**บทที่ 3**

**วิธีการดำเนินการวิจัย**

การดำเนินการวิจัยการสร้างชุดข้อมูลในการฝึกสอนไฟร์วอลล์ปัญญาประดิษฐ์ด้วยเทคโนโลยีโครงข่ายประสาทเทียมจากกฏของไฟร์วอลล์ มีจุดประสงค์เพื่อพัฒนาชุดข้อมูลฝึกสอนที่สร้างจากกฎของไฟร์วอลล์ เพื่อให้ชุดข้อมูลฝึกสอนสามารถสอนโมเดลได้ถูกต้องและแม่นยำอย่างมีประสิทธิภาพ

**3.1 การศึกษาค้นคว้าเทคโนโลยีและเครื่องมือที่ใช้ในการพัฒนาโมเดล**

ในการดำเนินการวิจัย ผู้จัดทำเลือกที่จะใช้ 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 แยกประเภทข้อมูลกลุ่มโครงสร้างเช่น 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 การตั้งสมมติฐานในงานวิจัย**

ข้อสรุปที่ได้จากการงานวิจัยชิ้นนี้จะมาจากผลการทดลองทั้งสิ้น โดยเป้าหมายหลักคือการวิเคราะห์ผลการทดลองจากการนำชุดข้อมูลฝึกสอนที่สร้างจากกฎของไฟร์วอลล์ที่สร้างไว้ ออกแบบให้ตรงตามจุดประสงค์ของสมมติฐาน โดยเราได้เริ่มจากการต่อยอดนำการทดลองและข้อสรุปจากเอกสารงานวิจัยอื่นมาใช้ในการตั้งสมมติฐานเบื้องต้นของงานวิจัยชิ้นนี้ ซึ่งประกอบไปด้วย

* โมเดลจะสามารถเรียนรู้ชุดข้อมูลจากชุดข้อมูลฝึกสอนที่สร้างจากกฎของไฟร์วอลล์และสามารถทำนายผลลัพธ์ได้
* เมื่อโมเดลเรียนรู้จากชุดข้อมูลที่มีจำนวนมากขึ้นและหลากหลายมากขึ้น โมเดลจะสามารถทำนายผลลัพธ์ได้แม่นยำมากขึ้น
* โมเดลเมื่อมีการเรียนรู้ถึงจุดๆหนึ่งจะไม่สามารถเพิ่มความแม่นยำในการทำนายผลลัพธ์ได้อีก
* โมเดลที่มีกฎของไฟร์วอลล์มากขึ้น มีความซับซ้อนมากขึ้น จะยังใช้เวลาในการประมวลผลเท่าเดิม

**3.3 วัฎจักรกระบวนการพัฒนางานวิจัยในการสร้างชุดข้อมูลฝึกสอน**

ในการวิจัยจะมุ่งเน้นไปที่การพัฒนาชุดข้อมูลฝึกสอน โดยการเปรียบเทียบหาผลลัพธ์จากการนำชุดฝึกสอนไปผ่านโมเดล DNN และได้ผลลัพธ์ออกมาที่มีความแม่นยำมากที่สุด ซึ่งการทดลองดังกล่าวจำเป็นต้องทำด้วยกันหลายครั้ง ทดสอบในหลายแง่มุมเพื่อให้สามารถวิเคราะห์และเปรียบเทียบหาข้อสรุปซึ่งในแต่ละครั้งการทดลองก่อนการนำมาเปรียบเทียบจะมีกระบวนการดำเนินงาน ดังนี้

Diagram

Description automatically generated

จากรูปภาพ ทำให้แบ่งขั้นตอนการทดลองหลักๆได้เป็น 6 ส่วนหลักตามการทำงานของโปรแกรม ได้แก่

* + การกำหนดกฏของไฟร์วอลล์และความเป็นไปได้ทั้งหมดของชุดข้อมูล Packet ในเครือข่าย
  + การสร้างชุดข้อมูลสำหรับการฝึกสอนและชุดข้อมูลสำหรับการทดสอบ
  + การนำชุดข้อมูลฝึกสอนผ่านโมเดลเพื่อเริ่มทำการเรียนรู้
  + การนำชุดข้อมูลทดสอบประมวลผลด้วยโมเดลที่ผ่านการเรียนรู้แล้ว
  + การเปรียบเทียบผลลัพธ์ค่าความถูกต้องของโมเดลที่ทดสอบกับชุดข้อมูลทดสอบ
  + การนำผลการเปรียบเทียบแต่ละครั้งมาสรุปเพื่อหาผลลัพธ์ที่ออกมาดีที่สุด

**ส่วนที่ 1** **การกำหนดกฏของไฟร์วอลล์และความเป็นไปได้ทั้งหมดของชุดข้อมูล Packet ในเครือข่าย**

เงื่อนไขหลักของการวิจัยคือการสร้างชุดข้อมูลฝึกสอนจากกฎของไฟร์วอลล์ เพื่อให้ได้ระบบการทำงานคัดกรองข้อมูล Packet ที่ได้มาตรฐานและเรียนรู้ได้เองอย่างมีประสิทธิภาพ ความแม่นยำสูง สิ่งที่ต้องทำในส่วนแรกคือการกำหนดขอบเขตความเป็นไปได้ที่ข้อมูลจะสามารถเกิดขึ้นในเครือข่าย และการกำหนดกฎของไฟร์วอลล์เพื่อให้สามารถสร้างชุดข้อมูล Packet ที่จะนำไปฝึกสอนให้กับโมเดล สร้างชุดข้อมูลทดสอบโมเดล ที่สามารถเปรียบเทียบความถูกต้องของผลลัพธ์ที่ได้จากโมเดลหลังผ่านการเรียนรู้แล้ว

การกำหนดขอบเขตความเป็นไปได้ที่จะเกิดชุดข้อมูล Packet ใดๆ จำเป็นต้องรู้ส่วนประกอบทั้งหมดและค่าความเป็นไปได้ของแต่ละ Label ที่จะนำมาพิจารณา เพื่อมาคำนวณต่อหา Sample Space หรือโอกาสที่เกิดขึ้น ถ้าหากมีข้อมูลภายใน Field เพียงชุดเดียวที่แตกต่างกัน ชุดข้อมูล Packet นั้นจะเหมือนเป็นชุดข้อมูลใหม่ แต่ถึงกระนั้นจะต้องดูความเข้ากันได้ของข้อมูลด้วย และข้อมูลนั้นจะต้องสามารถเกิดขึ้นได้จริง ยกตัวอย่างเช่น ผู้รับและส่งไม่สามารถเป็น IP Address เดียวกันได้ หรือโปรโตคอล FTP จะต้องจับคู่กันระหว่าง Port 20 และ Port 21 เท่านั้น เป็นต้น

**จากการแจกแจงความเป็นไปได้ของข้อมูลใน Field ทำให้ได้ส่วนประกอบของ Packet ดังนี้**

* Source Address

ความเป็นไปได้ขึ้นอยู่กับ mask เช่น /24 จะเป็นไปได้ทั้งหมด 2^(32-24) ความเป็นไปได้

* Source Port

ความเป็นได้ขึ้นอยู่กับจำนวน port ใน pull ที่กำหนดไว้

* Destination Address

ความเป็นไปได้ขึ้นอยู่กับ mask เช่น /24 จะเป็นไปได้ทั้งหมด 2^(32-24) ความเป็นไปได้

* Destination Port

ความเป็นได้ขึ้นอยู่กับจำนวน port ใน pull ที่กำหนดไว้

* Protocol

ประกอบไปด้วย TCP และ UDP

ขั้นตอนต่อมาคือการสร้างกฎของไฟร์วอลล์ ในขั้นตอนนี้จะเป็นการกำหนดกระบวนการทำ Packet Filtering ที่จะเป็นการตัดสินว่า ข้อมูล Packet ชุดดังกล่าวจะสามารถถูกตัดสินให้ผ่านหรือไม่ ซึ่ง Packet ทุกชุดจะถูกตรวจสอบในทุกกฎของไฟร์วอลล์โดยมี 2 คำสั่งหลัก ได้แก่ “Allow” ปล่อยผ่านให้ข้อมูลชุดนั้นเข้าสู่ระบบหรือ “Deny” ไม่ปล่อยให้ข้อมูลชุดนั้นผ่านเข้าสู่ระบบ ในขั้นตอนนี้จะเป็นสิ่งสำคัญมากในขั้นตอนการสร้างชุดข้อมูลที่ใช้ในการฝึกสอนและการทดสอบ

**ตัวอย่างของการสร้างกฎของไฟร์วอลล์**

**Table

Description automatically generated**

เมื่อสร้างกฎของไฟร์วอลล์เป็นที่เรียบร้อยแล้ว จะต้องนำค่าความเป็นไปได้และกฎของไฟร์วอลล์ที่ตั้งไว้ไปเป็น Parameter ในโปรแกรม Packet Generator

**ส่วนที่ 2 การสร้างชุดข้อมูลสำหรับการฝึกสอนและชุดข้อมูลสำหรับการทดสอบ**

การสร้างชุดข้อมูลฝึกสอนและชุดข้อมูลทดสอบจะถูกสร้างโดยโปรแกรม Packet Generator ที่สร้างขึ้นเอง ข้อมูลที่ถูกสร้างขึ้นจะถูกจัดระเบียบอยู่ใน Cell ของไฟล์นามสกุล CSV ทำให้ง่ายแก่การดึงข้อมูลกลับมาใช้ต่อในขั้นตอนถัดไป แต่ก่อนที่จะสร้างชุดข้อมูล Packet นั้นจะต้องทราบความต้องการและจุดประสงค์ของโมเดล ว่าโมเดลดังกล่าวมีการต้องการชุดข้อมูลที่มีความสัมพันธ์และมีจำนวน Input และ Output อย่างไร การสร้างชุดข้อมูล Packet จะถูกคำนวณจากความเป็นไปได้ทั้งหมดของชุดข้อมูล Packet ทั้งหมด และหลังจากนั้นจะเป็นการเพิ่ม Decision Field เข้าไปในชุดข้อมูล Packet แต่ละชุด เพื่อให้โมเดลนำไปเข้ากระบวนการเรียนรู้ และเปรียบเทียบผลลัพธ์ในขั้นตอนหลังการทดสอบ (Evaluate) หาค่า Reference Variant Set โดยค่าภายใน Decision Field จะถูกสร้างอ้างอิงกับกฎของไฟร์วอลล์ในขั้นตอนแรก

**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 ต้องทำการเรียนรู้แบบ 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

**หลักในการออกแบบชุดข้อมูลทดสอบ**

ในการตรวจสอบความถูกต้องของโมเดลที่ผ่านการเรียนรู้แล้ว เราใช้อัตราส่วน 50:50 แบ่งมาจากตัวอย่างของการให้ชุดข้อมูลผ่านและไม่ให้ชุดข้อมูลผ่าน โดยชุดข้อมูลทดสอบจะต้องครอบคลุมกฎของไฟร์วอลล์ทั้งหมดด้วย

ผลลัพธ์ในขั้นตอนนี้จะได้ผลลัพธ์ออกมาเป็นไฟล์นามสกุล CSV ที่ประกอบด้วย Packet จำนวนมาก ที่มี Decision Field ในการตัดสินใจว่าชุดข้อมูล Packet นั้นจะสามารถถูกตัดสินให้ผ่านไปได้หรือไม่

**ส่วนที่ 3 การนำชุดข้อมูลฝึกสอนผ่านโมเดลเพื่อเริ่มทำการเรียนรู้**

เป็นขั้นตอนการนำชุดข้อมูลฝึกสอนที่สร้างขึ้นไปประมวลผลผ่านโมเดลให้เกิดการเรียนรู้ โดยขั้นตอนการฝึกโมเดลจะต้องมีการกำหนดค่าพารามิเตอร์ ฟังก์ชั่นการเรียนรู้ และปรับปรุงการประมวลผลหาคำตอบขึ้นอยู่กับขอบเขตของงาน ซึ่งในส่วนนี้เราสามารถหาวิธีการที่ได้มาตรฐานจากงานวิจัยต่างๆ มาใช้อ้างอิงกับงานวิจัย พัฒนาให้โมเดลสามารถเรียนรู้ผ่านชุดข้อมูลฝึกสอนได้อย่างมีประสิทธิภาพขึ้นได้

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

**ข้อมูลการตั้งค่าที่สำคัญภายในโมเดล**

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

  Description automatically generated
* เครื่องมือเสริมประสิทธิภาพในการประมวลผล: Adam Optimizer

กระบวนการทำงานในขั้นตอนนี้ จะเป็นการแยกส่วนข้อมูลที่จะใช้พิจารณาแยกกันในไฟล์นามสกุล CSV ที่สร้างจากขั้นตอนที่แล้ว โดยแบ่งออกเป็นชุดข้อมูลฝึกสอนและชุดข้อมูลทดสอบในอัตราส่วนที่ได้จาก 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%

**ส่วนที่ 4 การนำชุดข้อมูลทดสอบประมวลผลด้วยโมเดลที่ผ่านการเรียนรู้แล้ว**

การทดสอบจริงโดยใช้ชุดข้อมูลทดสอบ (Evaluate) จะต้องมีโปรแกรมอีกตัวหนึ่งเพื่อทดสอบความแม่นยำของโมเดลโดยเฉพาะ โปรแกรมส่วนนี้จะทำหน้าที่การนำชุดข้อมูลทดสอบมาประมวลผลผ่านโมเดลที่บันทึกไว้ แล้วทำการตรวจคำตอบที่ถูกและผิดด้วยฟังก์ชั่น model.predict\_class ซึ่งมีอยู่ 2 ประเภท คือ Allow และ Deny ผลลัพธ์สุดท้ายจะได้เป็นจำนวนของ Packet ที่มีการตัดสินผ่าน Reference Variant Set ในรูปแบบต่างๆ ในขั้นตอนนี้จะต้องมีการจับเวลาเพื่อหาความสัมพันธ์ระหว่างเวลาและจำนวนของข้อมูลด้วย

**ส่วนที่ 5 การเปรียบเทียบผลลัพธ์ค่าความถูกต้องของโมเดลที่ทดสอบกับชุดข้อมูลทดสอบ**

เมื่อการขั้นตอนของการทดสอบเสร็จสิ้น ให้นำค่าที่ได้จากขั้นตอนที่ 4 มาเปรียบเทียบกับชุดข้อมูลทดสอบที่มีเพียง Decision Field กระบวนการนี้จะเป็นการเปรียบเทียบว่ามีค่าตรงกันหรือไม่ และเมื่อเปรียบเทียบผลลัพธ์มีโอกาสออกมา 4 รูปแบบด้วยกัน ซึ่งทำให้ไปอ้างอิงกับการคำนวณผลลัพธ์ต่อได้ว่า มีการตัดสินใจออกมาเป็นอย่างไรตามหลัก Reference Variant Set

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

ทำให้ผลลัพธ์ที่ได้จะเป็นไปตามสูตร

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

**ส่วนที่ 6 การนำผลการเปรียบเทียบแต่ละครั้งมาสรุปเพื่อหาผลลัพธ์ที่ออกมาดีที่สุด**

หากย้อนกลับไปที่จุดประสงค์ของงานวิจัยหลัก คือการหาวิธีการที่สามารถทำให้ชุดข้อมูลฝึกสอนสามารถสอนโมเดลได้อย่างมีประสิทธิภาพ เราจำเป็นต้องนำผลลัพธ์ของการทดสอบในแต่ละครั้งของการทดลองมาบันทึกผล แล้วสรุปให้อยู่ในรูปกราฟที่ประกอบไปด้วยผลลัพธ์จากการทดลองภายใต้สภาพแวดล้อมเดียวกัน เพื่อหาว่าผลลัพธ์ออกมาตรงตามสมมติฐานหรือสามารถบอกมุมมองใหม่ได้หรือไม่

ตัวแปรที่มีการเปลี่ยนค่าไปตามการทดลอง

* จำนวนของ Packet ที่นำเข้าระบบ หรือ Sample(N)

ได้แก่ ชุดข้อมูลฝึกสอนและชุดข้อมูลทดสอบที่สร้างจากกฎของไฟร์วอลล์

* จำนวน Node ของแต่ละ Layer

Node จะถูกกำหนดค่าไว้ตั้งแต่แรกตามจำนวนขนาดของชุดข้อมูล

* จำนวนรอบการทดสอบ หรือ Epoch

ผลลัพธ์ที่ค่าจะต้องเปลี่ยนแปลงไปตามการทดสอบแต่ละครั้ง

* เวลาที่โมเดลใช้ในการเรียนรู้จากชุดข้อมูลฝึกสอน หรือ Training
* เวลาที่โมเดลใช้ในการตัดสินใจจากชุดข้อมูลทดสอบ หรือ Predict
* ค่าความแม่นยำ หรือ Accuracy
* ค่าอัตราการสูญเสีย หรือ Loss
* อัตราความผิดพลาดที่อ้างอิงจาก Reference Variant Set