**บทที่ 1**

**บทนำ**

**1.1 ความเป็นมาของโครงงาน**

Firewall ถูกสร้างขึ้นเพื่อจุดประสงค์ทางด้านความปลอดภัยทางเครือข่าย มีหน้าที่เปรียบเสมือนยามเฝ้าประตู โดยข้อมูลภายในเครือข่ายจะผ่านการคัดกรองข้อมูลด้วยหลักการของ Packet Filtering เมื่อเวลาผ่านไป การพัฒนาของเทคโนโลยีใหม่ๆและรูปแบบการโจมตีทางเครือข่ายที่มีมากขึ้น Firewall แบบเก่าที่กาหนดโดยผู้ควบคุมระบบเพียงอย่างเดียว ไม่สามารถตอบโจทย์ทางด้านความปลอดภัยได้ ทำให้มีการนำปัญญาประดิษฐ์ หรือ AI มาประยุกต์ใช้งานกับ Firewall ให้มีความคิดและตัดสินใจเลือกคัดกรอง Packet ได้เอง ผู้จัดทำมีความคิดที่จะพัฒนา AI Firewall ให้มีประสิทธิภาพมากยิ่งขึ้น โดยนำมาประยุกต์ใช้กับกระบวนการเรียนรู้แบบ Deep Neural Network และมีชุดข้อมูล Packet ฝึกสอนที่สร้างขึ้นอ้างอิงตามนโยบายข้อกำหนดจาก Firewall Rules เพื่อแก้ปัญหาข้อมูลฝึกสอนที่ไม่ได้เป็นไปตามนโยบายข้อกำหนด ที่แต่เดิมต้องเอาข้อมูลการโจมตีที่เคยเกิดขึ้นมาก่อนเป็นข้อมูลฝึกสอน

* 1. **วัตถุประสงค์**

1. เพื่อให้เข้าใจหลักการทำงานของ Neural Network ที่จะใช้พัฒนาปัญญาประดิษฐ์
2. เพื่อพัฒนาสร้างชุดข้อมูลฝึกสอนให้เป็นไปตามนโยบายข้อกำหนดตาม Firewall Rules
3. เพื่อให้ชุดข้อมูล Network Packet ที่สร้างขึ้นสามารถฝึกสอนได้อย่างถูกต้องและแม่นยำ เมื่อนำไปใช้กับ AI ที่มีการเรียนรู้แบบ Deep Neural Network Model
4. เพื่อแก้ไขข้อจำกัดของข้อมูลฝึกสอน Firewall ให้ผ่านเงื่อนไขที่กำหนด เช่น เวลาที่ใช้ หรือปริมาณของข้อมูล Packet

**1.3 วิธีการดำเนินงาน**

พัฒนาสร้างชุดข้อมูลฝึกสอน Network Packet ที่สร้างขึ้นโดยมีการอ้างอิงจาก Firewall Rules ไปใช้กับ AI Firewall ที่มีการเรียนรู้แบบ Neural Network Model และทำการตรวจสอบความถูกต้อง ความผิดพลาดที่ได้ เปรียบเทียบกับ Firewall Rules ที่กำหนด โดยทำการทดลองหลายๆครั้ง เปลี่ยนตัวแปรและปัจจัยต่างๆ เพื่อหาวิธีการที่ทาให้ระบบสามารถทำงานได้อย่างถูกต้องและมีประสิทธิภาพมากที่สุด

**1.4 ขอบเขตของโครงงาน**

พัฒนา Neural Network Model และชุดข้อมูลฝีกสอน Network Packet ที่สร้างขึ้นโดยอ้างอิงจาก Firewall Rules นำไปผ่านการเรียนรู้และทำการทดสอบ ลองเปลี่ยนปัจจัยและค่าตัวแปรต่างๆ เปรียบเทียบผลลัพธ์ในแต่ละรูปแบบ ใช้ความถูกต้อง ความผิดพลาดที่อิงจากกฏของ Firewall Rules เป็นเกณฑ์ในการวัดผล ศึกษาหาวิธีการและผลลัพธ์ที่ดีที่สุดภายใต้การทำงานของโปรแกรมคอมพิวเตอร์

* 1. **ประโยชน์ที่คาดว่าจะได้รับ**

1. พัฒนาทักษะการเขียนโปรแกรมที่เขียนด้วยภาษา Python
2. เรียนรู้วิธีการสร้างชุดข้อมูลฝึกสอน สามารถประยุกต์ใช้กับปัญญาประดิษฐ์ได้
3. เรียนรู้วิธีการพัฒนาอัลกอรึทึ่มที่ช่วยลดเวลา เพิ่มประสิทธิภาพในการคำนวณข้อมูลได้
4. สามารถประยุกต์ learning model ไปใช้กับปัญญาประดิษฐ์รูปแบบอื่น เช่น การทำแชทบอท โปรแกรมวิเคราะห์ข้อมูล หรือ ระบบปฏิบัติการตอบโต้อัตโนมัติ

# บทที่ 2 ทฤษฏีการนำโครงข่ายระบบประสาทเชิงลึก มาใช้ในการทำงานของไฟร์วอลล์

## เทคโนโลยีของไฟร์วอลล์และโครงข่ายระบบประสาทเชิงลึก

* + 1. **Firewall**

Firewall เป็นระบบควบคุมและรักษาความปลอดภัยของระบบเครือข่าย คัดกรองข้อมูลเข้าออกในช่องทางอินเตอร์เน็ต เปรียบเสมือนยามเฝ้าประตู คอยป้องกันการโจมตี สแปม ผู้บุกรุกต่างๆที่ไม่หวังดีต่อระบบ และยังสามารถใช้ควบคุมการใช้งานของโปรแกรมที่ต้องการ ในปัจจุบันมีการใช้งานได้ทั้งระบบ Hardware และ Software ขึ้นอยู่กับความเหมาะสม ผลลัพธ์ที่ออกมาจาก Firewall จะพิจารณาการกระทำของ Packet ออกมาเป็น Allow หรือ Deny

**Packet Filtering**

ระบบการทำงานของ Firewall ทำงานในระบบ Internet Layer และ Transport Layer ตรวจสอบและคั่นกรอง Packet ที่เข้ามาในเครือข่าย โดยพิจารณาจาก Packet Header ตัดสินใจว่าจะทำการ Allow หรือ Deny โดยใช้กฎของ Firewall ในการอ้างอิง ซึ่ง Firewall แบ่งประเภทตามลักษณะการทำงาน ได้แก่

Diagram

Description automatically generated

**รูปที่ 2.1** กระบวนการทำงานของกลไก Packet Filtering Firewall

* + - 1. **Stateful Filtering**

Stateful Filtering จะมีเก็บสถานะ Packet ใดที่เคยถูกปล่อยผ่านและเก็บบันทึกไว้ใน State Table ทำให้การทำงานของ Firewall นี้จะถูกตรวจสอบเริ่มจากที่ State Table ก่อน ถ้าหาก Packet ที่กำลังถูกตรวจสอบอยู่ยังไม่เคยถูกปล่อยผ่านยังไม่มีการเก็บสถานะเอาไว้ถึงจะไปพิจารณากฎของไฟรวอลล์เป็นอันดับถัดไป กลไกนี้จะช่วยไฟร์วอลล์ทำงานได้เร็วขึ้น เพราะช่วยลดระยะเวลาในการทำงานไม่ต้องเสียเวลาพิจารณาทุก Packet Header ในกลไก Packet Filtering

* + - 1. **Application Firewall**

มีชื่อเรียกได้อีกอย่างหนึ่งว่า “Application-level Firewall” หรือ “Application Gateway” เป็น Firewall ชนิดที่ติดตั้งบนเครื่องคอมพิวเตอร์แยกต่างหาก ทำให้คอมพิวเตอร์เครื่องดังกล่าวทำหน้าที่เป็น Firewall โดยเฉพาะ อย่างไรก็ตามApplication Firewall สามารถกรอง Packet ที่จะผ่านเข้ามาในเครือข่ายอีกทั้งยังตรวจสอบเนี้อหาใน Packet ได้เช่นเดียวกับ Stateful Filtering Firewall นอกจากนี้ Application Firewall ยังทำหน้าที่คล้ายกับ Proxy Server ในการให้บริการคำร้องขอของผู้ใช้ได้อีกด้วย โดยความสามารถของ Application Firewall สามารถแบ่งทำได้ดังนี้

**Security**

การยืนยันตัวตนด้วย AAA คือ Authentication, Authorization และ Audit โดยการสร้าง Token ไปให้ทั้งผู้รับ และผู้ส่ง มีการกำหนด Policy เพื่อการเข้าถึงข้อมูล และยังทำการเก็บข้อมูลการเข้าออกของ Policyนั้นๆ อีกทั้งยังมีการป้องกันด้วยการตรวจสอบข้อมูลที่ได้รับก่อนว่าถูกต้องตามโครงสร้างที่ได้กำหนดไว้หรือไม่

**Integration**

การสร้างการเชื่อมต่อเข้ากับระบบต่างๆให้สามารถทำงานร่วมกันได้ เช่น ถ้าหากระบบที่ใช้มีโปรโตคอลที่แตกต่างกัน มันจะทำการแปลงโครงสร้างข้อความโดยการจับคู่ข้อมูล

**Control and Managing**

การควบคุมปริมาณของข้อความที่จะวิ่งเข้าไปหา Server โดยการกำหนด Policy แยกตามประเภทของ API และประเภทของข้อมูล สำหรับการควบคุมปริมาณข้อความนี้จะเป็นการป้องกันการถูกผู้ไม่หวังดีโจมตีจากช่องโหว่ของระบบได้ เช่น เรามี API ที่เปิดให้ลูกค้าหรือบุคคลอื่นๆเข้ามาใช้งานได้ ถ้าหากไม่มีการกำหนดปริมาณการเรียกใช้ API หรือเส้นทางของข้อมูล ก็จะเกิดช่องโหว่ของระบบที่ผู้ไม่หวังดีสามารถทำการ DOS ได้

**Optimizing**  
  การลดภาระการทำงาน ของ Server โดยการทำ SSL และนำภาระงานจากการถอดรหัสที่ Server ไปให้ไฟร์วอลล์ทำงานแทน จะทำให้ Server มีทรัพยากรเหลือพอที่จะรองรับการทำงานมากขึ้น

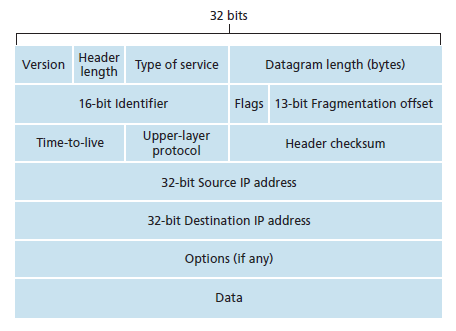
**A picture containing application

Description automatically generated**

**รูปที่ 2.2** กระบวนการทำงานของ Application Firewall

### Packet Header

### Packet Header เป็นโปรโตคอลอินเตอร์เน็ต มาตรฐานที่ทำให้อินเตอร์เน็ตสามารถเชื่อมต่อเข้าหากัน ติดต่อสื่อสารข้อมูลได้ด้วยการกำหนดวิธีการติดต่อสื่อสารร่วมกัน ในส่วนของ Packet Header จะเป็นลำดับชั้นโครงสร้างประกอบไปด้วย Field ข้อมูลที่บ่งบอกถึงวัตถุประสงค์และลักษณะการทำงานของ Packet โดยองค์ประกอบของ Packet Header มีดังนี้



**รูปที่ 2.3** ส่วนประกอบที่สำคัญของ Packet Header Datagram

* Version ส่วนที่ระบุเวอร์ชั่นโปรโตคอลของ Datagram
* Header length ส่วนที่ระบุขนาดของ Datagram Header
* Type of service ส่วนที่ระบุประเภทของ Datagram
* Datagram length ส่วนที่ระบุขนาดของ Datagram ทั้งหมดรวมถึง Datagram Header
* Identifier ส่วนที่มีไว้เพื่อยืนยันตัว หากมีการทำ Fragmention
* Flags ส่วนที่ระบุว่า Datagram นี้จะทำการ Fragmention หรือไม่
* Fragmentation offset ส่วนที่แสดงให้เห็นถึงจำนวนของข้อมูลก่อนทำการ Fragmention
* Time-to-live ส่วนที่กำหนดวงจรชีวิตของ Datagram เพื่อป้องกันไม่เกิด Loop ในเครื่อข่าย
* Protocol ส่วนที่ระบุโปรโตคอลที่ใช้ใน Datagram นี้
* Header checksum ส่วนที่ใช้สำหรับตรวจสอบความถูกต้อง Datagram Header
* Source and destination IP addresses ส่วนที่ระบุที่อยู่ของ IP ต้นทางกับ IP ปลายทาง
* Options ส่วนเพิ่มเติมที่คอยเก็บข้อมูลเช่น เส้นทางที่ใช้โดยเก็บไว้เพื่อตรวจสอบการทำงาน

### Artificial Intelligent

Artificial Intelligence คือ เครื่องจักรอัจฉริยะที่มีความสามารถในการทำความเข้าใจ เรียนรู้องค์ความรู้ต่างๆ เช่น การรับรู้ การให้เหตุผล ในการแก้ไขปัญหาต่างๆเพื่อปฎิบัติงานตามความต้องการของมนุษย์ เครื่องจักรที่มีความสามารถนี้ถูกเรียกอีกชื่อหนึ่งว่า “ปัญญาประดิษฐ์”

A picture containing diagram

Description automatically generated

**รูปที่ 2.4** ขั้นตอนกระบวนการฝึกฝนปัญญาประดิษฐ์

AI ถูกจำแนกเป็น 3 ระดับตามความสามารถดังนี้

**Narrow Artificial Intelligent** ปัญญาประดิษฐ์เชิงแคบ คือ AI ที่มีความสามารถเฉพาะทางได้ดีกว่ามนุษย์ เช่น เครื่องจักรที่ใช้ในการผ่าตัด

**General Artificial Intelligent** ปัญญาประดิษฐ์ทั่วไป คือ AI ที่มีความสามารถระดับเดียวกับมนุษย์สามารถทำทุกอย่างในประประสิทธิภาพที่ใกล้เคียงกับมนุษย์

**Strong Artificial Intelligent** ปัญญาประดิษฐ์แบบเข้ม คือ AI ที่มีความสามารถมากกว่ามนุษย์ในหลายๆด้าน

และจากการนำปัญญาประดิษฐ์มาประยุกต์ใช้ในการแก้ไขปัญหา มุมมองต่อ AI ที่แต่ละคนมีอาจไม่เหมือนกัน ขึ้นอยู่กับว่า เราต้องการความฉลาดโดย คำนึงถึงพฤติกรรมที่มีต่อสิ่งแวดล้อมหรือคำนึงการคิดได้ของผลผลิต AI ดังนั้นจึงมีคำนิยาม AI ตามความสามารถที่มนุษย์ต้องการ ให้มันแบ่งได้ 4 กลุ่ม ดังนี้

**Thinking humanly** (การคิดคล้ายมนุษย์)

natural language processing สื่อสารกับ มนุษย์ได้ด้วยภาษาที่มนุษย์ใช้ เช่น ภาษาอังกฤษ เป็นการประมวลผลภาษาธรรมชาติ

computer vision มีประสาทรับสัมผัสคล้ายมนุษย์ เช่นคอมพิวเตอร์วิทัศน์ รับภาพได้โดยใช้อุปกรณ์รับสัญญาณภาพ

machine learning เพื่อปรับให้เข้ากับสถานการณ์ใหม่และ ตรวจจับและคาดการณ์รูปแบบ

**Thinking rationally (คิดอย่างมีเหตุผล)**

คิดอย่างมี เหตุผล หรือคิดถูกต้อง โดยใช้หลักตรรกศาสตร์ในการคิดหาคำตอบอย่างมีเหตุผล เช่น ระบบผู้เชี่ยวชาญ

**Acting humanly (การกระทำคล้ายมนุษย์)**

การคิดคล้าย มนุษย์ ก่อนที่จะทำให้เครื่องคิดอย่างมนุษย์ได้ ต้องรู้ก่อนว่ามนุษย์มีกระบวนการคิดอย่างไร ซึ่งการวิเคราะห์ลักษณะการคิดของมนุษย์เป็นศาสตร์ด้าน cognitive science เช่น ศึกษาโครงสร้างสามมิติของเซลล์สมอง การแลกเปลี่ยนประจุไฟฟ้าระหว่างเซลล์สมอง วิเคราะห์การเปลี่ยนแปลงทางเคมีไฟฟ้าในร่างกายระหว่างการคิด ซึ่งจนถึงปัจจุบันเราก็ยังไม่รู้แน่ชัดว่า มนุษย์เรา คิดได้อย่างไร

**Acting rationally (การกระทำอย่างมีเหตุผล)**

กระทำอย่างมีเหตุผล เช่น agent (agent เป็นโปรแกรมที่มีความสามารถในการกระทำ หรือเป็นตัวแทนในระบบอัตโนมัติต่าง ๆ ) สามารถกระทำอย่างมีเหตุผลคือ agent ที่กระทำการเพื่อบรรลุเป้าหมายที่ได้ตั้งไว้ เช่น agent ใน ระบบขับรถอัตโนมัติที่มีเป้าหมายว่าต้องไปถึงเป้าหมายในระยะทางที่สั้นที่ สุด ต้องเลือกเส้นทางที่ไปยังเป้าหมายที่สั้นที่สุดที่เป็นไปได้จึงจะเรียกได้ ว่า agent กระทำอย่างมีเหตุผล อีกตัวอย่างเช่น agent ใน เกมหมากรุกมีเป้าหมายว่าต้องเอาชนะคู่ต่อสู้ ต้องเลือกเดินหมากที่จะทำให้คู่ต่อสู้แพ้ให้ได้ เป็นต้น

* + 1. **Machine Learning**

Machine Learning คือ ส่วนการเรียนรู้ของเครื่อง ถูกใช้งานเสมือนเป็นสมองของปัญญาประดิษฐ์ในการสร้างความฉลาด มักจะใช้เรียกโมเดลที่เกิดจากการเรียนรู้ของปัญญาประดิษฐ์ โดยมนุษย์มีหน้าที่เขียนโปรแกรมให้เรียนรู้จากชุดข้อมูลฝึกสอนหรือ Training set และอาศัยกลไกที่เป็นโปรแกรม หรือเรียกว่า Algorithm ที่มีหลากหลายแบบ โดยมี Data Scientist เป็นผู้ออกแบบ หนึ่งใน Algorithm ที่ได้รับความนิยมสูง คือ Deep Learning ซึ่งถูกออกแบบมาให้ใช้งานได้ง่าย และประยุกต์ใช้ได้หลายลักษณะงาน อย่างไรก็ตาม ในการทำงานจริง Data Scientist จำเป็นต้องออกแบบตัวแปรต่างๆ ทั้งในตัวของ Deep Learning เอง และต้องหา Algorithm อื่นๆ มาเป็นคู่เปรียบเทียบ เพื่อมองหา Algorithm ที่เหมาะสมที่สุดในการใช้งานจริง โดยตามหลักแล้วจะแบ่งประเภทของ Machine Learning ได้ดังนี้

* + - 1. **Supervised**

การทำให้เครื่องคอมพิวเตอร์สามารถเรียนรู้ได้จากชุดข้อมูลฝึกสอนหรือ Training set ก่อนที่จะประมวลผล โดยมนุษย์จะเป็นผู้กำหนดคุณลักษณะความสัมพันธ์เฉพาะของข้อมูลที่ต้องการให้เครื่องคอมพิวเตอร์เรียนรู้ หรือที่เรียกว่า Label และเมื่อโมเดลผ่านการเรียนรู้แล้ว จะสามารถแยกแยะประเภทมีวิธีการคิดที่เริ่มมีเหตุผล เมื่อข้อมูลที่ต้องการวิเคราะห์มีจำนวนที่มากขึ้นจำเป็นต้องมีข้อมูลที่เป็น Training set มากขึ้นเช่นเดียวกัน โดยการเรียนรู้แบบ Supervised Learning นี้จะประกอบไปด้วยดังนี้

* + - * 1. **Classification**

คือการสอนโมเดลให้สามารถแบ่งหรือแยกประเภทกลุ่มข้อมูลได้ โดยอ้างอิงจากความสัมพันธ์และน้ำหนักของข้อมูลแต่ละ Label ตัวอย่างเช่น การแยกกลุ่มผู้ป่วยว่าเป็นเนื้องอกในสมอง ซึ่งจะมีปัจจัยต่างๆมากมายไม่ว่าจะเป็น ขนาด, รูปร่าง, ตำแหน่ง หรือ สีผิว ซึ่งถ้าหากมีข้อมูลเพียงแค่ Label เดียว ไม่สามารถพิสูจน์หรือแบ่งกลุ่มได้

* + - * 1. **Regression**

การสอนโมเดลโดยอิงจากผลลัพธ์ที่ผ่านมา โดยผลลัพธ์จะเป็นการประมาณค่าความเป็นไปได้ที่จะเกิดขึ้นต่อ ทำให้เหมาะแก่การวิเคราะห์ความสัมพันธ์ของตัวแปรที่อยู่ในรูปกราฟ เช่น การหาความสัมพันธ์ระหว่างขนาดของบ้านและราคา การประเมินราคาหุ้น

* + - 1. **Unsupervised**

รูปแบบการเรียนรู้ที่ไม่จำเป็นต้องใช้ชุดข้อมูลฝึกสอน แต่เป็นการป้อนข้อมูล Test set ไปประมวลผลเพียงอย่างเดียว ทำให้ผลลัพธ์ที่ออกมาไม่รู้ผลลัพธ์แน่ชัด ซึ่งอัลกอรึทึ่มจะวิเคราะห์และหาโครงสร้างของข้อมูลเอง

* + - * 1. **Clustering**

เป็นการกำหนดให้เครื่องคอมพิวเคอร์หาวิธีแบ่งกลุ่มหรือจัดกลุ่มข้อมูลเอง เปรียบเสมือนการลด Label ของข้อมูลที่มีปริมาณมาก จัดกลุ่มหาข้อมูลที่มีความสัมพันธ์ใกล้เคียงกัน ผลลัพธ์ที่ได้ออกมาจะมีปริมาณ Label ที่น้อยลงเป็นอย่างมาก

* + - * 1. **Dimensionality Reduction**

เป็นการกลไกการบีบอัดและลดมิติข้อมูลจำนวนมากให้มีจำนวนลดลงโดยที่ข้อมูลยังครบถ้วน และยังสามารถนำไปจำแนกข้อมูลได้เหมือนเดิม

Diagram

Description automatically generated

**รูปที่ 2.5** ขั้นตอนการแยกหมวดหมู่และรูปแบบโมเดลที่จะศึกษา

* + - 1. **Reinforcement Learning**

เป็นการเรียนรู้ด้านหนึ่งของ Machine Learning มักใช้พัฒนาหุ่นยนต์หรือการเรียนรู้ที่อยู่ภายในเกมคอมพิวเตอร์ เช่น การลองผิดลองถูกไปเรื่อยเพื่อหาผลลัพธ์ที่ดีที่สุดประเมินออกมาเป็นคะแนน โดยชุดข้อมูลทดสอบจะเป็นสภาพแวดล้อมโดยรอบขึ้นอยู่กับความต้องการของผู้พัฒนา

* + 1. **Deep Learning**

Deep learning คือ อัลกอริทึมการเรียนรู้เชิงลึกโดยใช้หลักการ Artificial Neural Networks ที่มีรูปแบบการทำงานคล้ายคลึงกับเซลล์ประสาทที่เชื่อมต่อกันเป็นโครงข่ายประสาทในสมองมนุษย์ เหมาะกับการวิเคราะห์ข้อมูลขนาดใหญ่ที่มีความซับซ้อน เช่น การจำแนกรูปภาพ การจำแนกใบหน้า ประกอบไปด้วย โครงสร้างของหน่วยประมวลผลจำนวนมากคือเซลล์ประสาท หรือ Neuron โดยอัลกอริทึ่มนี้จะประกอบไปด้วยชั้นต่างๆ ดังนี้

Input Layer มีหน้าที่รับข้อมูลเข้ามาประมวลผลและส่งต่อให้ Hidden Layer

Hidden Layer มีหน้าที่คำนวณและประมวลผลข้อมูลโดยสามารถมีได้หลายชั้น หลายขนาดขึ้นอยู่กับความซับซ้อนของข้อมูล

Output Layer มีหน้าที่ส่งผลลัพธ์ข้อมูลที่ผ่านการประมวลผลแล้วออกมา

เมื่อเริ่มการฝึกฝนจะเริ่มจากการสุ่มค่าถ่วงน้ำหนัก (Weight) และจะเริ่มปรับผลลัพธ์เอามาคูณกับค่าถ่วงน้ำหนักแล้วบวกด้วยค่าความเอนเอียงของข้อมูล (Bias) หลังจากนั้นจะนำผลลัพธ์ที่ได้มาในแต่ละขาของ Neural Network มารวมกันแล้วมาผ่านฟังก์ชั่นส่งต่อให้ลำดับชั้นถัดไปประมวลผลมีการใช้วิธีการประมวลผลทางคณิตศาสตร์ (Activation Function) โดยทุกวันนี้มีการะประยุกต์ใช้อย่างแพร่หลาย แบ่งชนิดโครงข่ายประสาทออกเป็นดังนี้

Chart, scatter chart

Description automatically generated

**รูปที่ 2.6** ความแตกต่างระหว่าง Machine Learning และ Deep Learning

* + - 1. **โครงข่ายประสาทแบบป้อนไปข้างหน้า (Forward Propagation)**

Feed-forward neural networks ถือเป็นโมเดลที่มีโครงสร้างที่เรียบง่ายที่สุด เพราะว่า การดำเนินการของข้อมูลจะเป็นไปในทิศทางเดียว ก็คือ รับข้อมูลจาก input layer แล้วส่งไปต่อไปยัง hidden layer เลื่อยๆ จนกระทั้งถึง output layer ก็จะหยุด สังเกตุได้ว่าจะไม่มีวงวน หรือ loop เกิดขึ้นเลย

* + - 1. **โครงข่ายแบบวนซ้ำ (Recurrent neural networks : RNN)**

Recurrent neural networks คือ neural networks หลายเลเยอร์ที่สามารถเก็บข้อมูล informationไว้ที่ node จึงทำให้มันสามารถรับข้อมูลเป็นเเบบลำดับ (data sequences) และให้ผลลัพธ์ออกเป็นลำดับของข้อมูลได้ อธิบายอย่างง่ายๆ RNN ก็คือ neural network เชื่อมต่อกันหลายๆอันเเละยังสามารถต่อกันเป็นวงวนหรือ loop ได้นั่นเอง เพราะฉะนั้น RNN จึงเหมาะสมในการประมวลผลข้อมูลที่เป็นลำดับอย่างมาก

* 1. **ทบทวนวรรณกรรม**
     1. **การนำเอาความสามารถของ GPU มาใช้ในการคำนวณ**

การที่เราเลือกใช้ GPU ในการทำ Machine Learning เนื่องจากตัว GPU นั้นมีหน่วยความจำที่ให้ค่าแบนด์วิดธ์ที่สูง และตัว GPU เองยังออกแบบให้สามารถแก้สมการทางคณิตศาสตร์ได้อย่างรวดเร็ว นอกจากนี้ยังมีจำนวนหน่วยประมวลผลที่มีมากกว่า CPU หลายเท่าตัว จึงทำให้มีอัตราการประมวลผลที่สูงกว่า CPU และยังมีแพลตฟอร์มของ Nvidia ที่รองรับอย่าง CUDA ซึ่งเป็น Parallel Computing แพลตฟอร์มเพื่อช่วยให้นักพัฒนาสร้าง Tools ในการเรียกใช้การประมวลผลของ GPU และยังมี library อย่าง NVDIA cuDNN ซึ่งรองรับการทำ Deep Neural Network โดยตัว cuDNN ได้อำนวยการปรับแต่งขั้นสูงสำหรับการทำงานของ DNN เช่น forward และ backward convolution pooling normalization activation layers เป็นต้น

* + 1. **ทฤษฏี Rule of Thumb ในการหาจำนวนของ Hidden Layer**

การตัดสินใจเลือกจำนวน Neurons ใน Hidden Layers ถือเป็นส่วนสำคัญในการตัดสินภาพรวมของสถาปัตยกรรมโครงข่ายประสาทเทียม โดย Hidden Layers นั้นจะไม่ค่อยมีผลกับองค์ประกอบภายนอกแต่จะมีผลอย่างมากกับผลลัทธ์ที่จะออกมา จึงทำให้การกำหนดจำนวน Hidden Layers และ จำนวน Neurons ต้องพิจาราณาอย่างระมัดระวัง หากเราใช้จำนวน Neurons น้อยเกินไปก็จะเกิดปัญหาที่เรียกว่า Underfitting โดยจะเกิดขึ้นเมื่อมีจำนวน Neurons ใน Hidden Layers น้อยเกินไปจนไม่สามารถตรวจจับสัญญาณในข้อมูลที่ซับซ้อนได้อย่างเพียงพอ แต่ในทางกลับกันหากเราใช้จำนวน Neurons มากเกินก็จะเกิดปัญหา Ovefitting โดยจะเกิดขึ้นเมื่อความจุของข้อมูลที่จะประมวลผลมีมากเกินไป ซึ่งจะไปจำกัดข้อมูลที่จะอยู่ในชุดฝึกสอนทำให้ไม่เพียงพอต่อการเรียนรู้ของ Neurons ดังนั้นทำให้ต้องการกำหนดจำนวน Neurons ที่ไม่น้อยเกินไปหรือมากเกินไป โดยมีหลักการอย่างง่ายในการกำหนดจำนวน Neurons ตามนี้

* จำนวน Neurons ควรอยู่ในช่วงขนาดของ Input Layer และ Output Layer
* จำนวน Neurons ควรมีขนาดเป็น 2 : 3 ของขนาด Input layer รวมกับ Output layer
* จำนวน Neurons ควรมีขนาดน้อยกว่า 2 เท่าของขนาด Input Layer

โดยกฎทั้งสามที่ยกมานั้นเป็นเพียงส่วนหนึ่งในตัวเลือกให้สามารถนำไปใช้เพื่อให้ไม่ต้องมาสุ่มจำนวน Neurons ใหม่ซึ่งเท่าทำให้ไม่เสียเวลาที่ต้องนำไปทดลองกับจำนวน Neurons ที่สุ่มขึ้นใหม่

**บทที่ 3**

**วิธีการดำเนินการวิจัย**

การดำเนินการวิจัยการสร้างชุดข้อมูลในการฝึกสอนไฟร์วอลล์ปัญญาประดิษฐ์ด้วยเทคโนโลยีโครงข่ายประสาทเทียมจากกฏของไฟร์วอลล์ มีเป้าหมายเพื่อพัฒนาชุดข้อมูลฝึกสอนที่สร้างจากกฎของไฟร์วอลล์ เพื่อให้ชุดข้อมูลฝึกสอนสามารถสอนโมเดลได้ถูกต้องและแม่นยำอย่างมีประสิทธิภาพ

**3.1. 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 อีกทั้งสามารถประมวลผลได้ทั้ง 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

**3.2. การกำหนดเครื่องมือและสภาพแวดล้อมที่ใช้ในการทดลองวิจัย**

3.2.1 ประสิทธิภาพของเครื่องคอมพิวเตอร์ที่ใช้ในงานวิจัย

ผลลัพธ์ที่ได้จากการทดลองมีเวลามาเกี่ยวข้องด้วย ดังนั้นประสิทธิภาพในการทดลองแต่ละครั้งจะจำเป็นต้องใช้เครื่องคอมพิวเตอร์เดียวกันในการประมวลผล

* Computer Specification (Hardware)

OS: Windows 10 Enterprise x64 bit operating system  
 CPU: Intel(R) Core(TM) i7-3770K CPU @ 3.50GHz   
 RAM: DDR3(1600) 16GB (8GB x 2)  
 Mainboard: Gigabyte H61M-DS2  
 VGA: Gigabyte Geforce GTX1060 6GB

3.2.2 โปรแกรมที่ต้องพัฒนาขึ้นเองเพื่อใช้ในงานวิจัย

* Packet Generator

โปรแกรมสำหรับสร้างชุดข้อมูลฝึกสอนและชุดข้อมูลทดสอบภายใต้เงื่อนไขที่กำหนด

* Deep Learning Model Engine

โปรแกรมสำหรับฝึกสอนและสร้างโมเดล DNN จากข้อมูลที่กำหนดไว้

* Evaluate / Comparing Program

โปรแกรมสำหรับสรุปผลประสิทธิภาพการทำงานและความแม่นยำของโมเดล

**3.3. วัฎจักรการพัฒนางานวิจัยในการสร้างชุดข้อมูลฝึกสอน**

ในการวิจัยจะมุ่งเน้นไปที่การพัฒนาชุดข้อมูลฝึกสอนที่ทำให้โมเดลสามารถประมวลผลและคาดเดาผลลัพธ์ได้อย่างมีประสิทธิภาพ เพื่อให้การทดลองสามารถชี้ประเด็นปัจจัยต่างๆที่ส่งผลให้ความแม่นยำเปลี่ยนแปลงได้ จึงต้องมีการเปรียบเทียบผลลัพธ์ที่มาจากการสร้างชุดข้อมูลฝึกสอนด้วยค่า Parameter ที่แตกต่างกัน ทดลองหลายครั้งในหลายแง่มุมเพื่อให้สามารถวิเคราะห์และเปรียบเทีบผลลัพธ์หาข้อสรุปได้ ซึ่งการทดลองในแต่ละสมมติฐานจะมีการดำเนินงานที่คล้ายคลึงกัน ดังนี้

Diagram

Description automatically generated

**รูปที่ 3.1** Block diagram วัฎจักรการพัฒนาสร้างชุดข้อมูลฝึกสอน

จากรูปภาพ Block Diagram ข้างต้น สามารถแบ่งกระบวนการทำงานออกเป็นขั้นตอนได้ 6 ขั้นตอน ดังนี้

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

**3.3.1. ขั้นตอนที่ 1 การกำหนดขอบเขตของ Data Field ที่จะพิจารณา และการกำหนดกฎไฟร์วอลล์**

**Diagram

Description automatically generated****รูปที่ 3.2** Block Diagram การกำหนดขอบเขตของข้อมูลทั้งหมดที่จะศึกษา

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 | ขนาดใน Packet Header (Bit) | ความเป็นไปได้ (N Possible) |
| Source Address | 32 | 232 |
| Source Mask | 32 | 32 |
| Destination Address | 32 | 232 |
| Destination Mask | 32 | 32 |
| Port | 16 | 216 |
| Protocol | 8 | 28 |

**ตารางที่ 3.1** ผลลัพธ์ความเป็นไปได้ที่เกิดขึ้นทั้งหมดจาก Data Field ที่กำหนด

Data Field ที่จะใช้พิจารณาแจกแจง Sample Space ของ Possible Packet

* Source Address (32 bits)

ความเป็นไปได้ทั้งหมดจะขึ้นอยู่กับ Mask ของ Source Address

* Source Mask (32 bits)
* Destination Address (32 bits)

ความเป็นไปได้ทั้งหมดจะขึ้นอยู่กับ Mask ของ Destination Address

* Destination Mask (32 bits)
* Port (16 bits)

ความเป็นได้ขึ้นอยู่กับจำนวน port ใน pull ที่กำหนดไว้

* Protocol (8 bits)

ประกอบไปด้วย TCP และ UDP

เมื่อนำมาลองวิเคราะห์หา Packet Possible แม้จะมี Data Field เพียงแค่ 6 Field ก็ยังมีจำนวนมากเกินไปที่จะสามารถคำนวณได้ หมายความว่า Sample Space ของชุดข้อมูลจะเท่ากับ

232 × 32 × 232 × 32 × 216 × 28 = 5.7089907708 × 1045

ตัวแปรที่สำคัญคือจำนวน Source Address, Destination Address และจำนวน Port ที่มีมากเกินไป ซึ่งเมื่อลองลดจำนวนลงแล้วค่าจะเปลี่ยนไปอย่างมาก

* IP อยู่ในวง Subnet Mask /16, มีปลายทางเดียว, จำกัด 4 Ports, จำกัด 2 Protocols

216 × 16 × 1 × 1 × 4 × 2 = 8,388,608

จะเห็นได้ว่าจำนวนของ Possible Packet ของ Default เริ่มสามารถคำนวณได้ เห็นภาพรวมของข้อมูลได้ง่ายขึ้นเนื่องจากลดค่าความคลาดเคลื่อนของชุดข้อมูล Packet ลง

**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****รูปที่ 3.3** Block Diagram การสร้างชุดข้อมูลฝึกสอนสำหรับโมเดล

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

Table

Description automatically generated**รูปที่ 3.4** ตัวอย่างชุดข้อมูล Data set ที่ถูกสร้างขึ้นเมื่อแสดงผลออกมาเป็น Plain text

A picture containing outdoor, text, standing, looking

Description automatically generated**รูปที่ 3.5** ตัวอย่างชุดข้อมูล Data set ที่ถูกสร้างขึ้นเมื่อแสดงผลออกมาเป็น Binary set

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

นอกจากกฎไฟร์วอลล์ที่กำหนดขึ้นทั่วไป ยังมีกฎของ Default Rule ซึ่งจำเป็นต้องพิจารณาแยกเป็นกรณีพิเศษ เนื่องจากจำนวนความเป็นไปได้ของข้อมูลของกฏไฟร์วอลล์ที่มีการกำหนดมีขนาดที่ต่างกับ Default Rule มาก จึงทำให้การทดสอบต้องแบ่งออกเป็น 2 แบบ ได้แก่ With Default Rule และ Without Default Rule ซึ่งเราได้ตั้ง Default Rule เป็น Deny any หรือ Deny ทุกข้อมูลที่นอกเหนือจากไฟร์วอลล์ที่เรากำหนดไว้

**3.3.2.3. การออกแบบชุดข้อมูลฝึกสอนในสมมติฐาน**

ประกอบไปด้วย 2 ชุดข้อมูลฝึกสอนใน 1 เซต โดยประกอบไปด้วยชุดข้อมูลฝึกสอนที่มีประเด็นการนำ Default Rule มาใช้ และชุดข้อมูลฝึกสอนที่ไม่มีการนำ Default Rule มาใช้ในการสร้าง โดยมีตัวแปรสำคัญในการสร้างชุดข้อมูลฝึกสอนเพื่ออิงตามประเด็นศึกษาในสมมติฐาน ดังนี้

* จำนวนและเงื่อนไขของแต่ละกฏไฟร์วอลล์ที่ใช้ภายใน Rule set
* จำนวนของ Packet ของแต่ละกฎไฟร์วอลล์ที่จะนำเข้าระบบ
* การนำประเด็น Default Rule มาใช้ด้วย ประกอบด้วย With Default และ Without Default

**3.3.2.4. กลไกการแบ่งชุดข้อมูลฝึกสอน**

เป็นวิธีการในการออกแบบการแบ่งชุดข้อมูลฝึกสอนที่ใช้ในงานวิจัยนี้โดยเฉพาะ มีเป้าหมายเพื่อพิสูจน์ว่าอัลกอริทึ่มที่สร้างขึ้นจากสมมติฐานแบบใดจะสามารถให้ประสิทธิภาพในการฝึกสอนได้ดีกว่า โดยอัลกอริทึ่มที่จะนำมาใช้พิจารณา ประกอบไปด้วยดังนี้

* **การแบ่งชุดข้อมูลฝึกสอนแบบ N Sample**

การแบ่งชุดข้อมูลแบบ N Sample หรือการแบ่งชุดข้อมูลฝึกสอนให้มีจำนวนเท่ากันทั้งหมดในแต่ละกฎไฟร์วอลล์แม้ความเป็นไปได้ของชุดข้อมูลฝึกสอนในแต่ละกฎไฟร์วอลล์จะมีขนาดไม่เท่ากันก็ตาม อัลกอริทึ่มนี้สามารถสร้างขึ้นโดยเริ่มจากกฎละ 1 ข้อมูลฝึกสอนได้ แต่เพื่อให้เห็นผลกราฟในระยะยาวที่มีจำนวนชุดข้อมูลฝึกสอนมากๆและลดเวลาที่ใช้ในการทดสอบ ทำให้เราเลือกใช้วิธีการเพิ่มชุดข้อมูลแบบก้าวกระโดด

โดยจำนวนชุดข้อมูลฝึกสอนที่เราได้ทำการเลือก ได้แก่ 10, 100, 300, 600, 1,000, 3,000, 6,000, 10,000 โดยมีหน่วยเป็น Sample per rule หรือ จำนวนชุดข้อมูลฝึกสอนต่อ 1 กฎไฟร์วอลล์

* **การแบ่งชุดข้อมูลฝึกสอนแบบอัตราส่วน Ratio**

การแบ่งชุดข้อมูลฝึกสอนแบบอัตราส่วน Ratio เป็นสมมติฐานที่ตั้งขึ้นในงานวิจัยจากการคาดเดาว่ากฎไฟร์วอลล์ที่มีจำนวนข้อมูลเข้าเงื่อนไขสูงมากๆ โมเดลจำเป็นต้องมีการเรียนรู้จากข้อมูลฝึกสอนจากกฎดังกล่าวที่มากกว่า เนื่องจากข้อมูลมีขนาดกว้างทำให้ต้องใช้ข้อมูลฝึกสอนมากขึ้น โดยเราได้ใช้วิธีการแบ่งให้แต่ละกฎไฟร์วอลล์ได้รับจำนวนชุดข้อมูลฝึกสอนแบบอัตราส่วน หมายความว่ากฎที่มีจำนวนข้อมูลเข้าเงื่อนไขมากจะได้จำนวนโควต้าในชุดข้อมูลฝึกสอนมาก กฎไฟร์วอลล์ที่มีจำนวนน้อยกว่าจะได้รับจำนวนชุดข้อมูลฝึกสอนที่น้อยกว่า ซึ่งทั้งหมดจะต้องได้รับตามอัตราส่วนจากความเป็นไปได้ทั้งหมดภายในกฎไฟร์วอลล์นั้นเท่าๆกัน

โดยตัวแปรที่เราได้ทำการลองทดสอบสร้าง ประกอบไปด้วย 0.01, 0.03, 0.05, 0.07, 0.09, 0.11, 0.13, 0.15 มีหน่วยเป็นอัตราส่วนจำนวนชุดข้อมูลฝึกสอนที่นำเข้าโมเดล ต่อจำนวนความเป็นไปได้ทั้งหมดของข้อมูลฝึกสอนที่เข้าเงื่อนไข

**ตัวอย่างการออกแบบชุดข้อมูลฝึกสอน 2 กฎไฟร์วอลล์**

|  |  |  |
| --- | --- | --- |
| Data Field | ตัวแปรที่ใช้ | จำนวนความเป็นไปได้ทั้งหมด |
| Source Address | Subnet 192.168.0.0/16 | 65,534 |
| Source Mask | ขึ้นอยู่กับ Source Address | 1 |
| Destination Address | 161.246.34.11 | 1 |
| Destination Mask | /32 | 1 |
| Port | 22, 80 | 2 |
| Protocol | TCP, UDP | 2 |
| ทุก Data Field | ทุกตัวแปรที่ใช้ | 262,016 |

**ตารางที่ 3.3** ตัวอย่างการออกแบบ Default pool ที่พิจารณา

|  |  |
| --- | --- |
| ชุดเงื่อนไขทั้งหมดที่สร้างขึ้นจากกฎไฟร์วอลล์ | จำนวนข้อมูลตรงตามเงื่อนไข  (Packet Possible) |
| Rule set   1. allow 192.168.0.0/16 to 161.246.34.11/24 port 80 tcp 2. deny 192.168.128.0/18 to 161.246.34.11/24 port 22 udp | 65,534 + 16,382 = 81,916 |

**ตารางที่ 3.4** ตัวอย่างกฎไฟร์วอลล์ที่ทำการออกแบบ

จำนวนข้อมูลที่จะเข้าเงื่อนไข Default Rule = 262,016 – 81,916 = 181,000 รูปแบบ

**ตัวอย่างการแบ่งจำนวนข้อมูลฝึกสอน 2 กฎไฟร์วอลล์ (N Sample)**

|  |  |  |  |
| --- | --- | --- | --- |
| Sample per rule (N) | Rule 1 (N)  65,534 | Rule 2 (N)  16,382 | Total Packet (N)  262,016 |
| 10 | 10 | 10 | 20 |
| 100 | 100 | 100 | 200 |
| 300 | 300 | 300 | 600 |
| 600 | 600 | 600 | 1,200 |
| 1,000 | 1,000 | 1,000 | 2,000 |
| 3,000 | 3,000 | 3,000 | 6,000 |
| 6,000 | 6,000 | 6,000 | 12,000 |
| 10,000 | 10,000 | 10,000 | 20,000 |

* N Sample without Default Rule

**ตารางที่ 3.5** ตัวอย่างการแบ่งจำนวนชุดฝึกสอนแบบ N Sample without Default

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sample per rule (N) | Rule 1 (N)  65,534 | Rule 2 (N)  16,382 | Default Rule (N)  181,000 | Total Packet (N)  262,016 |
| 10 | 10 | 10 | 10 | 30 |
| 100 | 100 | 100 | 100 | 300 |
| 300 | 300 | 300 | 300 | 900 |
| 600 | 600 | 600 | 600 | 1,800 |
| 1,000 | 1,000 | 1,000 | 1,000 | 3,000 |
| 3,000 | 3,000 | 3,000 | 3,000 | 9,000 |
| 6,000 | 6,000 | 6,000 | 6,000 | 18,000 |
| 10,000 | 10,000 | 10,000 | 10,000 | 30,000 |

* N Sample with Default Rule

**ตารางที่ 3.6** ตัวอย่างการแบ่งจำนวนชุดฝึกสอนแบบ N Sample with Default

**ตัวอย่างการแบ่งจำนวนข้อมูลฝึกสอน 2 กฎไฟร์วอลล์ (Ratio)**

* Ratio without Default Rule

|  |  |  |  |
| --- | --- | --- | --- |
| Ratio per rule (Ratio) | Rule 1 (N)  65,534 | Rule 2 (N)  16,382 | Total Packet (N)  262,016 |
| 0.01 | 655 | 164 | 819 |
| 0.03 | 1,965 | 492 | 2,457 |
| 0.05 | 3,276 | 819 | 4,095 |
| 0.07 | 4,586 | 1,147 | 5,733 |
| 0.09 | 5,897 | 1,475 | 7,372 |
| 0.11 | 7,208 | 1,802 | 9,010 |
| 0.13 | 8,518 | 2,130 | 10,648 |
| 0.15 | 9,829 | 2,458 | 12,287 |

**ตารางที่ 3.7** ตัวอย่างการแบ่งจำนวนชุดฝึกสอนแบบ Ratio without Default

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ratio per rule (Ratio) | Rule 1 (N)  65,534 | Rule 2 (N)  16,382 | Default Rule (N)  181,000 | Total Packet (N)  262,016 |
| 0.01 | 655 | 164 | 1,810 | 2,629 |
| 0.03 | 1,965 | 489 | 5,430 | 7,884 |
| 0.05 | 3,275 | 815 | 9,050 | 13,140 |
| 0.07 | 4,585 | 1,141 | 12,670 | 18,396 |
| 0.09 | 5,895 | 1,467 | 16,290 | 23,652 |
| 0.11 | 7,205 | 1,793 | 19,910 | 28,908 |
| 0.13 | 8,515 | 2,119 | 23,530 | 34,164 |
| 0.15 | 9,825 | 2,445 | 27,150 | 39,420 |

* Ratio with Default Rule

**ตารางที่ 3.8** ตัวอย่างการแบ่งจำนวนชุดฝึกสอนแบบ Ratio with Default

**3.3.2.5. อัลกอริทึ่มในการสร้างชุดข้อมูลฝึกสอน**

เป็นกลไกในการจัดการรวมชุดข้อมูลฝึกสอนทั้งหมดก่อนเข้าไปฝึกในโมเดล ซึ่งจะประกอบไปด้วยกฎไฟร์วอลล์ที่กำหนด หรืออาจมีกฎจาก Default Rule ด้วย ซึ่งในโปรแกรม Packet Generator มีกลไกการทำงาน ดังนี้

1. สร้าง List ที่ประกอบไปด้วยจำนวนข้อมูลทั้งหมดที่สามารถเกิดขึ้นได้ใน pool
2. สร้าง List ที่ประกอบไปด้วยจำนวนข้อมูลทั้งหมดที่ตรงเงื่อนไขของแต่ละ rule set
3. สุ่มหยิบข้อมูลฝึกสอนจาก List rule set ให้มีจำนวนขนาดตามที่ต้องการเก็บไว้ใน list rule set quota
4. ถ้าหากต้องการ Default ให้ทำการสุ่มหยิบจาก pool เลยและเปรียบเทียบกับ list rule set ถ้าหากมีตรงกันให้หยิบใหม่ แต่ถ้าหากไม่ตรงกันให้เลือกเข้ามาให้ได้จำนวนตามที่ต้องการไว้
5. นำ quota มารวมกันเป็น list เดียวและบันทึกลงไฟล์ CSV

**ตัวอย่างการสร้างชุดข้อมูลฝึกสอนแบบมี Default, 2 rules**

* M = จำนวนข้อมูลที่ต้องการจากกฎไฟร์วอลล์ที่ 1
* N = จำนวนข้อมูลที่ต้องการจากกฎไฟร์วอลล์ที่ 2
* T = จำนวนข้อมูล Default ที่ต้องการ และไม่ตรงกับกฎใดๆเลย

Diagram

Description automatically generated

**รูปที่ 3.6** Block Diagram ขั้นตอนการสร้างชุดข้อมูลฝึกสอนแบบมี Default

**3.3.3. ขั้นตอนที่ 3 การนำโมเดลไปผ่านการเรียนรู้ด้วยชุดข้อมูลสำหรับฝึกสอน**

**Diagram

Description automatically generated**

**รูปที่ 3.7** Block Diagram ขั้นตอนการนำโมเดลไปฝึกฝนด้วยชุดข้อมูลฝึกสอน

เป็นขั้นตอนการนำชุดข้อมูลฝึกสอนที่สร้างขึ้นไปประมวลผลผ่านโมเดลให้เกิดการเรียนรู้ โดยขั้นตอนการฝึกโมเดลจะต้องมีการกำหนดค่าพารามิเตอร์และปรับปรุงแก้ไขการประมวลผลหาคำตอบที่ขึ้นอยู่กับขอบเขตของงานหรือข้อมูลที่จะพิจารณา ซึ่งในส่วนนี้เราสามารถหาหลักการได้ จากคำแนะนำของผู้พัฒนาโมเดล หรืองานวิจัยที่มีการใช้งานใกล้เคียงกัน โดยมีจุดประสงค์เพื่อ พัฒนาให้โมเดลสามารถเรียนรู้ผ่านชุดข้อมูลฝึกสอนได้อย่างมีประสิทธิภาพขึ้นได้

เราได้ตัดสินใจเลือกโมเดลที่มีการเรียนรู้แบบ Sequential Logistic Regression มีฟังก์ชั่นการประมวลผลแบบ Sigmoid สมการถดถอยที่มีการเรียนรู้ในเชิงคุณภาพหรือเชิงกลุ่ม โดยที่ตัวแปรที่ออกมามีอยู่ 2 ค่า คือมีค่าเป็น 0 กับ 1 ทำให้รูปแบบการเรียนรู้นี้เหมาะกับการแก้ปัญหาตามโจทย์ Binary Classification Problem ที่คำตอบจะถูกตัดสินใจแบบ Two-Class-Label แบ่งออกเป็น 2 ตัวเลือก ได้แก่ Allow หรือ Deny ตามที่เรากำหนดไว้ตั้งแต่แรกภายในการทดสอบ

**ข้อมูลการตั้งค่าที่สำคัญภายในโมเดล**

1. รูปแบบการเรียนรู้: Sequential Logistic Regression
2. ฟังก์ชั่นการประมวลผล: Sigmoid A picture containing object, clock

   Description automatically generated
3. เครื่องมือเสริมประสิทธิภาพในการประมวลผล: Adam Optimizer

กระบวนการทำงานในขั้นตอนนี้ จะเป็นการแยกส่วนข้อมูลที่จะใช้พิจารณาแยกกันในไฟล์นามสกุล CSV ที่สร้างจากขั้นตอนที่แล้ว โดยแบ่งออกเป็นชุดข้อมูลฝึกสอนและชุดข้อมูลทดสอบสำหรับการสรุปผลการเรียนรู้ในอัตราส่วนที่ได้จาก Rule of Thumb คือ 8:2 และแบ่งชุดข้อมูลดังกล่าวออกอีก ได้แก่

1. ชุดข้อมูลฝึกสอน ที่ประกอบไปด้วย Data Field ภายใน Packet ทั้งหมด
2. ชุดข้อมูลฝึกสอน ที่ประกอบไปด้วย Decision ที่เป็นผลลัพธ์ตัดสินใจว่าจะปล่อยผ่าน
3. ชุดข้อมูลทดสอบ ที่ประกอบไปด้วย Data Field ภายใน Packet ทั้งหมด
4. ชุดข้อมูลทดสอบ ที่ประกอบไปด้วย Decision ที่เป็นผลลัพธ์ตัดสินใจว่าจะปล่อยผ่าน

นำข้อมูลข้างต้นมาทำ Data Preprocessing หรือการจัดข้อมูลชุดให้อยู่ในรูป Matrix เปลี่ยนค่าภายในในกลายเป็นค่าถ่วงน้ำหนัก เป็นค่าที่โมเดลจะนำไปเรียนรู้ต่อและหาค่าความสัมพันธ์ว่าชุดข้อมูลดังกล่าวจะถูกตัดสินว่าเป็น Allow หรือ Deny โดยชุดข้อมูลที่จะต้องนำไปทำ Data Preprocessing ได้แก่

* ชุดข้อมูลฝึกสอน ที่ประกอบไปด้วย Data Field ภายใน Packet ทั้งหมด
* ชุดข้อมูลทดสอบ ที่ประกอบไปด้วย Data Field ภายใน Packet ทั้งหมด

**การออกแบบ MLP Architecture ในงานวิจัย**

โครงสร้างของชุดข้อมูลฝึกสอนมีผลอย่างมากในการเลือกโมเดลที่จะนำมาใช้ เนื่องจากข้อมูล Packet ของเราทั้งหมดจะอยู่ในรูปแบบเลขฐานสอง ทำให้มีหน่วยตั้งเป็นค่า Bit ซึ่งเมื่อถ้าหากนำไปอ้างอิงกับบทประพันธ์ที่ผ่านมาข้างต้น จะได้จำนวน Neuron กับจำนวน Hidden Layer ที่ต้องการได้

* **Input**: Source Address + Mask + Destination Address + Mask + Port + Protocol   
  = 32+32+32+32+16+8 = 152 Neurons
* **Output:** 2 Neurons (Allow, Deny)
* **Hidden Layer:** 3 Layers

กระบวนการเรียนรู้ในขั้นตอนนี้จะหยุดลงเมื่อข้อผิดพลาดในชุดการตรวจสอบความถูกต้องคงที่ {เมื่อค่าความคลาดเคลื่อนระหว่างข้อผิดพลาดก่อนหน้าและปัจจุบันหารด้วยข้อผิดพลาดปัจจุบันต่ำกว่าค่าคงที่เล็กน้อย ในกรณีของเราค่าคงที่นี้ถูกตั้งค่าเป็น 0.1%

**3.3.4. ขั้นตอนที่ 4 การสร้างชุดข้อมูลสำหรับการทดสอบโมเดล**

**Diagram

Description automatically generated**

**รูปที่ 3.8** Block Diagram การสร้างชุดข้อมูลทดสอบโมเดล

**หลักการออกแบบชุดข้อมูลทดสอบ**

ในการสร้างชุดข้อมูลทดสอบที่สามารถวัดผลความแม่นยำของโมเดลจากการทดลองได้ ในการออกแบบนั้นถือว่ามีความท้าทายในระดับหนึ่ง เพราะมีประเด็นสำคัญที่จำเป็นต้องพิจารณาดังต่อไปนี้

* จะทราบได้อย่างไรว่า โมเดลสามารถทำนายผลลัพธ์ได้ดีในทุกกฎไฟร์วอลล์
* จะทราบได้อย่างไรว่า โมเดลติดปัญหา Underfitting หรือ Overfitting

เราได้ทำการสร้างชุดข้อมูลทดสอบ **แบ่งจำนวนชุดข้อมูลออกเป็นจำนวนที่เท่าๆกัน ในแต่ละเงื่อนไขกฎของไฟร์วอลล์** เพื่อให้สามารถทราบได้ว่าภาพรวมที่โมเดลทำนายผลมานั้นให้ความถูกต้องแม่นยำเป็นอย่างไร ซึ่งถ้าหากไฟร์วอลล์นั้นสามารถทำนายผลได้เพียงบางเงื่อนไข ความแม่นยำที่ได้จากชุดข้อมูลทดสอบเดียวกันแต่โมเดลต่างกันจะต้องเห็นผลลัพธ์ที่สามารถสังเกตได้อย่างแน่นอน

ในความเป็นจริงแล้ว เพื่อให้มีการทดสอบและวิเคราะห์ได้ดียิ่งขึ้น อาจต้องสร้างชุดข้อมูลทดสอบหลายๆแบบที่มีความแตกต่างกัน เพื่อให้สามารถจับประเด็นสำคัญหรือปัญหาที่เกิดขึ้นจากโมเดลได้ เช่น การทดสอบว่าโมเดลมีปัญหา Overfitting หรือมีวิธีการตรวจสอบที่ดีหรือไม่

**3.3.5. ขั้นตอนที่ 5 การนำโมเดลไปประมวลผล ทำนายผลลัพธ์จากชุดข้อมูลสำหรับทดสอบ**

**Diagram

Description automatically generated****รูปที่ 3.9** Block Diagram การนำโมเดลไปประมวลผลหรือ Evaluate

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  
**รูปที่ 3.10** Reference Set ในการวิเคราะห์ความถูกต้องของโมเดล

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

ซึ่งผลลัพธ์ที่ได้จะเป็นไปตามสูตร

ความแม่นยำ (Accuracy) = SUM(TP, TN) / SUM(TP, TN, FP, FN)

**3.3.6. ขั้นตอนที่ 6 บันทึกผลลัพธ์จากการทดสอบโมเดล**

**Diagram

Description automatically generated****รูปที่ 3.11** Block Diagram ขั้นตอนการนำผลลัพธ์มาบันทึกผล

การหาวิธีการที่สามารถทำให้ชุดข้อมูลฝึกสอนสามารถสอนโมเดลได้อย่างมีประสิทธิภาพ เราจำเป็นต้องนำผลลัพธ์ของการทดสอบในแต่ละครั้งของการทดลองมาบันทึกผล แล้วสรุปให้อยู่ในรูปกราฟเปรียบเทียบที่ประกอบไปด้วยผลลัพธ์จากการทดลองภายใต้สภาพแวดล้อมเดียวกัน เพื่อหาว่าผลลัพธ์ออกมาตรงตามสมมติฐานหรือมีความสัมพันธ์กันในแต่ละตัวแปรอย่างไรบ้าง

**A close up of a building

Description automatically generated****รูปที่ 3.12** ตัวอย่างของตารางที่จะนำมาบันทึกผลลัพธ์การทดลอง

**บทที่ 4**

**ผลการดำเนินงานวิจัย**

การทดลองจะเป็นไปตามวัฎจักรการดำเนินงานวิจัยข้างต้น โดยชุดข้อมูลฝึกสอนที่ทำการพัฒนาขึ้นมีรูปแบบโครงสร้างจำลองมาจาก Packet Header และสร้างขึ้นผ่านโปรแกรม Packet Generator ที่ออกแบบขึ้นเอง ชุดข้อมูลฝึกสอนและชุดข้อมูลทดสอบจะมีการออกแบบให้มีความแตกต่างกันตามสมมติฐานที่กำหนด สังเกตกระบวนการทำงานของโมเดล และรูปแบบความสัมพันธ์ของตัวแปรที่ได้หลังโมเดลทำการเรียนรู้และประมวลผล และทำการสรุปผลลัพธ์ที่ได้หลังเสร็จสิ้นการทดลอง

**การออกแบบเงื่อนไขของชุดข้อมูล**

ค่าความเป็นไปได้ทั้งหมดที่สามารถเกิดขึ้นได้ในแต่ละส่วนของข้อมูลที่ออกแบบ มีดังนี้

|  |  |  |
| --- | --- | --- |
| Data Field | ตัวแปรที่ใช้ | จำนวนความเป็นไปได้ทั้งหมด |
| Source Address | Subnet 192.168.0.0/16 | 65534 |
| Source Mask | ขึ้นอยู่กับ Source Address | 1 |
| Destination Address | 161.246.34.11 | 1 |
| Destination Mask | /32 | 1 |
| Port | 22, 80 | 2 |
| Protocol | TCP, UDP | 2 |

**ตารางที่ 4.1** ตารางการจำแนกความเป็นไปได้ของแต่ละ Data Field

ดังนั้น ข้อมูล Packet ที่เกิดขึ้นได้ทั้งหมด จะเท่ากับ 65534 \* 1 \* 1 \* 1 \* 2 \* 2 = 262,016

**การจำแนกชุดกฎไฟร์วอลล์ที่จะทำการทดสอบ**

|  |  |
| --- | --- |
| ชุดเงื่อนไขทั้งหมดที่สร้างขึ้นจากกฎไฟร์วอลล์ | จำนวนข้อมูลที่สามารถเกิดขึ้นตรงตามเงื่อนไขของไฟร์วอลล์ที่กำหนด |
| Rule set ที่ 1   1. allow 192.168.0.0/16 to 161.246.34.11/24 port 80 tcp 2. deny 192.168.128.0/18 to 161.246.34.11/24 port 22 udp | 65,534 + 16,382 = 81,916 |
| Rule set ที่ 2   * allow 192.168.0.0/16 to 161.246.34.11/24 port 80 tcp * deny 192.168.128.0/18 to 161.246.34.11/24 port 22 udp * allow 192.168.64.0/24 to 161.246.34.11/24 port 22 tcp * deny 192.168.64.0/24 to 161.246.34.11/24 port 80 udp | 65,534 + 16,382 + 254 + 254 = 82,424 |
| Rule set ที่ 3   * allow 192.168.0.0/16 to 161.246.34.11/24 port 80 tcp * deny 192.168.128.0/18 to 161.246.34.11/24 port 22 udp * allow 192.168.64.0/24 to 161.246.34.11/24 port 22 tcp * deny 192.168.64.0/24 to 161.246.34.11/24 port 80 udp * allow 192.168.192.0/18 to 161.246.34.11/24 port 22 udp * allow 192.168.128.0/18 to 161.246.34.11/24 port 22 tcp | 65,534 + 16,382 + 254 + 254 + 16,382 + 16,382 = 115,188 |

**ตารางที่ 4.2** ตารางการจำแนกความเป็นไปได้ของแต่ละกฎไฟร์วอลล์

**การออกแบบชุดข้อมูลทดสอบ**

ประกอบไปด้วย 2 ชุดข้อมูลเช่นเดียวกับชุดฝึกสอน โดยประกอบไปด้วยชุดข้อมูลฝึกสอนที่มีประเด็นการนำ Default Rule มาใช้ และ ไม่มีการนำ Default Rule มาใช้ในการสร้าง

**ผลลัพธ์ที่คาดว่าค่าจะต้องเปลี่ยนแปลงไปตามการทดสอบแต่ละครั้ง**

* เวลาที่โมเดลใช้ในการเรียนรู้จากชุดข้อมูลฝึกสอน หรือ Training
* เวลาที่โมเดลใช้ในการตัดสินใจจากชุดข้อมูลทดสอบ หรือ Predict
* ค่าความแม่นยำโดยรวม หรือ Accuracy
* อัตราความผิดพลาดที่อ้างอิงจาก Reference Variant Set

**4.1 สมมติฐานการทดลองที่ 1**

ในสมมติฐานการทดลองที่ 1 เป็นการทดลองใช้ชุดข้อมูลฝึกสอนและชุดข้อมูลทดสอบที่สร้างขึ้น และเพื่อเป็นการพิสูจน์ว่าโมเดลสามารถประยุกต์ใช้ในงานวิจัยได้จริง มีหลักการทำงานและผลลัพธ์ที่คล้ายคลึงกับปัญญาประดิษฐ์ที่พบได้ทั่วไป โดยวางสมมติฐานเบื้องต้นไว้ ดังนี้

* โมเดลจะสามารถเรียนรู้จากชุดข้อมูลฝึกสอนที่สร้างจากกฎของไฟร์วอลล์และสามารถทำนายผลลัพธ์ได้
* เมื่อโมเดลเรียนรู้จากชุดข้อมูลฝึกสอนที่มีจำนวนมากขึ้นในแต่ละกฎไฟร์วอลล์ โมเดลจะสามารถทำนายผลลัพธ์ได้แม่นยำมากขึ้น
* โมเดลเมื่อมีการเรียนรู้ถึงจุดๆหนึ่งจะไม่สามารถเพิ่มความแม่นยำในการทำนายผลลัพธ์ได้อีก
* โมเดลจะใช้เวลาในการทดสอบประมวลผลข้อมูลเท่าเดิม แม้จะผ่านการเรียนรู้จากข้อมูลฝึกสอนที่มีจำนวนต่างกัน

**4.1.1. หลักการออกแบบชุดข้อมูลฝึกสอน**

ชุดข้อมูลฝึกสอนในแต่ละกฎไฟร์วอลล์ จะมีจำนวน N Sample เท่ากันทั้งหมด โดยจำนวนที่ของข้อมูลฝึกสอนที่ใช้ทดสอบ ประกอบไปด้วย 10, 100, 300, 600, 1,000, 3,000, 6,000, 10,000 ในแต่ละกฎไฟร์วอลล์

**4.1.2. ผลลัพธ์ที่ได้จากการทดลอง**

4.1.2.1. ตารางผลการทดลองแบบ N Sample เงื่อนไข Rule set ที่ 1 (2 กฎไฟร์วอลล์)

Table

Description automatically generated  
**ตารางที่ 4.3** ตารางผลการทดลองแบบ N Sample เงื่อนไข Rule set ที่ 1 (2 กฎ)

4.1.2.2. ตารางผลการทดลองแบบ N Sample เงื่อนไข Rule set ที่ 2 (4 กฎไฟร์วอลล์)

Table

Description automatically generated  
**ตารางที่ 4.4** ตารางผลการทดลองแบบ N Sample เงื่อนไข Rule set ที่ 2 (4 กฎ)

4.1.2.3. ตารางผลการทดลองแบบ N Sample เงื่อนไข Rule set ที่ 3 (6 กฎไฟร์วอลล์)

Table

Description automatically generated  
**ตารางที่ 4.5** ตารางผลการทดลองแบบ N Sample เงื่อนไข Rule set ที่ 3 (6 กฎ)

**4.1.3. กราฟผลลัพธ์การทดลองแบบ N Sample**

4.1.3.1. กราฟผลลัพธ์ เวลาในการฝึกสอนโมเดล : ชุดข้อมูลฝึกสอนต่อ 1 กฎไฟร์วอลล์

**รูปที่ 4.1** กราฟเวลาในการฝึกโมเดล : ชุดข้อมูลฝึกสอนต่อ 1 กฎ (N Sample)

4.1.3.2. กราฟผลลัพธ์ เวลาในการทำนายชุดทดสอบ : จำนวนชุดข้อมูลฝึกสอนต่อ 1 กฎไฟร์วอลล์

**รูปที่ 4.2** กราฟเวลาทำนายข้อมูลทดสอบ : จำนวนชุดฝึกสอนต่อ 1 กฎ (N Sample)

4.1.3.3. กราฟผลลัพธ์ ความแม่นยำในการประมวลผล : จำนวนชุดข้อมูลฝึกสอนต่อ 1 กฎไฟร์วอลล์

**รูปที่ 4.3** กราฟความแม่นยำในการประมวลผล : จำนวนชุดฝึกสอนต่อ 1 กฎ (N Sample)

**4.2 สมมติฐานการทดลองที่ 2**

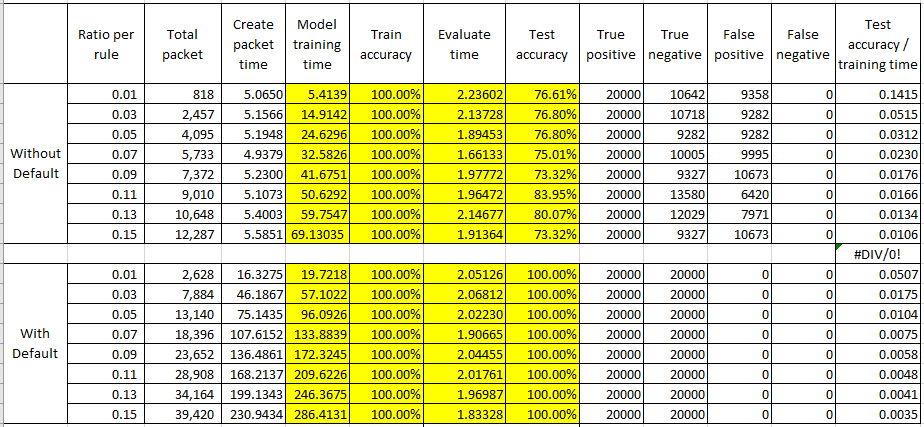
จากสมมติฐานแรกจะเห็นได้ว่า ความเป็นไปได้ของชุดข้อมูลฝึกสอนในแต่ละกฎไฟร์วอลล์มีจำนวนไม่เท่ากัน ดังนั้นในสมมติฐานนี้จึงเป็นการตั้งข้อสันนิษฐานว่า ถ้าหากตั้งเงื่อนไขให้กฎไฟร์วอลล์แต่ละกฎได้รับจำนวนชุดข้อมูลฝึกสอนไม่เท่ากัน จะส่งผลต่อความแม่นยำของโมเดลอย่างไร มีการพัฒนาโมเดลในทางที่ดีขึ้นหรือแย่ลงอย่างไร

**4.2.1. หลักการออกแบบชุดข้อมูลฝึกสอน**

สมมติฐานนี้จะเป็นการใช้อัตราส่วนมาเป็นหลักเกณฑ์ในการแบ่งจำนวนชุดข้อมูลฝึกสอนที่แต่ละกฎจะได้รับ โดยจำนวนของข้อมูลฝึกสอนที่ใช้ทดสอบ เพิ่มขึ้นด้วยอัตราส่วน Ratio ที่เท่าๆกัน โดยอัตราส่วนที่ใช้ประกอบไปด้วย 0.01, 0.03, 0.05, 0.07, 0.09, 0.11, 0.13, 0.15 โดยมีหน่วยเป็นจำนวนชุดข้อมูลฝึกสอนที่ใช้ในแต่ละกฎไฟร์วอลล์ต่อจำนวนความเป็นไปได้ทั้งหมดของชุดฝึกสอนในกฎไฟร์วอลล์นั้นๆ ยกตัวอย่างเช่น เงื่อนของกฎของไฟล์วอลล์หนึ่งมีจำนวนความเป็นไปได้คือ 16,382 ความเป็นไปได้ ชุดข้อมูลฝึกสอนที่กฎไฟร์วอลล์นั้นจะได้รับหากมีอัตราส่วน Ratio ที่ 0.01 คือ 163 แพ็คเกต

**4.2.2. ผลลัพธ์ที่ได้จากการทดลอง**

4.2.2.1. ตารางผลการทดลองแบบอัตราส่วน Ratio เงื่อนไข Rule set ที่ 1 (2 กฎไฟร์วอลล์)

  
**ตารางที่ 4.6** ตารางผลการทดลองแบบอัตราส่วน Ratio Rule set ที่ 1 (2 กฎ)

4.2.2.2. ตารางผลการทดลองแบบอัตราส่วน Ratio เงื่อนไข Rule set ที่ 2 (4 กฎไฟร์วอลล์)

Table

Description automatically generated  
**ตารางที่ 4.7** ตารางผลการทดลองแบบอัตราส่วน Ratio Rule set ที่ 2 (4 กฎ)

4.2.2.3. ตารางผลการทดลองแบบอัตราส่วน Ratio เงื่อนไข Rule set ที่ 3 (6 กฎ)

Table

Description automatically generated  
**ตารางที่ 4.8** ตารางผลการทดลองแบบอัตราส่วน Ratio Rule set ที่ 3 (6 กฎ)

**4.2.3. กราฟผลลัพธ์การทดลองแบบอัตราส่วน Ratio**

4.2.3.1. กราฟผลลัพธ์ เวลาในการฝึกสอนโมเดล : อัตราส่วนข้อมูลฝึกสอนต่อ 1 กฎไฟร์วอลล์

Chart, line chart

Description automatically generated

**รูปที่ 4.4** กราฟเวลาในการฝึกสอนโมเดล : อัตราส่วนข้อมูลฝึกสอนต่อ 1 กฎ (Ratio)

4.2.3.2. กราฟผลลัพธ์ เวลาในการฝึกสอนโมเดล : อัตราส่วนข้อมูลฝึกสอนต่อ 1 กฎไฟร์วอลล์

**รูปที่ 4.5** กราฟเวลาในการทำนายชุดทดสอบ : อัตราส่วนข้อมูลฝึกสอนต่อ 1 กฎ (Ratio)

4.2.3.3. กราฟผลลัพธ์ เวลาในการฝึกสอนโมเดล : อัตราส่วนข้อมูลฝึกสอนต่อ 1 กฎไฟร์วอลล์

Chart, line chart

Description automatically generated

**รูปที่ 4.6** กราฟเวลาในการฝึกสอนโมเดล : อัตราส่วนข้อมูลฝึกสอนต่อ 1 กฎ (Ratio)