บทที่ 3

วิธีการดำเนินการวิจัย

การดำเนินการวิจัยการสร้างชุดข้อมูลในการฝึกสอนไฟร์วอลล์ปัญญาประดิษฐ์ด้วยเทคโนโลยีโครงข่ายประสาทเทียมจากกฎของไฟร์วอลล์ มีจุดประสงค์เพื่อพัฒนาชุดข้อมูลฝึกสอนที่สร้างจากกฎของไฟร์วอลล์ เพื่อให้ชุดข้อมูลฝึกสอนสามารถสอนโมเดลได้ถูกต้องและแม่นยำอย่างมีประสิทธิภาพ

3.1 การศึกษาค้นคว้าเทคโนโลยีและเครื่องมือที่ใช้ในการพัฒนาโมเดล

ในการดำเนินการวิจัย ผู้จัดทำเลือกที่จะใช้ Python เป็นภาษาหลักในการพัฒนาโปรแกรมสร้างชุดข้อมูลฝึกสอนและส่วนของโมเดลเรียนรู้ 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

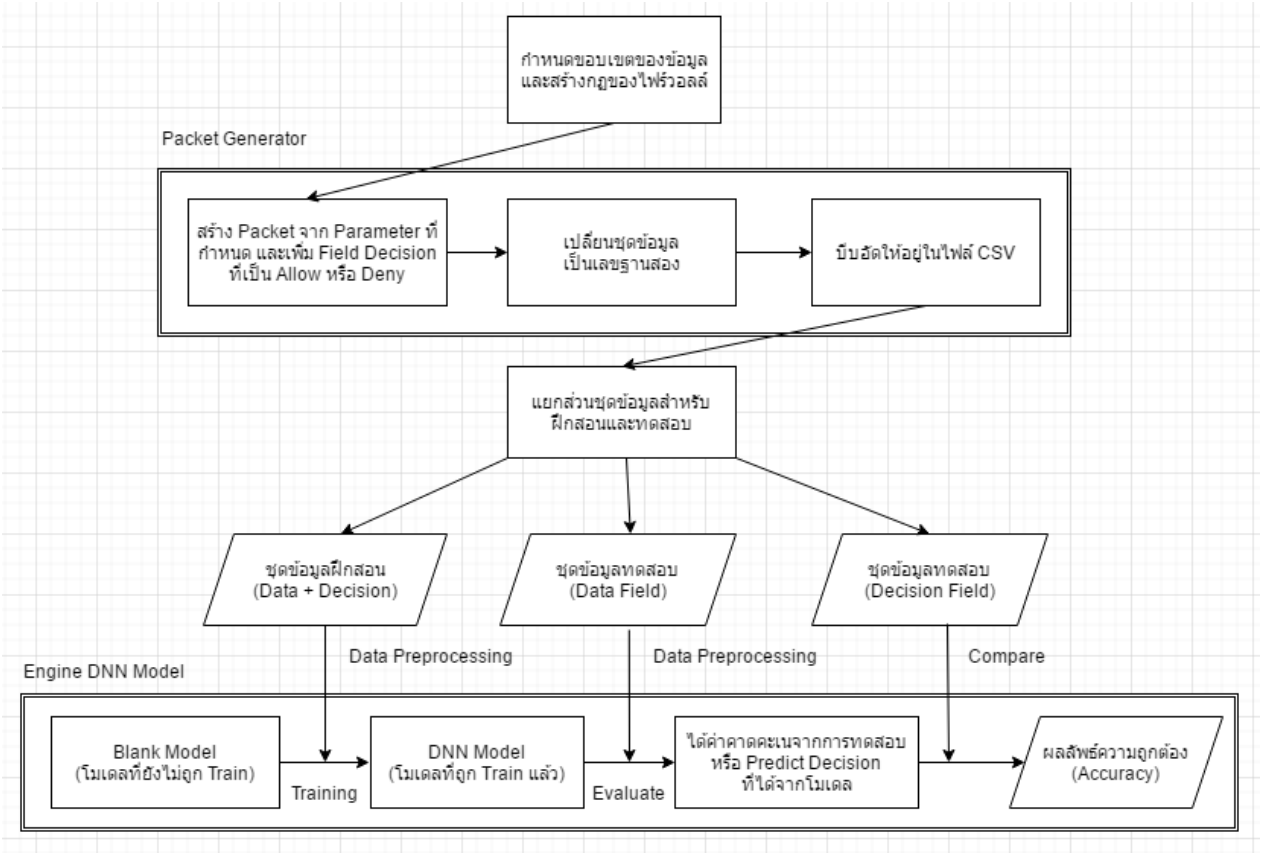
  Version: Tk 8.6.10
- NVIDIA CUDA เครื่องมือช่วยให้คอมพิวเตอร์สามารถประมวลผลผ่าน GPU ได้

  Version: CUDA 11.1.0
- NVIDIA cuDNN เครื่องมือช่วยในการประมวลผล DNN ผ่าน GPU

  Version: cuDNN 8.0

3.2 กระบวนการพัฒนาชุดข้อมูลฝึกสอน Training model

ในการวิจัยจะมุ่งเน้นไปที่การพัฒนาชุดข้อมูลฝึกสอน โดยการเปรียบเทียบหาผลลัพธ์จากการนำชุดฝึกสอนไปผ่านโมเดล DNN และได้ผลลัพธ์ออกมาที่มีความแม่นยำมากที่สุด ซึ่งการทดลองดังกล่าวจำเป็นต้องทำด้วยกันหลายครั้ง ซึ่งในแต่ละครั้งการทดลองก่อนการนำมาเปรียบเทียบจะมีกระบวนการดำเนินงาน ดังนี้



จากรูปภาพ ทำให้แบ่งขั้นตอนการทดลองหลักๆได้เป็น 6 ส่วนหลักตามการทำงานของโปรแกรม ได้แก่

- การกำหนดกฎของไฟร์วอลล์และความเป็นไปได้ทั้งหมดของชุดข้อมูล Packet ในเครือข่าย
- การสร้างชุดข้อมูลสำหรับการฝึกสอนและชุดข้อมูลสำหรับการทดสอบ

- การนำชุดข้อมูลฝึกสอนผ่านโมเดลเพื่อเริ่มทำการเรียนรู้
- การนำชุดข้อมูลทดสอบประมวลผลด้วยโมเดลที่ผ่านการเรียนรู้แล้ว
- การเปรียบเทียบผลลัพธ์ค่าความถูกต้องของโมเดลที่ทดสอบกับชุดข้อมูลทดสอบ
- การนำผลการเปรียบเทียบแต่ละครั้งมาสรุปเพื่อหาผลลัพธ์ที่ออกมาดีที่สุด

ส่วนที่ 1 การกำหนดกฎของไฟร์วอลล์และความเป็นไปได้ทั้งหมดของชุดข้อมูล 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 21 และ Port 22 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 ปล่อยผ่านหรือ Deny ไม่ปล่อยให้ผ่าน ในขั้นตอนนี้จะสำคัญมากในขั้นตอนการสร้างชุดข้อมูลที่ใช้ในการฝึกสอนและการทดสอบ

ตัวอย่างของการสร้างกฎของไฟร์วอลล์

Allow 192.168.0.0 192.168.1.0 any

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

- Decision Field

  Allow แทนค่า เป็น 1

  Deny แทนค่า เป็น 0

ในขั้นตอนนี้จะได้ผลลัพธ์ออกมาเป็นไฟล์นามสกุล CSV ที่ประกอบด้วย Packet จำนวนมาก ที่มี Decision Field ในการตัดสินใจว่าชุดข้อมูล Packet นั้นจะสามารถถูกตัดสินให้ผ่านไปได้หรือไม่

ส่วนที่ 3 การนำชุดข้อมูลฝึกสอนผ่านโมเดลเพื่อเริ่มทำการเรียนรู้

เป็นส่วนที่ทำให้โมเดลเกิดการเรียนรู้จากชุดข้อมูลฝึกสอน Packet ที่สร้างขึ้นจากกฎของไฟร์วอลล์ แบ่งส่วนข้อมูลที่จะนำมาพิจารณาและผลลัพธ์การตัดสินใจ ทำการจัดข้อมูลให้อยู่ในรูปของ Matrix ตัวโมเดลจะทำการเลือกรูปแบบที่ให้ผลลัพธ์ที่ดีที่สุดโดยวัดผลจากค่าความแม่นยำและอัตราการสูญเสียข้อมูล เมื่อวิเคราะห์จากความต้องการและจุดประสงค์การเลือกใช้ของโมเดลแล้ว ทำให้สรุปได้ว่า Sequential Model ที่มีการ Compile แบบ Binary Classification Problem สามารถตอบโจทย์ได้ดีที่สุด เนื่องจากผลลัพธ์ Output สุดท้ายจะเข้าข่ายการตัดสินใจแบบ Two-Class-Label หมายความว่าที่โมเดลจะทำการตัดสินใจจะมีเพียง 2 ตัวเลือก ซึ่งในงานวิจัยจะมีเพียง Allow หรือ Deny เท่านั้นภายในการทดสอบ

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

ต่อมาจะเป็นการทำ 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

ต่อมาจะเป็นการทำ Compile และการเลือกสร้าง Model (กลไกการทำงานในชั้น Code จะเขียนในส่วนของโปรแกรม) ซึ่งค่า Parameter ที่เราต้องเป็นผู้กำหนดคือ จำนวน Layer, จำนวน Node ในแต่ละ Layer, ขนาดของ Batch, และจำนวนรอบ Epoch

ซึ่ง Parameter ที่เรากำหนด มีดังนี้ –

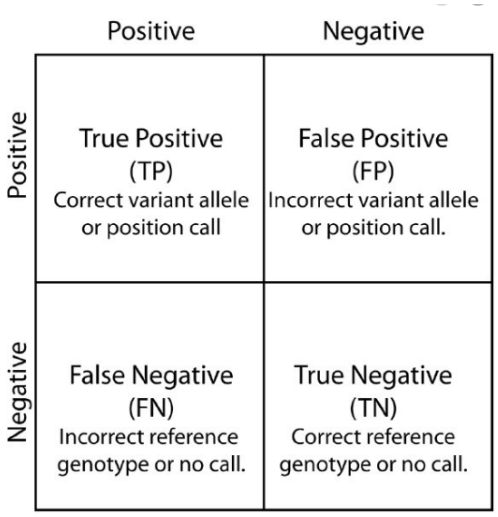
เมื่อได้โมเดลที่ผ่านการเรียนรู้แล้ว บันทึกโมเดลถือเป็นอันเสร็จสิ้น

ส่วนที่ 4 การนำชุดข้อมูลทดสอบประมวลผลด้วยโมเดลที่ผ่านการเรียนรู้แล้ว

การทดสอบหรือ Evaluate เป็นส่วนที่โมเดลที่ผ่านการเรียนรู้แล้วเริ่มทดสอบกับชุดข้อมูลทดสอบที่ประกอบไปด้วย Data Field ภายใน Packet เข้าฟังก์ชั่นการทำงาน model predict จะได้เป็นค่าคาดคะเนว่าจะเกิดขึ้นระหว่าง Allow หรือ Deny มากกว่ากัน ค่าที่เห็นจากตัวแปรจะเป็นตัวเลขทศนิยมที่อยู่ระหว่าง 0 ถึง 1 และเมื่อเข้าฟังก์ชั่น model.predict_class จะเป็นการให้โมเดลอ่านค่าผลลัพธ์ออกมาเป็นแค่ 0 หรือ 1 ในขั้นตอนนี้จะต้องมีการจับเวลาเพื่อหาความสัมพันธ์ระหว่างเวลาและจำนวนของข้อมูลด้วย

ส่วนที่ 5 การเปรียบเทียบผลลัพธ์ค่าความถูกต้องของโมเดลที่ทดสอบกับชุดข้อมูลทดสอบ

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



ผลลัพธ์ที่ได้จะประกอบไปทั้งหมด 4 ค่า ได้แก่

True Positive โมเดลอนุญาติให้ข้อมูลผ่านตรงตามกฎของไฟร์วอลล์ ให้ Allow ถูกต้อง
True Negative โมเดลไม่อนุญาติให้ข้อมูลผ่านตรงตามกฎของไฟร์วอลล์ ให้ Deny ถูกต้อง
False Positive โมเดลอนุญาติให้ข้อมูลผ่าน ไม่ตรงตามกฎของไฟร์วอลล์ ให้ Allow ผิดพลาด
False Negative โมเดลไม่อนุญาติให้ข้อมูลผ่าน ไม่ตรงตามกฎของไฟร์วอลล์ ให้ Deny ผิดพลาด

ผลลัพธ์ที่ได้จะเป็นไปตามสูตร

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

ส่วนที่ 6 การนำผลการเปรียบเทียบแต่ละครั้งมาสรุปเพื่อหาผลลัพธ์ที่ออกมาดีที่สุด

เป็นการนำผลลัพธ์ของการทดสอบในแต่ละครั้งของการทดลองมาบันทึกผล แล้วสรุปให้อยู่ในรูปกราฟที่ประกอบไปด้วยผลลัพธ์จากการทดลองภายใต้สภาพแวดล้อมเดียวกัน

ตัวแปรที่มีการเปลี่ยนค่าไปตามการทดลอง

- จำนวนของ Packet ที่นำเข้าระบบ หรือ Sample(N)
- จำนวน Node ของแต่ละ Layer
- จำนวนรอบการทดสอบ หรือ Epoch

ผลลัพธ์ที่ค่าจะต้องเปลี่ยนแปลงไปตามการทดสอบแต่ละครั้ง

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

3.3 กระบวนการสร้างโปรแกรมและเครื่องมือที่เกี่ยวข้อง

Packet Generator

เป็นโปรแกรมที่ใช้ในการสร้างชุดข้อมูล Packet โดยสุ่มจากพารามิเตอร์ที่กำหนดจากกฎของไฟร์วอลล์โดยชุดข้อมูลที่ได้จากการสุ่มจะถูกนำไปแปลงค่าข้อมูลเป็นเลขฐานสอง บันทึกเก็บไว้ในไฟล์นามสกุล CSV ก่อนจะนำไปเรียกใช้ต่อในโมเดล Depp Neural Network โดยโปรแกรมนี้จะถูกแบ่งไปใช้ในการทำงาน 2 ส่วน ได้แก่ ส่วนที่ใช้ในการสร้างชุดข้อมูลฝึกสอน และ ส่วนที่ใช้ในการสร้างชุดข้อมูลทดสอบ

Deep Neural Network Model Engine

เป็นเครื่องมือสร้าง Artificial Intelligent ที่พัฒนาขึ้นเอง โดยพัฒนาและประยุกต์โมเดลให้สามารถเรียนรู้กับชุดข้อมูลฝึกสอนที่ป้อนเข้าไป นำไปประมวลผล ตัดสินใจได้ว่าจะชุดข้อมูลที่ป้อนค่าเข้าไปนั้นเป็น Allow หรือ Deny และทำการตรวจสอบผลลัพธ์ที่ได้