าวมาใช้ จึงจะสามารถพิสูจน์ได้ โดยวิธีการแบ่งสร้างชุดฝึกสอนตามกฎไฟร์วอลล์ที่เราได้เสนอขึ้นมา ประกอบไปด้วย 3 วิธี ได้แก่

**3.3.1.1. อัลกอริทึ่มการแบ่งชุดฝึกสอนแบบ N Sample (งานวิจัยเดิม)**

การแบ่งชุดข้อมูลแบบ N Sample หรือการแบ่งชุดข้อมูลฝึกสอนให้มีจำนวนเท่ากันทั้งหมดในแต่ละกฎไฟร์วอลล์ อัลกอริทึ่มนี้ถูกใช้ในงานวิจัยเดิมที่เราได้อ้างอิง ซึ่งเราได้นำมาเป็นตัวต้นแบบในการเลือกวิธีการแบ่งชุดฝึกสอนที่สามารถให้ผลลัพธ์ได้ดีกว่า จำนวนที่จะนำมาใช้สร้างชุดฝึกสอนโมเดลจากกฎไฟร์วอลล์ของอัลกอริทึ่มนี้ไม่ต้องผ่านการคำนวณความซับซ้อนหรือขนาดของแต่ละกฎแต่จะมาจากเราเป็นผู้กำหนดเองทั้งหมด

เมื่อเรากำหนดค่า N หรือก็คือจำนวนที่เรากำหนดจำนวนชุดฝึกสอนต่อไฟร์วอลล์ 1 กฎ ตัวโปรแกรมจะเริ่มทำงานโดยการวนกฎไฟร์วอลล์ไปทีละกฎและสุ่มข้อมูลของแต่ละแพ็คเกตที่เข้าเงื่อนไขในกฎไฟร์วอลล์นั้นจนครบ N ตัว และจะวนเข้าไฟร์วอลล์กฎใหม่วนจนครบทุกกฎ

Graphical user interface, text, application

Description automatically generatedรูปที่ 3.5 อัลกอริทึ่มที่ใช้แบ่งชุดฝึกสอนโมเดลของ N Sample

Diagram

Description automatically generated

รูปที่ 3.6 ตัวอย่างวิธีการสร้างชุดข้อมูลฝึกสอนจากไฟร์วอลล์ที่มี 3 กฎ (Sample per Rule)

เมื่อจำนวนแพ็คเกตฝึกสอนต่อ 1 กฎไฟร์วอลล์เท่ากับ N Sample และทุกกฎไฟร์วอลล์ใช้จำนวนที่เท่ากันดังนั้นชุดข้อมูลแพ็คเกตฝึกสอนทั้งหมดที่ใช้สอนโมเดล 1 โมเดลจึงเข้าตามสูตรดังนี้

total packet (t) = จำนวนแพ็คเกตฝึกสอนที่จะนำไปสอนโมเดล

N Sample (n) = จำนวนแพ็คเกตฝึกสอนต่อ 1 กฎไฟร์วอลล์ที่เราเป็นผู้กำหนด

K = จำนวนกฎไฟร์วอลล์ที่ออกแบบมาทั้งหมดใน 1 เซต

**3.3.1.2. อัลกอริทึ่มการแบ่งชุดฝึกสอนแบบอัตราส่วน Ratio**

การแบ่งชุดข้อมูลฝึกสอนแบบอัตราส่วน Ratio เป็นสมมติฐานที่ตั้งขึ้นในงานวิจัยโดยใช้ความเป็นไปได้ทั้งหมดของกฎไฟร์วอลล์แต่ละกฎมาเป็นตัวแปรในการแบ่ง โดยมีหลักการสำคัญคือ กฎขนาดใหญ่ที่มีข้อมูลที่แตกต่างกันเข้าเงื่อนไขมากจะได้จำนวนโควต้าในชุดข้อมูลฝึกสอนมาก กฎไฟร์วอลล์ที่มีขนาดเล็กกว่าหรือมีจำนวนข้อมูลที่แตกต่างกันเข้าเงื่อนไขน้อยกว่าจะได้รับจำนวนชุดข้อมูลฝึกสอนที่น้อยกว่า ซึ่งทั้งหมดจะต้องได้รับตามอัตราส่วนจากความเป็นไปได้ทั้งหมดภายในกฎไฟร์วอลล์นั้นเท่าๆกัน

ตัวแปรที่สำคัญในอัลกอริทึ่มนี้จะได้แก่ sample space (s) จำนวนของความเป็นไปได้ของแพ็คเกตทั้งหมดที่เกิดขึ้นได้ในกฎไฟร์วอลล์นั้นๆ และ ratio (R) อัตราส่วนที่เรากำหนดให้แต่ละกฎไฟร์วอลล์ใช้ parameter นี้เท่ากันโดยมีหน่วยเป็นชุดแพ็คเกตฝึกสอนต่อ sample space เช่น ถ้าหากเรากำหนดค่าอัตราส่วนเป็น 1 ต่อ 10,000 หมายถึง โปรแกรมจะสร้างชุดข้อมูลฝึกสอน 1 แพ็คเกต เมื่อจำนวน sample space ของกฎไฟร์วอลล์นั้นมีขนาดเป็น 10,000 โดยมีสูตรเป็น

total packet (t) = จำนวนแพ็คเกตฝึกสอนที่จะนำไปสอนโมเดล

S(i) = จำนวน Sample Space แพ็คเกตฝึกสอนของกฎไฟร์วอลล์ลำดับที่ i

K = จำนวนกฎไฟร์วอลล์ที่ออกแบบมาทั้งหมดใน 1 เซต

Ratio (R) = อัตราส่วนที่กำหนด กำหนดเป็น 0.1, 0.001, 0.0001 เป็นต้น

จากสูตรข้างต้น วิธีการคำนวณเริ่มจากคำนวณหา Sample Space ของแพ็คเกตฝึกสอนในแต่ละกฎไฟร์วอลล์ซึ่งมาจากการหาผลคูณของขอบเขต Data Field ในแต่ละ Packet Header หลังจากนั้นให้นำมาคูณกับอัตราส่วนหรือตัวแปร R ที่กำหนดขึ้น ผลลัพธ์ที่ได้คือจำนวนโควต้าชุดฝึกสอนในกฎนั้นๆ โดยเมื่อคำนวณกับกฎไฟร์วอลล์อื่นก็จะยังใช้ค่า R ตัวเดียวกัน

Text

Description automatically generated  
รูปที่ 3.7 อัลกอริทึ่มที่ใช้แบ่งชุดฝึกสอนโมเดลของอัตราส่วน Ratio

การแบ่งแบบอัตราส่วนนี้ทำให้เราได้พบกับปัญหาจากอัลกอริทึ่มนี้ เมื่อเราได้นำกฎที่เราออกแบบมาลองใช้ สิ่งที่เห็นได้ชัดคืออัตราส่วนของแต่ละกฎของไฟร์วอลล์มีความแตกต่างมากเกินไป การกำหนดตัวแปร Ratio (R) เพื่อให้ได้กฎขนาดน้อยที่สุด 1 แพ็คเกต เมื่อนำมาใช้กับ Default Rule จะได้จำนวนแพ็คเกตที่ต้องสร้างสำหรับกฎนั้นมหาศาลมาก

จากปัญหาข้างต้นที่ทำให้เห็นถึง Scope ของ Default Rule ที่อัตราส่วนมีขนาดต่างกับกฎไฟร์วอลล์ทั่วไปมากเกินไป นอกจากควบคุมตัวแปรได้ยากและการที่จะต้องสร้างชุดฝึกสอนจำนวนมากตามวิธีการแบ่งนี้ จะเป็นภาระอย่างมากให้กับตัวผู้สร้างและฮาร์ดแวร์ที่ใช้ทดลอง จึงทำให้สรุปได้ว่าอัลกอริทึ่มนี้ไม่เหมาะแก่การที่จะนำมาใช้แบ่งชุดข้อฝึกสอน

Table

Description automatically generated  
ตารางที่ 3.2 จำนวนความเป็นไปได้ของกฎทั่วไปเทียบกับ Default Rule

**3.3.1.3. อัลกอริทึ่มการแบ่งชุดฝึกสอนด้วยการเข้า Log Function ในแต่ละ Data Field**

วิธีการสร้างชุดข้อมูลฝึกสอนด้วยวิธีการเข้า Log Function เป็นอีกอัลกอริทึ่มที่เสนอใหม่ จำนวนการสร้างชุดข้อมูลฝึกสอน (Sample) จะแปรผันตามค่าที่เป็นไปได้ของกฏ โดยใช้สูตรสมการที่กำหนดขึ้นที่เป็น Log function โดยจำนวนที่จะสร้างสำหรับแต่ละกฏเท่ากับผลคูณของค่า Log Function ที่หาได้จากแต่ละ Field สร้างตามกฏที่กำหนด มีการแยกการพิจารณาแบบรวม Default Rule และไม่รวม Default Rule วิธีนี้คาดว่าจะสามารถแก้ไขปัญหาในเรื่องของจำนวน Default ได้ เพราะจำนวนความเป็นไปได้ของแต่ละกฎไฟร์วอลล์นั้นผ่าน Log Function จะทำให้ค่าที่จะนำมาใช้สร้างจำนวนชุดฝึกสอนลดลงจนเป็นค่าที่พอรับมือได้

ขั้นแรกคือการหาความเป็นไปได้ของแต่ละฟิลด์ของกฎไฟร์วอลล์นั้นๆ และเมื่อหาได้แล้วให้นำค่าที่ได้ของแต่ละฟิลด์มาเข้า Log Function โดยมีสูตรเป็น

x(j) = ความเป็นไปได้ของข้อมูลทั้งหมดที่สามารถเกิดขึ้นได้ภายในฟิลด์ j

f(j) = ค่าของความเป็นไปได้ของแต่ละฟิลด์หลังผ่าน Log Function แล้ว

b = เลขฐานของฟังก์ชั่น Log โดยกำหนดเป็น 2, 4, 8, 16 เป็นต้น

และเมื่อได้ค่าของความเป็นได้ของแต่ละฟิลด์เมื่อผ่านสมการนี้แล้ว ให้นำผลที่ได้ทั้งหมด หรือ f (j) ของแต่ละฟิลด์ของกฎเดียวกันมาคูณกัน จะทำให้ได้จำนวนชุดแพ็คเกตฝึกสอนที่จะทำการสร้างภายในกฎไฟร์วอลล์นั้นๆ และทำการรวมผลบวกของจำนวนในแต่ละกฎไฟร์วอลล์เพื่อให้ได้จำนวน total packet ที่สามารถนำไปเทียบกับวิธีการแบ่งอื่นต่อไป

**3.3.2. การสุ่มสร้างชุดแพ็คเกตฝึกสอนด้วยโปรแกรม Packet Generator**

หลังจากคำนวณจนได้จำนวนชุดฝึกสอนในแต่ละกฎไฟร์วอลล์ที่ต้องการแล้ว จะเป็นขั้นตอนการสุ่มสร้างแพ็คเกตและเปลี่ยนรูปแบบให้กลายเป็นชุดฝึกสอนที่สามารถสอนโมเดลที่ออกแบบได้ โดยการทำงานทั้งหมดนี้จะทำโดยโปรแกรม Packet Generator

**โปรแกรม Packet Generator**

จุดมุ่งหมายของโปรแกรมนี้คือการสร้างชุดข้อมูลฝึกสอนที่อ้างอิงมาจากไฟร์วอลล์ตามจำนวนที่ให้ไว้ในแต่ละกฎ และปรับเปลี่ยนรูปแบบของชุดข้อมูลให้สามารถส่งต่อให้โมเดลโครงข่ายประสาทเทียมเชิงลึกที่ออกแบบขึ้นสามารถเรียนรู้ชุดข้อมูลแพ็คเกตฝึกสอนที่สร้างได้ รูปแบบชุดข้อมูลฝึกสอนสำหรับโมเดลจะประกอบไปด้วย 2 ส่วน ได้แก่ Data Field หรือส่วนข้อมูลดิบที่โมเดลจะทำการหาความสัมพันธ์ของข้อมูล โดยสุ่มมาจาก Packet Header ที่กฎของไฟร์วอลล์เป็นตัวกำหนด และ Decision Field หรือส่วนของการตัดสินใจที่โมเดลจะนำมาเปรียบเทียบคำตอบและเรียนรู้จาก Data Field

Diagram

Description automatically generated

รูปที่ 3.8 Block Diagram การทำงานของโปรแกรม Packet Generator

**หลักการออกแบบโครงสร้างชุดข้อมูลฝึกสอน**

เพื่อให้ได้รูปแบบที่เหมาะสมของชุดข้อมูลฝึกสอนที่โมเดลต้องการและสามารถนำมาประมวลผลได้ จำเป็นต้องมีการเปลี่ยนแปลงรูปแบบและมีการแทนค่าข้อมูลแพ็คเกตฝึกสอน ดังนี้

การแทนค่าเป็นเลขฐานสองใน Decision Field มีขนาด 1 bit

* Allow แทนค่า เป็น **1**
* Deny แทนค่า เป็น **0**

Data Field ที่ใช้ทั้งหมด 11 field รวมเป็น 125 bits

* 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)

และนอกจากนั้นจำเป็นต้องมีการกำหนดขอบเขตของแต่ละ Packet Header เพื่อลดเวลาการฝึกและเพิ่มความแม่นยำของโมเดล ตามข้อแนะนำการสร้างชุดฝึกสอนของงานวิจัย เนื่องจากเราทราบดีว่าแต่ละ Packet Header ไม่ได้มีขนาดความเป็นไปได้ทั้งหมดที่ 2n เสมอไปตามจำนวนขนาดของ field การกำหนดขอบเขตของ Packet Header จะส่งผลกระทบต่อกฎที่มีการระบุเงื่อนไขของ Data Field เป็น **any** โดยเฉพาะกฎ Default Rule หรือกฎที่มีข้อมูลใดๆก็ตามที่ไม่ได้อยู่ในเงื่อนไขของกฎที่เราทำการออกแบบ โดยในงานวิจัยนี้เรามองเห็นถึงปัญหาที่ถ้าหากกฎที่ออกแบบมีจำนวนมาก การเปรียบเทียบกฎทั้งหมดทีละกฎเพื่อหาว่าชุดข้อมูลนั้นเป็นแพ็คเกตจาก Default Rule หรือไม่จึงไม่ใช่ผลลัพธ์ที่ดีเนื่องจากเสียเวลามาก ดังนั้นการสร้างชุดฝึกสอนจาก Default Rule เราจึงทำการสร้างมาจาก Sample Space หรือความเป็นไปได้ของ Packet ทั้งหมด

|  |  |  |
| --- | --- | --- |
| Packet Header | จำนวนความเป็นไปได้ | หมายเหตุ |
| Version | 1 | ทุกแพ็คเกตเป็น IPv4 |
| IHL | 11 | มีค่าเป็นได้ตั้งแต่ 5-15 |
| Direction | 2 | In และ out |
| InterfaceID | 4 | กำหนดให้มี 4 Interfaces, eth 0-4 |
| DSCP | 22 | แพ็คเกตกำหนดให้มี 22 ค่า |
| ECN | 4 | เป็นได้ 4 ค่า |
| SourceIP | 232- 2 | จำนวน Host ของ /32 ทั้งหมด |
| SourcePort | 65535 | จำนวน port ทั้งหมด |
| DestinationIP | 232 – 2 | จำนวน Host ของ /32 ทั้งหมด |
| DestinationPort | 65535 | จำนวน port ทั้งหมด |
| Protocol | 2 | ได้แก่ TCP และ UDP |

ตารางที่ 3.3 ตารางความเป็นไปได้ของแต่ละ Data Field (Default Rule) งานวิจัย

เมื่อข้อมูลทั้ง Decision Field และ Data Field ถูกแปลงเป็นเลขฐานสองแล้ว ก็จะทำการแปลงเป็นชุดฝึกสอนให้อยู่ในรูปแบบ binary input ทั้งหมด โดยแปลง Data Field ที่มีข้อมูลภายในเป็น Packet Header แบ่งเลขฐานสองออกมาอีกและแบ่งข้อมูลแต่ละ bit ออกให้เหลือเพียง Data Field ละ 1 bit รูปแบบของชุดข้อมูลฝึกสอนที่ได้ก่อนนำเข้าโมเดล จำนวน Data Field ทั้งหมดจะเท่ากับขนาดของขนาดของ Packet Header ที่เรากำหนดใน 1 แพ็คเกตรวมกัน

A screenshot of a computer

Description automatically generated with medium confidenceรูปที่ 3.9 ข้อมูลแต่ละ Data Field ที่ได้จากการสุ่มเป็นเลข binary

จากรูปโปรแกรมทำการสุ่มแพ็คเกตจาก pool ความเป็นไปได้ที่กำหนดแยก Data Field ให้อยู่ในรูปของ Packet Header และหลังจากนั้นทำการแยก Data Field ให้เหลือขนาด 1 bit ตามหลักการแปลงรูปข้อมูลก่อนเข้าโมเดล จากนั้นบันทึกลงไฟล์เอกสารเพื่อเตรียมเป็นชุดฝึกสอนในรูปของ Input ที่จะนำมาให้โมเดลทำการอ่านข้อมูล

**A screenshot of a computer

Description automatically generated**รูปที่ 3.10 ข้อมูลแต่ละ Data Field ที่แยกมาจาก Packet Header

A screenshot of a computer

Description automatically generated with low confidenceรูปที่ 3.11 การจัดเก็บแพ็คเกตให้อยู่ในรูปแบบ binary ก่อนที่จะนำเข้าโมเดล

**3.3.3. การนำชุดฝึกสอนฝึกโมเดล Deep Neural Network ที่ออกแบบ**

**Deep Neural Network Model Engine**

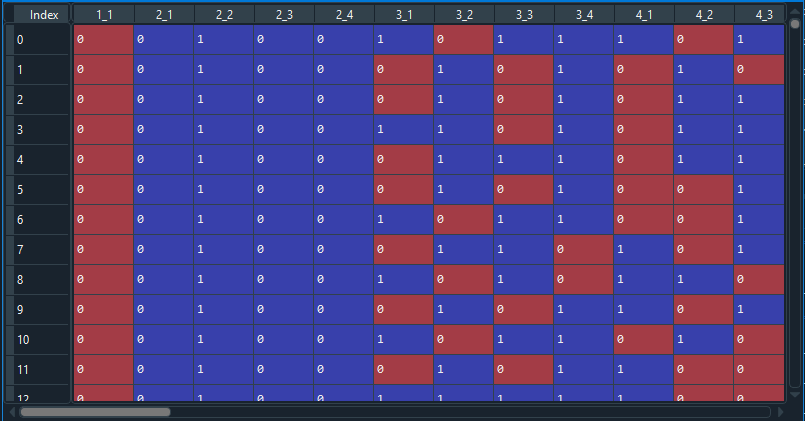
Deep Neural Network Model คือโมเดลที่สามารถหาความสัมพันธ์ของข้อมูล วิเคราะห์ตัวแปรต่างๆและทำนายออกมาเป็นผลลัพธ์ผ่านชุดข้อมูลที่เราเป็นผู้กำหนด ซึ่งเมื่อเราได้ออกแบบและได้ Deep Neural Network Model หลังจากที่ผ่านการฝึกด้วยชุดฝึกสอนเสร็จเป็นที่เรียบร้อยแล้ว ตัวของโมเดลจะสามารถทำงานได้คล้ายคลึงกับ Intrusion Detection System Firewall ที่มีความสามารถในการตรวจดักจับแพ็คเกตที่เข้ามาในระบบ โดยในงานวิจัยนี้คือการให้โมเดลทำนายแพ็คเกตว่าจะอนุญาติให้ผ่าน (Allow) หรือไม่อนุญาติให้ผ่าน (Deny)

Diagram

Description automatically generated

รูปที่ 3.12 Block Diagram ขั้นตอนการนำชุดฝึกสอนมาฝึกโมเดลที่ออกแบบ

ขั้นตอนการฝึกสอนโมเดลจะเริ่มจากกำหนดค่าพารามิเตอร์สำหรับการออกแบบโครงสร้างโมเดล ซึ่งโมเดลที่ถูกสร้างขึ้นใหม่นี้จะถูกเรียกว่า Blank Model หรือโมเดลที่ยังไม่ผ่านการฝึก และเราจะนำชุดฝึกสอนที่ได้จากโปรแกรมก่อนหน้ามาทำการฝึก Blank Model นี้ด้วย Deep Neural Network ซึ่งจะมีวิธีการคำนวณความสัมพันธ์และน้ำหนักของข้อมูลเชิงลึกผ่านไลบราลี Tensorflow และ Keras และสรุปความสัมพันธ์ที่ได้ผลลัพธ์ออกมาเป็นโมเดล ถือเป็นอันเสร็จในขั้นตอนนี้



รูปที่ 3.13 รูปแบบ Datagram ที่ใช้คำนวณค่าน้ำหนักของชุดฝึกสอนโมเดล

โครงสร้างของโมเดลที่จะนำมาฝึกจะถูกกำหนดขึ้นอยู่กับรูปแบบความสัมพันธ์ของข้อมูล รูปแบบของคำตอบที่ต้องการ ซึ่งค่าเหล่านี้จะเป็นการกำหนดรูปแบบอัลกอริทึ่มเชิงลึกคำนวณค่าน้ำหนักด้วยไลบราลี Tensorflow และ Keras ที่ใช้สำหรับพัฒนาโครงข่ายประสาทเทียม ในปัจจุบันนั้นจำนวน Nodes และ Epochs ที่เหมาะสมที่จะเป็นตัวแปรในการนำมาสร้างและฝึกโมเดลยังไม่มีค่าที่ระบุมาชัดเจน แต่จะมีเพียงแค่คำแนะนำในการเลือกจำนวนที่ไม่ก่อให้เกิดปัญหาในการฝึกเช่น Underfitting หรือ Overfitting โดยคำแนะนำนี้ถูกเรียกว่า Rule of Thumbs ซึ่งโครงสร้างโมเดลที่จะนำมาใช้ในงานวิจัยนี้ เคยมีงานวิจัยที่ศึกษาเกี่ยวกับการสร้างชุดฝึกสอนไฟร์วอลล์ซึ่งได้ความว่า จำนวน Nodes กับ Epochs ไม่ค่อยส่งผลกับความแม่นยำมากแม้เพิ่มจำนวนมากขึ้น ดังนั้นเราจึงเลือกใช้ Node และ Epochs ตามงานวิจัยดังกล่าวเลือกใช้ คือ 3 layers, 40 nodes, 30 epochs

**3.3.4. การประมวลผลชุดทดสอบ และสรุปผลการดำเนินงาน**

ในขั้นตอนการทดสอบโมเดลจะมีการสร้างชุดข้อมูลสำหรับทดสอบโดยเฉพาะ ชุดข้อมูลทดสอบจะมีโครงสร้างเหมือนกับชุดฝึกสอนโมเดลยกเว้น Decision Field ที่จะไม่ใส่ลงไปด้วย โดยโมเดลที่ผ่านฝึกแล้วจะวิเคราะห์จาก Data Field ที่เหลือเพื่อทำนายว่า Decision Field จะให้ผลลัพธ์อะไรระหว่าง Allow และ Deny โดยขั้นตอนนี้จะทำผ่านโปรแกรม Evaluate Engine

**3.3.4.1. Evaluate Engine / Compare Engine**

โปรแกรม Evaluate หรือ Compare Engine เป็นการทดสอบด่านสุดท้ายว่าโมเดลมีผลลัพธ์การให้ค่าความแม่นยำเป็นอย่างไร การวัดผลความถูกต้องของโมเดลจะวัดโดยการเปรียบเทียบระหว่างผลลัพธ์การตัดสินใจของโมเดลและ Decision Field ของแพ็คเกตที่ทำการทดสอบ การทดสอบการประมวลผลเพื่อหาค่าความแม่นยำของโมเดล เราจะใช้ชุดอ้างอิงจากชุดกฎของไฟร์วอลล์เดียวกัน แต่ไม่ได้ใช้ชุดข้อมูลที่สุ่มมาเหมือนกัน เพื่อให้เราสามารถวิเคราะห์ได้ว่าโมเดลสามารถตัดสินใจทำนายข้อมูลรูปแบบใหม่ที่มีความสัมพันธ์กันได้อย่างถูกต้องด้วยจำนวนชุดข้อมูลฝึกสอนที่กำหนดให้ไปอย่างจำกัดตามอัลกอริทึ่มการแบ่งชุดข้อมูลฝึกสอน

Diagram

Description automatically generated

รูปที่ 3.14 Block Diagram การทำงานของโปรแกรม Compare Engine

**3.3.4.2. การออกแบบชุดข้อมูลทดสอบ**

ชุดข้อมูลสำหรับทดสอบโมเดลเราจะใช้วิธีคิดเดียวกันกับงานวิจัยเดิม คือ แบ่งชุดข้อมูลทดสอบออกเป็น 3 ชุด ได้แก่

* ชุดของแพ็คเกตที่อยู่ภายใต้เงื่อนไขกฎไฟร์วอลล์ที่เรากำหนด
* ชุดของแพ็คเกตที่ไม่ได้อยู่ภายใต้เงื่อนไขกฎไฟร์วอลล์ที่เรากำหนด หรือ Default Rule
* ชุดของแพ็คเกตที่มีทั้ง 2 แบบแรก หรือ แบบผสม

แนวความคิดการออกแบบชุดทดสอบออกเป็น 3 ชุดนั้น เพื่อให้สามารถทราบว่าการออกแบบชุดฝึกสอนโดยอ้างอิงกับกฎไฟร์วอลล์เกี่ยวข้องอย่างไรกับกฎไฟร์วอลล์ที่ออกแบบ จึงต้องแบ่งชุดทดสอบเป็นชุดทดสอบในกฎไฟร์วอลล์ ชุดทดสอบนอกกฎไฟร์วอลล์ และชุดข้อมูลทดสอบแบบผสม โดยชุดทดสอบจะสามารถอธิบายความสัมพันธ์กับ Default Rule ได้อีกทีหนึ่ง

เมื่อโมเดลทำนายผลลัพธ์จากชุดทดสอบจะสามารถแจกแจงความแม่นยำได้ออกเป็น 4 รูปแบบ โดยทั้ง 4 รูปแบบนี้จะบอกว่าโมเดลมีการทำนายถูกต้องและผิดพลาดอย่างไรกับชุดข้อมูลทดสอบเมื่อนำมาเปรียบเทียบกับคำตอบที่แท้จริง รูปแบบการเปรียบเทียบนี้ถูกเรียกว่า Reference Variant Set ซึ่งเป็น Matrix ที่ใช้ในการทดสอบความแม่นยำ มักถูกใช้กับโมเดลที่มีการแก้ปัญหาแบบแบ่งกลุ่มเชิงคุณภาพ โดยผลลัพธ์ที่ได้ประกอบไปด้วย

* True Positive

โมเดลอนุญาติให้ข้อมูลผ่านตรงตามกฏของไฟร์วอลล์ หรือให้ Allow ถูกต้อง

* True Negative

โมเดลไม่อนุญาติให้ข้อมูลผ่านตรงตามกฏของไฟร์วอลล์ หรือให้ Deny ถูกต้อง

* False Positive

โมเดลอนุญาติให้ข้อมูลผ่านไม่ตรงตามกฏของไฟร์วอลล์ หรือให้ Allow ผิดพลาด

* False Negative

โมเดลไม่อนุญาติให้ข้อมูลผ่านไม่ตรงตามกฏของไฟร์วอลล์ หรือให้ Deny ผิดพลาด

Table

Description automatically generated

รูปที่ 3.15 Reference Variant Set ที่ใช้เปรียบเทียบความถูกต้องโมเดล

**3.3.4.3. การบันทึกผลลัพธ์ทั้งหมดและวิเคราะห์จากข้อมูลที่ได้**

ขั้นตอนสุดท้ายเป็นการสรุปผลที่ได้จากโปรแกรมทั้งหมดและรวมไปถึงการทำงานของโมเดลที่สร้างมาจากอัลกอริทึ่มและตัวแปรที่ต่างกัน โดยค่าผลลัพธ์ที่ได้ทั้งหมดนั้นจะสรุปลงตารางและกราฟในภายหลัง เพื่อให้สามารถนำมาวิเคราะห์ให้เห็นถึงความสัมพันธ์ระหว่างตัวแปรต่างๆและข้อมูลได้ง่ายขึ้น โดยผลการดำเนินงานทั้งหมดในขั้นตอนนี้จะอธิบายเพิ่มในบทถัดไป

ผลลัพธ์ที่ได้จากโปรแกรม Packet Generator

* Generating Time เวลาที่ใช้ในการสร้างชุดข้อมูลทั้งหมด

ผลลัพธ์ที่ได้จากโปรแกรม DNN Model Engine

* Training Time เวลาที่ใช้ในการฝึกโมสอนเดล
* ค่า Loss หรืออัตราข้อมูลที่สูญเสีย
* Train Accuracy ความแม่นยำของข้อมูลในการฝึกสอนโมเดล

ผลลัพธ์ที่ได้จากโปรแกรม Compare Engine

* Evaluate Time เวลาในการประมวลผลของโมเดล
* Test Accuracy ความแม่นยำในการทำนายชุดทดสอบ
* ค่าความแม่นยำจาก Reference Variant Set 4 ค่า

**ตัวอย่างรูปแบบผลลัพธ์ที่ได้จากผลการทดลอง**

รูปที่ 3.16 ตัวอย่างตารางเก็บบันทึกผลการดำเนินการ

รูปที่ 3.17 ตัวอย่างกราฟวิเคราะห์ที่ได้จากการบันทึกผลการดำเนินการ