**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

**กลไกในการออกแบบชุดข้อมูลฝึกสอน**

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

**3.3.3. ขั้นตอนที่ 3 การนำโมเดลไปผ่านการเรียนรู้ด้วยชุดข้อมูลสำหรับฝึกสอน**

**Diagram

Description automatically generated  
รูปที่ 3.6** 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**

* รูปแบบการเรียนรู้: Sequential Logistic Regression
* ฟังก์ชั่นการประมวลผล: Sigmoid A picture containing object, clock

  Description automatically generated
* เครื่องมือเสริมประสิทธิภาพในการประมวลผล: Adam Optimizer

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

* ชุดข้อมูลฝึกสอน ที่ประกอบไปด้วย Data Field ภายใน Packet ทั้งหมด
* ชุดข้อมูลฝึกสอน ที่ประกอบไปด้วย Decision ที่เป็นผลลัพธ์ตัดสินใจว่าจะปล่อยผ่าน
* ชุดข้อมูลทดสอบ ที่ประกอบไปด้วย Data Field ภายใน Packet ทั้งหมด
* 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 3.7** Block Diagram การสร้างชุดข้อมูลทดสอบโมเดล

**หลักการออกแบบชุดข้อมูลทดสอบ**

ในการสร้างชุดข้อมูลทดสอบที่สามารถวัดผลความแม่นยำของโมเดลจากการทดลองได้ ในการออกแบบนั้นถือว่ามีความท้าทายในระดับหนึ่ง เพราะมีประเด็นสำคัญที่จำเป็นต้องพิจารณาดังต่อไปนี้

* จะทราบได้อย่างไรว่า โมเดลสามารถทำนายผลลัพธ์ได้ดีในทุกกฎไฟร์วอลล์
* จะทราบได้อย่างไรว่า โมเดลติดปัญหา Underfitting หรือ Overfitting

เราได้ทำการสร้างชุดข้อมูลทดสอบ และแบ่งจำนวนชุดข้อมูลออกเป็นจำนวนที่เท่าๆกัน ในแต่ละเงื่อนไขกฎของไฟร์วอลล์ เพื่อให้สามารถทราบได้ว่าภาพรวมที่โมเดลทำนายผลมานั้นให้ความถูกต้องแม่นยำเป็นอย่างไร ซึ่งถ้าหากไฟร์วอลล์นั้นสามารถทำนายผลได้เพียงบางเงื่อนไข ความแม่นยำที่ได้จากชุดข้อมูลทดสอบเดียวกันแต่โมเดลต่างกันจะต้องเห็นผลลัพธ์ที่สามารถสังเกตได้อย่างแน่นอน

ในความเป็นจริงแล้ว เพื่อให้มีการทดสอบและวิเคราะห์ได้ดียิ่งขึ้น อาจต้องสร้างชุดข้อมูลทดสอบหลายๆแบบที่มีความแตกต่างกัน เพื่อให้สามารถจับประเด็นสำคัญหรือปัญหาที่เกิดขึ้นจากโมเดลได้ เช่น การทดสอบว่าโมเดลมีปัญหา Overfitting หรือมีวิธีการตรวจสอบที่ดีหรือไม่

**3.3.5. ขั้นตอนที่ 5 การนำโมเดลไปประมวลผล ทำนายผลลัพธ์จากชุดข้อมูลสำหรับทดสอบ**

**Diagram

Description automatically generated  
รูปที่ 3.8** 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 3.9** 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 3.10** Block Diagram ขั้นตอนการนำผลลัพธ์มาบันทึกผล

การหาวิธีการที่สามารถทำให้ชุดข้อมูลฝึกสอนสามารถสอนโมเดลได้อย่างมีประสิทธิภาพ เราจำเป็นต้องนำผลลัพธ์ของการทดสอบในแต่ละครั้งของการทดลองมาบันทึกผล แล้วสรุปให้อยู่ในรูปกราฟเปรียบเทียบที่ประกอบไปด้วยผลลัพธ์จากการทดลองภายใต้สภาพแวดล้อมเดียวกัน เพื่อหาว่าผลลัพธ์ออกมาตรงตามสมมติฐานหรือมีความสัมพันธ์กันในแต่ละตัวแปรอย่างไรบ้าง

**A close up of a building

Description automatically generated  
รูปที่ 3.11** ตัวอย่างของตารางที่จะนำมาบันทึกผลลัพธ์การทดลอง