**การศึกษาวิจัยเพื่อพัฒนาสร้างชุดข้อมูลในการฝึกสอนไฟร์วอลล์ปัญญาประดิษฐ์ด้วยเทคโนโลยีโครงข่ายประสาทเทียมจากกฏของไฟร์วอลล์**

**Researching for developing training sets   
with artificial neural network technology based on firewall rules**

**โดย**

**ฐิติโชติ ใจเมือง**

**Thitichote Chaimuang**

**พิพัฒน์บุญ พุทธคุณ**

**Pipatboon Buddhakul**

**อาจารย์ที่ปรึกษา**

**ผู้ช่วยศาสตราจารย์ อัครินทร์ คุณกิตติ**

**ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต**

**สาขาวิชาเทคโนโลยีสารสนเทศ**

**คณะเทคโนโลยีสารสนเทศ**

**สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง**

**ภาคเรียนที่ 2 ปีการศึกษา 2563**

**การศึกษาวิจัยเพื่อพัฒนาสร้างชุดข้อมูลในการฝึกสอนไฟร์วอลล์ปัญญาประดิษฐ์  
ด้วยเทคโนโลยีโครงข่ายประสาทเทียมจากกฏของไฟร์วอลล์**

**Researching for developing training sets   
with artificial neural network technology based on firewall rules**

**โดย**

**ฐิติโชติ ใจเมือง**

**พิพัฒน์บุญ พุทธคุณ**

**อาจารย์ที่ปรึกษา**

**ผู้ช่วยศาสตราจารย์ อัครินทร์ คุณกิตติ**

**ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต**

**สาขาวิชาเทคโนโลยีสารสนเทศ**

**คณะเทคโนโลยีสารสนเทศ**

**สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง**

**ภาคเรียนที่ 2 ปีการศึกษา 2562**

**RESEARCHING FOR DEVELOPING TRAINING SETS   
WITH ARTIFICIAL NEURAL NETWORK TECHNOLOGY   
BASED ON FIREWALL RULES**

**THITICHOTE CHAIMUANG**

**PIPATBOON BUDDHAKUL**

**A PROJECT SUBMITTED IN PARTIAL FULFILLMENT**

**OF THE REQUIREMENT FOR THE DEGREE OF**

**BACHELOR OF SCIENCE PROGRAM IN INFORMATION TECHNOLOGY**

**FACULTY OF INFORMATION TECHNOLOGY**

**KING MONGKUT’S INSTITUTE OF TECHNOLOGY LADKRABANG**

**2/2019**

**COPYRIGHT 2020**

**FACULTY OF INFORMATION TECHNOLOGY**

**KING MONGKUT’S INSTITUTE OF TECHNOLOGY LADKRABANG**

**ใบรับรองปริญญานิพนธ์ ประจำปีการศึกษา 2562**

**คณะเทคโนโลยีสารสนเทศ**

**สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง**

|  |  |
| --- | --- |
| **เรื่อง** | **การศึกษาวิจัยเพื่อพัฒนาสร้างชุดข้อมูลในการฝึกสอนไฟร์วอลล์ปัญญาประดิษฐ์ด้วยเทคโนโลยีโครงข่ายประสาทเทียมจากกฏของไฟร์วอลล์** |
| **RESEARCHING FOR DEVELOPING TRAINING SET WITH ARTIFICIAL NEURAL NETWORK TECHNOLOGY BASED ON FIREWALL RULES** |

**ผู้จัดทำ**

1. **นายฐิติโชติ ใจเมือง รหัสประจำตัว  60070019**
2. **นายพิพัฒน์บุญ พุทธคุณ รหัสประจำตัว  60070065**

**…………………………………………. อาจารย์ที่ปรึกษา**

**(ผู้ช่วยศาสตราจารย์ อัครินทร์ คุณกิตติ)**

**ใบรับรองโครงงาน (Project)**

**เรื่อง**

**การศึกษาวิจัยเพื่อพัฒนาสร้างชุดข้อมูลในการฝึกสอนไฟร์วอลล์ปัญญาประดิษฐ์  
ด้วยเทคโนโลยีโครงข่ายประสาทเทียมจากกฏของไฟร์วอลล์**

**Researching for developing training sets  
with artificial neural network technology based on firewall rules**

**นายฐิติโชติ ใจเมือง รหัสประจำตัว  60070019**

**นายพิพัฒน์บุญ พุทธคุณ รหัสประจำตัว  60070065**

**ขอรับรองว่ารายงานฉบับนี้ ข้าพเจ้าไม่ได้คัดลอกมาจากที่ใด**

**รายงานฉบับนี้ได้รับการตรวจสอบและอนุมัติให้เป็นส่วนหนึ่งของ**

**การศึกษาวิชาโครงงาน หลักสูตรวิทยาศาสตรบัณฑิต (เทคโนโลยีสารสนเทศ)**

**ภาคเรียนที่ 2 ปีการศึกษา 2562**

**..........................................................**

**(นายฐิติโชติ ใจเมือง)**

**.........................................................**

**(นายพิพัฒน์บุญ พุทธคุณ)**

|  |  |  |  |
| --- | --- | --- | --- |
| **หัวข้อโครงงาน** | การศึกษาวิจัยเพื่อพัฒนาสร้างชุดข้อมูลในการฝึกสอนไฟร์วอลล์ปัญญา ประดิษฐ์ด้วยเทคโนโลยีโครงข่ายประสาทเทียมจากกฏของไฟร์วอลล์ | | |
| **นักศึกษา** | ฐิติโชติ พิพัฒน์บุญ | ใจเมือง  พุทธคุณ | รหัสนักศึกษา  60070019 รหัสนักศึกษา  60070065 |
| **ปริญญา** | วิทยาศาสตรบัณฑิต | | |
| **สาขาวิชา** | เทคโนโลยีสารสนเทศ | | |
| **ปีการศึกษา** | 2563 | | |
| **อาจารย์ที่ปรึกษา** | ผู้ช่วยศาสตราจารย์ อัครินทร์ คุณกิตติ | | |

**บทคัดย่อ**

|  |  |  |  |
| --- | --- | --- | --- |
| **Project Title** | Researching for developing training set with artificial neural network  technology based on firewall rules | | |
| **Student** | Thitichote Pipatboon | Chaimuang  Buddhakul | Student ID  60070019 Student ID  60070065 |
| **Degree** | วิทยาศาสตรบัณฑิต | | |
| **Program** | เทคโนโลยีสารสนเทศ | | |
| **Academic Year** | 2020 | | |
| **Advisor** | ผู้ช่วยศาสตราจารย์ อัครินทร์ คุณกิตติ | | |

**ABSTRACT**

**กิตติกรรมประกาศ**

ปริญญานิพนธ์ฉบับนี้สาเร็จลุล่วงไปได้ด้วยดี ทางผู้จัดทาขอขอบพระคุณเป็นอย่างสูงกับความกรุณาช่วยเหลือและการให้คาปรึกษาของ ผู้ช่วยศาสตราจารย์อัครินทร์ คุณกิตติ ที่ช่วยชี้แนะแนวทาง ตั้งแต่วันแรกถึงวันสุดท้าย และขอบพระคุณอาจารย์ คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระ จอมเกล้าเจ้าคุณทหารลาดกระบังทุกๆท่าน ที่ให้ความรู้อันเป็นประโยชน์ยิ่ง ต่อการพัฒนาต่อยอดองค์ความรู้

ขอขอบคุณครอบครัวที่ให้การสนับสนุนอย่างดีเสมอมา

ขอขอบคุณคู่โครงงานที่อดทนและร่วมแรงร่วมใจช่วยกันมาจนถึงทุกวันนี้

ฐิติโชติ ใจเมือง

พิพัฒน์บุญ พุทธคุณ

**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

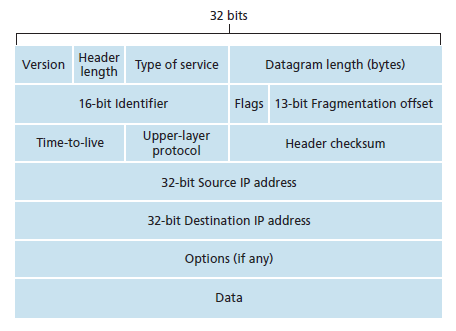
**A picture containing application

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**

**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

**บทที่ 4**

**ผลการดำเนินงานวิจัย**

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

**4.1. 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

**4.2. ผลการทดสอบการทำนายผลของโมเดลที่เรียนรู้ชุดฝึกสอนแบบ N Sample**

**4.3. ผลการทดสอบการทำนายผลจากโมเดลที่เรียนรู้ชุดฝึกสอนแบบ Ratio**

**บทที่ 5**

**ผลการวิเคราะห์การทดลอง**

เป้าหมายหลักของบทนี้คือการวิเคราะห์ผลการทดลองจากการนำชุดข้อมูลฝึกสอนที่สร้างจากกฎของไฟร์วอลล์ที่ออกแบบให้ตรงตามจุดประสงค์ของสมมติฐาน เพื่อหาชุดข้อมูลฝึกสอนที่สามารถทำให้โมเดลมีประสิทธิภาพในด้านความแม่นยำในการทำนายและเวลาที่ใช้ได้ดีที่สุด จึงจำเป็นต้องมีการวิเคราะห์ในเชิงเปรียบเทียบ ปรับรูปแบบกราฟเพื่อหาความสัมพันธ์ของตัวแปรต่างๆ

**5.1. การวิเคราะห์กลไกการทำงานโดยรวมของโมเดล**

**5.2.** **การวิเคราะห์ประสิทธิภาพการทำงานของโมเดล**

**บทที่ 6**

**สรุปผลและข้อเสนอแนะ**

**6.1. สรุปผลการดำเนินงานวิจัย**

**6.2. ปัญหาและอุปสรรคที่พบในงานวิจัย**

**6.3. ข้อเสนอแนะและแนวทางการพัฒนางานวิจัยในอนาคต**

**บรรณานุกรม**

[1] TensorFlow Teams. **“Essential Documentation”** [Online]. Available : <https://www.tensorflow.org/guide>. 2020

[2] nessessence. **“ปัญญาประดิษฐ์ (AI: Artificial Intelligence) คืออะไร”** [Online]. Available : <https://www.thaiprogrammer.org/2018/12/whatisai/>. 2018

[3] Rene Molenaar. “**IPv4 Packet Header”** [Online]. Available : <https://networklessons.com/cisco/ccna-routing-switching-icnd1-100-105/ipv4-packet-header>. 2020

[4] Sci-kit learn developers. **“scikit classification model”** [Online]. Available : <https://scikit-learn.org/stable/search.html?q=classification>. 2020

[5] TensorFlow Teams. **“พื้นฐาน Deep Learning”** [Online]. Available :<https://www.tensorflow.org/guide>. 2020

[6] sinlapachai lorpaiboon. “**มาเรียนรู้คำสั่งของ Pandas ใน Python ที่เอาไว้ใช้สำหรับจัดการข้อมูลกัน”** [Online]. Available : <https://medium.com/@sinlapachai.hon/มาเรียนรู้การใช้-การทำความสะอาดข้อมูลด้วย-python-โดยการใช้-pandas-กัน-2f5049640e70>. 2020

[7] T. Hastie, R. Tibshirani, J. Friedman. **The Elements of Statistical Learning (Second Edition).** : Springer-Verlag. 2009

[8] Saishruthi Swaminathan. “**Logistic Regression — Detailed Overview”** [Online]. Available : <https://towardsdatascience.com/logistic-regression-detailed-overview-46c4da4303bc>. 2018

[9] Anas Al-Masri. “**What Are Overfitting and Underfitting in Machine Learning?”** [Online]. Available : <https://towardsdatascience.com/what-are-overfitting-and-underfitting-in-machine-learning-a96b30864690>. 2019

[10] Will Koehrsen. “**Overfitting vs. Underfitting: A Complete Example”** [Online]. Available : <https://towardsdatascience.com/overfitting-vs-underfitting-a-complete-example-d05dd7e19765>. 2018

[11] Ahmed Gad. “**Beginners Ask ‘How Many Hidden Layers/Neurons to Use in Artificial Neural Networks?’”** [Online]. Available : <https://towardsdatascience.com/beginners-ask-how-many-hidden-layers-neurons-to-use-in-artificial-neural-networks-51466afa0d3e>. 2018

[12] Aurélien Géron. **Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow.** Sixth Release. United States of America: O’Reilly Media, Inc. 2019

[13] D. STATHAKIS. “How many hidden layers and nodes?” **International Journal of Remote Sensing,** Vol. 30, No. 8, 20 April 2009. pp2133–2147

[14] Jeff Heaton. **“Heaton Research The Number of Hidden Layers”** [online]. Available : [The Number of Hidden Layers | Heaton Research](https://www.heatonresearch.com/2017/06/01/hidden-layers.html). 2017

**ประวัติผู้เขียน**

A person posing for the camera

Description automatically generatedชื่อ – นามสกุล นาย ฐิติโชติ ใจเมือง

รหัสนักศึกษา 60070019

วัน เดือน ปีเกิด 7 พฤศจิกายน 2541

ประวัติการศึกษา

วุฒิ ม.6 โรงเรียนเตรียมอุดมศึกษาพัฒนาการ

ภูมิลำเนา จังหวัดกรุงเทพมหานคร

เบอร์โทร 08-6778-7397 E-Mail 60070019@it.kmitl.ac.th

สาขาที่จบ วิทยาศาสตร์ - คณิตศาสตร์ รุ่นที่ 34 ปีการศึกษา 2559

A person wearing a blue shirt

Description automatically generatedชื่อ – นามสกุล นาย พิพัฒน์บุญ พุทธคุณ

รหัสนักศึกษา 60070065

วัน เดือน ปีเกิด 25 เมษายน 2542

ประวัติการศึกษา

วุฒิ ม.6 โรงเรียนเซนต์ดอมินิก

ภูมิลำเนา จังหวัดกรุงเทพมหานคร

เบอร์โทร 08-6058-0919 E-Mail 60070065@it.kmitl.ac.th

สาขาที่จบ ศิลป์​-คำนวณ รุ่นที่ 48 ปีการศึกษา 2559

**ภาคผนวก**

**ขั้นตอนการติดตั้งไลบราลีและเครื่องมือสำหรับการใช้งานโครงข่ายประสาทเชิงลึกด้วยคอมพิวเตอร์ส่วนบุคคล**

**ส่วนที่ 1 ส่วนประกอบที่จำเป็นในการติดตั้งโปรแกรม**

* 1. ส่วนประกอบที่จำเป็นในการติดตั้งโปรแกรม

1.1.1. Windows 10 x64 bits

1.1.2. Python 3.7

1.1.3. Anaconda Navigator

**ส่วนที่ 2 ขั้นตอนการใช้งานและการทำงานของโปรแกรมที่เกี่ยวข้อง**

* 1. การติดตั้งสภาพแวดล้อมที่จำเป็นโดยใช้ Anaconda Navigator
     1. เข้าเว็บไซต์ และเลือกดาวน์โหลดแอพพลิเคชั่นสำหรับ Windows 64 bit

Graphical user interface, text, application

Description automatically generated

**รูปที่ ผ.1** การโหลดแอพพลิเคชั่น Anaconda Navigator ผ่านเว็บไซต์

* + 1. สร้างสภาพแวดล้อมใหม่เลือกเป็น Python เวอร์ชั่น 3.7

Graphical user interface

Description automatically generated  
**รูปที่ ผ.2** การสร้าง Environment เพื่อใช้งานโปรแกรมทั้งหมดในการทำวิจัย

2.1.3. ติดตั้งไลบราลีที่จำเป็น อย่างน้อยจะต้องมี Tensorflow และ Keras จึงจะสามารถทำงานได้

Graphical user interface, text, application, email

Description automatically generated

**รูปที่ ผ.3** การค้นหาเครื่องมือ Tensorflow และ Keras

Graphical user interface, text, application, email

Description automatically generated  
**รูปที่ ผ.4** รูปไลบรารีที่จำเป็นหลังติดตั้งเสร็จสิ้นแล้ว

* + 1. เมื่อติดตั้งเสร็จ ให้เปิดด้วยโปรแกรม Spyder ผ่านสภาพแวดล้อมที่ Anaconda สร้างเอาไว้

Graphical user interface, application

Description automatically generated

**รูปที่ ผ.5** การเปิดแอพพลิเคชั่น Spyder ผ่าน Anaconda Navigator

* 1. โปรแกรม Packet Generator

Source code ในงานวิจัย: https://github.com/Kodashi/AI-Firewall-Training-set-Researching-main