โมเดลการเรียนรู้แบบผสมและแบบเพิ่มเติมได้สำหรับการตรวจจับการบุกรุกเครือข่าย

HYBRID AND INCREMENTAL LEARNING MODELS FOR NETWORK INTRUSION DETECTION

พลอยพรรณ สอนสุวิทย์
PLOYPHAN SORNSUWIT

วิทยานิพนธ์นี้สำหรับการศึกษาตามหลักสูตร
ปริญญาปรัชญาดุษฎีบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์
ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2562
KMITL-2019-SC-D-002-077

โมเดลการเรียนรู้แบบผสมและแบบเพิ่มเติมได้สำหรับการตรวจจับการบุกรุกเครือข่าย

HYBRID AND INCREMENTAL LEARNING MODELS FOR NETWORK INTRUSION DETECTION

พลอยพรรณ สอนสุวิทย์
PLOYPHAN SORNSUWIT

วิทยานิพนธ์นี้สำหรับการศึกษาตามหลักสูตร
ปริญญาปรัชญาดุษฎีบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์
ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2562
KMITL-2019-SC-D-002-077

HYBRID AND INCREMENTAL LEARNING MODELS FOR NETWORK INTRUSION DETECTION

PLOYPHAN SORNSUWIT

A THESIS SUBMITTED IN FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF DOCTOR OF PHILOSOPHY PROGRAM IN COMPUTER SCIENCE
DEPARTMENT OF COMPUTER SCIENCE FACULTY OF SCIENCE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
2019
KMITL-2019-SC-D-002-077

หัวข้อวิทยานิพนธ์	โมเดลการเรียนรู้แบบผสมและแบบเพิ่มเติมได้สำหรับการตรวจจับการบุกรุกเครือข่าย
ชื่อนักศึกษา	นางสาวพลอยพรรณ สอนสุวิทย์
รหัสประจำตัว	56605011
ปริญญา	ปรัชญาดุษฎีบัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชา	วิทยาการคอมพิวเตอร์
พ.ศ.	2562
อาจารย์ที่ปรึกษาวิทยานิพนธ์	ผู้ช่วยศาสตราจารย์ ดร. สายชล ใจเย็น

บทคัดย่อ

การตรวจจับการบุกรุกทางเครือข่ายเป็นสิ่งที่จำเป็นในการป้องกันความปลอดภัยของเครือข่ายคอมพิวเตอร์ในปัจจุบัน ซึ่งการตรวจจับการบุกรุกที่มีประสิทธิภาพมีหลากหลายวิธี เช่น การตรวจจับการจราจร (Traffic) เครือข่ายอินเทอร์เน็ต ซึ่งจะทำการรู้จำรูปแบบการบุกรุกและสามารถตรวจจับทางเครือข่ายได้อย่างมีประสิทธิภาพเมื่อพบเจอสิ่งผิดปกติ งานวิจัยนี้นำเสนอการพัฒนาสองโมเดลที่มีจุดเด่นต่างกัน ได้แก่ โมเดลการเรียนรู้แบบผสม ซึ่งเป็นโมเดลที่สามารถเรียนรู้การตรวจจับการบุกรุกแบบผสมที่มีประสิทธิภาพสูง มีการคัดเลือกคุณลักษณะด้วยวิธีการคัดเลือกคุณลักษณะบนพื้นฐานของความสัมพันธ์ (Correlation-base Feature Selection) และโมเดลที่มีการเรียนรู้แบบเพิ่มเติมได้ โดยจะสามารถเรียนรู้ข้อมูลที่เข้ามาใหม่โดยไม่ต้องนำไปคำนวณกับข้อมูลเดิมที่สร้างโมเดลไว้ ซึ่งเป็นการเป็นการละทิ้งข้อมูลเดิมที่เคยเรียนรู้แล้ว จึงมีเพียงโมเดลที่เป็นตัวแทนการตัดสินใจของข้อมูลทั้งหมด มีการคัดเลือกคุณลักษณะด้วยวิธีการสหสัมพันธ์แบบเพียร์สัน (Pearson Correlation) ผลการทดลองพบว่าวิธีการที่นำเสนอทั้งสองโมเดลสามารถปรับปรุงประสิทธิภาพในการทำการจำแนกซึ่งมีค่าความถูกต้องสูงเมื่อเทียบกับวิธีการอื่นและเมื่อพิจารณาถึงระดับกลุ่มของการบุกรุกก็มีประสิทธิภาพโดยรวมสูงที่สุดในการจำแนกรายกลุ่ม รวมไปถึงเมื่อทดสอบกับฐานข้อมูลการบุกรุกเครือข่ายอื่นพบว่ามีประสิทธิภาพสูงที่สุดเช่นกัน

คำสำคัญ : การบุกรุกเครือข่าย, การเรียนรู้ของเครื่อง, การเรียนรู้ที่เพิ่มขึ้น, การเรียนรู้แบบผสม

Thesis Title	Hybrid and Incremental Learning Models for Network Intrusion Detection
Student Name	Ploypan Sornsuwit
Student ID	56605011
Degree	Doctor of Philosophy (Computer Science)
Department	Computer Science
Year	2019
Thesis Advisor	Assistant Professor Dr. Saichon Jaiyen

ABSTRACT

Network-based intrusion detection system plays an essential role in protecting threats and providing the security for today's computer networks. Effective intrusion detection methods are various, such as traffic or internet networks. These methods are able to recognize the intrusion patterns and detect the network effectively when encountering abnormalities. The objective of this research was to propose the development of two models with different strengths. The first model is the hybrid learning model with its capacity to perform highly effective intrusion detection and to conduct correlation-base feature selection. The second one is the incremental learning model with its capacity to learn new and incremental data and input without any effort for calculating incremental data with previous data as well as discarding previously learned data. For this reason, the model is a single element as a mechanism for making decision of all data. In addition, pearson correlation feature selection is also conducted for this model. The experimental results indicated that two proposed models could be applied to improve classification performance with higher accuracy than other methods. When the efficiency of instruction detection at a class level was considered, the proposed models had the highest accuracy. Besides, when the proposed models were tested with other network intrusion detection datasets, they had the highest efficiency as well.

Keywords: Hybrid Learning, Incremental Learning, Intrusion Detection System, Machine Learning

กิตติกรรมประกาศ

งานวิจัยนี้เสร็จลุล่วงไปได้ด้วยความกรุณาจาก ผู้ช่วยศาสตราจารย์ ดร. สายชล ใจเย็น อาจารย์ที่ปรึกษา ที่ให้คำแนะนำ ความรู้และแนวทางการทำวิจัยจนเสร็จสมบูรณ์ ผู้วิจัยขอกราบขอบพระคุณเป็นอย่างสูงมา ณ โอกาสนี้

ขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.ศรัณย์ อินทโกสุม ผู้ช่วยศาสตราจารย์ ดร.อนันตพร หรรษคุณาฒัย ผู้ช่วยศาสตราจารย์ ดร.กุลสวัสดิ์ จิตขจรวาณิช และ ผู้ช่วยศาสตราจารย์ ดร.ศุภกานต์ พิมลเรศ กรรมการสอบวิทยานิพนธ์ ที่กรุณาให้คำแนะนำข้อและข้อชี้แนะ จนทำให้วิทยานิพนธ์สำเร็จลงได้ และขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.นวลสวาท หิรัญสกลวงศ์ รวมถึงผู้บริหาร และบุคลากรประจำคณะวิทยาศาสตร์ ซึ่งได้ให้คำแนะนำและความช่วยเหลือในทุกขั้นตอนจนการเรียนและการทำวิทยานิพนธ์สำเร็จลุล่วงไปได้ด้วยดี

ขอขอบพระคุณ รองศาสตราจารย์ พรทิพย์ ไวแสง และ อาจารย์ ดร.มณฑา พลรักษ์ คณะศิลปศาสตร์ ผู้ที่ให้ความรู้ด้านภาษาอังกฤษอย่างลึกซึ้ง อันเป็นประโยชน์ต่อการทำวารสาร และเอกสารทางวิชาการอย่างดียิ่ง

ขอขอบพระคุณ รองศาสตราจารย์ ดร. ณัฐรดา วงษ์นายะ ผู้บังคับบัญชา ผู้ที่ให้โอกาส ให้ความช่วยเหลือ และให้กำลังใจในทุกด้าน ขอขอบคุณ อาจารย์ ดร.พิมกาญดา จันดาหัวดง รวมถึงบุคลากรมหาวิทยาลัยราชภัฏกำแพงเพชร ที่แบ่งเบาภาระงานตามพันธกิจที่ได้รับมอบหมาย เพื่อให้ผู้วิจัยสามารถใช้เวลาทำวิทยานิพนธ์ได้อย่างเต็มศักยภาพทั้งยังให้โอกาสที่ดีเสมอมา

ท้ายที่สุด ขอขอบพระคุณบิดามารดา ที่ให้การสนับสนุนด้านการเรียนในระดับที่ตั้งใจ ทั้งยังช่วยเหลือทุดด้าน และเป็นกำลังใจที่ดีเสมอมา

คุณงามความดีและประโยชน์อันใดที่เกิดขึ้นจากวิทยานิพนธ์ฉบับนี้ข้าพเจ้าขอมอบให้บิดามารดา คณาจารย์ ญาตพี่น้อง และผู้ที่ช่วยเหลือทุกคน

พลอยพรรณ สอนสุวิทย์

สารบัญ

หน้า

สารบัญตาราง

ตารางที่ **หน้า**

สารบัญรูป

รูปที่ หน้า

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบัน การใช้งานอินเทอร์เน็ตมีความสำคัญกับผู้ใช้งานในชีวิตประจำวัน ทั้งในการปฏิบัติทำงานประจำ (Routine) และการใช้งานส่วนตัว เช่น การสื่อสารข้อมูลระหว่างคู่ค้าในภาคธุรกิจ การศึกษา การแพทย์ หรือ การสื่อสารข้อมูลภาครัฐ เป็นต้น ส่วนหนึ่งของการสื่อสารที่มีความสำคัญ คือการรักษาความมั่นคงปลอดภัยของสารสนเทศ ทั้งในส่วนของข้อมูลที่เก็บรักษา ความปลอดภัยระหว่างการรับส่ง และ ความปลอดภัยของนโยบายในการรักษาความมั่นคงปลอดภัย เพื่อให้ได้ตามหลักการของแนวคิดหลักด้านความมั่นคงปลอดภัยของสารสนเทศ เรียกว่า CIA Triad (Pfleeger and Pfleeger, 2012) ในการรับประกันความปลอดภัยของการรับส่งข้อมูล นอกจากนี้ในปัจจุบัน ยังมีเทคโนโลยีใหม่ที่เกิดขึ้นและมีบทบาทในด้านการสื่อสารเพื่อใช้งานเพิ่มมากขึ้น เช่น อินเทอร์เน็ตของทุกสิ่ง (Internet of Things) เป็นต้น ซึ่งมีแนวโน้มในการนำมาใช้งานสูงขึ้น และมีแนวโน้มการบุกรุกระบบที่สูงขึ้นเช่นกัน

การใช้งานเทคโนโลยีในปัจจุบันจึงเป็นเป้าหมายของผู้บุกรุก (Intruder) ในการเข้าถึงข้อมูลที่สำคัญ ทั้งในส่วนที่เป็นการเก็บรักษาและอยู่ในระหว่างการรับส่งมากมาย และการรักษาความมั่นคงปลอดภัยในปัจจุบัน อาจไม่เพียงพอต่อการป้องกันการถูกโจมตีทางไซเบอร์ (Cyber Attack) เพราะแม้มีเทคโนโลยีที่ดีขึ้น แต่ผู้บุกรุกก็ยังคงพัฒนาขั้นตอนวิธีในการเอาชนะการป้องกัน เพื่อคาดหวังผลทางการค้า หรือทดสอบขีดความสามารถของระบบอยู่อย่างต่อเนื่อง จึงเป็นการยากในการจะรู้ได้ว่าจะมีขั้นตอนวิธีใหม่ๆแบบใด ในการเอาชนะกฎของไฟร์วอลล์ (Firewall) ได้ และจะมีพฤติกรรมการบุกรุกใดบ้างที่สามารถหลบเลี่ยงการตรวจจับของระบบตรวจจับการบุกรุก (Intrusion Detection System: IDS) ได้ เป็นต้น ประกอบกับพฤติกรรมการบุกรุกของผู้โจมตีในปัจจุบันมักมีพฤติกรรมการบุกรุกที่ตรวจจับได้ยาก มีความสามารถในการอำพรางตัว ปะปน และหลบเลี่ยง กฎที่ตั้งไว้ได้ แม้เป็นการตรวจจับด้วยขั้นตอนวิธีทางหลักสถิติ (Statistical) ในระบบตรวจจับการบุกรุกที่มีในปัจจุบัน จะยังคงไม่สามารถปรับปรุงต้นแบบหรือโมเดล (Model) ที่ใช้ในการตรวจจับได้เองขณะที่มีข้อมูลใหม่ที่ต้องการเรียนรู้เพิ่มเข้ามาอยู่เสมอ เนื่องจากขั้นตอนวิธีแบบเดิมจะต้องเกิดการคำนวณโมเดลใหม่ทั้งหมด ซึ่งในการใช้งานจริงมักมีการบุกรุกแบบใหม่ๆเกิดขึ้นอยู่เสมอ จึงทำให้เป็นข้อจำกัดของการใช้งานระบบตรวจจับการบุกที่มีการใช้ขั้นตอนวิธีแบบเดิมที่เคยมีมา ประกอบกับหากมีการบุกรุกจากผู้บุกรุกที่มีความชำนาญสูง อาจทำให้ระบบตรวจจับได้คลาดเคลื่อน หรือไม่สามารถตรวจจับได้เลย ส่งผลต่อความปลอดภัยในองค์กรที่ต้องการป้องกันข้อมูลสำคัญจากการถูกบุกรุก ทั้งนี้ระบบจึงควรมีความสามารถในการเรียนรู้แบบเพิ่มเติมได้จากข้อมูลเครือข่ายใหม่ๆที่มีสิ่งผิดปกติผ่านเครือข่ายเข้าออกองค์กรตลอดเวลา

จากที่กล่าวมาจะพบได้ว่า เทคโนโลยีที่มีการใช้งานในปัจจุบันจะต้องอาศัยความน่าเชื่อถือสูงในการรับส่งข้อมูล เช่น การโอนเงินผ่านธนาคารออนไลน์ การยื่นภาษาออนไลน์ หรือการส่งข้อมูลทางธุรกิจอีเมล์ (Email) เป็นต้น ซึ่งจะมีความเสี่ยงสูงต่อการถูกดักจับข้อมูล ขโมยข้อมูล เปลี่ยนแปลงข้อมูล หรือหยุดให้บริการเครื่องแม่ข่ายได้ เสมือนมีเครื่องเป้าหมายที่ถูกโจมตีได้ตลอดเวลา การตรวจจับการบุกรุกจึงมีความสำคัญทั้งในด้านความสามารถในการจำแนก (Classification) ที่แม่นยำ

สูง และความสามารถในการเรียนรู้ที่เพิ่มเติมได้ (Incremental Learning) ในขณะที่กำลังใช้งานอยู่ จึงเป็นที่มาของงานวิจัยนี้ในการพัฒนาขึ้นตอนวิธีในการตรวจจับการบุกรุกสองขั้นตอนวิธี ได้แก่ การเรียนรู้แบบเพิ่มเติมได้สำหรับความปลอดภัยของเครือข่าย บนพื้นฐานของการวิเคราะห์ดิสคริมิแนนต์เชิงเส้นแบบเพิ่มเติมได้ (Incremental Linear Discriminant Analysis: ILDA) เพื่อให้สามารถจำแนกการบุกรุกประเภทต่างๆได้ และสามารถเรียนรู้โมเดลโดยไม่ต้องนำข้อมูลเดิมมาเรียนรู้ซ้ำ ณ ขณะเวลาใดๆ ซึ่งจะทำให้มีโมเดลในการตัดสินใจที่เป็นตัวแทนของข้อมูลทั้งหมด และพัฒนาขั้นตอนวิธีการเรียนรู้แบบผสม (Hybrid Learning) ในการตรวจจับการบุกรุกซึ่งจะเน้นประสิทธิภาพในสามารถจำแนกประเภทของการบุกรุกที่แม่นยำสูงเมื่อเปรียบเทียบกับวิธีการอื่นๆ

1.2 วัตถุประสงค์ของงานวิจัย

1) เพื่อพัฒนาขั้นตอนวิธีการเรียนรู้แบบเพิ่มเติมได้ สำหรับความปลอดภัยของเครือข่ายบนพื้นฐานของการวิเคราะห์ดิสคริมิแนนต์เชิงเส้นแบบเพิ่มเติมได้ โดยทำการคัดเลือกคุณลักษณะ (Feature Selection) ด้วยวิธีสหสัมพันธ์แบบเพียร์สัน

2) เพื่อพัฒนาขั้นตอนวิธีการเรียนรู้แบบผสม สำหรับความปลอดภัยของเครือข่ายโดยทำการคัดเลือกคุณลักษณะด้วยวิธีการคัดเลือกคุณลักษณะบนพื้นฐานของความสัมพันธ์

3) เพื่อเปรียบเทียบประสิทธิภาพของขั้นตอนวิธี (Algorithm) กับวิธีการของการเรียนรู้ของเครื่อง (Machine Learning) อื่นๆ ได้แก่ ซัพพอร์ตเวกเตอร์แมทชีน (Support Vector Machine: SVM) ต้นไม้ตัดสินใจ (Decision Tree) เพอร์เซ็ปตรอนหลายชั้น (Multilayer Perceptron: MLP) เพื่อนบ้านใกล้ที่สุด (*k*-Nearest Neighbor: *k*-NN) การวิเคราะห์การจำแนกประเภทเชิงเส้น (Linear Discriminant Analysis: LDA) และ นาอีฟเบย์ (Naïve Bayes)

4) เพื่อเปรียบเทียบประสิทธิภาพของขั้นตอนวิธีเมื่อใช้ฐานข้อมูลการบุกรุกอื่นๆ

1.3 ขอบเขตของงานวิจัย

1) ศึกษาการพัฒนาขั้นตอนการเรียนรู้ตามลำดับชั้นแบบเพิ่มเติมได้สำหรับความปลอดภัยของเครือข่าย บนพื้นฐานของการวิเคราะห์ดิสคริมิแนนต์เชิงเส้นแบบเพิ่มเติมได้ ที่คัดเลือกคุณลักษณะด้วยวิธีสหสัมพันธ์แบบเพียร์สันเท่านั้น

2) ศึกษาการพัฒนาขั้นตอนวิธีการเรียนรู้แบบผสมสำหรับความปลอดภัยของเครือข่าย ที่คัดเลือกคุณลักษณะด้วยวิธีการคัดเลือกคุณลักษณะบนพื้นฐานของความสัมพันธ์เท่านั้น

3) ชุดข้อมูล (Dataset) ที่ใช้ในการทดลองเป็นชุดข้อมูลมาตรฐานที่ผู้พัฒนาได้เผยแพร่ไว้ทั้งหมด

4) วัดผลความแม่นยำของผลลัพธ์ เปรียบเทียบกับวิธีการเรียนรู้ของเครื่องวิธีการของการเรียนรู้แบบมีผู้สอน (Supervised Learning) เท่านั้น

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1) ได้พัฒนาขั้นตอนวิธีใหม่ของการเรียนรู้ตามลำดับชั้นแบบเพิ่มเติมได้สำหรับความปลอดภัยของเครือข่าย บนพื้นฐานของการวิเคราะห์ดิสคริมิแนนต์เชิงเส้นแบบเพิ่มเติมได้

2) ได้พัฒนาขั้นตอนวิธีการเรียนรู้แบบผสมสำหรับความปลอดภัยของเครือข่าย

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ความปลอดภัยของระบบคอมพิวเตอร์ ได้มีการนำเสนอแนวคิดโดย James P. Anderson (Anderson, 1980) โดยนำเสนอถึงปัญหาด้านความปลอดภัยของระบบคอมพิวเตอร์ ในช่วงแรกภัยคุกคามต่อความมั่นคงปลอดภัยของระบบคอมพิวเตอร์ จะเป็นภัยคุกคามที่เกิดทางกายภาพ เช่น การลักขโมยเครื่องคอมพิวเตอร์ อุปกรณ์ต่อพ่วง และข้อมูลเป้าหมายที่ต้องการ เป็นต้น ต่อมาเมื่อมีการใช้งานเทคโนโลยีสารสารเทศและการสื่อสารเพิ่มมากขึ้นจนถึงปัจจุบัน การป้องกันรักษาความปลอดภัยจึงเป็นสิ่งสำคัญต่อการรักษาข้อมูลสารสนเทศที่มีการใช้งานตลอดเวลา เช่น การซื้อสินค้าออนไลน์ การเก็บรักษาฐานข้อมูลองค์กร การแชท (Chat) หรือส่งจดหมายอิเล็กทรอนิกส์ (e-Mail) ไปยังผู้รับในงานประจำวัน ซึ่งการรักษาความปลอดภัยของระบบดังกล่าว จำเป็นต้องมีฮาร์ดแวร์ (Hardware) หรือ ซอฟต์แวร์ (Software) ที่สามารถป้องกันได้ จึงได้มีการพัฒนาตัวแบบของระบบตรวจจับการบุกรุก (Intrusion Detection System: IDS) ที่มีความสามารถในการวิเคราะห์และป้องกันผู้บุกรุก (Intruder) แบบมีประสิทธิภาพสูง และสามารถวิเคราะห์ได้หลากหลายขั้นตอนวิธี จะมีความฉลาดในการจำแนกระหว่าง ข้อมูลที่ปกติและข้อมูลที่ไม่ปกติได้

ในบทนี้ จะอธิบายถึงหลักการพื้นฐานที่เกี่ยวข้องกับงานวิจัยทั้งในส่วนของหลักการด้านความมั่นคงปลอดภัยของสารสนเทศ หลักทฤษฎีของขั้นตอนวิธีที่ใช้ในงานวิจัย การวัดประสิทธิภาพ และงานวิจัยที่เกี่ยวข้อง มีรายละเอียดดังต่อไปนี้

2.1 แนวคิดหลักด้านความมั่นคงปลอดภัย

ความปลอดภัย (Security) หมายถึง คุณภาพ หรือสถานะที่ปลอดภัยจากอันตราย องค์กรที่ประสบความสำเร็จในการดำเนินงาน โดยที่มีการใช้ข้อมูลและสารสนเทศทั้งภายในและภายนอกองค์กร มักมีการวางแผนรักษาความปลอดภัย และป้องกันโอกาสที่อาจเกิดขึ้นได้จากทุกกระบวนการ โดยมีการคำนึงถึงส่วนประกอบด้านความปลอดภัยหลายชั้น ดังต่อไปนี้ (Whitman and Mattord, 2012)

แนวคิดหลักด้านความมั่นคงปลอดภัยของสารสนเทศ เรียกว่าสามเหลี่ยมด้านความปลดภัย หรือ CIA Triad เป็นหลักการรักษาความมั่นคงปลอดภัยของสารสนเทศที่ดี ประกอบไปด้วย 3 ประการ ได้แก่ ความลับ (Confidentiality) ความสมบูรณ์ (Integrity) และ ความพร้อมใช้ (Availability) (Pfleeger and Pfleeger, 2012)

2.1.1 ความลับ (Confidentiality): หมายถึง ความสามารถของระบบ ที่รับประกันได้ว่าสารสนเทศจะดูได้ได้โดยผู้ที่มีสิทธิ์เท่านั้น ดังนั้น ผู้ใช้งานจะสามารถเข้าถึงสารสนเทศที่ถูกป้องกันไว้ได้ ข้อมูลที่เป็นความลับอาจเป็นข้อมูลส่วนตัว เช่น ใบรายงานผลการเรียนของนักศึกษา ประวัติการรักษาพยาบาล หรือ การใช้งานอีเมล์ (e-Mail) ซึ่งข้อมูลเหล่านี้จะต้องเป็นผู้ที่มีสิทธิ์เท่านั้น ถึงจะเข้าถึงได้ เช่น ผู้ป่วยมีประวัติการรักษาพยาบาลบางโรค ผู้ป่วยมีสิทธิ์ไม่เปิดเผยข้อมูลการรักษาความปลอดภัยแก่ผู้อื่น แต่หากผู้ป่วยต้องการแจ้งต่อผู้อื่นเอง นั่นคือสิทธิ์ที่จะบอกได้ ดังนั้นในด้านการรักษาความปลอดภัย จะต้องทำทุกวิถีทางในการป้องกันความลับของข้อมูลดังกล่าวให้ได้อย่างสมบูรณ์ อาจต้องมีการลงทุนซื้ออุปกรณ์ ตั้งค่า หรือพัฒนาระบบ เพื่อให้สามารถรักษาความลับของข้อมูลได้

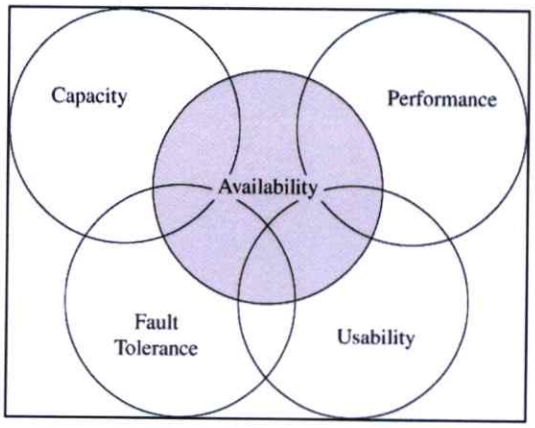
แต่ปัจจุบัน การใช้งานเทคโนโลยี คือการแบ่งปัน (Share) มากกว่าป้องกัน เช่น นักศึกษาใช้สื่อสังคมออนไลน์ (Social Network) ในการนำเสนอข้อความเกรดของตนเอง และตั้งค่าประวัติส่วนตัวโดยบอกเบอร์โทรศัพท์ และเปิดเผยอีเมล์ในข้อมูลส่วนตัว กรณีดังกล่าวอาจเป็นช่องทางให้ผู้ไม่หวังดี นำข้อมูลส่วนตัวไปลองลงชื่อเข้าใช้ได้ Social Network ได้สำเร็จ มีกรณีที่อาจทำให้การรักษาความลับไม่ประสบความสำเร็จ ดังต่อไปนี้ (Pfleeger and Pfleeger, 2012)

- เข้าถึงข้อมูลโดยผู้ที่ไม่มีสิทธิ์
- เข้าถึงข้อมูลโดยกระบวนการหรือโปรแกรมที่ไม่มีสิทธิ์
- ผู้ที่มีสิทธิ์เข้าถึงข้อมูล แต่สามารถเข้าถึงข้อมูลอื่นได้ด้วยที่ตนไม่มีสิทธิ์
- ผู้ที่ไม่มีสิทธิ์เข้าถึงข้อมูลโดยประมาณได้ เช่น ช่วงของข้อมูลเงินเดือน หรือ แผนกของกลุ่มโรคที่รักษา ซึ่งแม้ไม่ทราบข้อมูลโดยตรงรายบุคคล แต่ทราบโดยประมาณ
- ผู้ที่ไม่มีสิทธิ์เรียนรู้โดยประสบการณ์จากบางส่วนของข้อมูล เช่น รู้ข้อมูลเกี่ยวกับการพัฒนาผลิตภัณฑ์ใหม่ จากการควบรวมกิจการ เป็นต้น

นอกจากนี้ ภัยคุกคามอื่นๆ ที่มีปัจจุบัน ก็เป็นสาเหตุให้สารสนเทศถูกเปิดเผยได้ เช่น ถูกขโมยฐานข้อมูลลูกค้าของเว็บไซต์ขายสินค้าออนไลน์ การติดไวรัสคีย์ล็อคเกอร์ (Key Logger) เพื่อดักเก็บรหัสผ่านบนเว็บบราวเซอร์ (Web Browser) ของผู้ใช้งาน ซึ่งอาจส่งรหัสผ่านกลับไปยังผู้โจมตี ทำให้สามารถเข้าถึงข้อมูลส่วนตัวได้ เป็นต้น

2.1.2 ความสมบูรณ์ (Integrity): หมายถึง หมายถึง ความสามารถของระบบ ที่รับประกันได้ว่าสารสนเทศจะถูกแก้ไขได้โดยผู้ที่มีสิทธิ์เท่านั้น มีการกล่าวว่า สารสนเทศที่มีความสมบูรณ์เมื่อสารสนเทศนั้น มีครบ ทั้งหมด สมบูรณ์ และไม่เสียหาย (Whitman and Mattord, 2012) ซึ่งสาเหตุหนึ่งที่ทำให้สารสนเทศขาดความสมบูรณ์คือการถูกโจมตีด้วย ไวรัส (Virus) เวิร์ม (Worm) โทรจัน (Trojan) และ มัลแวร์ (Malware) ที่ทำให้ข้อมูลที่ถูกต้องกลายเป็นข้อมูลที่ถูกแก้ไขผิดไปจากข้อมูลเดิม ซึ่งวิธีการหนึ่งในการยืนยันว่าข้อมูลเป็นข้อมูลที่ถูกแก้ไขหรือไม่นั้นคือ การใช้ฟังก์ชันแฮช (Hash) ซึ่งจะเป็นขั้นตอนวิธีที่เอาไว้ย่อยข้อมูลแล้วจะได้ค่าที่แฮชออกมาจะต้องมีค่าแตกต่างกันกับข้อมูลอื่นอย่างสิ้นเชิง ดังนั้นหากข้อมูลใดๆถูกแก้ไขเปลี่ยนแปลงหรือขาดความสมบูรณ์ เมื่อนำไปคำนวณด้วยฟังก์ชันแฮชจะได้ค่าออกมาที่แตกต่างกับค่าที่แฮชได้กับข้อมูลต้นฉบับ

2.1.3 ความพร้อมใช้ (Availability): หมายถึง ความสามารถของระบบ ที่รับประกันได้ว่าสารสนเทศจะถูกใช้ได้โดยผู้ที่มีสิทธิ์เท่านั้น ซึ่งต้องสามารถใช้ได้ตลอดเวลาและมีความน่าเชื่อถือการเข้าใช้ (Stallings and Brown, 2015) ซึ่งในความพร้อมใช้จะหมายถึงข้อมูลและบริการ (Services) ซึ่งจะต้องใช้งานได้ตลอดเวลา ระบบมีประสิทธิภาพเพียงพอต่อการให้บริการของ Services และ Services จะต้องให้บริการอย่างสมบูรณ์ในคาบเวลาที่ยอมรับได้ (Pfleeger and Pfleeger, 2012) เป้าหมายของความพร้อมใช้ จะประกอบไปด้วยคุณสมบัติบางประการที่ซ้อนทับกัน ได้แก่ สมรรถนะ (Capacity) ประสิทธิภาพที่ดี (ประสิทธิภาพ) ทนต่อความผิดพลาด (Fault Tolerance) และใช้งานได้ (Usability) ทั้งหมดนี้เป็นคุณสมบัติที่ทำให้สารสนเทศสามารถมีความพร้อมใช้ เช่น แสดงดังรูปที่ 2.1



รูปที่ 2.1 คุณสมบัติที่ซ้อนทับกันของความพร้อมใช้

แต่ในปัจจุบัน หลักการของ CIA-Triad นั้นไม่เพียงพอต่อการรักษาความปลอดภัย เนื่องจากการใช้งานสารสนเทศในปัจจุบันมีขอบเขตที่กว้างขึ้น มีกรณีความเสียหายที่เพิ่มมากขึ้น เช่น การทำลาย การขโมย หรือการลุกล้ำสิทธิ์การใช้งานผู้อื่น เป็นต้น (Whitman and Mattord, 2012) จึงมีการขยายหลักการของ CIA-Triad ออกไปเพิ่มขึ้นอีก 2 หัวข้อ (Whitman and Mattord, 2012) ดังนี้

2.1.4 ความถูกต้อง (Accuracy) หมายถึง ผู้ใช้งานจะต้องได้รับสารสนเทศที่ถูกต้องตามที่คาดหวัง นั่นคือปราศจากข้อผิดพลาดทั้งโดยที่ตั้งใจหรือไม่ตั้งใจก็ตาม เช่น เมื่อผู้ใช้งานถอนเงินจากเครื่องรับจ่ายเงินอัตโนมัติ (Automatic Teller Machine: ATM) แล้วพบว่ายอดเงินในบัญชีไม่ตรงกับยอดเงินจริงที่มีอยู่ นั่นอาจหลายถึงสารสนเทศไม่ถูกต้องในขั้นตอนใดๆของการประมวลผล หรืออาจผิดพลาดจากพนักงานในขั้นตอนนำเงินเข้าฝากก็เป็นได้

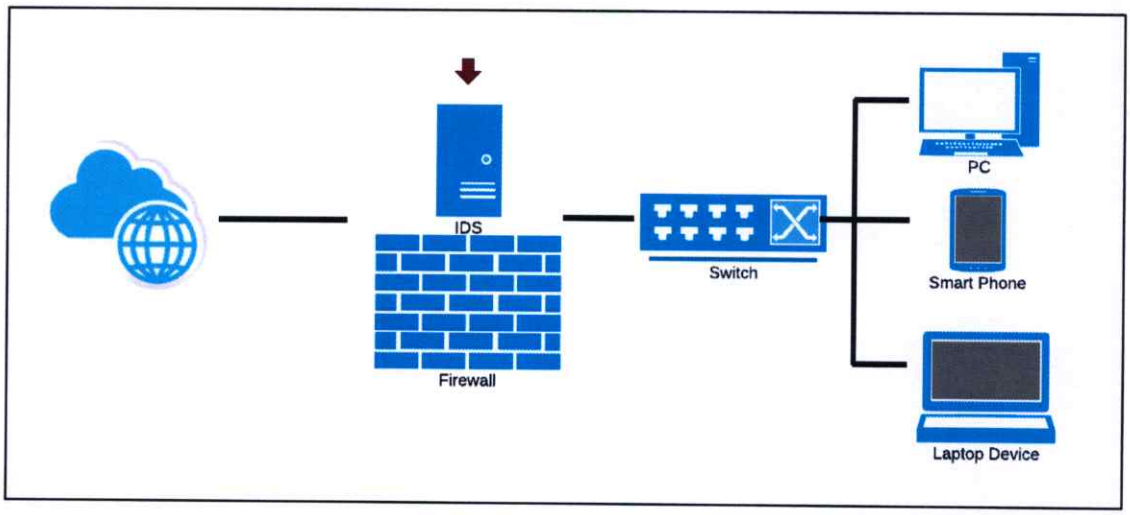
หรือระบบรายงานผลเฉลี่ยบนเว็บไซต์ของมหาวิทยาลัย พบว่ามีการคำนวณเกรดและหน่วยกิตผิดพลาดทำให้แสดงผลค่าเกรดเฉลี่ยรวมผิดพลาด เป็นต้น

2.1.5 ความเป็นของแท้ (Authenticity) หมายถึง ผู้ใช้งานจะต้องได้รับสารสนเทศที่มาจากต้นกำเนิน ไม่ใช้ได้สารสนเทศที่สำเนามาจากแหล่งอื่น หรือประดิษฐ์ขึ้นมาใหม่ ตัวอย่างเช่น การส่งอีเมล์นั้น ข้อมูลที่ส่งจากต้นทางไปยังปลายทาง ต้องเป็นข้อมูลที่ถูกต้องทั้งหมด กรณีที่ขาดความเป็นของแท้ ได้แก่ การทำอีเมล์สวมรอย (e-Mail Spoofing) ซึ่งในการส่งจะปลอมแปลงแหล่งที่มา โดยอาจมีการแนบไฟล์ หรือเขียนที่อยู่ปลายทางหลอกลวงไว้ในข้อความจดหมายแล้วปลอมแหล่งที่มาให้เป็นที่อยู่อีเมล์ที่มีความน่าเชื่อถือ เป็นต้น

2.2 ระบบตรวจจับการบุกรุก

การตรวจจับการบุกรุก (Intrusion Detection) คือ กระบวนการในการติดตามและระบุชนิดของสิ่งผิดปกติในการจราจรเครือข่าย (Traffic) ได้ ดังนั้น ระบบตรวจจับการบุกรุก (Intrusion Detection System: IDS) จึงเป็นเครื่องมือในการตรวจสอบสิ่งผิดปกติทุกอย่างแล้วแจ้งเตือนให้ผู้ดูแลระบบทราบ เช่น สิทธิ์ในการเข้าถึงที่ได้รับอนุญาต (Anther Permission) การเปลี่ยนแปลงค่า รีจิสตรีโดยไม่ได้รับอนุญาต (Unauthorized Registry Change) หรือ การย้ายไฟล์ที่ผิดปกติ (Malicious File Manipulation) เป็นต้น (Rhodes-Ousley, 2013)

ดังนั้น เครือข่ายคอมพิวเตอร์ จึงจำเป็นต้องมีระบบตรวจจับการบุกรุก เพื่อให้สามารถติดตามตรวจจับ และแจ้งเตือนให้กับผู้ดูแลระบบได้ เสมือนกับระบบแจ้งเตือนอัคคีภัย ซึ่งจะแจ้งเตือนเมื่อมีเหตุเพลิงไหม้ เพื่อให้ทราบและป้องกันความเสียหายได้อย่างทันท่วงที แสดงตำแหน่งของระบบตรวจจับการบุกรุก ดังรูปที่ 2.2



รูปที่ 2.2 ตำแหน่งของระบบตรวจจับการบุกรุกในเครือข่าย ในโครงสร้างของเครือข่าย

2.2.1 ประเภทของระบบตรวจจับการบุกรุก

2.2.1.1 ระบบตรวจจับการบุกรุกทางเครือข่าย (Network-based Intrusion Detection)

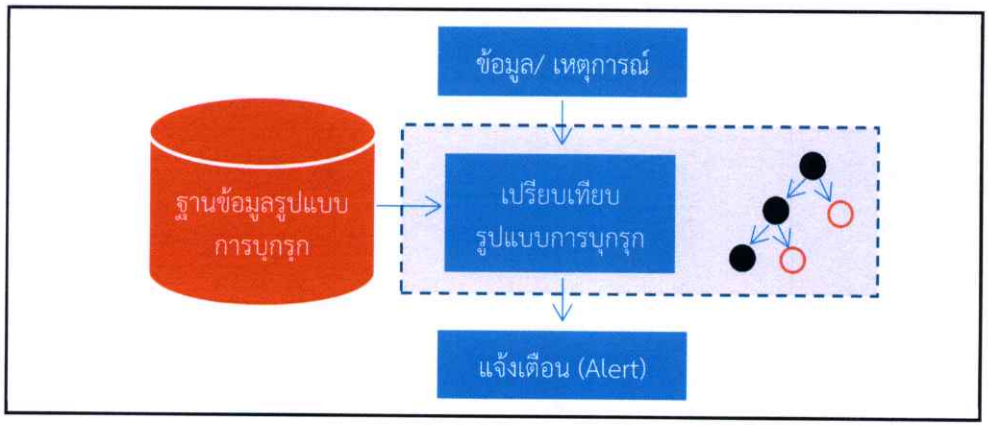
เป็นระบบที่ตรวจจับแพคเก็ต (Packet) ข้อมูลที่รับส่งในระบบเครือข่ายได้ และวิเคราะห์ข้อมูลในแพคเก็ตนั้นๆว่า เป็นรูปแบบของการบุกรุกหรือไม่ หากใช่ก็จะแจ้งเตือนทันที ซึ่งระบบนี้จะสามารถป้องกันเครือข่ายได้หลายเครื่อง เช่น เป็นกลุ่มของเครื่องแม่ข่าย (Server Farm) กลุ่มของเครื่องบริหารลูกข่าย (Server Host) หรืออาจป้องกันเครือข่ายได้ทั้งองค์กรได้

2.2.1.1 ระบบตรวจจับการบุกรุกด้วยเครื่อง (Host-based Intrusion Detection)

เป็นระบบที่ตรวจจับการบุกรุก บนเครื่องคอมพิวเตอร์โฮส (Host Computer) เครื่องเดี่ยวโดยตรง ซึ่งจะต้องติดตั้งระบบตรวจจับการบุกรุกด้วยเครื่อง ไว้ที่เครื่องคอมพิวเตอร์ในแต่ละเครื่อง เพื่อคอยวิเคราะห์ประวัติการใช้งานของระบบ (System Log) เหตุการณ์บนเครื่อง (Event) และประวัติความปลอดภัยของเครื่อง (Security Log) เมื่อมีการตรวจจับเจอไฟล์ใดๆระบบตรวจจับการบุกรุกด้วยเครื่องจะตรวจสอบประวัติการใช้งานที่เกิดขึ้น แล้ววิเคราะห์ว่าเป็นการบุกรุกหรือไม่

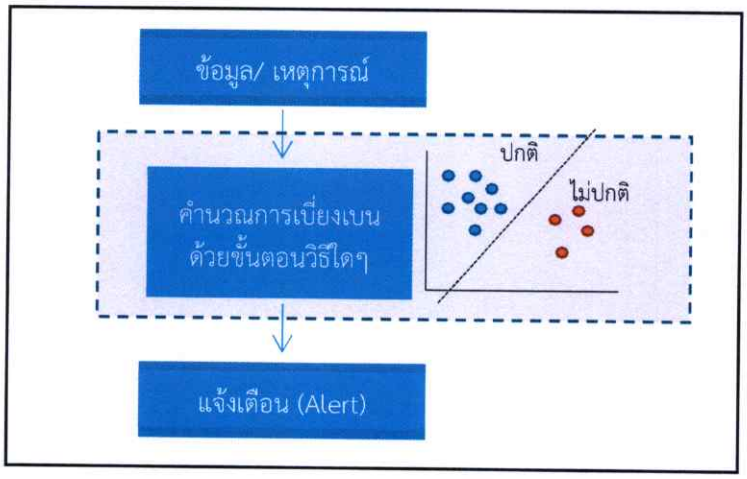
2.2.2 วิธีในการตรวจจับการบุกรุก

2.2.2.1 ระบบตรวจจับการบุกรุกแบบซิกเนเจอร์ (Signature-based IDS) เป็นวิธีการวิเคราะห์ที่ตรวจสอบการใช้งานที่ผิดไปจากรูปแบบที่มีอยู่ เรียกอีกชื่อหนึ่งว่าการตรวจจับโดยใช้กฎ (Rule- based Detection) โดยจะใช้การเปรียบเทียบรูปแบบ (Pattern Matching) เหตุการณ์ที่เกิดขึ้นในระบบ กับฐานข้อมูลของรูปแบบการบุกรุก (Signature) ที่มีอยู่ หากพบว่าเกตุการณ์ดังกล่าวตรงกับรูปแบบของรูปแบบการบุกรุกใดๆในฐานข้อมูล ระบบก็จะตอบสนองแล้วแจ้งเตือนทันที แสดงดังรูปที่ 2.3



รูปที่ 2.3 วิธีการของตรวจจับการบุกรุกแบบซิกเนเจอร์

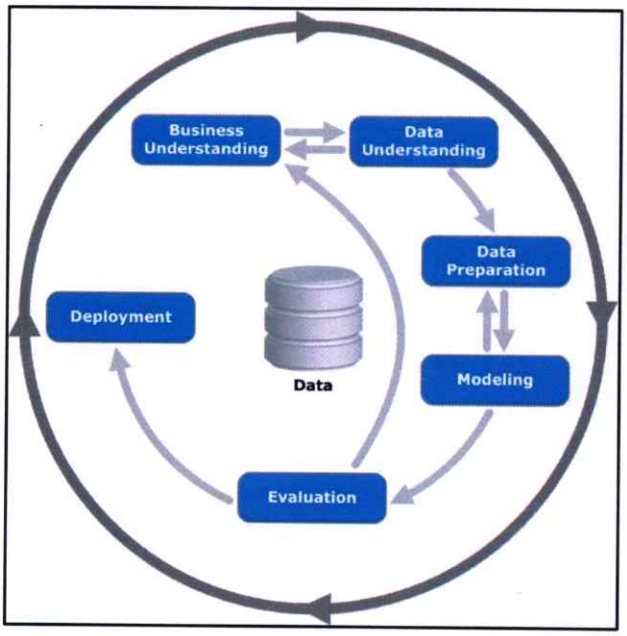
2.2.2.2 ระบบตรวจจับการบุกรุกแบบตรวจความไม่ปกติ (Anomaly-based IDS) เป็นวิธีการตรวจจับและแจ้งเตือนการบุกรุกที่ตั้งอยู่บนสมมติฐานที่ว่า เหตุการณ์ใดๆที่ไม่ปกติ จะมีรูปแบบพฤติกรรมเบี่ยงเบนออกไปจากเหตุการณ์ที่ระบบรู้จำว่าปกติซึ่งระบบตรวจจับการบุกรุกแบบตรวจความไม่ปกติ จะสามารถวิเคราะห์และแจ้งเตือนได้ในช่วงกว้าง นั้นหมายถึงอาจไม่มีฐานข้อมูลซิกเนเจอร์ สำหรับเปรียบเทียบเหตุการณ์ใดๆ แต่ระบบจะมีขั้นตอนวิธีในการคำนวณว่า เหตุการณ์ที่มีนั้นเมื่อวิเคราะห์ค่าออกมาแล้ว ตัดสินสินใจได้ว่าเป็นเหตุการณ์ที่มีรูปแบบเข้าข่ายปกติ หรือเข้าข่ายไม่ปกติ ซึ่งหากพบว่าค่าที่ได้เข้าข่ายไม่ปกติ ระบบจะตรวจจับและแจ้งเตือนทันที แสดงดังรูปที่ 2.4



รูปที่ 2.4 วิธีการของจับการบุกรุกแบบตรวจความไม่ปกติ

2.3 เทคนิคการทำเหมืองข้อมูล (Data Mining Technique)

เทคนิคการทำเหมืองข้อมูลที่มีมาตรฐานในการวิเคราะห์ข้อมูล และใช้กันกว้างขวาง เรียกว่า Cross-Industry Standard Process for Data Mining หรือ CRISP-DM ได้รับการพัฒนาขั้นมาจากผู้เชี่ยวชาญด้านเหมืองข้อมูล (Data Mining) ในปี ค.ศ. 1996 เปรียบเสมือนเป็นพิมพ์เขียว (Blue Print) สำหรับงานด้านเหมืองข้อมูล มีขั้นตอนทั้งหมด 6 ขั้นตอน (Shearer, 2000) แสดงรูปที่ 2.5 ดังต่อไปนี้



รูปที่ 2.5 กระบวนการ CRISP-DM

2.3.1 ทำความเข้าใจในธุรกิจ (Business Understanding)

ขั้นตอนแรกนี้ เป็นขั้นตอนการทำความเข้าใจกับวัตถุประสงค์และความต้องการเพื่อนำไปสู่รูปแบบการแก้ปัญหาทางธุรกิจด้วยเหมืองข้อมูล

2.3.2 ทำความเข้าใจข้อมูล (Data Understanding)

ขั้นตอนนี้เป็นการทำความเข้าใจกับข้อมูล เก็บรวบรวมข้อมูล ศึกษาข้อมูล ระบุปัญหา ประเมินคุณภาพของข้อมูลเชิงลึก และพิจารณาความจำเป็นในการนำข้อมูลไปใช้ว่าใช้ทั้งหมดหรือใช้บางส่วน

2.3.3 การจัดเตรียมข้อมูล (Data Preparation)

ขั้นตอนนี้ เป็นการเตรียมข้อมูลดิบให้เป็นข้อมูลที่สามารถนำไปใช้ในการสร้างแบบจำลอง ในขั้นตอนต่อไปได้ จะต้องมีการทำความสะอาดข้อมูล (Data Cleaning) เช่น การทำให้มูลให้อยู่ในรูปแบบเดียวกัน หรือการทำคัดเลือกคุณลักษณะ เป็นต้น

2.3.4 การสร้างแบบจำลอง (Modeling)

ขั้นตอนนี้จะเป็นการสร้างแบบจำลอง โดยเลือกใช้แบบจำลองที่เหมาะสมจากหลากหลายเทคนิคเพื่อให้ได้คำตอบที่ดีที่สุด ในขั้นตอนนี้อาจมีการย้อนกลับไปทำในขั้นตอน การจัดเตรียมข้อมูลเพื่อจัดการกับข้อมูลบางส่วนให้เหมาะสมกับแต่ละเทคนิคที่ใช้ได้อีกด้วย

2.3.5 การประเมินแบบจำลอง (Evaluation)

ขั้นตอนนี้เป็นการวัดประสิทธิภาพของผลลัพธ์ที่วิเคราะห์ได้ว่ามีความน่าเชื่อถือ และตรงตามวัตถุประสงค์ทางในขั้นตอนแรกหรือไม่ บางครั้งอาจพบว่ามีค่าประสิทธิภาพที่ยังไม่ดีเพียงพอหรือไม่ตรงตามวัตถุประสงค์อาจต้องย้อนไปขั้นตอนก่อนหน้าเพื่อให้ได้ผลลัพธ์ที่ต้องการก่อนนำไปใช้

2.3.6 การนำแบบจำลองไปใช้ (Deployment)

ขั้นตอนนี้เป็นการนำแบบจำลองไปใช้งานจริง เช่น ใช้กับข้อมูลที่ต้องการวิเคราะห์จริง หรือใช้กับระบบออนไลน์จริง เป็นต้น

2.4 ขั้นตอนวิธีที่ใช้ในงานวิจัย

2.4.1 สหสัมพันธ์ของเพียร์สัน

สหสัมพันธ์ของเพียร์สัน คือการวัดค่าความแข็งแรงของความสัมพันธ์ระหว่างตัวแปร X และตัวแปร Y ตามวิธีการแปรปรวนร่วม (Covariance) ของทั้งสองค่าหารด้วยผลคูณของค่าเบี่ยงเบนมาตรฐาน (Lena and Margara, 2010) มีประสิทธิภาพในการหาความสัมพันธ์ของข้อมูลการบุกรุก (Eid, Hassanien, Kim and Banerjee, 2013) แสดงการคำนวณดังสมการที่ 2.1

กำหนดให้ $X = \{x_1, x_2, \dots, x_n\}$ และ $Y = \{y_1, y_2, \dots, y_n\}$

$$r_{XY} = \sum_{i=1}^{n} \frac{(X_i - \bar{X})}{\sqrt{\sum_{i=1}^{n}(X_i - X)^2}} \cdot \frac{Y_i - \bar{Y}}{\sqrt{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}} \quad (2.1)$$

r_{XY} คือค่าสัมประสิทธิ์สหสัมพันธ์ซึ่งมีค่าอยู่ระหว่าง -1 ถึง 1

2.4.2 Sequential Incremental LDA

กำหนดให้ $X = \{x_1, x_2, \dots, x_N\}$ คือเซตของข้อมูลตัวอย่าง ที่มีจำนวน M กลุ่ม และ N คือจำนวนของข้อมูลตัวอย่าง. ให้ y คือข้อมูลใหม่ที่เข้ามา โดยมีชื่อกลุ่มเป็น k, ค่า Eigen space โมเดลใหม่ คือ $\Omega' = (Sw', Sb', \bar{x}, N+1)$ จะถูกปรับปรุงจากค่า Ω เดิม และจากข้อมูลใหม่ yสำหรับการคำนวณค่าเฉลี่ยใหม่ $\bar{x}'$ สามารถคำนวณดังสมการ 2.2 ต่อไปนี้ (Pang, Ozawa and Kasabov, 2005)

$$\bar{x}' = \frac{(N\bar{x}+y)}{(N+1)} \quad (2.2)$$

สำหรับ Between-class Scatter Matrix Sb' ถ้า $k = M+1$ เป็นข้อมูลใหม่ที่เข้ามาในกลุ่มดังนั้นจะคำนวณและปรับปรุงค่า ดังสมการ 2.3-2.4

$$Sb' = \sum_{c=1}^{M} n_c(\overline{x_c} - \bar{x}')(\overline{x_c} - \bar{x}')^T + (y - \bar{x}')(y - \bar{x}')^T \quad (2.3)$$

$$Sb' = \sum_{c=1}^{M+1} n_c'(\overline{x_c} - \bar{x}')(\overline{x_c} - \bar{x}')^T \quad (2.4)$$

เมื่อ n_c' คือจำนวนของข้อมูลตัวอย่างในกลุ่ม C หลังจากมีข้อมูล y ปรากฏ, $\mathrm{n_c}'$=$\mathrm{n_c}$ เมื่อ $1 \le \mathrm{c} \le \mathrm{M}$, $\mathrm{n}_c' = 1$ เมื่อ $\mathrm{c} = \mathrm{M}+1$ และ $\overline{\mathrm{x}_c} = \mathrm{y}$ เมื่อ $\mathrm{c} = \mathrm{M}+1$

เมื่อ $1 \le c \le M$ ดังนั้น Sb' จะปรับปรุงดังสมการ 2.5

$$Sb' = \sum_{c=1}^{M} n_c'(\bar{x}_c - \bar{x}')(\bar{x}_c - \bar{x}')^T \quad (2.5)$$

ในกรณีที่ $\overline{\mathrm{x}_c} = \big(1/(\mathrm{n_c}+1)\big)(\mathrm{n_c}\overline{\mathrm{x}_c} + \mathrm{y})$ และ $\mathrm{n_c}'$=$\mathrm{n_c}+1$ ถ้า y เท่ากับ Class c; ดังนั้น $\overline{\mathrm{x}_c}' = \overline{\mathrm{x}_c}$ และ $\mathrm{n}_c' = \mathrm{n_c}'$

สำหรับ Within-class Scatter Matrix Sw ถ้า y คือกลุ่มใหม่ ซึ่งหมายถึง k คือ $(M+1)$ กลุ่ม ดังนั้น การปรับปรุง Within-class Scatter Matrix จะไม่ต้องเปลี่ยนแปลงใดๆดังสมการ 2.6

$$Sw' = \sum_{c=1}^{M} \sum_c + \sum_k = \sum_{c=1}^{M+1} \sum_c = \sum_{c=1}^{M} \sum_c \quad (2.6)$$

ในกรณีที่ $1 \leq c \leq M$ จะปรับปรุง s_w ในสมการ จะแสดงการ Proof ใน Appendix.

$$Sw' = \sum_{c=1, c \neq k}^{M} \sum_c + \sum_k{}' \quad (2.7)$$

$$\Sigma_k' = \Sigma_k + \frac{n_k}{n_k+1}(y - \bar{x}_k)(y - \bar{x}_k)^T \quad (2.8)$$

ตัวอย่าง

กำหนดให้ ข้อมูลมี 2 กลุ่มดังต่อไปนี้

$$C_1 = [(1,2), (2,3), (3,3), (4,5), (5,5)]$$
$$C_2 = [(1,0), (2,1), (3,1), (3,2), (5,3), (6,5)]$$

หาค่าเฉลี่ย จากสมการ (2) $\bar{x}' = \frac{(N\bar{x}+y)}{(N+1)}$

ค่าที่ 1 (1,2) $\bar{x}' = \frac{(0+(1,2))}{(0+1)} = (1,2)$

ค่าที่ 2 (2,3) $\bar{x}' = \frac{(1*(1,2)+(2,3))}{(1+1)} = \frac{(1,2)+(2,3)}{2} = \frac{(3,5)}{2} = (1.5,2.5)$

ค่าที่ 3 (3,3) $\bar{x}' = \frac{(2*(1.5,2.5)+(3,3))}{(2+1)} = \frac{(3,5)+(3,3)}{3} = \frac{(6,8)}{3} = (2,2.67)$

ค่าที่ 4 (4,5) $\bar{x}' = \frac{(3*(2,2.67)+(4,5))}{(3+1)} = \frac{(6,8)+(4,5)}{4} = \frac{(10,13)}{4} = (2.5,3.25)$

ค่าที่ 5 (5,5) $\bar{x}' = \frac{(4*(2.5,3.25)+(5,5))}{(4+1)} = \frac{(10,13)+(5,5)}{5} = \frac{(15,18)}{5} = (3,3.6)$

จากตัวอย่างการคำนวณค่าเฉลี่ย (Mean) แบบ Incremental Learning จะพบว่า จะใช้ค่า Mean จากขั้นตอนก่อนหน้ามาคำนวณต่อได้ทันที เช่น คำนวณค่าที่ 5 ก็นำค่า Mean จากค่าที่ 4 มาใช้คำนวณต่อได้ทันที ไม่จำเป็นต้องเก็บข้อมูลไว้ตั้งแต่ค่าแรกเพื่อมารอหารเฉลี่ยถึงค่าสุดท้าย ดังนั้น ค่า Mean ที่ได้จากการคำนวณ $\bar{x}_{c1}$ เป็น (3,3.6) และคำนวณเช่นเดียวกันกับค่า C_2 ซึ่งได้ค่า$\bar{x}_{c2}$ เป็น (3.33,2) และค่า Mean รวมทั้งหมด $\bar{x}'$ เป็น (3.18, 2.72)

หลังจากได้ค่า Mean จากนั้นจะคำนวณค่า Sb' จากสมการ (2.9)

$$Sb' = \sum_{c=1}^{M} n_c{}' (\bar{x}_c - \bar{x}')(\bar{x}_c - \bar{x}')^T \quad (2.9)$$

ซึ่งในการคำนวณ จะแยกคำนวณเป็น 2 กลุ่ม ได้แก่ Sb′ ของกลุ่มที่ 1 และ Sb′ ของกลุ่มที่ 2 แล้วจึงนำผลลัพธ์มารวมกัน โดย

$$\mathrm{Sb'}_{c1} = 5 \times [(3,3.6) - (3.18, 2.72)] * [(3,3.6) - (3.18, 2.72)]^{\mathrm{T}} = 5 \times \begin{bmatrix} 0.0324 & -0.1584 \\ -0.1584 & 0.7744 \end{bmatrix}$$

$$\mathrm{Sb'}_{c1} = \begin{bmatrix} 0.1620 & -0.7920 \\ -0.7920 & 3.8720 \end{bmatrix}$$

$$Sb'_{c2} = 6 \times [(3.33,2) - (3.18, 2.72)] * [(3.33,2) - (3.18, 2.72)]^{T}$$
$$= 6 \times \begin{bmatrix} 0.0229 & -0.1101 \\ -0.1101 & 0.5289 \end{bmatrix}$$

$$Sb'_{c2} = \begin{bmatrix} 0.1350 & -0.6480 \\ -0.6480 & 3.1104 \end{bmatrix}$$

ดังนั้น

$$\mathrm{Sb'} = \begin{bmatrix} 0.1620 & -0.7920 \\ -0.7920 & 3.8720 \end{bmatrix} + \begin{bmatrix} 0.1350 & -0.6480 \\ -0.6480 & 3.1104 \end{bmatrix} = \begin{bmatrix} \mathbf{0.2970} & \mathbf{-1.4400} \\ \mathbf{-1.4400} & \mathbf{6.9824} \end{bmatrix}$$

จากนั้น หาค่า Sw' จากสมการ (2.7) จะมีการคำนวณและปรับปรุงค่า Σ'_k

$$\Sigma'_k = \Sigma_k + \frac{n_k}{n_k + 1}(y - \bar{x}_k)(y - \bar{x}_k)^T$$

รอบที่ 1 = $\left(\frac{0}{0+1}\right) * \left[((1,2) - (0,0))' * ((1,2) - (0,0))\right] = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$

รอบที่ 2 = $\left(\frac{1}{1+1}\right) * \left[((2,3) - (1,2))' * ((2,3) - (1,2))\right] = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$

รอบที่ 3 = $\left(\frac{2}{2+1}\right) * \left[((3,3) - (1.5,2.5))' * ((3,3) - (1.5,2.5))\right] = \begin{bmatrix} 1.5 & 0.5 \\ 0.5 & 0.16667 \end{bmatrix}$

ดังนั้น รอบที่ 3 จะมีค่า $\begin{bmatrix} 01.5 & 0.5 \\ 0.5 & 0.16667 \end{bmatrix} + \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 1 & 0.67 \end{bmatrix}$

รอบที่ 4 = $\left(\frac{3}{3+1}\right) * \left[((4,5) - (2,2.667))' * ((4,5) - (2,2.667))\right] = \begin{bmatrix} 3 & 3.4995 \\ 3.4995 & 4.0822 \end{bmatrix}$

ดังนั้น รอบที่ 3 จะมีค่า $\begin{bmatrix} 3 & 3.4995 \\ 3.4995 & 4.0822 \end{bmatrix} + \begin{bmatrix} 2 & 1 \\ 1 & 0.67 \end{bmatrix} = \begin{bmatrix} 5 & 4.5 \\ 4.5 & 4.75 \end{bmatrix}$

รอบที่ 5 = $\left(\frac{4}{4+1}\right) * \left[((5,5) - (2.5,3.25))' * ((5,5) - (2.5,3.25))\right] = \begin{bmatrix} 3 & 3.5 \\ 3.5 & 2.45 \end{bmatrix}$

ดังนั้น รอบที่ 4 จะมีค่า $\begin{bmatrix} 3 & 3.5 \\ 3.5 & 2.45 \end{bmatrix} + \begin{bmatrix} 5 & 4.5 \\ 4.5 & 4.75 \end{bmatrix} = \begin{bmatrix} 10 & 8 \\ 8 & 7.2 \end{bmatrix}$

ดังนั้น จะมีค่า Sw' ของกลุ่ม C1 เป็น $\begin{bmatrix} 10 & 8 \\ 8 & 7.2 \end{bmatrix}$ ทำเช่นนี้กับกลุ่ม C2 เช่นกัน จะได้ผลลัพธ์ค่า

Sw' ของกลุ่ม C2 เป็น $\begin{bmatrix} 17.33 & 16 \\ 16 & 16 \end{bmatrix}$

เมื่อคำนวณค่า Sw' ทั้งหมด จะมีค่า $Sw' = \begin{bmatrix} 10 & 8 \\ 8 & 7.2 \end{bmatrix} + \begin{bmatrix} 17.33 & 16 \\ 16 & 16 \end{bmatrix} = \begin{bmatrix} \mathbf{27.33} & \mathbf{24} \\ \mathbf{24} & \mathbf{23.20} \end{bmatrix}$

ดังนั้น จะมีค่า $Sw'^{-1} = \begin{bmatrix} 0.399 & -0.4128 \\ -0.4128 & 0.47 \end{bmatrix}$

จากนั้น หาค่า Weight (w) ดังนี้

$$w = Sw'^{-1} * \mathrm{Sb}'$$

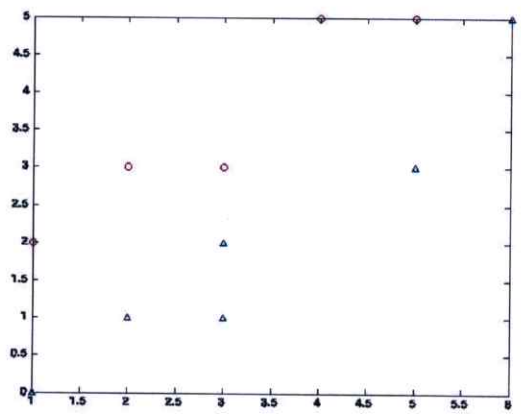
$$w = \begin{bmatrix} 0.399 & -0.4128 \\ -0.4128 & 0.47 \end{bmatrix} * \begin{bmatrix} 0.2970 & -1.4400 \\ -1.4400 & 6.9824 \end{bmatrix}$$

$$w = \begin{bmatrix} 0.7214 & -3.4628 \\ -0.8090 & 3.8832 \end{bmatrix}$$

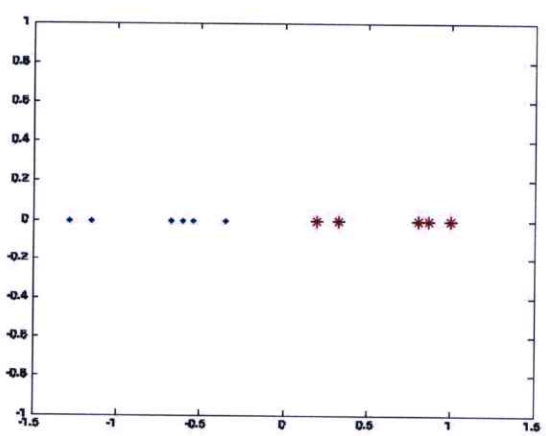
จากนั้นจึงหาค่า Max Eigenvalue ได้เป็น

Eigen vector = $\begin{bmatrix} -0.9789 & 0.6655 \\ -0.2039 & -0.7463 \end{bmatrix}$ และ Eigen value = $\begin{bmatrix} 0 & 0 \\ 0 & 4.6047 \end{bmatrix}$

เมื่อได้ค่า Eigen จึงสามารถคำนวณจุดของข้อมูลใหม่ได้ เปรียบเทียบจุดของข้อมูลเดิมดังรูปที่ 2.6 และแสดงจุดของข้อมูลใหม่ดังรูปที่ 2.7



รูปที่ 2.6 จุดของข้อมูลเดิมก่อนการคำนวณ



รูปที่ 2.7 จุดของข้อมูลใหม่หลังการคำนวณ

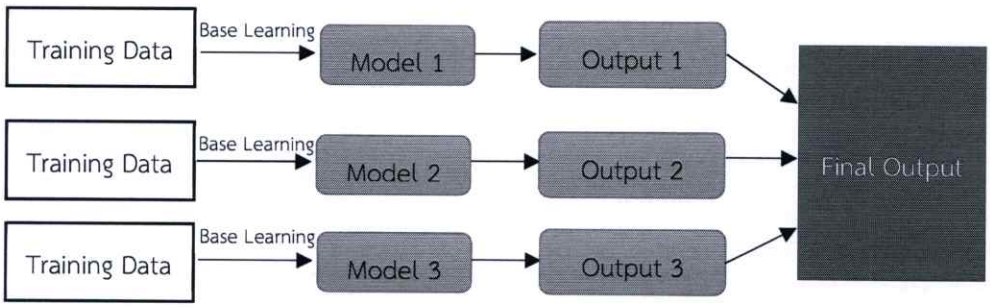
2.4.3 การหาระยะทางแบบมาหาลาโนบิส (Mahalanobis Distance)

การหาระยะทางมาหาลาโนบิส คือ วัดระยะห่างระหว่างสองจุดในพื้นที่หลายตัวแปร เป็นวิธีการที่น่าสนใจและถูกนำมาใช้เป็นจำนวนมาก การคำนวณจะกำหนดให้ d คือ ระยะทางมาหาลาโนบิส x คือข้อมูลตัวอย่างใดๆที่ต้องการวัดระยะห่าง และ μ คือค่าเฉลี่ยนของกลุ่มข้อมูล ตัวอย่าง S คือค่าโควาเรียนซ์ (Covariance) (De Maesschalck, 2000) แสดงการคำนวณดังดังสมการ (2.10)

$$d(x,\mu) = \sqrt{(x-\mu)^T S^{-1}(x-\mu)} \tag{2.10}$$

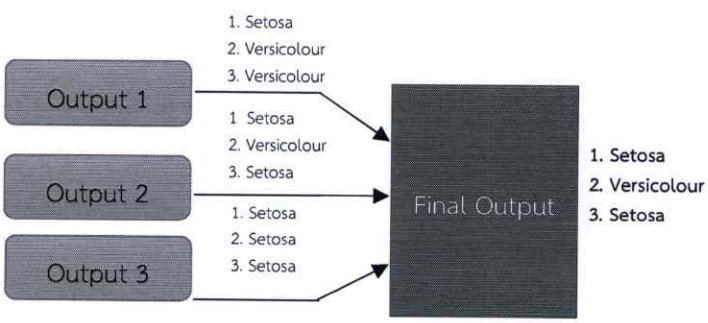
2.4.4 การเรียนรู้แบบผสม

การเรียนรู้แบบผสมเป็นวิธีการที่ใช้ Base Learning หรือ Weak Learner มาคำนวณโมเดลหลายๆโมเดลแล้วจึงให้เสียงข้างมากในการทายคำตอบ (Dietterich and Thomas, 2002) ตัวอย่างการสร้างโมเดล แสดงดังรูปที่ 2.8



รูปที่ 2.8 วิธีการของการเรียนรู้แบบผสม

ตัวอย่าง เช่น ฝึกสอน (Training) ชุดข้อมูล Iris ซึ่งมีการใช้ Base Learner เป็น Naïve Bayes กรณีดังภาพ จะใช้ Naïve Bayes ในการฝึกสอนชุดข้อมูล Fisher iris จำนวน 10 โมเดล ซึ่งในแต่ละโมเดลจะได้คำตอบออกมา อาจทายคำตอบคล้ายกันบ้าง หรือแตกต่างกันบ้าง ซึ่งในท้ายที่สุดก็จะให้เสียงข้างมากเพื่อให้ได้คำตอบเดียวเป็นคำตอบสุดท้าย (Final Output) จากการคำนวณทั้งหมด แสดงตัวอย่างการให้เสียงข้างมาก 3 ระเบียน (Record) ดังรูปที่ 2.7



รูปที่ 2.9 การให้เสียงข้างมากของการเรียนรู้แบบผสม

จากรูปที่ 2.7 พบว่าผลลัพท์ ทั้ง 3 คำตอบ มีการทายที่แตกต่างกัน

- คำตอบสุดท้ายของข้อมูลที่ 1 คือ Setosa นั่นมาจากการให้เสียงข้างมากจากผลลัพท์ ที่ 1,2 และ 3
- คำตอบสุดท้ายของข้อมูลที่ 2 คือ Versicolour นั่นมาจากการให้เสียงข้างมากจากผลลัพท์ ที่ 1 และ 2
- คำตอบสุดท้ายของข้อมูลที่ 3 คือ Setosa นั่นมาจากการให้เสียงข้างมากจากผลลัพท์ ที่ 2 และ 3

ความแม่นยำในการการจำแนกด้วยการเรียนรู้แบบผสมนั้น ขึ้นอยู่กับ Base Learner และจำนวนโมเดลที่กำหนด หากกำหนดให้มีโมเดลเป็น 10 นั่นคือ จะมีจำนวน 10 ผลลัพท์ แล้วจึงนำทั้ง 10 ผลลัพท์มาให้เสียงข้างมากเพื่อให้ได้เพียงคำตอบสุดท้ายคำตอบเดียวเท่านั้น ดังนั้น เสมือนมีการทายหลายคำตอบที่รอการคำนวณผลสรุปจากการให้เสียงข้างมาก และหาก Base Learner มีความแม่นยำในการจำแนกยิ่งทำให้คำตอบสุดท้ายมีความแม่นยำสูง ในทางตรงกันข้าม หาก Base Learner ไม่มีความแม่นยำในการจำแนก แม้มีจำนวนโมเดลมาก นั่นไม่ได้หมายถึงว่าจะมีประสิทธิภาพในการทำการจำแนกที่ดี การนำผลลัพท์ ที่ไม่แม่นยำมาให้เสียงข้างมากเข้าด้วยกัน จึงมักได้คำตอบสุดท้ายจากการให้เสียงข้างมากที่ไม่แม่นยำนักไปในทางเดียวกัน

แม้มีการใช้การเรียนรู้แบบผสม ในปัจจุบัน เนื่องจากมีประสิทธิภาพสูง แต่มีข้อเสียในด้านการใช้เวลาในการประมวลผลมาก เพราะต้องอาศัยการฝึกสอนจากหลายโมเดล (Ren, 2014; Kumar and Selvakumar, 2013; Kuncheva nd Rodríguez, 2014) เพื่อให้ได้คำตอบสุดท้ายจึงอาจเหมาะกับการประยุกต์ใช้กับบางงานที่ไม่มีข้อจำกัดเรื่องเวลา

2.4.5 Adaboost.m1

Adaboost.m1 คือ การเรียนรู้แบบผสมแบบหนึ่ง ซึ่งพัฒนามาจาก Adaboost แบบดั้งเดิมที่ทำการจำแนกได้เพียงสองกลุ่ม โดย Adaboost.m1 จะสามารถทำการ การจำแนก ได้แบบหลายกลุ่ม และยังมีการคำนวณค่าน้ำหนัก (Weight) ที่แตกต่างจากเดิม (Galar et al., 2014) Adaboost.m1 จะมีการคำนวณค่าผิดพลาด (Error) จากคำตอบสุดท้ายของ Weak Learner ทุกรอบของการคำนวณ หากมีค่ามากกว่า 0.5 จะย้อนกลับไปให้ Weak Learner ทำการฝึกสอนใหม่ จนกว่าจะได้ค่าความผิดพลาด น้อยกว่า 0.5 จึงจะปรับปรุงค่า Distribution แล้วจึงให้เสียงข้างมากได้ในที่สุด

กำหนดให้

ข้อมูลจำนวน m ข้อมูล $X = \{(x_1, y_1), \dots, (x_m, y_m)\}$

ε คือ ค่าความผิดพลาด

β คือ ค่าคงที่

โดยที่ y_i คือ ชื่อกลุ่ม $y_i \epsilon Y = \{1, \dots, k\}$

T คือ จำนวนรอบของการวนซ้ำ

D คือ ค่า Distribution

ซึ่งเมื่อทำการฝึกสอน ด้วย Weal Learner แล้ว จะต้องคำนวณค่าความผิดพลาดทุกครั้ง แล้วจึงปรับปรุงค่า Distribution เพื่อ ให้เสียงข้างมากหาคำตอบ ดังสมการที่ 2.11-2.14

$$\varepsilon_t = \sum_{i:h_t(x)\neq y} D_t(i) \quad (2.11)$$

จากนั้นคำนวณค่า β

$$\beta_t = \frac{\varepsilon_t}{1-\varepsilon_t} \quad (2.12)$$

จากนั้น ปรับปรุงค่า Distribution

$$D_{t+1}(i) = \frac{D_t(i)}{Z_i} \times \begin{cases} \beta_t \; if \; h_t(x_i) = y_i \\ 1 \; otherwise \end{cases} \quad (2.13)$$

ทำการให้เสียงข้างมากเพื่อหา Final Hypothesis

$$H(x) = \underset{y\in C}{argmax} \sum_{t=1}^{T} \log(\frac{1}{\beta_t}) [h_t(x) = y] \quad (2.14)$$

2.4.6 วิธีการคัดเลือกคุณลักษณะบนพื้นฐานของความสัมพันธ์

การทำคัดเลือกคุณลักษณะแบบวิธีการคัดเลือกคุณลักษณะบนพื้นฐานของความสัมพันธ์เป็นวิธีการที่นิยมในการใช้เพื่อเลือกคุณลักษณะที่มีความสำคัญในการคำนวณจากจำนวนคุณลักษณะทั้งหมดเพื่อเป็นการเพิ่มประสิทธิภาพในการทายคำตอบและลดระยะเวลาในการประมวลผลมีการคำนวณดังสมการ 2.15 (Hall, 1999)

$Merit_s$ คือค่า ความสัมพันธ์ระหว่างคุณลักษณะ

k คือจำนวนขององค์ประกอบ

$\bar{r}_{cf}$ คือค่าเฉลี่ยของความสัมพันธ์กลุ่มของคุณลักษณะ $(f \epsilon S)$

$\bar{r}_{ff}$ is คือค่าเฉลี่ยของค่า Inter-Correlation Between Components

$$Merit_s = \frac{k\bar{r}_{cf}}{\sqrt{k+k(k-1)\bar{r}_{ff}}} \quad (2.15)$$

2.4.7 วิธีการที่ใช้เปรียบเทียบประสิทธิภาพ

2.4.7.1 Naïve Bayes

Naïve Bayes เป็นวิธีการหาค่าความน่าจะเป็นโดยทฤษฎีของ Bayes' Theorem (Jensen, 1996) เป็นวิธีที่ได้รับความนิยมในการจำแนกกับข้อมูลที่ไม่รู้จักมาก่อน ซึ่งมีสมการในการคำนวณค่าความน่าจะเป็น (Probability) ดังสมการ 2.16 ดังต่อไปนี้

C_i คือ กลุ่มใดๆ

X คือ ข้อมูลที่ต้องการจำแนก

$$P(C_i|X) = \frac{P(C_i)\prod_{k=1}^{n} P(X_k|C_i)}{P(X)} \quad (2.16)$$

หลังจากคำนวณค่าความน่าจะเป็นจากสมการนี้แล้ว จะเลือกค่าความน่าจะเป็นที่สูงที่สุดเป็นคำตอบ

2.4.7.2 *k*-NN

k-NN เป็นวิธีการแบ่งกลุ่มข้อมูลโดยวัดระยะห่างระหว่าง ระหว่างข้อมูลที่ต้องการทำนายกับข้อมูลที่อยู่ใกล้เคียงจำนวน k ข้อมูล คำตอบของการทำนาย คือคำตอบที่ได้จำนวนกลุ่ม มากที่สุด โดยทั่วไปใช้การหาระยะทางแบบยูคลิด (Eucledian Distance) ระหว่างจุดสองจุดในการคำนวณ (Galit, Nitin and Peter, 2010) มีสมการ 2.17 ดังต่อไปนี้

$$d(\mathbf{x},\mathbf{y}) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots .. + (x_p - y_p)^2} \qquad (2.17)$$

2.4.7.3 Decision Tree

Decision Tree มีหลักการจำแนก คล้ายกับโครงสร้างต้นไม้ตัดสินใจ มีการใช้งานอย่างแพร่หลาย เนื่องจากประสิทธิภาพสูงทำการทำนาย และสามารถแปลความหมายให้เข้าใจได้โดยง่าย ซึ่งการสร้างโมเดลของ Decision Tree จะคัดเลือกคุณลักษณะที่มีความสัมพันธ์กับกลุ่มมากที่สุด มาเป็นโหนดบนสุด (Root Node) ของต้นไม้จากนั้นก็จะหาคุณลักษณะอื่นถัดไปเรื่อยๆตามลำดับ (Quinlan, 1996) มีสมการคำนวณ 2.18-2.20 ดังต่อไปนี้

$\mathbf{P_i}$ คือค่าความน่าจะเป็นของ Field D กับกลุ่ม $\mathbf{C_i}$

หาค่า Entropy จากสมการดังต่อไปนี้

$$Info(D) = \sum_{i=1}^{m} P_i \, log_2(P_i) \qquad (2.18)$$

$$Info_A(D) = \sum_{i=1}^{v} \frac{D_j}{D} * ID_j \qquad (2.19)$$

ค่า Information Gain สำหรับ Field A เป็น

$$Gain_A = Info_D - Info_A D \qquad (2.20)$$

2.4.7.4 Multilayer Perceptron (MLP)

Multilayer Perceptron (MLP) เป็นวิธีการหนึ่งของ Neural Network ที่มีการ Map เชิงเส้นจากอินพุตสเปซ (Input Space) ไปยังชั้นฮิดเดนสเปซ (Hidden Space) แล้วส่งออกไปยังพื้นที่ส่งออก (Output Space) ซึ่งชั้นสุดท้ายหรือ Output จะเป็นชั้นที่ทำนายกลุ่มของข้อมูล โดย MLP สามารถเรียนรู้และย้อนไปกลับเพื่อฝึกสอนได้ เสมือนเลียนแบบการทำงานของสมองมนุษย์ ซึ่งเป็นขั้นตอนวิธี ที่มีความแม่นยำในการทำนาย มีการคำนวณดังสมการ 2.21-2.25 (Simon, 1998)

η คือ ค่าอัตราการเรียนรู้ (Learning Rate)

x_i คือข้อมูลนำเข้า (Input)

y_k คือข้อมูลนำเข้า (Output)

w_{ij} คือ น้ำหนัก (Weight)

θ_j คือค่า Bias

δ_k คือ Error Gradient

กำหนดค่านำหนักและอัตราการเรียนรู้

$$y_k = \sum_{i=1}^{p} w_{jk} x_i + \theta_j \quad (2.21)$$

สำหรับผลลัพท์โหนด k จะคำนวณค่า Error Gradient δ_k

$$\delta_k = (y_{target} - y_k) y_k (1 - y_k) \quad (2.22)$$

สำหรับ Hidden Node j คำนวณ ค่า Error Gradient δ_j

$$\delta_j = o_j (1 - o_j) \sum_k w_{jk} \delta_k \quad (2.23)$$

หาค่า Weight จาก

$$w_{ji} = w_{ji} + \Delta w_{ji} \quad (2.24)$$

โดยที่

$$\Delta w_{ji} = \eta \delta_i o_j \quad (2.25)$$

2.4.7.5 Support Vector Machine (SVM)

SVM เป็นวิธีการจำแนกประเภทที่มีการใช้ระนาบ (Hyperplane) ที่เหมาะสมที่สุดในการจำแนกข้อมูลที่มีมิติจำนวนมากได้ (Brereton, 2010) โดยวิธีการจำแนกเชิงเส้น (Linear Classifier) เป็นวิธีการที่ได้รับการยอมรับอย่างกว้างขวางและประยุกต์ใช้กับงานหลากหลายด้าน มีสมการคำนวณดังสมการ 2.26-2.28

x คือ ข้อมูล

w คือ ค่าน้ำหนักของเส้น

b คือ ค่าคงที่

ในกรณีที่ x คือ ข้อมูลของกลุ่ม +1

$$w^T x + b \geq +1 \quad (2.26)$$

ในกรณีที่ x คือ ข้อมูลของกลุ่ม -1

$$w^T x + b \leq -1 \quad (2.27)$$

เมื่อต้องการหาค่า Hyperplane ที่เหมาะสม ที่มีค่า Margin กว้างที่สุดจะใช้ฟังก์ชันการตัดสินใจ

$$f(x) = sign(w^t x + b) \quad (2.28)$$

2.5 งานวิจัยที่เกี่ยวข้อง

นักวิจัยได้มีการศึกษาการเรียนรู้ของเครื่องที่ใช้ในการตรวจจับการบุกรุกหลายขั้นตอนวิธี ได้แก่ การเรียนรู้แบบไม่มีผู้สอน (Unsupervised Learning) (Syarif, Prugel-Bennett and Wills, 2012: Kumar and Chacko, 2016; Bohara, Thakore and Sanders, 2016) จะเป็นการจัดกลุ่ม (Cluster) โดยไม่มีการระบุผล (Target) ไว้ก่อน และการเรียนรู้แบบมีผู้สอน ซึ่งจะทำการฝึกสอนข้อมูล เพื่อสร้างโมเดลในการทำนายก่อน แล้วจึงทำนายข้อมูลใหม่ในขั้นตอนทดสอบ (Testing) รวมไปถึงแบบการเรียนรู้แบบกึ่งผู้สอน (Semi-supervised learning) (Ashfaq, Wang, Huang, Abbas

and He, 2017), (Gao, Huang, Gao, Shen and Zhang, 2015), (Wurzenberger et al., 2017: Xue, Shang and Feng, 2010: Yuan, Kaklamanos and Hogrefe, 2016) คืออีกประเภทหนึ่งซึ่งจะเกี่ยวข้องกับฟังก์ชันที่ใช้ทำนายและจัดกลุ่ม โดยที่ทั้งทราบชื่อกลุ่ม และไม่ทราบชื่อกลุ่ม จัดเป็นขั้นตอนวิธีที่อยู่ระหว่างการเรียนรู้แบบไม่มีผู้สอน และการเรียนรู้แบบมีผู้สอนซึ่งเป็นที่นิยมในการนำมาใช้งาน นอกจากนี้ยังมีการเรียนรู้แบบผสม (Jabbar, Aluvalu and Reddy, 2017; Kumar and Selvakumar, 2013; Miller and Busby-Earle, 2017) ใช้โมเดลในการจำแนกหลายๆ โมเดลเพื่อมาใช้ในการให้เสียงข้างมากทำนายคำตอบ เป็นวิธีการที่มีประสิทธิภาพสูง เนื่องจากมีการให้ค่าน้ำหนัก (Weight) กับคำตอบแล้วจึงนำไป ให้เสียงข้างมาก โอกาสตอบถูกจึงสูงมากหากมี Weak Learner ที่แม่นยำเหมาะสมกับข้อมูล แต่ในการใช้งานจริงบนเครือข่าย จะเป็นการเพิ่มภาระการประมวลผลที่ต้องใช้ความเร็วในการตรวจจับการบุกรุก

หลากหลายวิธีดังที่กล่าวมา พยายามจะเพิ่มประสิทธิภาพในการตรวจจับคือ เพิ่มความถูกต้อง และลดอัตราการแจ้งเตือนที่ผิดพลาด (False Alarm Rate) ซึ่งมีหลายงานวิจัยจากหลากหลายด้าน มีการใช้การทำงานร่วมกันในแบบผสม ซึ่งเสมือนเป็นการดึงความสามารถของหลายๆวิธีการ มาทำงานร่วมกันเพื่อให้มีประสิทธิภาพในการตรวจจับสูงสุด พบว่ามีงานวิจัย (Datti and Verma, 2010) ได้ใช้ LDA ในการทำการลดจำนวนคุณลักษณะ กับชุดข้อมูล NDL-KDD จากนั้นทำการจำแนกด้วย Neural Network ซึ่งพบว่าสามารถลดจำนวนคุณลักษณะลงได้ ทำให้มีเวลาในการฝึกสอนต่ำ และมีประสิทธิภาพสูงในการจำแนกทุกประเภทของการบุกรุก เช่นเดียวกับ งานวิจัย (Aburomman and Reaz, 2016) ได้มีการใช้ PCA และ LDA โดยการหาคู่ของกลุ่ม (Class-Pair) ซึ่งจะใช้ทั้ง PCA และ LDA หาค่าของคุณลักษณะที่ดีที่สุดจากทั้งสองวิธี จากนั้นนำคุณลักษณะที่ได้มาจำแนกด้วย SVM ผลการทดลองพบว่า สามารถปรับปรุงประสิทธิภาพได้ นอกจากนี้ ยังมีแนวคิดเพิ่มประสิทธิภาพของ LDA เอง ดังงานวิจัย (Saad, Khalid and Mohamed, 2015) ได้พัฒนา Direct LDA โดยการปรับปรุงการคำนวณค่า S_b และ S_w ซึ่งช่วยเพิ่มประสิทธิภาพของอัตราการตรวจจับ (Detection Rate) และลดอัตราการแจ้งเตือนที่ผิดพลาดลง. ไม่เพียงเท่านี้ LDA ยังใช้ป้องกันการโจมตีแบบ DoS และ Black Hole Attack ของการสื่อสารแบบ Self-Driving and Semi Self-Driving Vehicles ในเครือข่าย VANETs (Alheeti, Gruebler and McDonald-Maier, 2017) ทั้งยังประสิทธิภาพดีกว่า QDA เรียกได้ว่าการทำการคัดเลือกคุณลักษณะกับชุดข้อมูลการบุกรุก ซึ่งช่วยเพิ่มประสิทธิภาพของการจำแนกได้เป็นอย่างดี (Sathya, Ramani and Sivaselvi, 2011), (Datti and Lakhina, 2012) ซึ่งนอกจากจะช่วยเพิ่มประสิทธิภาพแล้ว ยังเป็นการลดเวลาในการประมวลผลอีกด้วย เพราะในการใช้งานจริงอาจมีข้อมูลเครือข่ายปริมาณมากส่งผ่านระบบตรวจจับการบุกรุก จึงจำเป็นต้องเลือกเพียงบางคุณลักษณะที่สำคัญและสามารถตรวจจับได้ทันเวลาเมื่อเกิดการบุกรุกระบบขึ้น

ปัจจุบัน ขั้นตอนวิธีที่การที่เหมาะแก่การตรวจจับการบุกรุก ที่อาจใกล้เคียงสภาพการใช้งานจริงมากขึ้น คือการเรียนรู้แบบเพิ่มขึ้น หรือ Incremental Learning เพราะสามารถเรียนรู้จากข้อมูลขนาดใหญ่ที่เป็นแบบ Dynamic Stream แล้วสร้างฐานความรู้เพื่อประโยชน์ในการเรียนรู้ (Learning) แล้วสามารถตัดสินใจได้ต่อไป (He, Chen, Li and Xu, 2011) หากเป็นการตรวจจับการบุกรุกบนการจราจรเครือข่ายที่ผ่านเข้าออก จำเป็นต้องเกิดการคำนวณทางสถิติที่เปลี่ยนแปลงไปตามเวลาอย่างต่อเนื่อง เพื่อทำการตัดสินใจ ณ เวลาใดๆ จำเป็นจะต้องวิเคราะห์และตรวจจับได้ว่า เป็นการบุกรุกหรือไม่ (Bhosale and Ade, 2015) ซึ่งงานวิจัย (Jin, Ding and Huang, 2010) ก็ได้นำเสนอ

Weight ILDA (WILDA) เพื่อนำมาใช้กับระบบ Online Hand-Written Chinese Character Recognition ซึ่ง WILDA ได้เป็นวิธีที่คำนึงถึงการเรียนรู้ที่เพิ่มขึ้นของข้อมูล โดยมีขั้นตอนการคำนวณ Weight ของ S_w และ S_b ซึ่งจะลดปัญหาเรื่องการเรียนรู้ข้อมูลที่เข้ามาใหม่ ซึ่งจะทำให้มีความถูกต้องต่ำลงได้ ซในการทดลองพบว่า WILDA สามารถแก้ปัญหาดังกล่าวได้ มีค่าความถูกต้องที่สูงขึ้น และมีประสิทธิภาพที่สูงกว่า ILDA นอกจากนี้ ยังมีงานวิจัย (Pang, Peng et al., 2015) ที่พัฒนาระบบออนไลน์ FNTAE ในการตรวจจับการบุกรุกแบบเวลาจริง โดยใช้หลักการ FIncLDA ในการเรียนรู้ แล้วใช้ *k*-NN ในการเป็นขั้นตอนวิธีของการตัดสินใจว่าใช่การบุกรุกหรือไม่ พบว่าระบบนี้ ได้ใช้ความสามารถของ Chunk LDA ในการเรียนรู้แบบออนไลน์ สามารถใช้งานได้จริง เพิ่มประสิทธิภาพในการตรวจจับการบุกรุกได้เป็นอย่างดี

การเพิ่มประสิทธิภาพในการตรวจจับนั้น ขั้นตอนเตรียมข้อมูล มีความสำคัญ หลายงานวิจัยได้มีการทำการคัดเลือกคุณลักษณะ ด้วยวิธีการที่แตกต่างกัน เช่น งานวิจัย (Ji, Jeong, Choi and Jeong, 2016) ได้ใช้ Discrete Wavelet Transform (DWT) ในการเพิ่มประสิทธิภาพ แล้วใช้ iPCA ในการทำ Factor Analysis งานวิจัยนี้ได้ใช้สำหรับ Visual Comparison ของคุณลักษณะต่างๆ ซึ่งผู้วิจัยได้ทำการทดลองเปรียบเทียบการ Projection ชุดข้อมูล NSL-KDD ระหว่างมีการใช้ DWT และไม่ใช้ DWT พบว่า การใช้ DWT นั้นมีประสิทธิภาพในการตรวจจับดีกว่า แต่มีบางกลุ่ม ได้แก่ R2L ตรวจจับได้ไม่ดีนัก เนื่องจากเป็นประเภทของการบุกรุกที่มีปริมาณน้อยปะปนในข้อมูลขนาดใหญ่ แต่เมื่อใช้ DWT สามารถตรวจจับได้และมีประสิทธิภาพสูงเมื่อนำไปทดสอบกับการเรียนรู้ของเครื่องวิธีการอื่นๆ นอกจากนี้ ยังมีการใช้ Chi Squared Attribute Evaluator ในการทำการคัดเลือกคุณลักษณะ แล้วทำการจำแนก ด้วย LDA และ Logistic Regression (LR) พบว่า ทั้งสองวิธีการมีประสิทธิภาพในการตรวจจับได้ดีทั้งแบบหลายกลุ่ม และแบบสองกลุ่ม ถึงแม้ว่าค่าความถูกต้องจะสูงกว่า Naïve Bayes แต่อาจจะไม่สูงกว่า SVM และ C4.5 แต่มีการประมวลผลที่ต่ำกว่าSVM มาก จึงมีความเหมาะสมในการนำไปพัฒนาให้สามารถตรวจติดตามเครือข่ายได้แบบเวลาจริง (Real-Time) ในอนาคต (Subba, Biswas and Karmakar, 2015)

Lashkari, Draper-Gil, Mamun and Ghorbani (2017) ได้ศึกษาการการจำแนก ของ Tor Traffic และ Nontor Traffic เพื่อเป็นการตรวจสอบกิจกรรมและรักษาความปลอดภัยในการใช้งานของผู้ใช้งาน ซึ่งผู้วิจัยได้เปรียบเทียบประสิทธิภาพของการจำแนก ด้วย Artificial Neural Network และ Support vector machine โดยใช้ชุดข้อมูล UNB-CIC Tor Network Traffic ในการทดลอง ผลการวิจัยพบว่า การใช้วิธีการคัดเลือกคุณลักษณะบนพื้นฐานของความสัมพันธ์เลือกคุณลักษณะ ได้จำนวน 10 คุณลักษณะ จากนั้นทำการจำแนกด้วย Artificial Neural Network พบว่ามีประสิทธิภาพสูงที่สุด โดยมีค่า ความถูกต้องสูงถึง 99.8% บางงานวิจัยให้ความสำคัญกับการตรวจจับ Tor บนเครือข่ายแบบ Real-time เพราะเล็งเห็นว่าเป็นภัยคุกคามที่ยากแก่การตรวจจับ ทั้งยังสร้างความเสียหายอย่างมาก (Ghafir, Svoboda and Prenosil, 2014)

Abdelhamid, Ayesh และ habtahb (2014) ที่ได้พัฒนา Multi-label Classifier based Associative Classification (MCAC) ในการจำแนกการโจมตีแบบ Phishing โดยมีการใช้ Chi-Square เป็นวิธีการในการทำการคัดเลือกคุณลักษณะ การทดลองได้เปรียบเทียบกับการเรียนรู้ของเครื่องวิธีการอื่นๆ พบว่า ขั้นตอนวิธี MCAC มีค่าความถูกต้องสูงที่สุดและสามารถตรวจจับประเภท “Suspicious” ได้ ซึ่งไม่มีการบุกรุกประเภทในอยู่ในข้อมูลสำหรับฝึกสอนอีกด้วย

จะเห็นได้ว่า งานวิจัยที่ผ่านมาต้องการพัฒนาขั้นตอนวิธีเพื่อเพิ่มประสิทธิภาพในการตรวจจับการบุกรุก และได้เล็งเห็นในการโอกาสในการพัฒนาต่อเพื่อประยุกต์ใช้งานได้ โดยการใช้การเรียนรู้แบบผสมและการเรียนรู้แบบเพิ่มเติมได้กับชุดข้อมูล Tor ซึ่งเป็นการบุกรุกที่ซ่อนบริการและการโจมตีอื่นๆภายใน การพัฒนานี้จะมีเป้าหมายเพื่อเพิ่มประสิทธิภาพในการเรียนรู้ที่เพิ่มเติมได้ของชุดข้อมูลใหม่ๆที่มีการเรียนรู้อย่างต่อเนื่องในโมเดล และสามารถจำแนกประเภทของการบุกรุกได้อย่างมีประสิทธิภาพอีกด้วย เป็นขั้นตอนวิธีที่สามารถต่อยอดการนำไปใช้ประโยชน์ในเครือข่ายจริงในอนาคต

บทที่ 3
วิธีการดำเนินงานวิจัย

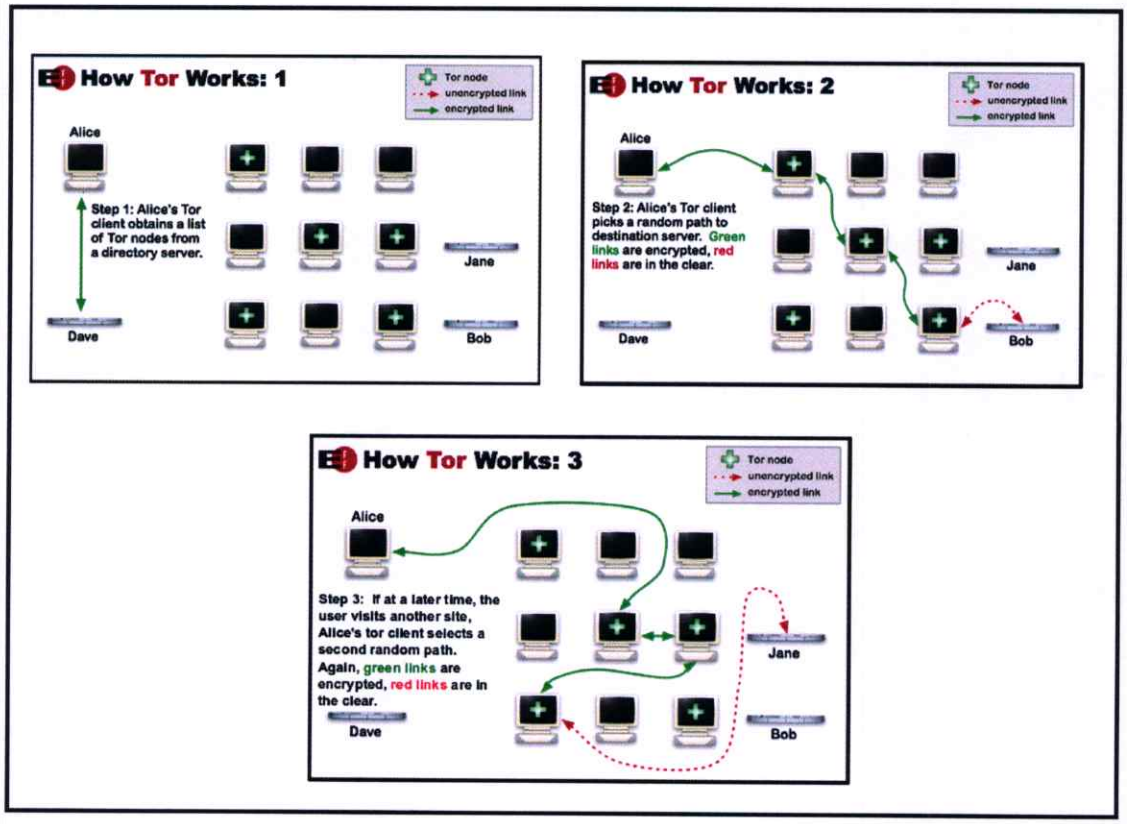
ในบทนี้จะเสนอแนวทางการดำเนินการวิจัย สำหรับการพัฒนาขั้นตอนวิธีใหม่ในการตรวจจับการบุกรุกเครือข่ายคอมพิวเตอร์แบบ ด้วยเรียนรู้ที่เพิ่มเติมได้บนพื้นฐานของการวิเคราะห์ดิส คริมิแนนต์เชิงเส้นแบบเพิ่มเติมได้และการเรียนรู้แบบผสม ซึ่งจะเป็นขั้นตอนวิธีที่มีการทำงานบนพื้นฐานของการเรียนรู้ที่เพิ่มขึ้น ซึ่งจะมีขั้นตอนการดำเนินการวิจัย ตามกระบวนการทำเหมืองข้อมูลดังที่กล่าวไว้ในบทที่ 2

3.1 ทำความเข้าใจในธุรกิจ (Business Understanding)

การรับส่งข้อมูลทางเครือข่ายในปัจจุบัน มีการปะปนของสิ่งผิดปกติในระหว่างการรับส่งมากมาย ซึ่งโดยปกติหากข้อมูลถูกส่งถึงปลายทางและถูกถอดรหัสในระดับหน้าเครื่องแล้ว นั่นหมายถึงสิ่งผิดปกติเหล่านั้น ถูกส่งสำเร็จและถอดรหัสได้เมื่อมาถึงเครื่องคอมพิวเตอร์ของผู้ใช้งาน เช่น มีการส่งไฟล์ไวรัสผ่านข้อความส่วนตัวทางสื่อสังคมออนไลน์ (Social Network) ทำให้ผู้ใช้งานดาวน์โหลดไฟล์ไวรัสลงสู่เครื่องในที่สุด เป็นต้น กรณีเป็นอุปสรรคต่อการตรวจจับด้วย Firewall หรือ IDS เนื่องจากมีการใช้โปรโตคอล SSL ในการรับส่งข้อมูล จึงทำให้ไม่สามารถถอดรหัสได้โดยง่ายในระหว่างการส่ง นอกจากนี้ยังมีโปรแกรมประยุกต์ (Application) อีกมากมาย ที่เข้ารหัสในการส่งแต่กลับถูกใช้เป็นช่องทางของผู้บุกรุกส่งสิ่งที่ไม่พึงประสงค์

การใช้ Tor ในการรับส่งข้อมูล เป็นการเข้ารหัสเพื่อปิดบังแหล่งที่มาที่แท้จริง (Hidden Services) มักเป็นเครื่องมือแรกของผู้โจมตี ใช้ปิดบังตนเองก่อนจะเริ่มโจมตีระบบเป้าหมายตามที่ต้องการ ทำให้ไม่สามารถติดตามและแกะรอยผู้โจมตีได้ ในกรณีที่เกิดความเสียหายขึ้นขึ้นแก่ระบบ เช่น ใช้ Tor แล้วทำการโจมตีด้วย Distributed Denial of Services (DDoS) หรือ ใช้ Tor ร่วมกับการทำ VPN จะเสมือนการเข้ารหัสไว้หลายชั้น ทำให้การรับส่งข้อมูลในขณะนั้นยากแก่การตรวจสอบแหล่งที่มามากขึ้น แสดงขั้นตอนการทำงานของ Tor ดังรูปที่ 3.1

งานวิจัยนี้ จึงมุ่งพัฒนาขั้นตอนวิธีใหม่ ที่สามารถตรวจจับการบุกรุกเครือข่าย Tor และการใช้งานที่มีการเข้ารหัสในการรับส่งข้อมูลอีกหลายรูปแบบโดยขั้นตอนวิธีที่นำเสนอจะมีการเรียนรู้ที่เพิ่มขึ้นของข้อมูลใดๆ ณ ขณะหนึ่งโดยเมื่อมีการเรียนรู้แล้วจะได้ค่าการคำนวณที่เป็นตัวแทนของโมเดลทั้งหมดและเมื่อมีข้อมูลใหม่เพิ่มขึ้นจะสามารถเรียนรู้ต่อได้ทันทีจากค่าการคำนวณสุดท้าย ไม่จำเป็นต้องใช้ข้อมูลเก่าเพื่อปรับปรุงโมเดลทั้งหมดเหมือนการเรียนรู้แบบมีผู้สอนทั่วไป ทั้งนี้ โครงสร้างการตรวจจับการบุกรุกที่นำเสนอจึงเหมาะแก่การเป็นขั้นตอนวิธีที่ใช้บนระบบตรวจจับการบุกรุกที่ต้องเรียนรู้และตรวจหาความผิดปกติตลอดเวลาที่มีการใช้งานเครือข่ายคอมพิวเตอร์ นอกจากนี้งานวิจัยนี้ยังพัฒนาขั้นตอนวิธีแบบผสมซึ่งมุ่งเน้นความสามารถในการจำแนกข้อมูลการบุกรุกเครือข่ายได้อย่างมีประสิทธิภาพ ซึ่งจะสามารถจำแนกประเภทของการบุกรุกข้อมูลเครือข่าย Tor ที่มีการเข้ารหัสอย่างซับซ้อนเช่นกัน



รูปที่ 3.1 การทำงานของ Tor Services

3.2 ทำความเข้าใจข้อมูล (Data Understanding)

ข้อมูลที่ใช้ในการทดลอง เป็นข้อมูลที่ถูกสร้างขึ้นจากเครือข่ายเสมือนจริง โดยใช้เครื่องมือในการสร้างเครือข่ายที่ไม่ใช่ Tor (Nontor) และสร้างเครือข่าย Tor ขึ้นมา (UNB CIC, 2017) โดยมี 8 การใช้งานในเครือข่าย ได้แก่ การใช้เว็บเบราว์เซอร์ (Browsing) ข้อมูลเสียง (Audio Streaming) ข้อความส่วนตัว (Chat) ข้อมูลวีดีโอ (Video Streaming) อีเมล์ (Mail) การโทรด้วยเสียงผ่านเครือข่ายอินเทอร์เน็ต (Voice over IP: VoIP) การส่งแบบ Peer-to-Peer (P2P) และ การส่งไฟล์ (File Transfer)

ในการเก็บข้อมูลระหว่างการรับส่งจะใช้ซอฟต์แวร์ Wireshark และซอฟต์แวร์ TCP Dump ในการเก็บข้อมูลงานวิจัยนี้จะใช้ชุดข้อมูล UNB-CIC Tor Network Traffic ซึ่งผู้พัฒนาได้จัดเตรียมข้อมูลไว้ 2 รูปแบบ ได้แก่

- ข้อมูลที่ 1 Tor Scenario A ซึ่งเป็นแบบสองกลุ่ม ได้แก่ Tor และ Nontor
- ข้อมูลที่ 2 Tor Scenario B ซึ่งเป็นแบบหลายกลุ่ม ได้แก่ Browsing, Audio Streaming, Chat, Video Streaming, Mail, VoIP, P2P และ File Transfer แสดงรายละเอียดดังตารางที่ 3.1

ตารางที่ 3.1 กลุ่มของข้อมูลในฐานข้อมูล Tor Scenario A และ Scenario B

Scenario	ลำดับที่	กลุ่ม	บริการ (Services)
A	1	Tor	8 ประเภทของบริการ
	2	Nontor	Normal
B	1	Web Browsing	HTTP, HTTPS traffic
	2	Email	SMTP/S, POP3/SSL
	3	Chat	Facebook, Hangouts, Skype, IAM, และ ICQ
	4	Audio-Streaming	Audio Applications บนข้อมูล Streaming
	5	Video-Streaming	HTML5 and flash versions
	6	FTP	Skype file transfers, FTP over SSH (SFTP) and FTP over SSL (FTPS)
	7	VoIP	Facebook, Hangouts and Skype
	8	P2P	Bittorrent

3.3 การจัดเตรียมข้อมูล (Data Preparation)

ในการจัดเตรียมข้อมูลมีการจัดการกับค่าที่สูญหาย (Missing Values) และการคัดเลือกคุณลักษณะ

3.1.1 จัดการข้อมูลสูญหาย (Missing Values)

ฐานข้อมูล Tor มี Missing Values จำนวน 6 Records ใน Scenario A ส่วนของ Scenario B ไม่มีข้อมูลสูญหายดังนั้นจึงใช้การประมาณค่าทีใกล้เคียงกับข้อมูลที่ไม่สูญหาย (Nearest Non-Missing Value)

3.2.2 การทำการคัดเลือกคุณลักษณะ

งานวิจัยนี้ได้ใช้สหสัมพันธ์แบบเพียร์สัน ในการทำการคัดเลือกคุณลักษณะเนื่องจากเป็นขั้นตอนวิธีที่มีการใช้งานกับข้อมูลการบุกรุกได้มีประสิทธิภาพ ซึ่งต้นฉบับฐานข้อมูล Tor จากผู้พัฒนามีทั้งหมด 27 คุณลักษณะ ทั้งสอง Scenarios แสดงดังตารางที่ 3.2 ซึ่งจะต้องทำการคัดเลือกเพื่อให้เหลือเพียงคุณลักษณะที่มีความสัมพันธ์กันสูงเพื่อให้มีประสิทธิภาพในการจำแนกและลดภาระการประมวลผลอีกด้วย

ตารางที่ 3.2 ชื่อและความหมายของคุณลักษณะ

ลำดับที่	คุณลักษณะ	ความหมาย
1	Source IP	หมายเลขไอพีต้นทาง
2	Source Port	หมายเลขพอร์ตต้นทาง

ตารางที่ 3.2 ชื่อและความหมายของคุณลักษณะ (ต่อ)

ลำดับที่	คุณลักษณะ	ความหมาย
3	Destination IP	หมายเลขไอพีปลายทาง
4	Destination Port	หมายเลขพอร์ตปลายทาง
5	Protocol	โปรโตคอล เช่น TCP UDP หรือ ICMP เป็นต้น
6	Flow Duration	ระยะเวลาในการส่ง
7	Flow Bytes/s	จำนวนไบต์ของข้อมูลในขั้นตอนส่ง
8	Flow Packets/s	จำนวนแพคเก็ตของข้อมูลในขั้นตอนส่ง
9	Flow IAT Mean	ค่าเฉลี่ยของเวลาในการส่งไปทิศทางใดทิศทางหนึ่ง
10	Flow IAT Std	ค่าส่วนเบี่ยงเบนมาตรฐานของเวลาในการส่งไปทิศทางใดทิศทางหนึ่ง
11	Flow IAT Max	ค่าสูงสุดของเวลาในการส่งไปทิศทางใดทิศทางหนึ่ง
12	Flow IAT Min	ค่าต่ำสุดของเวลาในการส่งไปทิศทางใดทิศทางหนึ่ง
13	Fwd IAT Mean	ค่าเฉลี่ยของเวลาในการส่งไปข้างหน้า
14	Fwd IAT Std	ค่าส่วนเบี่ยงเบนมาตรฐานของเวลาในการส่งไปข้างหน้า
15	Fwd IAT Max	ค่าสูงสุดของเวลาในการส่งไปข้างหน้า
16	Fwd IAT Min	ค่าต่ำสุดของเวลาในการส่งไปข้างหน้า
17	Bwd IAT Mean	ค่าเฉลี่ยของเวลาในการส่งกลับ
18	Bwd IAT Std	ส่วนเบี่ยงเบนมาตรฐานของเวลาในการส่งกลับ
19	Bwd IAT Max	ค่าสูงสุดของเวลาในการส่งกลับ
20	Bwd IAT Min	ค่าต่ำสุดของเวลาในการส่งกลับ
21	Active Mean	จำนวนเวลาเฉลี่ยที่ทำงาน ก่อนจะหยุดนิ่ง
22	Active Std	ค่าส่วนเบี่ยงเบนมาตรฐานของเวลาที่ทำงาน ก่อนจะหยุดนิ่ง
23	Active Max	จำนวนเวลาสูงสุดที่ทำงาน ก่อนจะหยุดนิ่ง
24	Active Min	จำนวนเวลาต่ำสุดที่ทำงาน ก่อนจะหยุดนิ่ง
25	Idle Mean	เวลาเฉลี่ยที่หยุดนิ่งก่อนจะกลับมาทำงาน
26	Idle Std	ค่าส่วนเบี่ยงเบนมาตรฐานของเวลาที่หยุดนิ่งก่อนจะกลับมาทำงาน
27	Idle Max	เวลาสูงสุดที่หยุดนิ่งก่อนจะกลับมาทำงาน

3.3.3 แปลงข้อมูล

ในการเตรียมข้อมูลจะต้องทำการแปลงข้อมูลที่อยู่ในรูปแบบข้อความสัญลักษณ์ ไปเป็นตัวเลขเพื่อให้สามารถนำไปใช้ในการคำนวณด้วยขั้นตอนวิธีที่นำเสนอได้ ตัวอย่าง เช่น โปรโตคอล (Protocol)

TCP	ซึ่งไม่สามารถนำไปคำนวณได้ จึงแปลงเป็น 1
UDP	ซึ่งไม่สามารถนำไปคำนวณได้ จึงแปลงเป็น 2 เป็นต้น
ICMP	ซึ่งไม่สามารถนำไปคำนวณได้ จึงแปลงเป็น 3 เป็นต้น

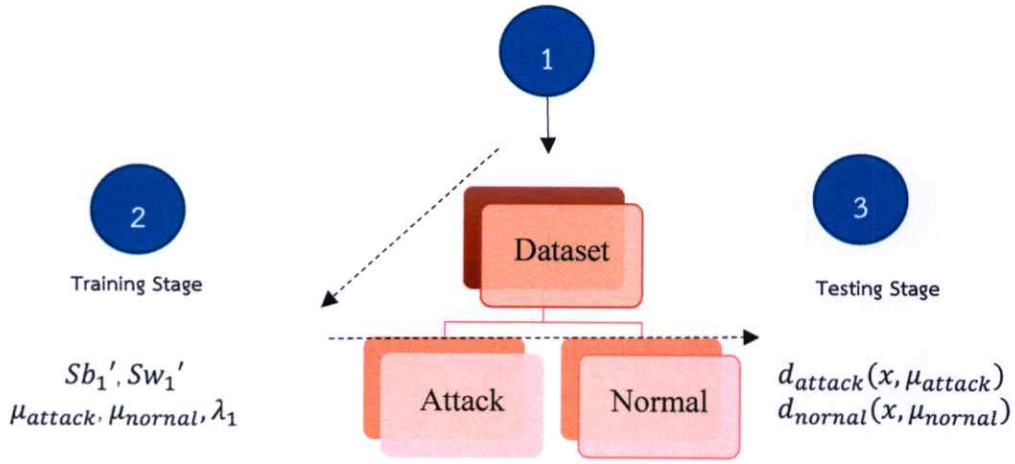
3.4 การสร้างแบบจำลอง (Modeling)

ในงานวิจัยจะมีการสร้างแบบจำลอง 2 แบบ คือ แบบจำลองที่นำเสนอคือ แบบจำลองการเรียนรู้ตามลำดับชั้นแบบเพิ่มเติมได้สำหรับความปลอดภัยของเครือข่าย บนพื้นฐานของการวิเคราะห์ดิสคริมิแนนต์เชิงเส้นแบบเพิ่มเติมได้ ซึ่งเป็นแบบจำลองในการจำแนก เพื่อตรวจจับการบุกรุกบนพื้นฐานของการเรียนรู้แบบเพิ่มเติมได้ และแบบจำลองการเรียนรู้แบบผสมชื่อว่า Hybrid Adaboost.m1 ซึ่งเป็นแบบจำลองในการจำแนก เพื่อตรวจจับการบุกรุกบนพื้นฐานของการเรียนรู้แบบผสม มีรายละเอียดดังต่อไปนี้

3.4.1 แบบจำลองการเรียนรู้ตามลำดับชั้นแบบเพิ่มเติมได้สำหรับความปลอดภัยของเครือข่าย บนพื้นฐานของการวิเคราะห์ดิสคริมิแนนต์เชิงเส้นแบบเพิ่มเติมได้

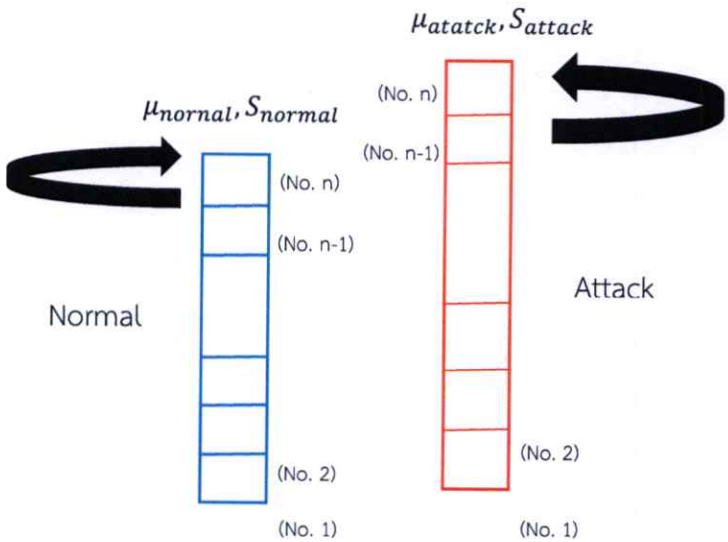
ขั้นตอนนี้ จะฝึกสอนด้วยการเรียนรู้ตามลำดับชั้นแบบเพิ่มเติมได้สำหรับความปลอดภัยของเครือข่าย บนพื้นฐานของการวิเคราะห์ดิสคริมิแนนต์เชิงเส้นแบบเพิ่มเติมได้ เมื่อได้ข้อมูลที่ผ่านการคัดเลือกคุณลักษณะมาแล้วนั้น ขั้นตอนนี้จะทำการแบ่งข้อมูลสำหรับฝึกสอนเป็นร้อยละ 70 และข้อมูลสำหรับทดสอบสอบเป็นร้อยละ 30 ซึ่งในส่วนของการฝึกสอน โดยการทดลอง 2 ส่วน คือ ทดลองด้วยชุดข้อมูล Tor Scenario A จะเป็นตัวแทนของการทำการจำแนกแบบสองกลุ่ม (Binary Classification) และ ชุดข้อมูล Tor Scenario B จะเป็นตัวแทนของการทำการจำแนกแบบหลายกลุ่ม จะใช้ข้อมูลสำหรับฝึกสอนโดยขั้นตอนของการเรียนรู้แบบเพิ่มเติมได้จะทำการรับเข้าข้อมูลทีละ 1 Record เข้าไปทำการเรียนรู้ ในกรณีแบบสองกลุ่ม แสดงขั้นตอนวิธีดังรูปที่ 3.2

จากรูปที่ 3.2 ขั้นตอนการฝึกสอนข้อมูลการคำนวณจะเริ่มต้นจากอ่านข้อมูลทีละ 1 Record ที่ทราบ แล้วว่าคือกลุ่มใด จากนั้นจะคำนวณด้วยขั้นตอนวิธีที่นำเสนอ โดยเริ่มต้น จะคำนวณค่าเฉลี่ยของทั้งสองกลุ่ม คือ μ_{attack} และ μ_{nornal} แบบเพิ่มเติมได้แล้วแยกเก็บไว้คนละตัวแปร สมมติว่าข้อมูลแรกที่คำนวณคือกลุ่ม Normal ดังนั้น ข้อมูลที่คำนวณได้จะถูกเก็บไว้ในตัวแปร ซึ่ง เป็นเป็นตัวแปรที่รู้จำว่าคือ Normal และเมื่อมีข้อมูลที่สองมาคำนวณ เป็นกลุ่ม Attack ข้อมูลที่ คำนวณได้จะถูกเก็บไว้ในตัวแปร ซึ่งเป็นเป็นตัวแปรที่รู้จำว่าคือ Attack ในขั้นตอนนี้จะคำนวณค่า Scatter Matrix (S) ของแต่ละกลุ่มแยกเก็บไว้ด้วยเพื่อเตรียมคำนวณค่า Sw ต่อไป แสดงดังรูปที่ 3.3



รูปที่ 3.2 การจำแนกแบบสองกลุ่มด้วยขั้นตอนวิธีที่นำเสนอ

ดังนั้น เมื่อมีการอ่านข้อมูลเข้ามาคำนวณเพิ่ม ก็จะวนซ้ำโดยที่ขั้นตอนวิธีจะนำค่าสุดท้ายที่เคยคำนวณเก็บไว้ในตัวแปรมาใช้คำนวณต่อทันที จากรูปที่ 3.3 คือ no. n-1 จะถูกนำมาใช้คำนวณต่อโดยที่ไม่มีการนำข้อมูลเดิม มาร่วมคำนวณใหม่ เช่น คำนวณข้อมูลลำดับที่ 500 (n) ของกลุ่ม Normal จะนำข้อมูลที่เก็บไว้ในตัวแปรที่คำนวณถึงลำดับที่ 499 (n-1) มาคำนวณต่อทันที ดังนั้น ข้อมูลใดๆที่เคยนำมาคำนวณแล้ว จะละทิ้งไม่นำกลับมาคำนวณอีก แตกต่างจากการสร้างโมเดลของการเรียนรู้แบบมีผู้สอนทั่วไป จากนั้นจึงคำนวณค่า Sb_1', Sw_1', λ_1 ต่อไปทันทีแบบเพิ่มเติมได้ในทุกๆข้อมูล ค่าที่ได้จากการคำนวณในขั้นตอนฝึกสอนคือ คือ Sb_1', Sw_1', λ_1 ซึ่งค่าที่นำมาใช้ได้คือ ค่าที่ ทำการเรียนรู้จนถึงข้อมูลฝึกสอนข้อมูลสุดท้ายแล้ว นั่นหมายถึง หากทำการเรียนรู้จนถึงข้อมูลที่ 450 จะได้ค่า Sb_1', Sw_1', λ_1 ที่จำนวนข้อมูล 450 Records แต่หากมีข้อมูลทั้งหมด 10,000 ข้อมูล ก็จะต้องเรียนรู้จนถึงลำดับที่ 10,000 จึงจะได้ค่า Sb_1', Sw_1', λ_1 ที่จำนวนข้อมูล 10,000 จึงจะนำมาใช้ได้

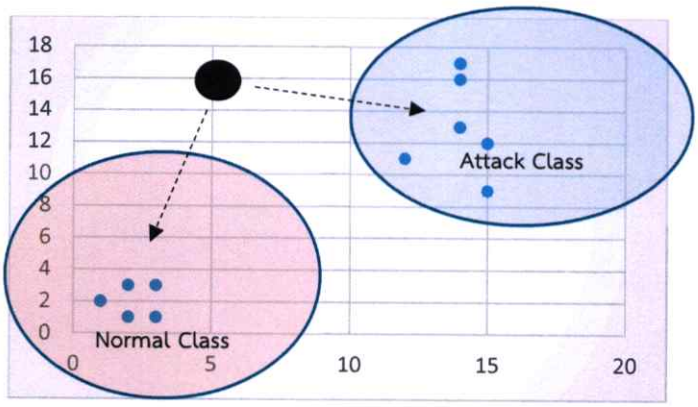


รูปที่ 3.3 การคำนวณเพื่อเก็บค่า μ และ S จากการคำนวณ ของทั้งสองกลุ่ม

สำหรับขั้นตอนการทดสอบ จะอ่านข้อมูลที่ต้องการทดสอบเข้ามาทีละ 1 เรคอร์ด เช่นกัน จากนั้น ค่า λ_1 และค่า μ_{nornal} และ μ_{atatck} ดังรูปที่ 3.3 จะถูกนำมาใช้ในการคำนวณด้วย การหาระยะทางมาหาลาโนบิสโดยข้อมูลจะถูกคำนวณทั้งคือ

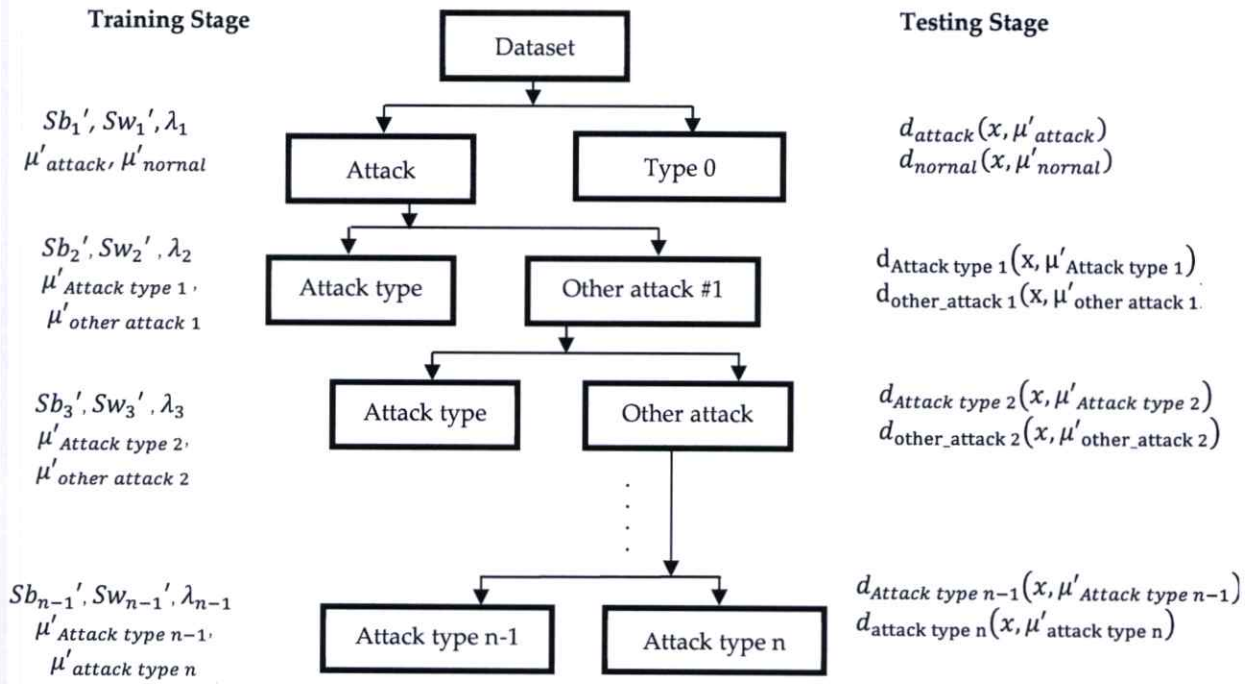
- ครั้งที่ 1 ข้อมูลคำนวณหาระยะห่างระหว่างกลุ่ม Normal โดยใช้ค่า μ_{nornal} และ λ_1 ได้ค่าระยะห่าง D_{nornal}
- ครั้งที่ 2 ข้อมูลคำนวณหาระยะห่างระหว่างกลุ่ม Attack โดยใช้ค่า μ_{attack} และ λ_1 ได้ค่าระยะห่าง D_{attack}

จากนั้น เปรียบเทียบว่าค่าระหว่าง D_{nornal} และ D_{attack} ให้ทำนายว่าเป็นกลุ่มนั้น เพราะว่ามีระยะห่างใกล้ที่สุด เช่น ถ้า $D_{attack} < D_{nornal}$ ก็สรุปได้ว่าข้อมูลนั้นๆคือกลุ่ม Attack หรือ Tor นั่นเอง

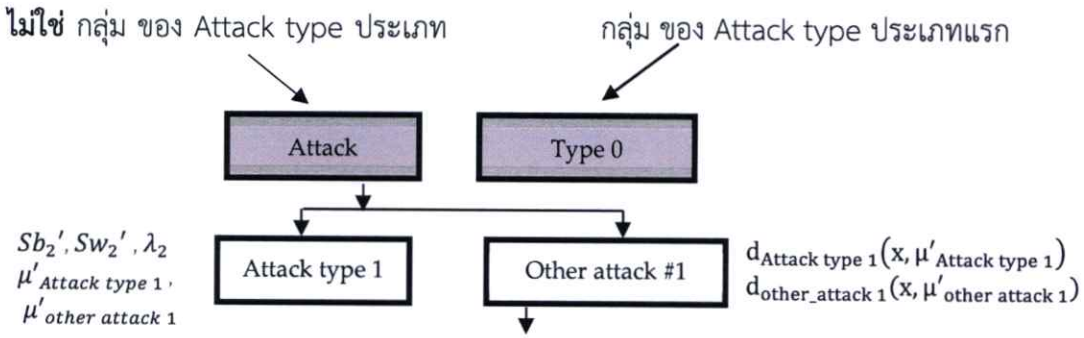


รูปที่ 3.4 การหาระยะห่างระหว่าง 2 กลุ่ม

ในกรณีการจำแนกแบบหลายกลุ่ม แสดงขั้นตอนวิธีดังรูปที่ 3.5 จะพบว่า ในขั้นตอนฝึกสอนจะมีการทำ Incremental Learning เป็นลำดับชั้น เพื่อทำการจำแนก ให้กับทุกกลุ่ม ของการบุกรุกใน Scenario B โดยในลำดับชั้นบนสุด จะคล้ายกับกรณีแบบสองกลุ่ม ซึ่งฝึกสอน โดยรู้จำกลุ่ม ที่มีปริมาณมากที่สุดก่อนเป็นลำดับชั้นแรก การเรียนรู้ในลำดับชั้นนี้ จะเหมือนกับการทำการจำแนกแบบสองกลุ่มดังที่กล่าวมาทุกประการ จากนั้น เมื่อต้องการฝึกสอนข้อมูลในกลุ่มอื่นที่เหลือ ก็จะวนซ้ำคำนวณในลำดับชั้นถัดไป ลงไปตามลำดับจนถึง กลุ่มสุดท้าย อธิบายเพิ่มเติมดังรูปที่ 3.6



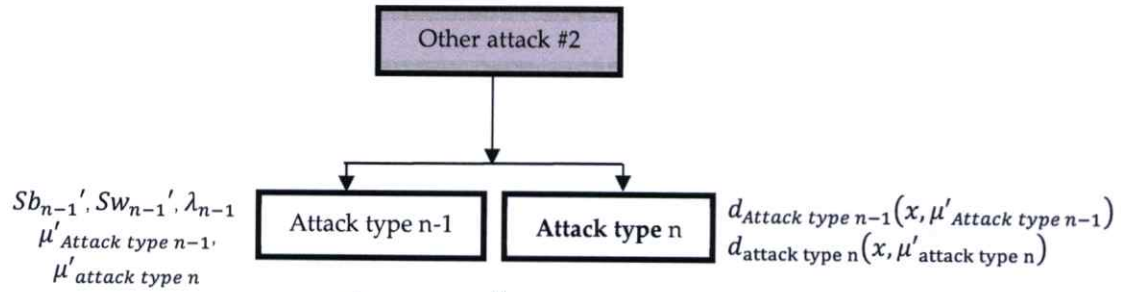
รูปที่ 3.5 การจำแนกแบบหลายกลุ่มด้วยขั้นตอนวิธีที่นำเสนอ



รูปที่ 3.6 ลำดับชั้นที่ 2 ของโครงสร้าง

พิจารณารูปที่ 3.6 จากลำดับชั้นบนสุดจะรู้จำ กลุ่มแรก หรือ Type 0 แต่ในส่วนของกลุ่ม Attack นั้น จะหมายถึงรวมทุกกลุ่ม ที่ไม่ใช่ กลุ่ม Type 0 เมื่อถึงลำดับชั้นถัดมา ก็จะฝึกสอนข้อมูลโดยรู้จำค่าของกลุ่ม ที่ 2 หรือชื่อว่า Attack Type 1 (ทราบกลุ่ม) และ Other Attack Type#1 นั้นจะเป็นกลุ่มที่รวมกันที่เหลือทั้งหมด

ดังนั้น ค่าจากการเรียนรู้ ได้แก่ $Sb_2', Sw_2', \lambda_2, \mu'_{Attack\ type\ 1}$ และ $\mu'_{other\ attack\ 1}$ ก็จะเป็นค่าที่รู้จำเฉพาะสองกลุ่ม ในลำดับชั้นนั้นๆ คือ Attack Type 1 และ Other Attack Type#1 ซึ่งค่า μ และ S จะคำนวณแบบการเรียนรู้แบบเพิ่มเติมได้เก็บไว้สำหรับการหาระยะทางมาหาลาโนบิสในลำดับชั้นนี้ของขั้นตอนทดสอบอีกด้วย ทำเช่นนี้ไปเรื่อยๆ เป็นลำดับชั้นแบบ Tree จนถึงกลุ่ม ของประเภทการบุกรุกสุดท้าย แต่ในลำดับชั้นสุดท้ายจะจำแนกเหลือเพียงสองกลุ่มเท่านั้น แสดงดังรูปที่ 3.7



รูปที่ 3.7 ลำดับชั้นสุดท้ายโครงสร้าง

พิจารณารูปที่ 3.7 ลำดับชั้นสุดท้าย การฝึกสอนเพื่อรู้จำจะเหลือเพียง 2 กลุ่ม เนื่องจากไม่มีกลุ่มอื่นใดอีก การฝึกสอนจึงแบ่งเป็นเพียงกลุ่ม Attack type n-1 และ Attack type n เท่านั้น ซึ่งงานวิจัยนี้ขั้นตอนวิธีที่นำเสนอจะคำนวณค่า Sb' แบบสองกลุ่ม โดยจะคำนวณระหว่าง 2 โหนดใดๆในแต่ละลำดับชั้นของต้นไม้ แตกต่างจากการคำนวณต้นฉบับของ Sequential Incremental LDA ที่สมการสำหรับการหาค่าเป็นแบบหลายกลุ่ม ซึ่งจะเหมาะสมกว่า ในการจำแนกที่พิจารณาเพียงสองกลุ่มใดๆ ณ ลำดับชั้นที่กำลังฝึกสอบข้อมูล แสดงดังสมการที่ 2.29

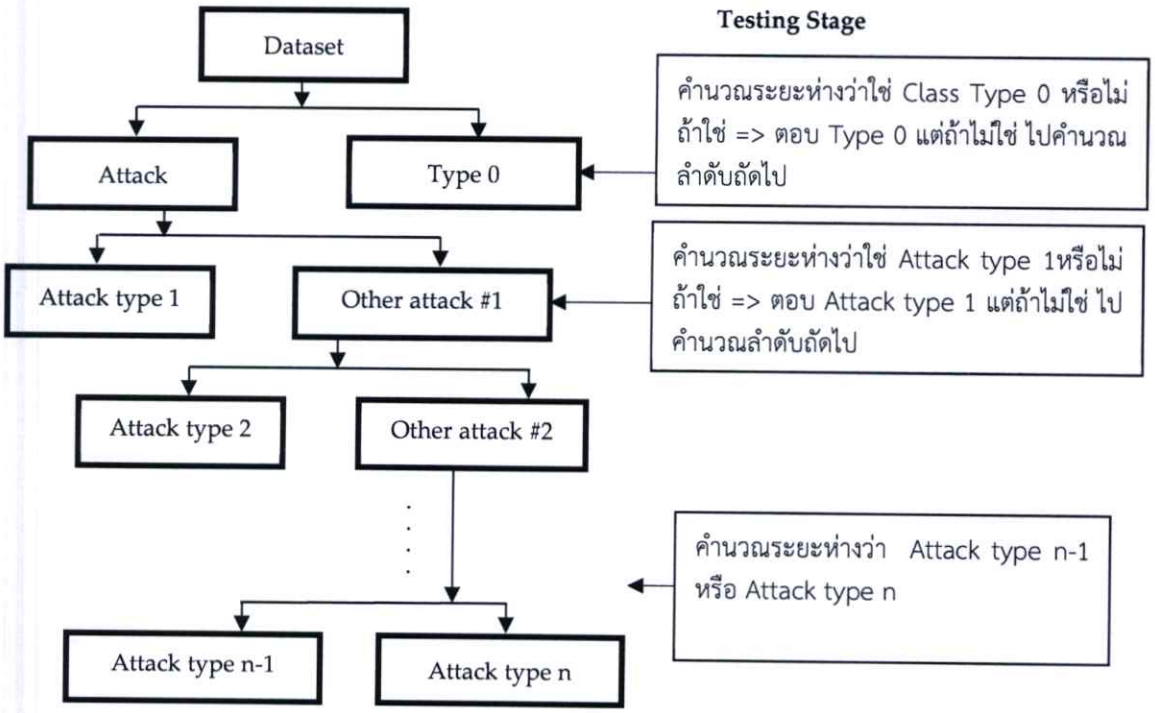
$$Sb' = \sum_{c=1}^{2}(\bar{x}'_1 - \bar{x}'_2)(\bar{x}'_1 - \bar{x}'_2)^T \qquad (2.29)$$

ดังนั้น ขั้นตอนการฝึกสอนจะรู้จำในทุกๆลำดับชั้น ว่าโหนดที่ต้องการจำแนก คือกลุ่ม ใด แล้วในกรณีขั้นตอนการทดสอบด้วยข้อมูลทดสอบ ข้อมูลใดๆก็จะเริ่มต้นไล่ลำดับในการหาค่าระยะห่างตั้งแต่ลำดับชั้นบนสุด ลงไปเรื่อยๆจนถึงลำดับชั้นสุดท้าย หากพบว่าข้อมูลใดๆนั้น จัดอยู่ใน กลุ่มใด ก็จะจำแนก ว่าข้อมูลนั้น เป็นการบุกรุกประเภทนั้นๆ

3.2.4 ทดสอบด้วยข้อมูลสำหรับทดสอบ และคำนวณค่าประสิทธิภาพ

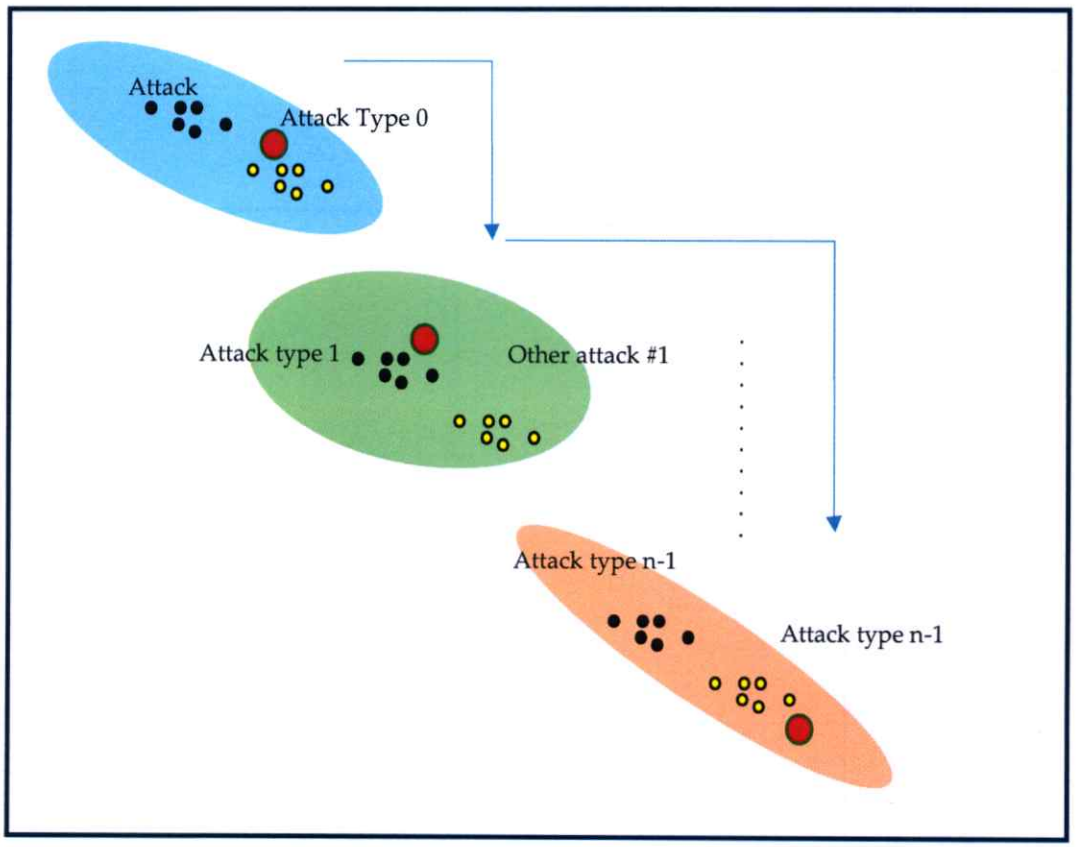
ในขั้นตอนการทดสอบ จะนำข้อมูลทดสอบที่เตรียมไว้ มาทดสอบกับขั้นตอนวิธีที่นำเสนอ โดยค่าที่ได้จากการคำนวณในขั้นตอนฝึกสอน ได้แก่ค่า Max eigenvalue ($\boldsymbol{\lambda}$) และค่า Mean (μ) ของแต่ละลำดับชั้น จะนำมาใช้ในการคำนวณแต่ละลำดับชั้นที่เรียนรู้ไว้ได้ ดังสมการ (2.30)

$$d(x, \mu) = \sqrt{(x-\mu)^T \lambda^{-1}(x-\mu)} \qquad (2.30)$$



รูปที่ 3.8 ขั้นตอนการหาระยะห่างเพื่อทำนายกลุ่มของข้อมูลทดสอบ ในแต่ละลำดับชั้นของต้นไม้

พิจารณารูปที่ 3.8 ข้อมูลทดสอบจะคำนวณระยะห่างโดยเสมือนการไล่ลำดับหาคำตอบว่าข้อมูลนั้นๆมีระยะห่างใกล้ Attack Type ใด โดยเริ่มต้นจะถามลำดับชั้นแรกสุด หากระยะห่างที่คำนวณได้ มีค่าใกล้ Type 0 ก็จะตอบ Type 0 ทันที แต่หากใกล้กลุ่ม Attack ก็จะต้องส่งไปคำนวณต่อในลำดับชั้นถัดไปนั่นคือ Attack type 1 เพื่อดูว่าค่าระยะห่างใกล้ Attack Type 1 หรือไม่ หากใกล้ก็สามารถตอบกลุ่ม ได้ทันที แต่หากไม่ใช่ จะต้องส่งไปคำนวณถามลำดับชั้นถัดไปเรื่อยๆจนถึงลำดับชั้นสุดท้ายของ Tree คล้ายการไล่ลำดับการสอบถามไปในทุกมิติของคู่ 2 กลุ่มใดๆที่กำลังพิจารณาในลำดับชั้นเดียวกัน แสดงได้ชัดเจนขึ้นดังรูปที่ 3.9



รูปที่ 3.9 การหาระยะห่างระหว่าง 2 กลุ่มในแต่ละลำดับชั้นของต้นไม้

การเรียนรู้แบบเพิ่มเติมได้สำหรับการตรวจจับการบุกรุกเครือข่าย แสดขั้นตอนวิธีตามรูปที่ 3.10 กำหนดให้

Sb' = Between-class Scatter Matrix
Sw' = Within-class Scatter Matrix
$\bar{\mu}'$ = ค่าเฉลี่ย
λ = Max Eigenvalue
x = ข้อมูล
y_i = จำนวนกลุ่ม
n = จำนวนข้อมูล

Input: Training set $= \{(x_1, y_1), (x_2, y_2), \ldots, (x_n y_n)\}$, with class label $y_i \in \{1, .., k\}$ of training dataset

Output: Incremental Hierarchical Learning for Network Security based on Incremental Linear Discriminant Analysis Model

For $i = 1, .., n$ // n is number of samples

  For $y = 1, \ldots, k$ // k is number of class label

    If class label belong to any k class. Then

      Update$\bar{\mu}'$, Sb' and Sw'

    Else

      Update $\bar{\mu}'$, Sb' and Sw' for other k class

      Calculate max eigenvalue for any k class // for binary classification

      If class label belong to any $k - 1$ class

        Update $\bar{\mu}'$, Sb' and Sw'

      Else

        Update $\bar{\mu}'$, Sb' and Sw' for k class

        Calculate max eigenvalue for any k class // for binary classification

      End

    End

  End

  Model with max eigenvalue of every class label for hierarchical distance measure

    $$d(x, \mu) = \sqrt{(x - \mu)^T \lambda^{-1} (x - \mu)}$$

End

รูปที่ 3.10 ขั้นตอนวิธีการเรียนรู้แบบเพิ่มเติมได้

3.4.2 แบบจำลอง Hybrid Adaboost.m1

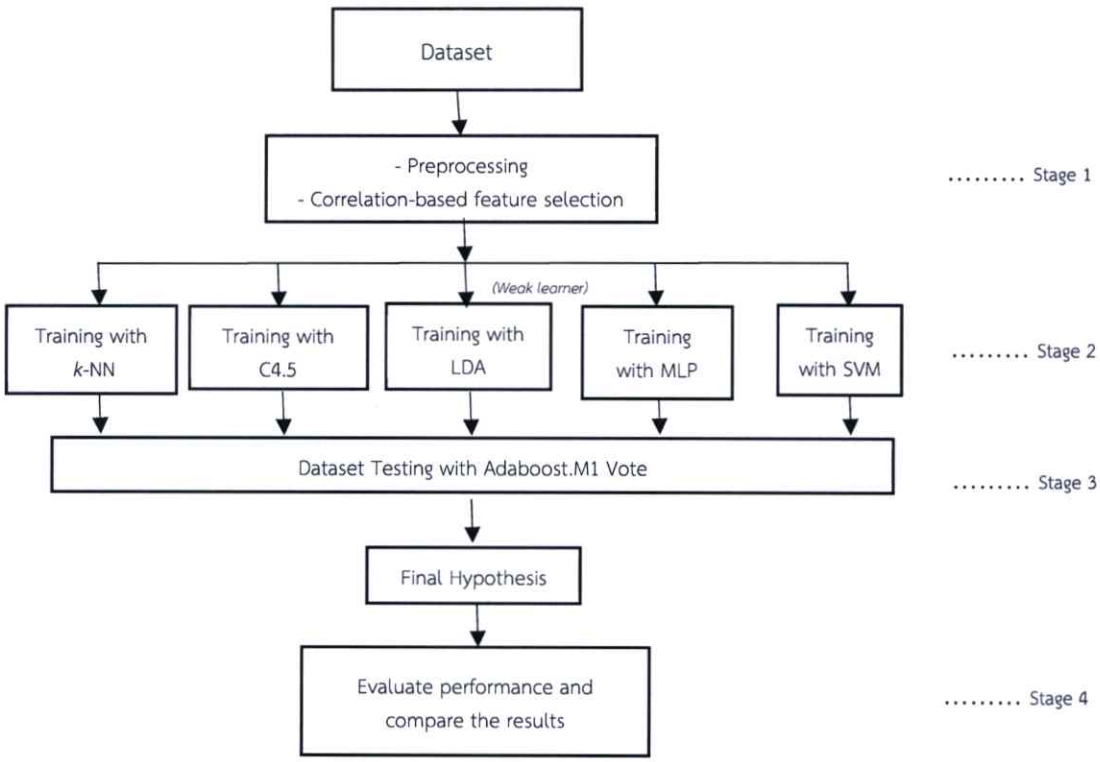
แบบจำลองของ Hybrid Adaboost.m1 จะมีขั้นตอนการจัดการกับข้อมูลในกระบวนการ Preprocessing เหมือนแบบจำลองเรียนรู้ตามลำดับชั้นแบบเพิ่มเติมได้ มีแสดงขั้นตอนของแบบจำลองดังรูปที่ 3.11 และ 3.12

Stage 1: แสดงขั้นตอนการจัดเตรียมข้อมูล ซึ่งเป็นการจัดการกับข้อมูลและมีการใช้วิธีการคัดเลือกคุณลักษณะบนพื้นฐานของความสัมพันธ์ในการคัดเลือกคุณลักษณะ เนื่องจากมีความเหมาะสมในการเลือกคุณลักษณะสำหรับชุดข้อมูลการบุกรุก

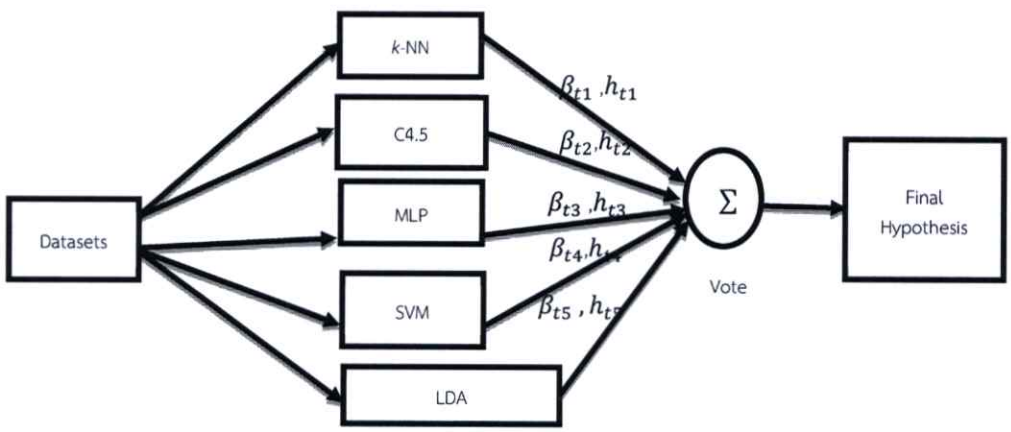
Stage 2: ทำการฝึกสอนด้วย Weak Learner ทั้งขั้นตอนวิธี โดยจะสร้างโมเดลและเก็บค่าคำตอบ (Hypothesis และ Weight) ไว้ เช่น ในการทดลองมี 5 Weak Leaner ได้แก่ *k*-NN, C4.5, LDA, MLP และ SVM ซึ่งจะมีค่า Hypothesis และ Weight ทั้งหมด 5 ค่า ของแต่ละ Weak Learner แสดงดังรูปที่ 3.11

Stage 3: ทำการให้เสียงข้างมากเพื่อหาค่า Final Hypothesis จากทุกๆ Hypothesis และ Weight ที่ได้จาก Stage 2

Stage 4: วิเคราะห์ประสิทธิภาพที่ได้จากการทำการจำแนก



รูปที่ 3.11 แบบจำลอง Hybrid Adaboost.m1



รูปที่ 3.12 การให้เสียงข้างมากจาก Weak Learner

การเรียนรู้แบบผสมสำหรับการตรวจจับการบุกรุกเครือข่าย แสดขั้นตอนวิธีตามรูปที่ 3.13

กำหนดให้

x = ข้อมูล

y = จำนวนกลุ่ม

m = จำนวนข้อมูล

T = จำนวนรอบของการวนซ้ำ

ε = Error

β = Weight

D = ค่า Distribution

Input: sequence of m example $((x_1, y_1), \ldots, (x_m, y_m))$ with labels $y_i \in$ Y=$\{1, \ldots, k\}$

Weak learning algorithm **Weaklearn**

Integer T specifying number of iterations / Weak Learner

Initialize $D_1(i) = 1/m$ for all$i$

Do for $t = 1.2, \ldots, T$

1. Call **Weaklearn** providing it with the distribution D_t
2. Get back hypothesis $h_t = X \rightarrow Y$
3. Calculate the error of h_t : $\varepsilon_t = \sum_{i:h_t(x_i) \neq y_i} D_t(i)$ if ε_t>1/2 then set $T = T - 1$

   and abort loop
4. Set $\beta_t = \varepsilon_t/(1 - \varepsilon_t)$
5. Update distribution D_t: $D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} \beta_t & if\ h_t(x_i) = y_i \\ 1 & otherwise \end{cases}$

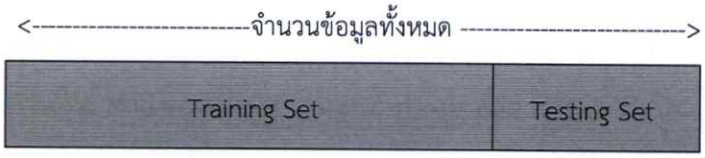
   Where Z_t is a normalization constant (Chosen so that D_{t+1} will be a distribution)

Output the final hypothesis: $h_{fin} = \underset{y \in Y}{argmax} \sum_{t:h_t(x)=y} \log \frac{1}{\beta_t}$

รูปที่ 3.13 ขั้นตอนวิธีวิธีการเรียนรู้แบบผสม

3.5 การประเมินแบบจำลอง (Evaluation)

งานวิจัยนี้ ได้ใช้ Split Test ในการแบ่งข้อมูลสุ่มออกมา ร้อยละ 70 และ ร้อยละ 30 โดยข้อมูลในส่วนร้อยละ 70 จะใช้เป็นข้อมูลสำหรับขั้นตอนการฝึกสอน และข้อมูลร้อยละ 30 เป็นข้อมูลสำหรับขั้นตอนการทดสอบ ซึ่งในการทดลอง จะทำการสุ่มออกมา 3 ครั้ง เพื่อทำการจำแนก แล้วจึงวิเคราะห์ประสิทธิภาพ โดยแต่ละครั้งที่สุ่มข้อมูลอกมา จะนำข้อมูลก่อนหน้า กลับไปรวมไว้แบบเดิมก่อนแล้วจึงสุ่มออกมาใหม่ เมื่อทำครบ 3 ครั้งแล้ว จึงนำค่าประสิทธิที่วิเคราะห์ได้มาหาค่าเฉลี่ยเพื่อเป็นคำตอบ วิธีการดังกล่าวเป็นการป้องกันกรณีที่สุ่มข้อมูลออกมาแล้ว ข้อมูลทดสอบคล้ายกับข้อมูลที่ใช้สร้างโมเดล หรือข้อมูลทดสอบแตกต่างจากข้อมูลที่ใช้สร้างโมเดลมาก จะทำให้วิเคราะห์ประสิทธิภาพออกมาแล้วไม่สามารถเชื่อถือได้ แสดงดังรูปที่ 3.14



รูปที่ 3.14 จำนวนการแบ่ง Split Test

งานวิจัยนี้ได้คำนวณค่าประสิทธิภาพจากตาราง Confusion Matrix ซึ่งจะถูกเปรียบเทียบคำตอบที่นำนายได้ กับกลุ่มที่เป็นบวก (Positive) และกลุ่มที่เป็นลบ (Negative) แล้วจึงวิเคราะห์ค่าออกมาว่า ผลจากการทำนายมีความถูกต้อง (Ture) หรือมีความผิดพลาด (Flase) แสดงดังตาราง 3.1

ตารางที่ 3.3 Confusion Matrix

Predict Value	Actual Value	
	Positive	Negative
Positive	True Positive: TP	False Positive: FP
Negative	False Negative: FN	True Negative: TN

True Positive: TP หมายถึง สิ่งที่ทำนายเป็นกลุ่มบวก และ ผลการทำนายเป็นกลุ่มบวก
True Negative: TN หมายถึง สิ่งที่ทำนายวกลุ่มลบ และ ผลการทำนายเป็นกลุ่มลบ
False Positive: FP หมายถึง สิ่งที่ทำนายเป็นกลุ่มบวก และ ผลการทำนายเป็นกลุ่มลบ
False Negative: FN หมายถึง สิ่งที่ทำนายเป็นกลุ่มลบ และ ผลการทำนายเป็นกลุ่มบวก

ค่าที่ใช้ในการวิเคราะห์ได้แก่ ค่าความแม่นยำ (Precision) ค่าความไว (Recall) ค่าถ่วงดุล (f-Measure) และค่าความถูกต้อง มีการคำนวณจากการนำค่าในตาราง Confusion Matrix ดังต่อไปนี้

Precision เป็นการวัดวัดค่าความแม่นยำในการดึงข้อมูลที่สนใจ ในจำนวนข้อมูลที่ถูกดึงออกมาทั้งหมด

$$Precision = \frac{TP}{TP+FP} \tag{2.31}$$

Recall เป็นการวัดค่าความสามารถในการดึงข้อมูลที่สนใจ ในจำนวนข้อมูลที่เกี่ยวข้องทั้งหมด

$$Recall = \frac{TP}{TP+FN} \tag{2.32}$$

F-measure เป็นค่าที่แสดงความสัมพันธ์ระหว่างค่า Precision และค่า Recall เพื่อหาความถูกต้องโดยรวม

$$f - Measure = \frac{2 \times Precision \times Recall}{Precision+Recall} \tag{2.33}$$

Accuracy เป็นการวัดค่าความถูกต้องโดยรวมของการทำนายทุกกลุ่มของข้อมูล

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{2.34}$$

3.6 การนำแบบจำลองไปใช้ (Deployment)

หลังจากที่ได้ใช้ เรียนรู้ตามลำดับชั้นแบบเพิ่มเติมบนพื้นฐานของการวิเคราะห์ดิสคริมิแนนต์เชิงเส้นแบบเพิ่มเติมได้ ในการตรวจจับข้อมูล Tor แล้วนั้น จะนำฐานข้อมูล Tor ไปทดสอบเพื่อเปรียบเทียบประสิทธิภาพในการตรวจจับกับวิธีการของการเรียนรู้ของเครื่องอื่นๆ เช่น SVM Decision Tree MLP *k*-NN และ Naïve Bayes เป็นต้น นอกจากนี้ ยังนำขั้นตอนวิธี นี้ ไปทดสอบกับฐานข้อมูลการบุกรุกอื่นๆ ที่มีประเภทของการบุกรุกแตกต่างกันไป ที่พบการบุกรุกได้ในปัจจุบัน เช่น NSL-KDD SAME Spambase และ Phishing โดยเป็นการยืนยันผลของประสิทธิภาพในการตรวจจับสิ่งผิดปกติทางเครือข่าย ว่าสามารถตรวจจับการบุกรุกประเภทอื่นๆได้ด้วยหรือไม่

บทที่ 4
ผลการวิจัยและการอภิปรายผล

ในบทนี้จะนำเสนอผลการวิจัยที่ได้จากวิธีการดำเนินงานวิจัย แบ่งเป็นผลการวิจัยจากการเรียนรู้แบบเพิ่มเติมได้ และผลการวิจัยจากการเรียนรู้แบบผสมมีหัวข้อดังต่อไปนี้

4.1 ชุดข้อมูล Tor

ตารางที่ 4.1 รายละเอียดชุดข้อมูล Tor

ชุดข้อมูล	Scenario A		Scenario B	
	กลุ่ม	จำนวน	กลุ่ม	จำนวน
ข้อมูลฝึกสอน	Tor	5631	Audio	505
	Normal	41853	Browsing	1,123
			Chat	226
			File-Transfer	605
			Mail	197
			P2P	760
			Video	612
			VOIP	1,604
ข้อมูลทดสอบ	Tor	2413	Audio	216
	Normal	17937	Browsing	481
			Chat	97
			File-Transfer	259
			Mail	85
			P2P	325
			Video	262
			VOIP	687
	รวม	67,843	รวม	8,044

ตัวอย่างข้อมูลในชุดข้อมูล Tor แสดงดังตารางที่ 4.1

1	Source IP	Source Port	Destination IP	Destination	Protocol	Flow Durati	Flow Bytes/	Flow IAT Me	Flow IAT Sti	Flow IAT Ma	Flow IAT Mi	Fwd IAT Mea	Fwd IAT Std	Fwd IAT Ma
2	10.8.0.10	44816	82.165.251.100	80	6	176437	0	176437	0	176437	176437	0	0	0
3	10.8.0.10	42566	173.194.123.70	80	6	25449	0	25449	0	25449	25449	0	0	0
4	10.8.0.14	58724	74.125.226.3	80	6	25263	0	25263	0	25263	25263	0	0	0
5	10.8.0.14	58724	74.125.226.3	80	6	25359	0	25359	0	25359	25359	0	0	0
6	10.8.0.14	33015	216.58.219.237	443	6	26060	0	26060	0	26060	26060	0	0	0
7	10.8.0.14	53933	173.194.123.5	443	6	25459	0	25459	0	25459	25459	0	0	0
8	10.8.0.14	57281	173.194.207.188	5228	6	38298	0	38298	0	38298	38298	0	0	0
9	10.8.0.14	33015	216.58.219.237	443	6	27120	0	27120	0	27120	27120	0	0	0
10	10.8.0.14	53933	173.194.123.5	443	6	25999	0	25999	0	25999	25999	0	0	0
11	10.8.0.14	57281	173.194.207.188	5228	6	38322	0	38322	0	38322	38322	0	0	0
12	10.8.0.14	33015	216.58.219.237	443	6	25835	0	25835	0	25835	25835	0	0	0
13	10.8.0.14	53933	173.194.123.5	443	6	25557	0	25557	0	25557	25557	0	0	0
14	10.8.0.14	57281	173.194.207.188	5228	6	38287	0	38287	0	38287	38287	0	0	0
15	10.8.0.14	33015	216.58.219.237	443	6	26015	0	26015	0	26015	26015	0	0	0
16	10.8.0.14	53933	173.194.123.5	443	6	25307	0	25307	0	25307	25307	0	0	0
17	10.8.0.14	33015	216.58.219.237	443	6	25950	0	25950	0	25950	25950	0	0	0
18	10.8.0.14	53933	173.194.123.5	443	6	25612	0	25612	0	25612	25612	0	0	0
19	10.8.0.14	57281	173.194.207.188	5228	6	39183	0	39183	0	39183	39183	0	0	0
20	10.8.0.14	57281	173.194.207.188	5228	6	38807	0	38807	0	38807	38807	0	0	0
21	10.8.0.14	57281	173.194.207.188	5228	6	38997	0	38997	0	38997	38997	0	0	0
22	10.8.0.14	57281	173.194.207.188	5228	6	39058	0	39058	0	39058	39058	0	0	0
23	10.8.0.14	32818	173.194.121.5	443	6	31349	0	31349	0	31349	31349	0	0	0
24	10.8.0.14	34564	173.194.123.8	443	6	25435	0	25435	0	25435	25435	0	0	0
25	10.8.0.14	57281	173.194.207.188	5228	6	39120	0	39120	0	39120	39120	0	0	0

รูปที่ 4.1 ตัวอย่างข้อมูลในชุดข้อมูล Tor

4.2 ผลการวิจัยการเรียนรู้แบบเพิ่มเติมได้

4.2.1 ผลการคัดเลือกคุณลักษณะของชุดข้อมูล Tor Scenario A ของการเรียนรู้แบบเพิ่มเติมได้

ในขั้นตอนการทำการคัดเลือกคุณลักษณะ ได้ใช้สหสัมพันธ์แบบเพียร์สันในการคัดเลือก จากจำนวนทั้งหมด 28 คุณลักษณะซึ่งในการทดลองจะแยกระหว่าง Tor Scenario A ซึ่งเป็นชุดข้อมูลแบบสองกลุ่มและ Tor Scenario B จะเป็นชุดข้อมูล แบบหลายกลุ่ม ได้ยุบรวมกลุ่มโดยตามลักษณะของการบุกรุกที่พบเจอคล้ายกัน เหมาะสมต่อการจัดกลุ่มเดียวกัน (He, G. Y., Yang, M., Lau, J. and Gu, X., 2015) คือ กลุ่มที่ 1 ได้แก่ Browsing กลุ่มที่ 2 ได้แก่ FTP และ P2P กลุ่มที่ 3 ได้แก่ Audio และ VOIP และ กลุ่มที่ 4 ได้แก่ Chat Email และ Video ผลการทดลองพบว่า ชุดข้อมูล Tor Scenario A ได้คัดเลือกคุณลักษณะมาได้จำนวน 9 คุณลักษณะ จากการเรียงลำดับคุณลักษณะคุณลักษณะที่มีความสัมพันธ์กันจากมากไปน้อย แล้วจึงทำคุณลักษณะเหล่านั้นทดสอบกับขั้นตอนวิธีที่นำเสนอเพื่อดูผลว่าคุณลักษณะใดบ้างที่ทำให้มีค่าประสิทธิภาพสูงที่สุด แสดงดังตารางที่ 4.1

ตารางที่ 4.2 คุณลักษณะที่คัดเลือกแล้วสำหรับชุดข้อมูล Tor Scenario A ของการเรียนรู้แบบเพิ่มเติมได้

ลำดับ	ลำดับที่ของคุณลักษณะ	ชื่อคุณลักษณะ	ความหมาย
1	8	Flow Packets/s	จำนวนแพคเก็ตของข้อมูลในขั้นตอนส่ง
2	7	Flow Bytes/s	จำนวนไบต์ของข้อมูลในขั้นตอนส่ง
3	6	Flow Duration	ระยะเวลาในการส่ง
4	5	Protocol	โปรโตคอล เช่น TCP UDP หรือ ICMP
5	19	Bwd IAT Max	ค่าสูงสุดของเวลาในการส่งกลับ
6	1	Source IP	หมายเลขไอพีต้นทาง
7	2	Source Port	หมายเลขพอร์ตต้นทาง

ตารางที่ 4.2 คุณลักษณะที่คัดเลือกแล้วสำหรับของชุดข้อมูล Tor Scenario A ของการเรียนรู้แบบเพิ่มเติมได้ (ต่อ)

8	12	Flow IAT Min	ค่าต่ำสุดของเวลาในการส่งไปทิศทางใดทิศทางหนึ่ง
9	9	Flow IAT Mean	ค่าเฉลี่ยของเวลาในการส่งไปทิศทางใดทิศทางหนึ่ง

ในส่วนของ Scenario B ได้คัดเลือกคุณลักษณะ มาจำนวน 11 คุณลักษณะจากการเรียงลำดับคุณลักษณะคุณลักษณะที่มีความสัมพันธ์กันจากมากไปน้อย แล้วจึงทำคุณลักษณะเหล่านั้นทดสอบกับขั้นตอนวิธีที่นำเสนอเพื่อดูผลว่าคุณลักษณะใดบ้างที่ทำให้มีค่าประสิทธิภาพสูงที่สุด แสดงดังตารางที่ 4.2

ตารางที่ 4.3 คุณลักษณะที่คัดเลือกแล้วสำหรับชุดข้อมูล Tor Scenario B ของการเรียนรู้แบบเพิ่มเติมได้

ลำดับ	คุณลักษณะที่	ชื่อคุณลักษณะ	ความหมาย
1	2	Source Port	หมายเลขพอร์ตต้นทาง
2	11	Flow IAT Max	ค่าสูงสุดของเวลาในการส่งไปทิศทางใดทิศทางหนึ่ง
3	15	Fwd IAT Max	ค่าสูงสุดของเวลาในการส่งไปข้างหน้า
4	19	Bwd IAT Max	ค่าสูงสุดของเวลาในการส่งกลับ
5	18	Bwd IAT Std	ส่วนเบี่ยงเบนมาตรฐานของเวลาในการส่งกลับ
6	14	Fwd IAT Std	ค่าส่วนเบี่ยงเบนมาตรฐานของเวลาในการส่งไปข้างหน้า
7	4	Destination Port	หมายเลขพอร์ตปลายทาง
8	10	Flow IAT Std	ค่าส่วนเบี่ยงเบนมาตรฐานของเวลาในการส่งไปทิศทางใดทิศทางหนึ่ง
9	6	Flow Duration	ระยะเวลาในการส่ง
10	1	Source IP	หมายเลขไอพีต้นทาง
11	9	Flow IAT Mean	ค่าเฉลี่ยของเวลาในการส่งไปทิศทางใดทิศทางหนึ่ง

4.2.2 ประสิทธิภาพของการจำแนกกลุ่มด้วยการเรียนรู้แบบเพิ่มเติมได้

4.2.2.1 การเปรียบเทียบประสิทธิภาพ

ในการทดลองจะเปรียบเทียบประสิทธิภาพในการทำการจำแนกด้วยค่า Precision Sensitivity Specificity f-Measure และค่าความถูกต้อง โดยทำการทดลองกับ Tor Scenario A ชุด

ข้อมูล ซึ่งเป็นแบบสองกลุ่ม และ Tor Scenario B ชุดข้อมูลซึ่งเป็นแบบหลายกลุ่ม มีการทำการจำแนก จะแบ่งออกเป็น 3 แบบย่อยแตกต่างกัน ดังนี้

- การใช้ ILDA ร่วมกับกับการหาระยะทางมาฮาลาโนบิส เป็นการทดลองที่ต้องการเปรียบเทียบการใช้ ILDA แบบดังเดิมที่มีการพัฒนาไว้ ร่วมกับการหาระยะทางมาฮาลาโนบิส
- การใช้ ILDA ทุก คุณลักษณะเป็นการทดลองที่ต้องการเปรียบเทียบการใช้ ILDA ที่มีการปรับปรุงค่า S_b ในทุกคุณลักษณะร่วมกับการหาระยะทางมาฮาลาโนบิส
- วิธีการที่นำเสนอที่ทำการคัดเลือกคุณลักษณะด้วยสหสัมพันธ์แบบเพียร์สัน เป็นการทดลองที่ต้องการเปรียบเทียบการใช้ ILDA ที่มีการปรับปรุงค่า S_b และทำการคัดเลือกคุณลักษณะ

แสดงผลการทดลองดังตารางที่ 4.3 และ 4.4 เมื่อพิจารณาตาราง 4.3 พบว่า วิธีการที่นำเสนอมีค่าความถูกต้องสูงที่สุดเป็นร้อยละ 86.14 สูงกว่าวิธีการแบบไม่คัดเลือกคุณลักษณะ ซึ่งมีค่าร้อยละ 58.71 และต่ำที่สุดคือวิธีการ ILDA แบบดั้งเดิมจากผู้พัฒนาร่วมกับการหาระยะทางมาฮาลาโนบิส มีค่าร้อยละ 28.74 เมื่อพิจารณารายกลุ่มพบว่า กลุ่ม Tor และกลุ่ม Non-Tor ในวิธีการที่นำเสนอมีค่าประสิทธิภาพสูงที่สุดเช่นกัน พิจารณาค่า f-Measure กลุ่ม Tor มีค่าร้อยละ 57.97 และกลุ่ม Non-Tor มีค่าร้อยละ 91.70 ซึ่งในส่วนวิธีการอื่นที่เปรียบเทียบจะมีค่าประสิทธิภาพเมื่อแยกรายกลุ่มน้อยกว่าตามลำดับ

ตารางที่ 4.4 การเปรียบเทียบประสิทธิภาพระหว่างขั้นตอนวิธี ที่นำเสนอกับวิธีการของ ILDA ดั้งเดิม และวิธีการที่ไม่ทำการคัดเลือกคุณลักษณะของชุดข้อมูล Scenario A

วิธีการ	กลุ่ม	
การใช้ ILDA แบบดั้งเดิมร่วมกับกับการหาระยะทางมาฮาลาโนบิส	Tor	Non-Tor
Precision	14.27	100
Sensitivity	100	19.15
Specificity	19.15	100
f-Measure	24.98	32.14
Accuracy	28.74	
การใช้ ILDA ทุกคุณลักษณะ	Tor	Non-Tor
Precision	22.3	99.99
Sensitivity	99.96	53.16
Specificity	53.16	99.96
f-Measure	36.47	69.42
Accuracy	58.71	
วิธีการที่นำเสนอที่ทำการคัดเลือกคุณลักษณะด้วยสหสัมพันธ์แบบเพียร์สัน	Tor	Non-Tor
Precision	44.68	97.42
Sensitivity	82.52	86.62

ตารางที่ 4.4 การเปรียบเทียบประสิทธิภาพระหว่างขั้นตอนวิธี ที่นำเสนอกับวิธีการของ ILDA ดั้งเดิม และวิธีการที่ไม่ทำการคัดเลือกคุณลักษณะของชุดข้อมูล Scenario A (ต่อ)

Specificity	86.62	82.52
f-Measure	57.97	91.70
Accuracy	86.14	

ตารางที่ 4.5 การเปรียบเทียบประสิทธิภาพระหว่างวิธีการที่นำเสนอกับวิธีการของ ILDA ดั้งเดิม และวิธีการที่ไม่ทำการคัดเลือกคุณลักษณะของชุดข้อมูล Scenario B

Methods	กลุ่ม			
การใช้ ILDA ร่วมกับกับการหาระยะทางมาฮาลาโนบิส	กลุ่มที่ 1	กลุ่มที่ 2	กลุ่มที่ 3	กลุ่มที่ 4
Precision	50.31	87.58	70.21	45.32
Sensitivity	51.35	66.44	77.52	49.1
Specificity	87.36	96.99	80.32	86.64
f-Measure	50.82	75.56	73.68	47.13
Accuracy	64.39			
การใช้ ILDA ทุกคุณลักษณะ	กลุ่มที่ 1	กลุ่มที่ 2	กลุ่มที่ 3	กลุ่มที่ 4
Precision	50.31	85.49	71.4	48.13
Sensitivity	51.35	66.61	77.96	52.03
Specificity	87.36	96.39	81.31	87.35
f-Measure	50.82	74.88	74.54	50.00
Accuracy	65.13			
วิธีการที่นำเสนอที่ทำการคัดเลือกคุณลักษณะด้วยสหสัมพันธ์แบบเพียร์สัน	กลุ่มที่ 1	กลุ่มที่ 2	กลุ่มที่ 3	กลุ่มที่ 4
Precision	55.6	98.55	97.72	75.37
Sensitivity	78.38	93.32	75.97	81.31
Specificity	84.41	99.56	98.94	94
f-Measure	65.05	95.86	85.48	78.23
Accuracy	81.63			

พิจารณาตาราง 4.4 พบว่า วิธีการที่นำเสนอมีค่าความถูกต้องสูงที่สุดเป็นร้อยละ 81.63 สูงกว่าวิธีการแบบไม่คัดเลือกคุณลักษณะ ซึ่งมีค่าร้อยละ 65.13 และต่ำที่สุดคือวิธีการ ILDA แบบดั้งเดิมจากผู้พัฒนาร่วมกับการหาระยะทางมาฮาลาโนบิส มีค่าร้อยละ 64.39 เมื่อพิจารณารายกลุ่มพบว่า

ในวิธีการที่นำเสนอมีค่าประสิทธิภาพสูงที่สุดในทุกกลุ่มเช่นกัน พิจารณาค่า f-Measure กลุ่มกลุ่มที่ 1 ค่าร้อยละ 65.05 กลุ่มที่ 2 ค่าร้อยละ 95.86 กลุ่มที่ 3 ค่าร้อยละ 85.48 และ กลุ่มที่ 4 มีค่า

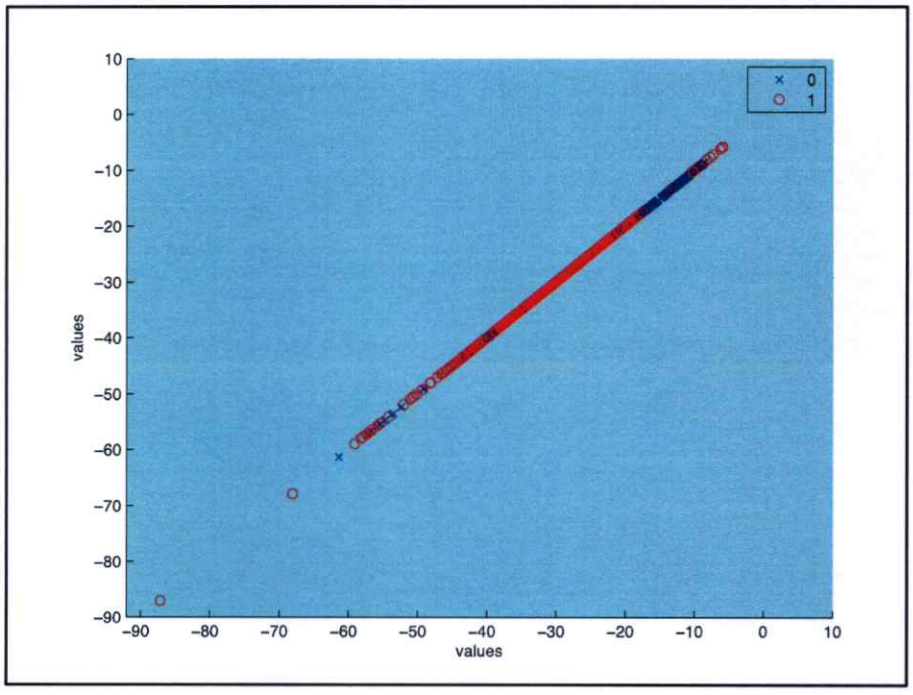
ร้อยละ 78.23 ซึ่งในส่วนวิธีการอื่นที่เปรียบเทียบจะมีค่าประสิทธิภาพเมื่อแยกรายกลุ่มน้อยกว่าตามลำดับในทุกกลุ่ม

4.2.2.2 การวิเคราะห์ค่าประสิทธิภาพระหว่างลำดับชั้นของหลายกลุ่ม

เมื่อพิจารณาถึงการตรวจจับประเภทของการบุกรุกแบบต่างๆที่เป็นลำดับชั้นในการทำการจำแนกแบบหลายกลุ่ม ของแต่ละประเภทของการบุกรุกรายกลุ่มในลำดับชั้นใดๆ โดยจะพิจารณาการวิเคราะห์แบบหนึ่งต่อกลุ่ม (1-vs-all) ในขั้นตอนหลังจากการฝึกสอนข้อมูล ซึ่งในขั้นตอนทดสอบข้อมูลที่ต้องการทำนาย จะต้องถูกคำนวณให้เป็น 1 มิติ แบบเส้นตรงแล้วจึงนำไปคำนวณหาระยะทางมาฮาลาโนบิส

ดังนั้น หากพิจารณาข้อมูลใดแต่ละลำดับชั้นเมื่อคำนวณเป็น 1 มิติ ก่อนหาระยะทางมาฮาลาโนบิส แสดงรูปข้อมูลแต่ละลำดับชั้นของกลุ่มที่ทำการจำแนกและกลุ่มที่เหลือดังต่อไปนี้

- ลำดับชั้นที่ 1 จำแนกกลุ่มที่ 1 และกลุ่มที่ไม่ใช่กลุ่มที่ 1 โดยกลุ่มที่ 1 แสดงวงกลมสีแดง แสดงดังรูปที่ 4.2

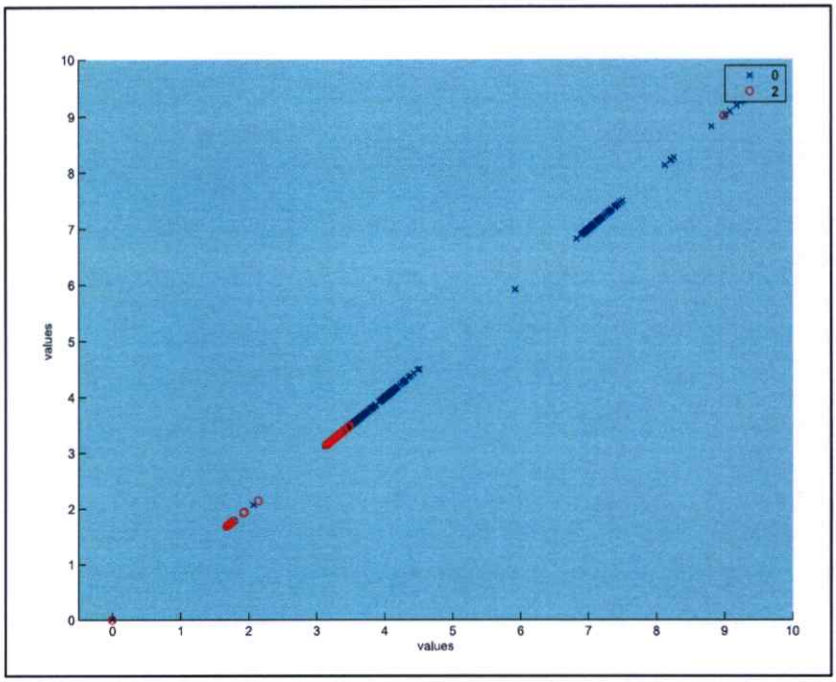


รูปที่ 4.2 ข้อมูลเมื่อถูกลดมิติของกลุ่ม 1 และ ไม่ใช่กลุ่ม 1

จากรูปที่ 4.2 พบว่า กลุ่มที่ 1 คือ Browsing มีการกระจายตัวของข้อมูลในแนวเส้นตรงทั้งที่มีทั้งจำแนกได้ชัดเจน และปะปนกับกลุ่มที่เหลืออยู่บางส่วน

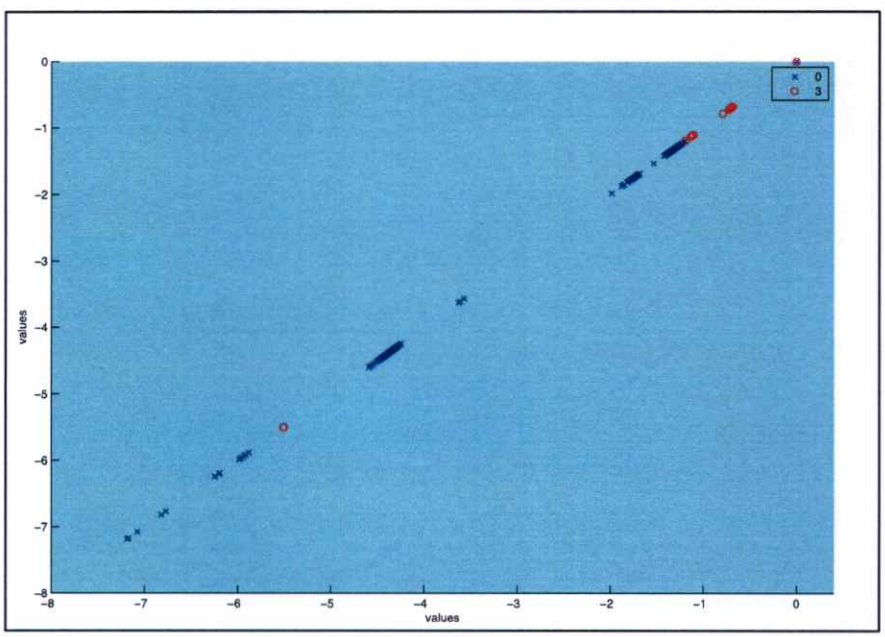
- ลำดับชั้นที่ 2 จำแนกกลุ่มที่ 2 และกลุ่มที่ไม่ใช่กลุ่มที่ 2 โดยกลุ่มที่ 2 แสดงวงกลมสีแดง แสดงดังรูป 4.3

จากรูปที่ 4.2 พบว่า กลุ่มที่ 2 คือ FTP และ P2P พบว่า การกระจายตัวของข้อมูลในแนวเส้นตรงทั้งที่มีระยะห่างของข้อมูลค่อนข้างชัดเจนบางระยะ แต่ข้อมูลค่อนข้างเกาะกลุ่มได้ดี และปะปนกับกลุ่มที่เหลืออยู่บางส่วน



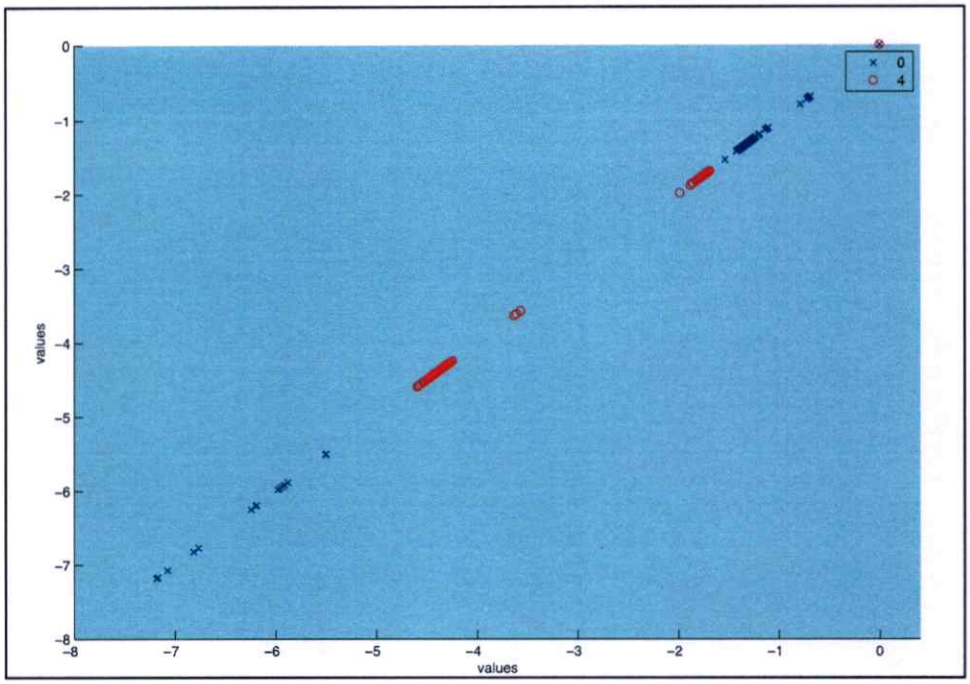
รูปที่ 4.3 ข้อมูลเมื่อถูกลดมิติของกลุ่ม 2 และ ไม่ใช่กลุ่ม 2

- ลำดับชั้นที่ 3 จำแนกกลุ่มที่ 3 และกลุ่มที่ไม่ใช่กลุ่มที่ 3 โดยกลุ่มที่ 3 แสดงวงกลมสีแดง แสดงดังรูปที่ 4.4



รูปที่ 4.4 ข้อมูลเมื่อถูกลดมิติของกลุ่ม 3 และ ไม่ใช่กลุ่ม 3

จากรูปที่ 4.3 พบว่า กลุ่มที่ 3 คือ Audio และ VOIP พบว่า การกระจายตัวของข้อมูลในแนวเส้นตรงทั้งที่มีระยะห่างของข้อมูลค่อนข้างชัดเจนหลายระยะ แต่ข้อมูลยังคงกลุ่มได้ดี และปะปนกับกลุ่มที่เหลืออยู่บางส่วน



รูปที่ 4.5 ข้อมูลเมื่อถูกลดมิติของกลุ่ม 4 และ ไม่ใช่กลุ่ม 4

- ลำดับชั้นที่ 3 จำแนกกลุ่มที่ 4 และกลุ่มที่ไม่ใช่กลุ่มที่ 4 โดยกลุ่มที่ 4 แสดงวงกลมสีแดง แสดงดังรูปที่ 4.5

จากรูปที่ 4.4 พบว่า กลุ่มที่ 3 คือ Chat Email และ Video พบว่า การกระจายตัวของข้อมูลในแนวเส้นตรงทั้งที่มีระยะห่างของข้อมูลค่อนข้างกระจายตัว เกาะกลุ่มค่อนข้างชันเจน แต่สลับตำแหน่งในแนวเส้นตรงเดียวอาจเนื่องจากมีกลุ่มย่อยปะปนในกลุ่มเดียวกัน

เมื่อข้อมูลได้ถูกเปลี่ยนเป็น 1 มิติแล้ว จากนั้นจึงทำการหาระยะทางมาฮาลาโนบิส แสดงค่าประสิทธิภาพ และค่าความ Error ดังต่อตารางที่ 4.6 ดังต่อไปนี้

ตารางที่ 4.6 การเปรียบเทียบประสิทธิภาพระในแต่ละลำดับชั้นของชุดข้อมูล Scenario B

ค่าประสิทธิภาพ	กลุ่ม	
	กลุ่ม 1	ไม่ใช่กลุ่ม 1
Precision	0.56	0.94
Sensitivity	0.78	0.84
Specificity	0.84	0.78
f-Measure	0.65	0.89
Accuracy	83.21	
Error	16.79	

ตารางที่ 4.6 การเปรียบเทียบประสิทธิภาพระในแต่ละลำดับชั้นของชุดข้อมูล Scenario B (ต่อ)

ค่าประสิทธิภาพ	กลุ่ม	
	กลุ่ม 2	ไม่ใช่กลุ่ม 2
Precision	0.99	0.98
Sensitivity	0.93	1
Specificity	1	0.93
f-Measure	0.96	0.99
Accuracy	98.05	
Error	1.95	
ค่าประสิทธิภาพ	กลุ่ม 3	ไม่ใช่กลุ่ม 3
Precision	0.91	0.87
Sensitivity	0.76	0.96
Specificity	0.96	0.76
f-Measure	0.83	0.91
Accuracy	88.39	
Error	11.61	
ค่าประสิทธิภาพ	กลุ่ม 4	ไม่ใช่กลุ่ม 4
Precision	0.84	0.96
Sensitivity	0.81	0.96
Specificity	0.96	0.81
f-Measure	0.82	0.96
Accuracy	93.70	
Error	6.30	

พิจารณาตารางที่ 4.5 พบว่า เมื่อคำนวณค่าประสิทธิภาพแยกตามรายกลุ่มของแต่ละลำดับชั้น พบว่าแต่ละลำดับชั้นมีค่าประสิทธิภาพสูงเกินร้อยละ 80 ทุกลำดับชั้น และลำดับชั้นที่ 2 มีค่า ค่าความถูกต้องสูงที่สุด คือ ร้อยละ 98.05 มีค่า Error 1.95 รองลงมาคือลำดับชั้นที่ 3 กลุ่มที่ 4 คือ ร้อยละ 93.70 มีค่า Error 6.30 ถัดมา ลำดับชั้นที่ 3 กลุ่ม 3 คือร้อยละ 88.39 มีค่า Error 11.61 และลำดับชั้นที่ 1 ร้อยละ 83.21 มีค่า Error 16.79 ตามลำดับ ในการพิจารณาค่า f-Measure แยกรายกลุ่ม ก็มีค่าประสิทธิภาพรายกลุ่มมีผลไปในทิศทางเดียวกัน

4.2.3 เปรียบเทียบประสิทธิภาพกับวิธีการอื่น

เมื่อทำการวิเคราะห์ประสิทธิภาพ เพื่อเปรียบเทียบกับวิธีการอื่นๆที่ใช้ในระบบตรวจจับการบุกรุกพบว่าขั้นตอนวิธีที่นำเสนอมีค่าประสิทธิภาพสูงที่สุดในการตรวจจับการบุกรุกทุกประเภทรายกลุ่มและมีค่าประสิทธิภาพรวมสูงที่สุดอีกด้วย แสดงดังตารางที่ 4.6 สำหรับชุดข้อมูล Scenario A และ ตารางที่ 4.7 สำหรับชุดข้อมูล Scenario B

ตารางที่ 4.7 การเปรียบเทียบประสิทธิภาพในการทำการจำแนกของชุดข้อมูล Scenario A

วิธีการ	ประสิทธิภาพ	Tor	Non-Tor
วิธีการที่นำเสนอ	Precision	44.68	97.42
	Sensitivity	82.52	86.62
	Specificity	86.62	82.52
	f-Measure	57.97	91.70
	Accuracy	86.14	
Decision Tree	Precision	11.86	100
	Sensitivity	100	0.07
	Specificity	0.07	100
	f-Measure	21.21	0.14
	Accuracy	11.92	
Naïve Bayes	Precision	10.86	87.49
	Sensitivity	36.3	59.9
	Specificity	59.9	36.3
	f-Measure	16.72	71.11
	Accuracy	57.11	
k-NN	Precision	16.55	97.66
	Sensitivity	93.49	36.6
	Specificity	36.6	93.49
	f-Measure	28.12	53.25
	Accuracy	43.35	
MLP	Precision	16.02	89.48
	Sensitivity	32.78	76.87
	Specificity	76.87	32.78
	f-Measure	21.52	82.70
	Accuracy	71.65	
SVM	Precision	3.03	86.63
	Sensitivity	3.73	89.93

ตารางที่ 4.7 การเปรียบเทียบประสิทธิภาพในการทำการจำแนกของชุดข้อมูล Scenario A (ต่อ)

Specificity	83.93	3.73
f-Measure	3.34	88.25
Accuracy	74.42	

จากตารางที่ 4.7 พบว่า วิธีการที่นำเสนอมีค่าประสิทธิภาพสูงที่สุด รองลงมาคือ MLP มีค่าค่าความถูกต้องร้อยละ 71.65 Naïve Bayes มีค่าร้อยละ 57.11 *k*-NN มีค่าร้อยละ 39.11 และ SVM มีค่าละ 33.33 ตามลำดับ เมื่อพิจารณาค่า f-Measure แยกรายกลุ่มพบว่า วิธีการที่นำเสนอมีค่าสูงที่สุดในทุกกลุ่มเช่นกัน

ตารางที่ 4.8 เปรียบเทียบประสิทธิภาพในการทำการจำแนกของชุดข้อมูล Scenario B

วิธีการ	ประสิทธิภาพ	กลุ่ม 1	กลุ่ม 2	กลุ่ม 3	กลุ่ม 4
วิธีการที่นำเสนอ	Precision	55.6	98.55	97.72	75.37
	Sensitivity	78.38	93.32	75.97	81.31
	Specificity	84.41	99.56	98.94	94
	f-Measure	65.05	95.86	85.48	78.23
	Accuracy	81.63			
Decision Tree	Precision	60.53	72.48	78.44	74.4
	Sensitivity	76.51	67.64	80.18	56.31
	Specificity	87.57	91.79	86.81	95.63
	f-Measure	67.59	69.98	79.30	64.10
	Accuracy	72.01			
Naïve Bayes	Precision	60	73.41	78.13	73.14
	Sensitivity	74.84	67.12	80.29	57.66
	Specificity	87.57	92.23	86.55	95.22
	f-Measure	66.60	70.12	79.20	64.48
	Accuracy	71.85			
k-NN	Precision	55.13	66.87	75.98	51.14
	Sensitivity	59.25	57.02	87.93	40.54
	Specificity	87.99	90.97	83.37	91.26
	f-Measure	57.12	61.55	81.52	45.23
	Accuracy	66.00			
MLP	Precision	66.49	62.93	80.31	67.38

ตารางที่ 4.8 เปรียบเทียบประสิทธิภาพในการทำการจำแนกของชุดข้อมูล Scenario B (ต่อ)

วิธีการ	ประสิทธิภาพ	กลุ่ม 1	กลุ่ม 2	กลุ่ม 3	กลุ่ม 4
MLP	Sensitivity	50.73	90.62	91.25	48.76
	Specificity	93.63	89.89	86.61	91.4
	f-Measure	57.55	74.28	85.43	56.58
	Accuracy	71.72			
SVM	Precision	63.42	99.38	92.68	55.87
	Sensitivity	66.32	82.53	77.08	84.68
	Specificity	90.47	99.84	96.36	84.91
	f-Measure	64.84	90.17	84.16	67.32
	Accuracy	77.65			

จากตารางที่ 4.8 พบว่า วิธีการที่นำเสนอมีค่าประสิทธิภาพสูงที่สุด รองลงมาคือ Decision Tree มีค่าความถูกต้อง ร้อยละ 72.01 Naïve Bayes มีค่าร้อยละ 71.85 *k*-NN มีค่าร้อยละ 66.00 MLP มีค่าร้อยละ 53.28 และ SVM มีค่าละ 46.64 ตามลำดับ เมื่อพิจารณาค่า f-Measure แยกรายกลุ่มพบว่า วิธีการที่นำเสนอมีค่าสูงที่สุดในทุกกลุ่ม ยกเว้น กลุ่มที่ 1 มีค่าประสิทธิภาพน้อยกว่า Decision Tree และ Naïve Bayes เพียงกลุ่มเดียว แต่ประสิทธิภาพโดยรวมยังคงสูงกว่า ซึ่งจากการวิเคราะห์ประสิทธิภาพจะพบว่า SVM มีประสิทธิภาพต่ำที่สุดในในการทำการจำแนก ด้วยข้อมูล Tor ที่มีการเข้ารหัสหลายชั้น

4.2.4 เปรียบเทียบประสิทธิภาพเมื่อใช้ฐานข้อมูลการบุกรุกอื่น

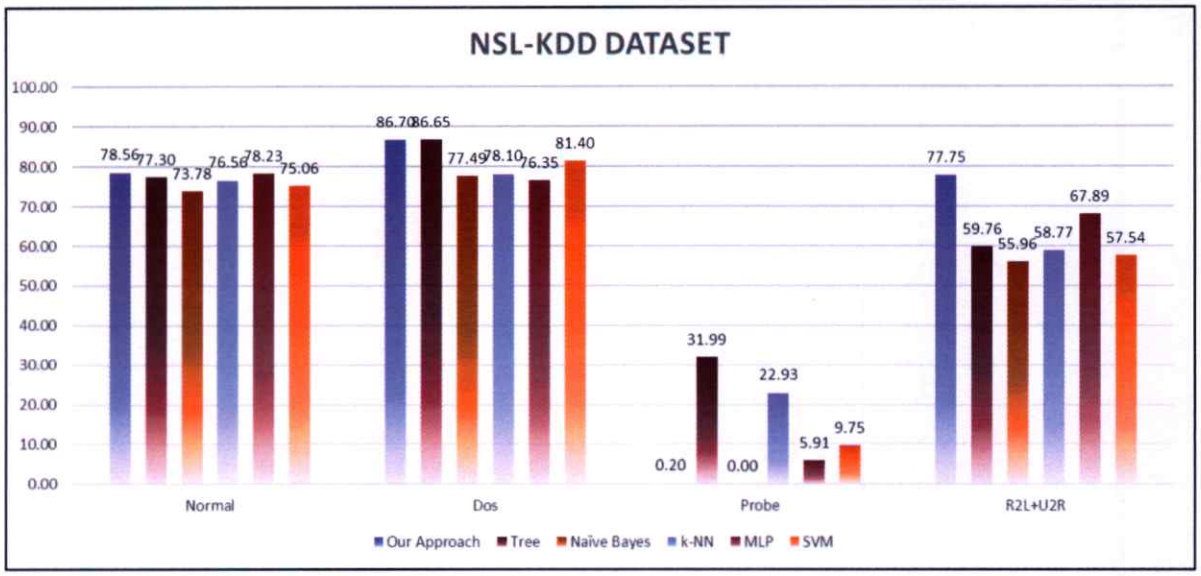
ในการทดลอง ได้ประสิทธิภาพของวิธีการที่นำเสนอกับข้อมูลการบุกรุกเครือข่ายคอมพิวเตอร์ในรูปแบบอื่นๆ ได้แก่ ชุดข้อมูล NSL-KDD เป็นการบุกรุกบนเครือข่ายที่มีประเภทของสิ่งผิดปกติ ได้แก่ การปฏิเสธการใช้บริการแบบกระจาย (Distributed-Denial of Service) การบุกรุกด้วยการตรวจสอบระบบ (Probing) การเรียกบริการระยะไกล (Remote to Local) และการแอบใช้สิทธิ์สูงสุดของระบบ (User to Root) ถัดมาเป็นชุดข้อมูล SAME เป็นการบุกรุกระบบปฏิบัติบนโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอย (Android) รุ่นที่ 5 มีกลุ่มปกติ และกลุ่มไม่ปกติ Phishing Dataset เป็นการส่งจดหมายหลอกหลวงผู้ใช้งานด้วยการขอข้อมูลส่วนตัวทางอีเมล์ มีกลุ่ม Phishing และไม่ใช่กลุ่ม Phishing และ Spambase เป็นชุดข้อมูลการส่งแสปมเพื่อก่อกวนผู้ใช้งานระหว่างกลุ่ม Spam และ ไม่ใช่ Spam และ UNSW-NB15 (UNSW, 2018) เป็นข้อมูลการโจมตีบนเครือข่ายประกอบไปด้วย 9 ประเภทของการบุกรุกย่อย ได้แก่ Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode และ Worms มีผลการเปรียบเทียบประสิทธิภาพดังตารางที่ 4.9

ตารางที่ 4.9 เปรียบเทียบประสิทธิภาพในการทำการจำแนกของชุดข้อมูล Scenario B

ชุดข้อมูล	วิธีการ/ ค่าความถูกต้อง					
	วิธีการที่นำเสนอ	C4.5	Naïve Bayes	k-NN	MLP	SVM
NSL-KDD	75.71	74.73	68.12	71.38	70.16	70.86
SAME	96.04	95.38	93.23	95.38	92.65	92.57
Phishing	91.05	89.93	87.13	83.54	91.19	71.30
Spambase	85.00	82.61	43.36	72.83	78.11	84.86
UNSW-NB15	85.02	73.71	82.02	66.80	69.54	75.17

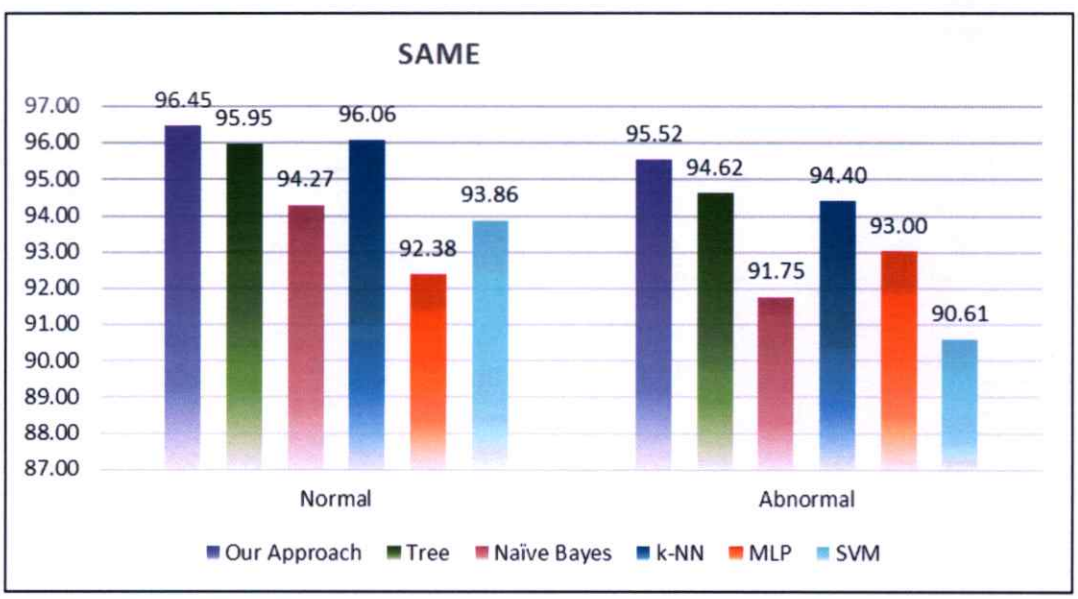
จากตาราง 4.9 พบว่า วิธีการที่นำเสนอมีค่าความถูกต้อง สูงที่สุด 4 ใน 5 ชุดข้อมูลที่ใช้ในการเปรียบเทียบ ได้แก่ NSL-KDD Dataset มีค่าความถูกต้อง ร้อยละ 75.71 ชุดข้อมูล SAME มีค่าความถูกต้อง ร้อยละ 96.04 ชุดข้อมูล Spambase มีค่าค่าความถูกต้อง ร้อยละ 85.00 และ UNSW-NB15 มีค่าค่าความถูกต้องร้อยละ 85.02 ในส่วนของชุดข้อมูล Phishing มีค่าความถูกต้องร้อยละ 91.05 ซึ่ง MLP มีค่าความถูกต้องสูงที่สุดเป็นร้อยละ 91.19

หากพิจารณาค่าการเปรียบเทียบ f-Measure ของวิธีการที่นำเสนอเปรียบเทียบกับวิธีการของการเรียนรู้ของเครื่องอื่นๆ แยกตามกลุ่มของการบุกรุกพบว่า สำหรับชุดข้อมูล NSL-KDD มีค่า f-Measure ในการตรวจจับกลุ่ม Normal DoS U2R และ R2L ได้สูงที่สุด แต่กลุ่ม Probing ตรวจจับได้ไม่ดีนัก โดย Decision Tree มีค่าที่สูงที่สุดในการตรวจจับการบุกรุกกลุ่มนี้ แต่ภาพรวมโดยค่าความถูกต้องยังคงสูงที่สุดอยู่ แสดงดังรูปที่ 4.6



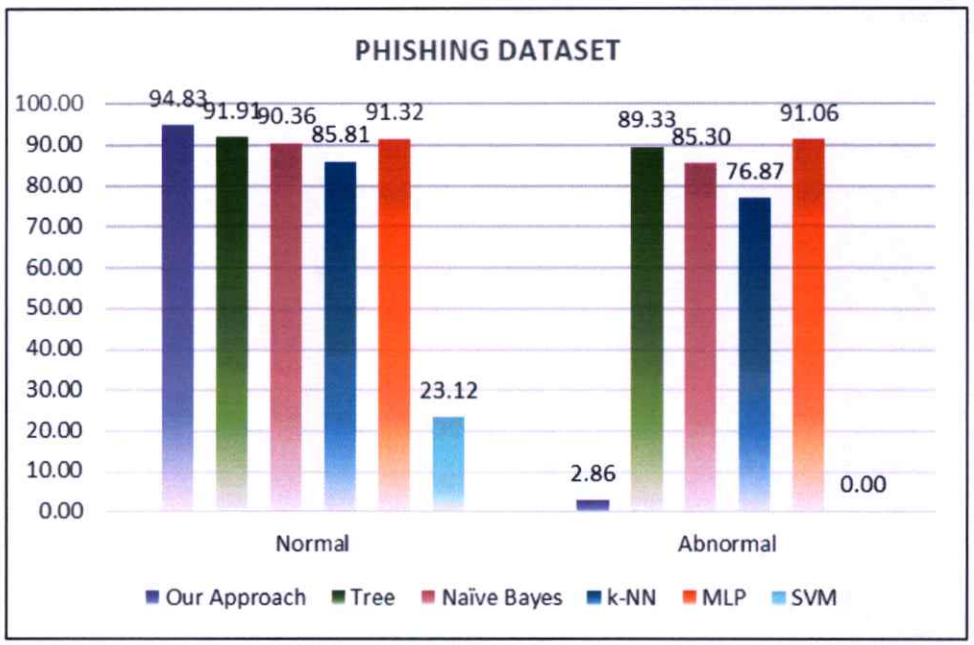
รูปที่ 4.6 เปรียบเทียบค่า f-Measure ของชุดข้อมูล NSL-KDD

พิจารณาชุดข้อมูล SAME พบว่าวิธีการที่นำเสนอมีค่า f-Measure สูงที่สุดในทุกกลุ่ม ซึ่งโดยส่วนใหญ่มีค่าสูงเกินร้อยละ 90 ซึ่งแม้ทุกวิธีการจะตรวจจับได้ค่าที่สูงแต่วิธีการที่นำเสนอยังคงมีประสิทธิภาพที่สูงที่สุดอยู่เมื่อพิจารณารายกลุ่มทั้งยังสามารถเรียนรู้แบบเพิ่มเติมได้อีกด้วย



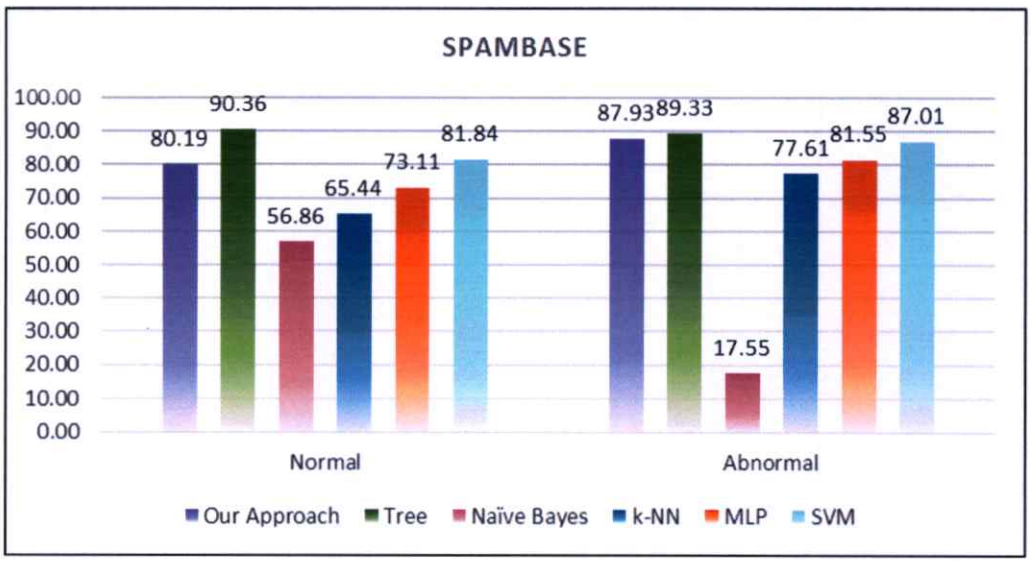
รูปที่ 4.7 เปรียบเทียบค่า f-Measure ของชุดข้อมูล SAME

พิจารณาชุดข้อมูล Phishing พบว่าวิธีการที่นำเสนอมีค่า f-Measure สูงที่สุดในกลุ่ม Normal แต่กลุ่ม Phishing มีค่าต่ำเพียงร้อยละ 2.86 ซึ่งชุดข้อมูลนี้ แม้วิธีการ MLP จะมีค่าความถูกต้องสูงที่สุดก็ตาม แต่หากพิจารณารายกลุ่มพบว่าวิธีการที่นำเสนอยังคงจะมีค่า f-Measure ที่สูงที่สุด Normal



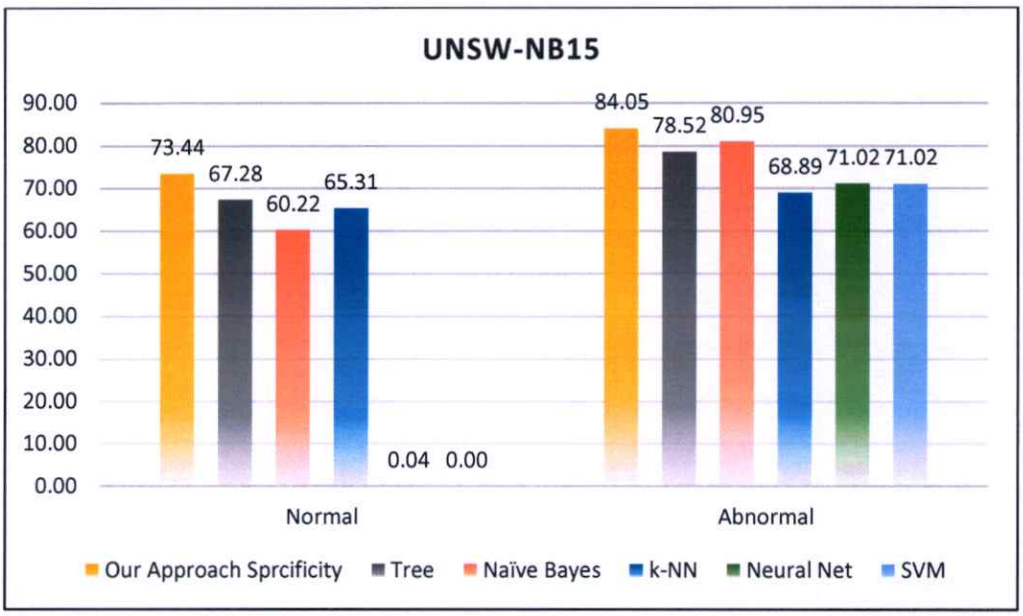
รูปที่ 4.8 เปรียบเทียบค่า f-Measure ของชุดข้อมูล Phishing

พิจารณาชุดข้อมูล Spambase พบว่าวิธีการที่นำเสนอมีค่า f-Measure ไม่ได้สูงที่สุดในสองกลุ่ม แต่เมื่อพิจารณากลุ่ม Normal มีค่าร้อยละ 80.19 ซึ่งยังคงสูงอยู่เป็นลำดับที่ 3 รองจาก Decision Tree และ SVM และกลุ่ม Spam มีค่าสูงเป็นลำดับที่สองรองจาก Decision Tree



รูปที่ 4.9 เปรียบเทียบค่า f-Measure ของชุดข้อมูล Spambase

พิจารณาชุดข้อมูล UNSW-NB15 พบว่าวิธีการที่นำเสนอมีค่า f-Measure สูงที่สุดในทุกกลุ่ม ซึ่งชุดข้อมูลนี้ กลุ่ม Abnormal ประกอบไปด้วยการบุกรุกย่อยๆหลายกลุ่ม ซึ่งจะพบได้ว่า MLP และ SVM มีความสามารถในการตรวจจับที่ต่ำมากเพียงร้อยละ 0.04 และ 0.00 เท่านั้นตามลำดับ



รูปที่ 4.10 เปรียบเทียบค่า f-Measure ของชุดข้อมูล UNSW-NB15

จากการวิเคราะห์ค่าประสิทธิภาพกับการบุกรุกชนิดอื่นๆพบว่า ขั้นตอนวิธีที่นำเสนอ มีประสิทธิภาพในการทำการจำแนก โดยรวมสูงที่สุดเมื่อพิจารณาจากค่าความถูกต้อง ซึ่งหากพิจารณาค่า f-Measure แยกตามกลุ่ม ประสิทธิภาพยังคงสูงอยู่แม้มีบางประเภทของการบุกรุกในบางชุดข้อมูลที่อาจมีขั้นตอนวิธีอื่นที่มีประสิทธิภาพสูงกว่าเล็กน้อย

4.3 ผลการวิจัยการเรียนรู้แบบผสม

4.3.1 ผลการทำการคัดเลือกคุณลักษณะของชุดข้อมูล Tor Scenario A ของการเรียนรู้แบบผสม

ในขั้นตอนการทำการคัดเลือกคุณลักษณะได้ใช้วิธีการคัดเลือกคุณลักษณะบนพื้นฐานของความสัมพันธ์ในการคัดเลือกคุณลักษณะ ซึ่งในการทดลองจะแยกระหว่าง Tor Scenario A ซึ่งเป็นชุดข้อมูล แบบสองกลุ่มและ Tor Scenario B จะเป็นชุดข้อมูลแบบหลายกลุ่ม ผลการทดลองพบว่า ชุดข้อมูล Tor Scenario A ได้คัดเลือกคุณลักษณะมาได้จำนวน 5 คุณลักษณะ และชุดข้อมูล Tor Scenario B ได้คัดเลือกคุณลักษณะมาได้จำนวน 6 คุณลักษณะ จากการเรียงลำดับคุณลักษณะคุณลักษณะที่มีความสัมพันธ์กันจากมากไปน้อย แล้วจึงทำคุณลักษณะเหล่านั้นทดสอบกับขั้นตอนวิธีที่นำเสนอเพื่อดูผลว่าคุณลักษณะใดบ้างที่ทำให้มีค่าประสิทธิภาพสูงที่สุด แสดงดังตารางที่ 4.10

4.3.2 ประสิทธิภาพของการจำแนกกลุ่มด้วยการเรียนรู้แบบผสม

เมื่อทำการวิเคราะห์ประสิทธิภาพ เพื่อเปรียบเทียบกับวิธีการอื่นๆที่ใช้ในระบบตรวจจับการบุกรุก พบว่าขั้นตอนวิธีที่นำเสนอ มีค่าประสิทธิภาพสูงที่สุดในการตรวจจับการบุกรุกทุกประเภทรายกลุ่ม และมีค่าประสิทธิภาพรวมสูงที่สุดอีกด้วย แสดงดังตารางที่ 4.11 สำหรับชุดข้อมูล Scenario A และ ตารางที่ 4.12 สำหรับชุดข้อมูล Scenario B

ตารางที่ 4.10 ผลการคัดเลือกคุณลักษณะของของชุดข้อมูล Tor Scenario A สำหรับการเรียนรู้แบบผสม

ชุดข้อมูล	ลำดับที่ของคุณลักษณะ	ชื่อคุณลักษณะ	ชุดข้อมูล	ลำดับที่ของคุณลักษณะ	ชื่อคุณลักษณะ
Scenario A	2	Source Port	Scenario B	1	Source IP
	5	Protocol		2	Source Port
	7	Flow Bytes/s		7	Flow Bytes/s
	14	Fwd IAT Std		11	Flow IAT Max
	17	Bwd IAT Mean		16	Fwd IAT Min
				20	Bwd IAT Min

ตารางที่ 4.11 ผลการเปรียบเทียบประสิทธิภาพการในการจำแนกของชุดข้อมูล Tor Scenario A สำหรับการเรียนรู้แบบผสม

วิธีการ	กลุ่ม	อัตราการตรวจจับ (Detection Rate)	False Positive Rate	ค่าความถูกต้อง
k-NN	Tor	15.64	1.13	37.36
	Non Tor	98.87	84.36	
C4.5	Tor	11.87	0	11.95
	Non Tor	100.00	88.13	
LDA	Tor	17.43	0	43.85
	Non Tor	100.00	82.57	
MLP	Tor	43.07	0	84.32
	Non Tor	100.00	56.93	
SVM	Tor	8.62	12.02	84.13
	Non Tor	87.98	91.38	
วิธีการที่นำเสนอ	Tor	99.97	0	99.98
	Non Tor	100	0.03	

จากตารางที่ 4.11 พบว่าวิธีการที่นำเสนอมีประสิทธิภาพด้านอัตราการตรวจจับกลุ่ม Tor เป็นร้อยละ 99.97 และอัตราการตรวจจับกลุ่ม Non Tor เป็นร้อยะ 100 มีค่าความถูกต้องเป็น 99.98 ซึ่งสูงที่สุด รองลงมาคือ MLP มีค่าความถูดต้อง 84.32 และ SVM 84.13 ตามลำดับ ต่ำที่สุดคือ C4.5 มีค่าความถูกต้องเพียงร้อยละ 11.95

ตารางที่ 4.12 ผลการคัดเลือกคุณลักษณะของของชุดข้อมูล Tor Scenario A สำหรับการเรียนรู้แบบผสม

วิธีการ	กลุ่ม	อัตราการตรวจจับ	False Positive Rate	ค่าความถูกต้อง
k-NN	AUDIO	66.24	2.8	66.79
	BROWSING	57.84	10.72	
	CHAT	16.05	3.6	
	FILE-TRANSFER	42.42	8.41	
	MAIL	27.69	2.85	
	P2P	87.22	0.54	
	VIDEO	43.42	7.46	
	VOIP	82.90	1.12	
C4.5	AUDIO	57.41	3.02	73.51
	BROWSING	72.40	14.65	
	CHAT	53.53	0.27	
	FILE-TRANSFER	100.00	7.99	
	MAIL	100.00	0.47	
	P2P	86.79	0.15	
	VIDEO	49.55	2.23	
	VOIP	85.52	0.12	
MLP	AUDIO	67.02	6.37	60.98
	BROWSING	52.00	1.89	
	CHAT	6.45	3.99	
	FILE-TRANSFER	61.46	8.65	
	MAIL	53.85	2.8	
	P2P	80.60	0.05	
	VIDEO	36.28	3.86	
	VOIP	98.91	16.09	
SVM	AUDIO	32.90	3.24	74.17
	BROWSING	83.28	10.15	
	CHAT	39.53	2.71	
	FILE-TRANSFER	100.00	8.15	

ตารางที่ 4.12 ผลการคัดเลือกคุณลักษณะของของชุดข้อมูล Tor Scenario A สำหรับการเรียนรู้แบบผสม (ต่อ)

วิธีการ	กลุ่ม	อัตราการตรวจจับ	False Positive Rate	ค่าความถูกต้อง
	MAIL	97.44	1.98	
	P2P	99.38	0.19	
	VIDEO	99.13	1.56	
	VOIP	77.22	0.59	
LDA	AUDIO	43.45	6.75	26.87
	BROWSING	57.99	15.17	
	CHAT	59.32	2.63	
	FILE-TRANSFER	28.27	8.83	
	MAIL	21.43	3.42	
	P2P	23.98	0.09	
	VIDEO	0.00	11.65	
	VOIP	0.00	30.68	
วิธีการที่นำเสนอ	AUDIO	100	0	100
	BROWSING	100	0	
	CHAT	100	0	
	FILE-TRANSFER	100	0	
	MAIL	100	0	
	P2P	100	0	
	VIDEO	100	0	
	VOIP	100	0	

จากตารางที่ 4.12 พบว่าวิธีการที่นำเสนอมีประสิทธิภาพด้านอัตราการตรวจจับเป็นร้อยละ 100 ในทุกประเภทของการบุกรุก และมีค่าความถูกต้องเป็นร้อยละ 100 ซึ่งสูงที่สุด รองลงมาคือ SVM มีค่าความถูดต้อง 74.17 และ C4.5 มีค่า 73.51 ตามลำดับ ต่ำที่สุดคือ LDA มีค่าความถูกต้องเพียงร้อยละ 26.87

บทที่ 5
สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

การตรวจจับการบุกรุกทางเครือข่ายเป็นสิ่งที่มีความจำเป็นในปัจจุบัน วิธีการตรวจจับการบุกรุกแบบเดิมบนระบบตรวจจับการบุกรุกมีการวิเคราะห์จราจรเครือข่ายด้วยขั้นตอนวิธีที่ไม่สามารถเรียนรู้เพิ่มเติมด้วยตนเองได้ตลอดเวลา เช่น การใช้กฎในการตรวจจับบน Snort หรือใช้การเรียนรู้แบบมีผู้สอน เป็นต้น ซึ่งทำให้เมื่อมีการบุกรุกหรือสิ่งผิดปกติแบบใหม่เข้ามาในระบบ ความสามารถในการตรวจจับและความแม่นยำของระบบจึงมีค่าเท่ากับโมเดลที่ระบบได้รู้จำ งานวิจัยนี้จึงได้นำเสนอขั้นตอนวิธีการเรียนรู้ตามลำดับชั้นแบบเพิ่มเติมได้สำหรับความปลอดภัยของเครือข่ายบนพื้นฐานของ ILDA ซึ่งจะเป็นการพัฒนาขั้นตอนวิธีที่สามารถเรียนรู้แบบเพิ่มเติมได้ตลอดเวลาที่มีการรับข้อมูลเข้ามา และเกิดการคำนวณเก็บค่าซึ่งเป็นตัวแทนของข้อมูลทั้งหมดไว้รอคำนวณต่อไป จะไม่นำโมเดลมาใช้ร่วมคำนวณอีกเหมือนการเรียนรู้แบบมีผู้สอนทั่วไป โดยขั้นตอนวิธีการเรียนรู้ตามลำดับชั้น จะสามารถตรวจจับการบุกรุกแต่ละประเภทได้เป็นลำดับชั้นเพื่อหาประเภทของการบุกรุกที่เข้ามาระบบ หากตรวจจับเจอในลำดับชั้นใด ก็จะวิเคราะห์ว่าเป็นการบุกรุกประเภทนั้นๆ

ในส่วนของการเรียนรู้แบบเพิ่มเติมได้ จะมีการทำการคัดเลือกคุณลักษณะด้วยสหสัมพันธ์แบบเพียร์สัน เพื่อลดจำนวนของคุณลักษณะเพื่อให้เหลือเพียงคุณลักษณะ ที่มีความสัมพันธ์กันเพื่อให้มีประสิทธิภาพในการทำการจำแนกที่สูง และลดภาระการประมวลผลอีกด้วย ผลจากการทำการคัดเลือกคุณลักษณะพบว่า การเลือกคุณลักษณะที่เหมาะสมช่วยให้มีประสิทธิภาพในการทำการจำแนกที่สูงขึ้น และขั้นตอนวิธีที่นำเสนอ มีประสิทธิภาพในการตรวจจับการบุกรุกสูงที่สุด มีค่าความถูกต้องของชุดข้อมูล Tor A เป็นร้อยละ 86.14 พิจารณาตามกลุ่มพบว่ามีค่า f-Measure กลุ่ม Tor เป็นร้อยละ 57.97 และกลุ่ม Non-Tor เป็นร้อยละ 91.70 ในส่วนของชุดข้อมูล Tor B เป็นร้อยละ 81.63 พิจารณาตามกลุ่มพบว่ามีค่า f-Measure กลุ่มที่ 1 เป็นร้อยละ 65.05 กลุ่มที่ 2 เป็นร้อยละ 95.86 กลุ่มที่ 3 เป็นร้อยละ 85.48 และ กลุ่มที่ 4 เป็นร้อยละ 78.23

การทดลองกับชุดข้อมูล ของการบุกรุกอื่นๆพบว่า วิธีการที่นำเสนอมีค่าความถูกต้องสูงที่สุดเมื่อเปรียบเทียบกับวิธีการอื่นๆ เมื่อพิจารณารายกลุ่มพบว่าโดยส่วนใหญ่มีค่า f-Measure สูงกว่าวิธีการอื่นๆ แต่มีบางกลุ่มที่อาจมีค่าต่ำกว่า อาจเนื่องมาจากลักษณะกลุ่มย่อยใน Dataset ที่มีรูปแบบ (Pattern) ในการบุกรุกแตกต่างกัน วิธีอื่นอาจมีความเหมาะสมกว่ากับการตรวจจับในรูปแบบนั้นๆ

ในส่วนของการเรียนรู้แบบผสมจะมีการทำการคัดเลือกคุณลักษณะด้วยวิธีการคัดเลือกคุณลักษณะบนพื้นฐานของความสัมพันธ์ ผลการทดลองพบว่าผลจากการทำการคัดเลือกคุณลักษณะพบว่า การเลือกคุณลักษณที่เหมาะสมช่วยให้มีประสิทธิภาพในการทำการจำแนกที่สูงขึ้นและขั้นตอนวิธีที่นำเสนอมีประสิทธิภาพในการตรวจจับการบุกรุกสูงที่สุด มีค่าความถูกต้อง ของชุดข้อมูล Tor A เป็นร้อยละ 99.98 พิจารณาตามกลุ่มพบว่ามีค่าอัตราการตรวจจับ กลุ่ม Tor เป็นร้อยละ 99.97 และกลุ่ม Non-Tor เป็นร้อยละ 100 ในส่วนของชุดข้อมูล Tor B เป็นร้อยละ พิจารณาตามกลุ่มได้แก่ AUDIO BROWSING CHAT FILE-TRANSFER MAIL P2P VIDEO และ VOIP พบว่ามีค่าทุกกลุ่มมีค่าอัตราการตรวจจับร้อยละ 100 ทั้งมด

5.2 อภิปรายผลการวิจัย

เมื่อวิเคราะห์ผลการทดลอง พบว่าในขั้นตอนการทำการคัดเลือกคุณลักษณะนั้น จะต้องหาคุณลักษณที่มีความสัมพันธ์กันกับข้อมูลที่มีอยู่ ดังนั้นในกรณีที่มีการเรียนรู้แบบเพิ่มเติมได้อย่างต่อเนื่อง แล้วมีข้อมูลใหม่ซึ่งอาจมีลักษณะข้อมูลเป็นการบุกรุกแบบใหม่เข้ามา ระบบจะยังคงสามารถเรียนรู้และปรับปรุงโมเดลได้ แต่นั่นหมายถึงคุณลักษณที่คัดเลือกไว้อาจไม่สามารถยืนยันได้ว่า เป็นคุณลักษณสัมพันธ์กันดีและเหมาะสมอยู่เพราะข้อมูลเกิดการปรับปรุงไปอย่างต่อเนื่อง เมื่อเกิดการรับเข้ามาคำนวณจากเครือข่ายคอมพิวเตอร์ในการใช้งานจริง

ชุดข้อมูลของการบุกรุกเครือข่ายที่นำมาใช้ในการทดลอง เป็นตัวแทนของการบุกรุกที่ใหม่และหลากหลายบนเครือข่ายคอมพิวเตอร์ ประสิทธิภาพในการตรวจจับในภาพรวมการบุกรุกแม้จะดีที่สุด แต่หากวิเคราะห์โดยละเอียด อาจมีบางกลุ่มของการบุกรุกที่มีวิธีการอื่นๆ วิเคราะห์ประเภทได้ดีกว่า อาจเนื่องมาจาก การบุกรุกบางประเภท เช่น Probing เป็นการ Ping Scan เพื่อเป็นการสำรวจเครือข่าย ในคาบเวลาที่ยาวนาน และมีแพคเก็ตที่น้อย ซึ่งวิธีการที่นำเสนอจะทำการเรียนรู้แบบเพิ่มเติมได้ตลอดเวลาอาจทำให้ระบบเกิดการคำนวณข้อมูล Probing ในคาบเวลาที่ยาวนานจากการปรับปรุงข้อมูลเดิม ทำให้การตรวจจับสามารถกระทำได้ยาก หากมีข้อมูลการบุกรุกกลุ่มอื่นเข้ามาปะปนแล้วเกิดการเรียนรู้ไปเป็นกลุ่มอื่นก่อนหน้า

5.3 ข้อเสนอแนะ

ควรพัฒนาให้ระบบสามารถตรวจจับการบุกรุกกับข้อมูลการบุกที่มีรูปแบบไม่ตายตัว มีปริมาณข้อมูลน้อย หรือสามารถตรวจจับการบุกรุกที่มีกลุ่มแตกต่างกันมากได้ (Class Imbalance) ได้

เอกสารอ้างอิง

Abdelhamid, N., Ayesh, A., and Thabtah, F. 2014. "Phishing detection based associative classification data mining." *Expert Systems with Applications*. 41(13): 5948-5959.

Aburomman, A. A., and Reaz, M. B. I. 2016. "Ensemble of binary SVM classifiers based on PCA and LDA feature extraction for intrusion detection." 636-640. In **2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)**. China: IEEE.

Alheeti, K. M. A., Gruebler, A., and McDonald-Maier, K. 2017. "Using discriminant analysis to detect intrusions in external communication for self-driving vehicles." *Digital Communications and Networks*. *3*(3). 180-187.

Anderson, J. P. 1980. **Computer security threat monitoring and surveillance**. Technical Report. Fort Washington: James P. Anderson Company.

Ashfaq, R. A. R., Wang, X. Z., Huang, J. Z., Abbas, H., & He, Y. L. 2017. "Fuzziness based semi-supervised learning approach for intrusion detection system." *Information Sciences*. 378: 484–497.

Bhosale, D., and Ade, R. 2015. "Intrusion detection using incremental learning from streaming imbalanced data." *International Journal of Managing Public Sector Information and Communication Technologies*. *6*(1): 9-20.

Bohara, A., Thakore, U., & Sanders, W. H. 2016. "Intrusion Detection in Enterprise Systems by Combining and Clustering Diverse Monitor Data.". 7–16. in **Proceedings of the Symposium and Bootcamp on the Science of Security**. Pittsburgh, Pennsylvania: ACM.

Brereton, R. G. and Loyd, G. R. 2010. "Support Vector Machines for classification and regression," *Analyst.* 135(2): 230-267.

Canadian Institute for Cybersecurity. 2017. or-nonTor dataset (ISCXTor2016). [Online]. Available : https://www.unb.ca/cic/datasets/tor.html.

Datti, R., and Verma, B. 2010. "B.: Feature reduction for intrusion detection using linear discriminant analysis." *International Journal on Engineering Science and Technology*. 1072–1078.

Datti, R., and Lakhina, S. 2012. "Performance Comparison of Features Reduction Techniques for Intrusion Detection System," 3(1): 332-335.

De Maesschalck, R., Jouan-Rimbaud, D., and Massart, D. L. 2000. "The mahalanobis distance." *Chemometrics and intelligent laboratory systems*, 50(1): 1-18.

Dietterich, T. G. 2002. Ensemble learning. *The handbook of brain theory and neural networks*, *2*, 110-125.

Eid, H. F., Hassanien, A. E., Kim, T. H., & Banerjee, S. 2013. Linear

correlation-based feature selection for network intrusion detection model. 240-248. In **International Conference on Security of Information and Communication Networks**. Heidelberg: Springer

Freund, Y. and Schapire, R.E.1996. "Experiments with a new boosting algorithm" In: **Proceedings of the Thirteenth International Conference**. 96. 148-156.

Galar, M., A. Fernandez, E. Barrenechea, and H. Bustince. 2014. "A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches." *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 42 (4): 463–84.

Galit, S. , Nitin, R. P. and Peter C. B. 2010. **Data Mining for Business Intelligence: Concepts, Techniques, and Applications in Microsoft Office Excel with XLMiner**: Wiley Publishing

Gao, Q., Huang, Y., Gao, X., Shen, W., and Zhang, H. 2015. "A novel semi-supervised learning for face recognition". *Neurocomputing*. 152: 69-76.

Ghafir, I., Svoboda, J., and Prenosil, V. 2014. "Tor-based malware and Tor connection detection." 1-6. in **International Conference on Frontiers of Communications, Networks and Applications**. Malaysia: ACM

Hall, M. A. 1999. "Correlation-based feature selection for machine learning." Phd. dissertation, University of Waikato.

He, G., Yang, M., Luo, J., & Gu, X. 2015. "A novel application classification attack against Tor." *Concurrency and Computation: Practice and Experience*. 27(18): 5640-5661.

He, H., Chen, S., Li, K., & Xu, X. 2011. "Incremental learning from stream data." *IEEE Transactions on Neural Networks*. *22*(12): 1901-1914.

Jabbar, M. A., Aluvalu, R., & Reddy, S. 2017. "Cluster based ensemble classification for intrusion detection system." 253-257. In **Proceedings of the 9th International Conference on Machine Learning and Computing**. Singapore: ACM.

Jensen, F. V. 1996. **Introduction to Bayesian Networks**. New York: Springer-Verlag Inc.,

Jin, L., Ding, K., and Huang, Z. 2010. "Incremental learning of LDA model for Chinese writer adaptation." *Neurocomputing*. 73(10-12): 1614-1623.

Ji, S. Y., Jeong, B. K., Choi, S., & Jeong, D. H. 2016. "A multi-level intrusion detection method for abnormal network behaviors." *Journal of Network and Computer Applications*. 62: 9-17.

Kumar, K. S., and Anitha Mary MO Chacko. 2016."Clustering Algorithms for Intrusion

Detection: A Broad Visualization." 135:1–135:4. in **Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies**. Udaipur, India: ACM.

Kumar, P. A. R., & Selvakumar, S. 2013. "Detection of distributed denial of service attacks using an ensemble of adaptive and hybrid neuro-fuzzy systems." *Computer Communications*. 36(3): 303-319.

Kuncheva, L. I., and J. J. Rodríguez. 2014. "A weighted voting framework for classifiers ensembles." *Knowledge and Information Systems*. 38 (2): 259–75.

Lashkari, A. H., Draper-Gil, G., Mamun, M. S. I., & Ghorbani, A. A. 2017. "Characterization of Tor Traffic using Time based Features." 253-262. In 3rd **International Conference on Information Systems Security and Privacy**. Portugal: SCITEPRESS

Lena, D., Pietro, and Margara, L. 2010. "Optimal Global Alignment of Signals by Maximization of Pearson Correlation." *Information Processing Letters*. 110(16): 679–686.

Miller, S. T., and Busby-Earle, C. 2017. "Multi-perspective machine learning a classifier ensemble method for intrusion detection." 7-12. In **Proceedings of the 2017 International Conference on Machine Learning and Soft Computing**. Vietnam: ACM.

Pang, S., Ozawa, S., and Kasabov, N. 2005. "Incremental linear discriminant analysis for classification of data streams." *IEEE Transactions on Systems, Man, and Cybernetics*. 35(5): 905-914.

Pang, S., Peng, Y., Ban, T., Inoue, D., and Sarrafzadeh, A. 2015. "A federated network online network traffics analysis engine for cybersecurity."1-8. In **2015 International Joint Conference on Neural Networks (IJCNN)**. Ireland: IEEE.

Pfleeger, C. P. & Pfleeger, S. L. 2013. **Analyzing Computer Security**. New Jersey: Pearson Education, Inc.

Quinlan, J. R. 1993. **C4.5: programs for machine learning.** Morgan Kaufmann Publishers Inc., 1993.

Rhodes-Ousley, M. 2013. **Information security**. 2nd Edition. USA: McGraw-Hill.

Ren, Y. 2014. "An integrated intrusion detection system by combining SVM with adaboost." *Journal of Software Engineering and Applications*. 7(12): 1031–38.

Saad, A. A., Khalid, C., & Mohamed, J. 2015. "Network intrusion detection system based on Direct LDA." 1-6. In **2015 Third World Conference on Complex Systems (WCCS)**. Morocco: IEEE.

Sathya, S. S., Ramani, R. G., and Sivaselvi, K. 2011. "Discriminant analysis based feature selection in kdd intrusion dataset". *International Journal of computer applications*. *31*(11): 1-7.

Shearer, C. 2000. "The CRISP-DM model: The new blueprint for data mining." *Journal of Data Warehousing*. 5(4): 13–22

Simon, H. 1998. **Neural Networks: A Comprehensive Foundation**. India: Prentice Hall PTR.

Stallings, W. and Brown, L. 2015. **Computer Security Principle and Practice**. London:Pearson Education, Inc

Subba, B., Biswas, S., and Karmakar, S. 2015. "Intrusion detection systems using linear discriminant analysis and logistic regression." 1-6. In **2015 Annual IEEE India Conference (INDICON)**. India: IEEE.

Syarif, I., Prugel-Bennett, A., & Wills, G. "Unsupervised Clustering Approach for Network Anomaly Detection." 135–145. In ***International Conference on Networked Digital Technologies***. Springer, Berlin: Heidelberg.

Tor. 2018. The Tor Project. [Online]. Available : https://www.torproject.org/about/overview.html.en.

UNSW Canberra Cyber. 2018. The UNSW-NB15 Dataset. [Online]. Available: https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/.

Whitman, M. E. and Mattord, H. J. 2012. **Principles of information security** (4thed.). Cengage Learning. Boston, USA: Cengage Learning.

Wurzenberger, M., Skopik, F., Landauer, M., Greitbauer, P., Fiedler, R., and Kastner, W. 2017. "Incremental clustering for semi-supervised anomaly detection applied on log data." 31-36. In **Proceedings of the 12th International Conference on Availability, Reliability and Security**. Italy: ACM.

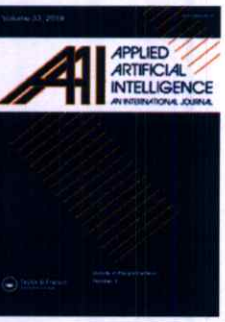
Xue, Z., Shang, Y., & Feng, A. 2010. "Semi-supervised outlier detection based on fuzzy rough C-means clustering." *Mathematics and Computers in Simulation*. 80(9): 1911-1921.

Yuan, Y., Kaklamanos, G., and Hogrefe, D. 2016. "A novel semi-supervised Adaboost technique for network anomaly detection." 111-114. In **Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems**. Malta: ACM.

ภาคผนวก ก
ผลงานวิจัยตีพิมพ์ที่ Applied Artificial Intelligence

Journal homepage: https://www.tandfonline.com/loi/uaai20

Sornsuwit, P., & Jaiyen, S. 2019. "A New Hybrid Machine Learning for Cybersecurity Threat Detection Based on Adaptive Boosting". *Applied Artificial Intelligence*, 33(5), 462-482.



Taylor & Francis
Taylor & Francis Group

Applied Artificial Intelligence
An International Journal

ISSN: 0883-9514 (Print) 1087-6545 (Online) Journal homepage: https://www.tandfonline.com/loi/uaai20

A New Hybrid Machine Learning for Cybersecurity Threat Detection Based on Adaptive Boosting

Ployphan Sornsuwit & Saichon Jaiyen

To cite this article: Ployphan Sornsuwit & Saichon Jaiyen (2019) A New Hybrid Machine Learning for Cybersecurity Threat Detection Based on Adaptive Boosting, Applied Artificial Intelligence, 33:5, 462-482, DOI: 10.1080/08839514.2019.1582861

To link to this article: https://doi.org/10.1080/08839514.2019.1582861

Published online: 01 Mar 2019.

APPLIED ARTIFICIAL INTELLIGENCE
2019, VOL. 33, NO. 5, 462–482
https://doi.org/10.1080/08839514.2019.1582861

Taylor & Francis
Taylor & Francis Group

A New Hybrid Machine Learning for Cybersecurity Threat Detection Based on Adaptive Boosting

Ployphan Sornsuwit and Saichon Jaiyen

Advanced Artificial Intelligence Research Laboratory, Department of Computer Science, Faculty of Science, King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand

ABSTRACT

A hybrid machine learning is a combination of multiple types of machine learning algorithms for improving the performance of single classifiers. Currently, cyber intrusion detection systems require high-performance methods for classifications because attackers can develop invasive methods and evade the detection tools. In this paper, the cyber intrusion detection architecture based on new hybrid machine learning is proposed for multiple cyber intrusion detection. In addition, the correlation-based feature selection is adopted for reducing the irrelevant features and the weight vote of adaptive boosting that is adopted to combine multiple classifiers is concentrated. In the experiments, UNB-CICT or network traffic dataset is used for evaluating the performance of the proposed method. The results show that the proposed method can achieve higher efficiency in every attack type detection. Furthermore, the experiments with Phishing website dataset UNSW-NB 15 dataset NSL-KDD dataset and KDD Cup'99 dataset are also conducted, and the results show that the proposed method can produce higher efficiency as well.

Introduction

In recent years, many applications of computer and network technologies have been used and added in daily life, including the use of data privacy, government data, or business data. Cybersecurity has become more important to prevent intrusion into systems. In the past, setting up a security policy on a firewall may not have enough protection against these intrusions because the invasion of new forms has been developed, including using the weaknesses of the operating system, including the settings in communication between networks. However, we can detect the malfunction as well as prevent the intrusion by using the Intrusion Detection System (IDS) (Anand and Patel 2012). Presently, intrusion focuses on commercial interests because there are many activities that are security risks, and important activities such as transferring money online, sending important files through emails or social networks, etc. Although there is

CONTACT Ployphan Sornsuwit ployphan.en@gmail.com

the use of https protocol, that does not mean that it can be protected completely, for example, website phishing activities that trick individuals into registering personal information, DDoS attacks to stop services of target machines, etc. One major attack that evades firewalls and IDS/IPS is Tor or "The Onion Router" which is a distributed overlay network designed to anonymize TCP-based applications like web browsing, secure shell, and instant messaging (Dingledine, Mathewson, and Syverson 2004). This is a service created to allow people to surf the Internet without revealing themselves. The user will need to connect to a network of other middleware that will hide the IP address from the website visited as a private route, so no one can trace your usage, even with Tor users it can be hard to detect. Currently, this is a challenge for these securities.

Machine Learning has been made for practical usage to enhance the detection capabilities of IDS, but still cannot detect them all. There still exist some errors (AbdElrahman and Abraham 2014; Amer, Goldstein, and Abdennadher 2013; Mascaro, Nicholson, and Korb 2014; Sagha et al. 2013; Sheykhkanloo 2014). Moreover, hybrid and ensemble systems are also used to increase the capability of the traditional IDS. The research (Aburomman and Reaz 2017) on the detection of abnormalities found that the ensemble implementation of IDS detection enhancements has been developed in two ways: the homogeneous ensemble method and the heterogeneous ensemble method. In the homogeneous ensemble method, a weak learner is used in the same way, but heterogeneous ensemble method will choose a different weak learner. Both methods must boost the weak learner to combine decisions to achieve better final results than the single learner. This is achieved when testing by doing the classification in each research by using the different methods. Voting found that homogeneous ensembles can frequently classify some classes into which the difficult class is. For heterogeneous ensemble, it has low false alarm detection but both still have the same disadvantage which cannot detect the new irregularities.

Therefore, through these studies, we can see that machine learning is widely used in classification problems. The problem of multiple intrusion detection can be considered as a multiclass classification problem. So, the objective of our research is to develop new effective Adaboost algorithm to classify multiclass intrusions by using UNB-CIC Tor Network Traffic dataset (UNB, 2017). In addition, a new hybrid classifier is developed for IDS dataset in which features are collected by correlation-based selection. The selected features will be trained with multiple weak learners and build a strong hypothesis by voting.

The rest of this paper is arranged as follows. "Review of Related Work" is a section describing recent related researches. "Ensemble Learning" and "correlation-based feature selection" are described briefly. Concepts of the two algorithms are proposed. "Proposed Method" presents an algorithm

and proposed hybrid method. "Experimental Results" section shows our experiment. The final section is "Conclusion and Future Work".

Related Work

Based on the research in the past, many current researchers have developed various studies to detect malfunctions on network-based investigations and applied a variety of machine learning techniques in anomaly detection (AbdElrahman and Abraham 2014; Amer, Goldstein, and Abdennadher 2013; Mascaro, Nicholson, and Korb 2014; Sagha et al. 2013; Sheykhkanloo 2014), mostly to improve classification efficiency. For example, Hussain and Lalmuanawma tested various methods of hybrid systems with different feature selections with a 4.5 weak learner which was adapted to the Adaboost algorithm. The experiment showed that the wrapper method and Adaboost with decision tree with weak learners gave the best efficiency (Hussain and Lalmuanawma 2014). Wahba et al. proposed hybrid feature selection methods by combining correlation-based and information-gained in selecting relevant features and classification steps using Adaboost.M1 with Naïve Bayes weak learners, the result showed a good detection rate and a low false positive rate (Wahba, ElSalamouny, and ElTaweel 2015). Aburomman and Reaz proposed a novel combination of multiple experts (SVM, k-NN, PSO) into one ensemble algorithm, they combined all results from different experts by using a weighted majority vote. The result showed that the novel approach gave better accuracy than other methods (Aburomman and Reaz 2016). Michael et al. proposed supervised machine learning with meta-classifiers, the results showed that the bagging with REPTree weak learners was more capable in predicting than other meta-classifiers (Michael, Kumaravel, and Chandrasekar 2015). Nejad and Abadi developed a security system with IG and GR feature reduction and applied features in Adaboost methods, IG and Adaboost with random tree gave the better performance than other methods (Nejad and Abadi 2014).

Other machine learnings and hybrid methods were improved performance of Adaboost such as SVM (Ren 2014), Neuro-Fuzzy (Kumar and Selvakumar 2013) or new weight vote framework (Kuncheva and Rodríguez 2014). In addition, classification is conducted by using Adaboost.m1, which is a multiclass boosting tool which is used to improve classification methods and show satisfactory performance more than other ensemble methods. Most of this research tries to improve the performance of Adaboost methods with several weak learners, but most of these are not effective to detect multiclass intrusion (Zhang and Xie 2010). However, there are various intruding ways and behaviors that avoid network detection, and conceal or prevent communication in order to make it difficult to trace internet activity or fraudulent websites, etc. Thus, some studies

made efforts to develop detection algorithms: Hodo et al. presented the process of classification of Tor Traffic and Non-tor traffic to monitor the activity and security of the user's usage. The researchers have compared the quality of classification with Artificial Neural Network and Support Vector Machine using UNB-CIC TOR Network Traffic dataset and resulted in the usage of Correlation-based feature selection (CFS) and can select 10 features then classify them with Artificial Neural Network. The results of this study had an accuracy of 99.8% (Hodo et al. 2014). The research of Ghafir, Svoboda, and Prenosil also presented a methodology for detecting Tor by applying our methodology on campus live traffic and showed that it can automatically detect Tor connections (Ghafir, Svoboda, and Prenosil 2014b). Some studies use hybrid feature selection with Mbox2xml tools to extract features, then use Bays Net Algorithm as a Classifier to analyze whether this is phishing email or not. Results found that when select features are left to only eight features and accuracy in classifying is as high as 94% (Hamid, Abawajy, and Kim 2013) as with the research (Abdelhamid, Ayesh, and Thabtah 2014) developed a Multi-label Classifier based Associative Classification (MCAC). This was used to classify phishing using Chi-square feature selection method to select features for the test compared to other machine learning and found that using MCAC has a higher accuracy and can detect a new class called "Suspicious" that was not originally in the training data set.

Ensemble Learning

Boosting is an important method in ensemble learning, Boosting is the method which was involved with the creation of different ensembles from many weak learners that were combined these weak learners into a single strong learner. The idea of Boosting, when we have distribution from various weak learners may have the answer of class that is correct or incorrect. Boosting can combine them to achieve single strong learners which is the final correct answer, according to the following procedure (Zhou 2012)

Adaboost.M1 is an extension of the original adaptive boosting method. It is extended to multiclass boosting with a different weight changing mechanism (Galar et al. 2014). The key idea of Adaboost.M1 is that it will update the distribution weights of samples that are classified by the current hypothesis. In Adaboost.M1, the weak learner requires errors less than 0.5% before adding to the ensemble. Adaboost.M1 will concentrate on difficulty classified instances by increasing weights of incorrectly classified samples. The details of this algorithm are shown in algorithm 1.

Algorithm 1: Adaboost.M1

Input: sequence of m examples $(x_1, y_1), \ldots, (x_m, y_m)$, $x_i \in X$, with labels $y_i \in Y = \{1, \ldots, k\}$

Weak learning algorithm (**Weaklearn**)

Integer T specifying number of iterations

Initialize $D_1(i) = 1/m$ for all$i$

Do for $t = 1.2, \ldots, T$

1. Call **Weaklearn** and provide it with the distribution D_t
2. Get back a hypothesis $h_t : X \rightarrow Y$
3. Calculate the error of h_t: $\varepsilon_t = \sum_{i:h_t(x_i) \neq y_i} D_t(i)$ if $\varepsilon_t > 1/2$ then set $T = T - 1$ and abort loop
4. Set $\beta_t = \varepsilon_t/(1 - \varepsilon_t)$
5. Update distribution D_t : $D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} \beta_t \text{ if } h_t(x_i) = y_i \\ 1 \text{ otherwise} \end{cases}$

   Where Z_t is a normalization constant (Chosen so that D_{t+1} will be a distribution)

Output the final hypothesis: $h_{fin} = \underset{yY}{argmax} \sum_{t:h_t(x)=y} \log \frac{1}{\beta_t}$

Correlation-Based Feature Selection

Correlation-based feature selection (CFS) is a principle of screening and ranking subgroups according to the relationship between features and classes by the good subgroups of features, it will have a high correlation with a class that will be selected for using in predicting the answer for class. In the case of features which have no correlation in redundant information should be eliminated as well.

Network Traffic dataset contains attributes that are correlated with each other, so we need to select only the high relationship features by using correlation-based feature selection. Correlation-based feature selection evaluates subsets of features by considering the individual predictive ability of each feature along with the degree of redundancy between them. Subsets of features that are highly correlated with the class while having low intercorrelation are preferred (Hall 1999).

The correlation can be calculated as

$$Merit_s = \frac{k\bar{r}_{cf}}{\sqrt{k + k(k-1)\bar{r}_{ff}}} \tag{1}$$

where

$Merit_s$ is the correlation between the summed components and the outside variable.

k is the number of components.

$\bar{r}_{cf}$ is the mean feature-class correlation ($f \epsilon S$)

$\bar{r}_{ff}$ is the average inter-correlation between components.

The heuristic metrics $\bar{r}_{zi}$ and $\bar{r}_{ii}$ are computed as the symmetrical uncertainty (SU)

$$SU = 2.0 \times \left[\frac{H(X) + H(Y) - H(X,Y)}{H(X) + H(Y)}\right] \tag{2}$$

where $H(X)$ is defined as entropy that can be calculated as

$$H(X) = -\sum_{x \epsilon X} p(x) log_2(p(x)) \tag{3}$$

Proposed Method

The proposed model will use five machine learning techniques as weak learners to build the model and combine the model to be the final hypothesis with Adaboost.M1 and then evaluate the efficiency of the algorithm and comparison. The processes of training the proposed model are consisting of four stages which are data preprocessing, hybrid weak classifier training, strong classifier training, and performance evaluation as shown in Figure 1. The first stage is the data pre-processing. Firstly, some symbolic features must be converted to numeric features such as Source IP and Destination I because they cannot be calculated by machine learning algorithms. Then, the correlation-based feature selection is applied for selecting the relevant features in the dataset in order to reduce the number of features. The second stage is to train various classifiers with the training set. In this stage, five classifiers including k-NN, C4.5, MLP, SVM, and LDA are adopted to build the weak classifiers. Each classifier is effective for detecting each type of intrusion. The third stage is to build a strong classifier by Adaboost.M1 (Freund and Schapire 1996). The final stage is to evaluate the performance of the classifier. The main idea is to build a strong classifier from various types of weak classifiers by adopting Adaboost.M1 (Galar et al. 2014). In Adaboost.M1 algorithms, the weak classifiers are the same type, and the strong classifier is built by the combination of the same weak classifiers. In our proposed method, the combination of the same type of weak classifiers is changed to the combination of various types of weak classifiers.

After learning processes, $\beta_t$will be obtained from every weak leaner as $\beta_1 - \beta_5$. After that, β_t is sent for calculation in the testing process. Testing data will be exploited to classify it with five weak learners to get a hypothesis h_t from $h_1 - h_5$. In this process, β_t and h_t will be employed to vote with the method of Adaboost.M1. The proposed hybrid machine learning for detecting cybersecurity intrusions is shown in Figure 2. This new model is designed

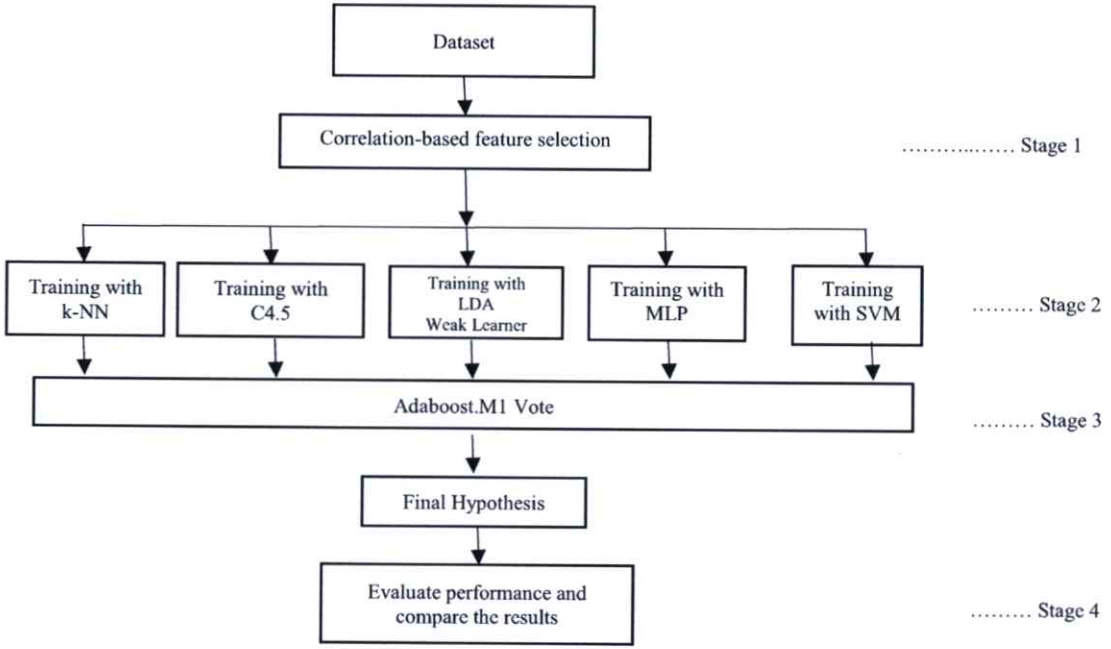


Figure 1. The learning processes of proposed hybrid machine learning.

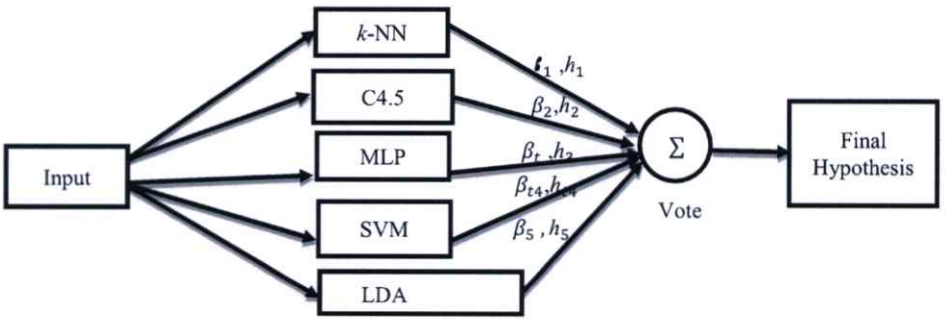


Figure 2. The proposed hybrid machine learning model.

to increase the efficiency of cybersecurity threat detection according to the ability of hybrid machine learning algorithms and correlation-based feature selection as mentioned above.

Experimental Results

Our research dataset was created from real-world traffic, defined as a set of tasks that created users, including Alice and Bob, to use different applications such as Skype, Facebook, and so forth to capture the traffic that occurs during eight service communications, which includes audio, browsing, chat, file transfer, mail, P2P, video, and VOIP (Lashkari et al. 2017) used in the dataset for experiments which contain two scenarios: Scenario A and Scenario B, in which both scenarios are different. Scenario A has two classes which are interested in classifying normal traffic and Tor traffic, but Scenario B is interested in classifying characterization of all eight services of Tor traffic as mentioned above. The details of the two scenarios can be found in Tables 1 and 2 as follows.

In addition, experiments will use the cybersecurity dataset to confirm the performance of our proposed algorithms, with a variety of attacks and dataset with traditional and existing intrusion including Phishing website, UNSW-NB15, NSL-KDD and KDD Cup'99.

Data Preprocess and Feature Selection

In the pre-processing step, the text features that are converted into numeric features are proto, service and state. In this process, the correlation-based

Table 1. Number of data in Scenario A.

Class	No. of data	Description
Tor	8,044	Tor Traffic
Nontor	59,790	Normal Traffic
Total	67,843	Tor and Normal Traffic

Table 2. Number of data in Scenario B.

Class	No. of data	Description
Browsing	1,604	HTTP and HTTPS traffic is generated by the user with two web browsers; Firefox and Chrome.
Email	282	Traffic sample created with Thunderbird client, both Alice and Bob accounts using Gmail in which the clients will send of SMTP/S and received with POP3/SSL.
Chat	323	Send Instant Message and specify a chat label for applications such as ICQ, Skype, IAM as well as Facebook and Hangout (on the web browser).
Audio	721	The traffic of the streaming data that is labelled as audio is stored in Spotify.
File Transfer	864	The traffic used to receive and send files, FTP over SSH (SFTP) and FTP over SSL (FTPS).
P2P	1,085	The traffic sharing using a protocol like Bit Torrent, which in order to generate traffic download the .torrent file and then capture the session traffic.
Video	874	The traffic of the video data that is labelled Video, which will be collected from YouTube and Vimeo Services.
VOIP	2,291	It is a VOIP application that is stored from Voice Call from Facebook, Skype and Hangout Application.
Total	8,044	All classes ofTor in Scenario B

feature selection is applied to select the relevant features, because it is an effective way to detect intrusion (Bahl and Sharma 2015; Eid et al. 2013; Nguyen, Franke, and Petrovic 2014; Shahbaz et al. 2016; Zhang et al. 2017). After pre-processing, we divide the UNB-CIC Tor Network Traffic dataset into Scenario A and Scenario B to the training dataset and testing dataset of 70:30, which gives the amount of data for both scenarios shown in Table 3. There are six features selected from this process, to be found in Table 4.

Performance Evaluation

In the experimental results, the comparative efficiency between single classifiers and the proposed multiple classifiers are done by using efficiency: precision, detection rate, specificity, FPR, f-Measure, and accuracy. Tables 5 and 6 shows confusion matrix of Scenario A and scenario B. Tables 7 and 8 show the efficiency analysis value from the confusion matrix from both Scenario A and scenario B. Based on the analysis, the analysis yielded 100% efficiency for Scenario B, which means that it is very efficient to classify Tor traffic. Comparing performance with other machine learning methods has been made for regular classification finds that the method we offer for detector intrusion efficiency is higher than Scenario A and Scenario B between weak learners before a vote and model and after a vote, as shown in Tables 9 and 10, which is most efficient when compared to other methods of Scenario A and scenario B.

In addition, our research offers comparisons with other Intrusion Datasets. The results show a comparison of Phishing web performance (Abdelhamid, Ayesh, and Thabtah 2014) in Table 11, UNSW-NB15 (Moustafa and Slay

Table 3. The details of the dataset for the Training dataset and the Testing dataset.

	Scenario A		Scenario B	
Class	Class	No. of data	Class	จำนวน
Training	Tor Traffic	5631	Audio	505
	Normal Traffic	1853		
			Browsing	1,123
			Chat	226
			File-Transfer	605
			Mail	197
			P2P	760
			Video	612
			VOIP	1,604
Testing	Tor Traffic	2413	Audio	216
	Normal Traffic	17937		
			Browsing	481
			Chat	97
			File-Transfer	259
			Mail	85
			P2P	325
			Video	262
			VOIP	687
	Total	67,843	Total	8,044

Table 4. The six features from the feature selection process.

Dataset	Number of features in dataset	Features name	Dataset	Number of features in dataset	Features name
Scenario A	2	Source Port	Scenario B	1	Source IP
	5	Protocol		2	Source Port
	7	Flow Bytes/s		7	Flow Bytes/s
	14	Fwd IAT Std		11	Flow IAT Max
	17	Bwd IAT Mean		16	Fwd IAT Min
				20	Bwd IAT Min

Table 5. The confusion matrix of our proposed method Scenario A.

	Predict Class	
Actual Classes	Normal	Tor
Normal	17937	0
Tor	5	2408

Table 6. The confusion matrix of our proposed method from Scenario B.

	Predict Class							
Actual Classes	Audio	Browsing	Chat	File-transfer	Mail	P2P	Video	VOIP
Audio	216	0	0	0	0	0	0	0
Browsing	0	481	0	0	0	0	0	0
Chat	0	0	97	0	0	0	0	0
File-transfer	0	0	0	259	0	0	0	0
Mail	0	0	0	0	85	0	0	0
P2P	0	0	0	0	0	325	0	0
Video	0	0	0	0	0	0	262	0
VOIP	0	0	0	0	0	0	0	687

Table 7. The efficiency of all classifiers when training with UNB-CIC Tor Network Traffic dataset for Scenario A form this table between weak learner before vote and model after vote.

Classifier		TP	FP	FN	TN	Precision	Detection Rate	Specificity	FPR	f-Measure	Accuracy
k-NN	Normal	17,759	178	167	2,246	99.01	99.07	92.66	7.34	99.04	98.03
	Tor	2,246	167	178	17,759	93.08	92.66	99.07	0.93	92.87	
C4.5	Normal	17,916	21	32	2,381	99.88	99.82	99.13	0.87	99.85	99.74
	Tor	2,381	32	21	17,916	98.67	99.13	99.82	0.18	98.90	
LDA	Normal	17,100	837	1391	1,022	95.33	92.48	54.98	45.02	93.88	89.05
	Tor	1,022	1391	837	17,100	42.35	54.98	92.48	7.52	47.85	
MLP	Normal	17,536	401	742	1,671	97.76	95.94	80.65	19.35	96.84	94.38
	Tor	1,671	742	401	17,536	69.25	80.65	95.94	4.06	74.52	
SVM	Normal	16,591	1346	53	2,360	92.5	99.68	63.68	36.32	95.96	93.19
	Tor	2,360	53	1346	16,591	97.8	63.68	99.68	0.32	77.14	
Our Approach	**Normal**	**17,937**	**0**	**5**	**2,408**	**100**	**99.97**	**100**	**0**	**99.98**	**99.98**
	Tor	**2408**	**5**	**0**	**17937**	**99**	**100**	**99.97**	**0.03**	**99.49**	

Table 8. The efficiency of all classifiers when training with UNB-CIC Tor Network Traffic dataset in scenario B from this table between weak learner before vote and model after vote.

Classifier		TP	FP	FN	TN	Precision	Detection Rate	Specificity	FPR	f-Measure	Accuracy
k-NN	AUDIO	166	50	37	2,159	76.85	81.77	97.74	2.26	79.23	91.83
	BROWSING	425	56	78	1,835	88.36	84.49	97.07	2.93	86.38	
	CHAT	84	13	9	2,306	86.6	90.32	99.44	0.56	88.42	
	FILE-TRANSFER	236	23	31	2,122	91.12	88.39	98.93	1.07	89.73	
	MAIL	61	24	8	2,319	71.76	88.41	98.98	1.02	79.22	
	P2P	317	8	7	2,080	97.54	97.84	99.62	0.38	97.69	
	VIDEO	240	22	23	2,127	91.6	91.25	98.98	1.02	91.42	
	VOIP	686	1	4	1,721	99.82	99.42	99.94	0.06	99.62	
C4.5	AUDIO	212	4	0	2,196	98.15	100	99.82	0.18	99.07	99.46
	BROWSING	478	3	5	1,926	99.38	98.96	99.84	0.16	99.17	
	CHAT	97	0	1	2,314	100	98.98	100	0	99.49	
	FILE-TRANSFER	259	0	0	2,153	100	100	100	0	100.00	
	MAIL	84	1	3	2,324	98.82	96.55	99.96	0.04	97.67	
	P2P	324	1	0	2,087	99.69	100	99.95	0.05	99.84	
	VIDEO	262	0	4	2,146	100	98	100	0	98.99	
	VOIP	683	4	0	1,725	99.42	100	99.77	0.23	99.71	
LDA	AUDIO	107	109	120	2,076	49.54	47.14	95.01	4.99	48.31	74.72
	BROWSING	267	214	119	1,812	55.51	69.17	89.44	10.56	61.59	
	CHAT	97	0	25	2,290	100	79	100	0	88.27	
	FILE-TRANSFER	248	11	64	2,089	95.75	79.49	99.48	0.52	86.87	
	MAIL	53	32	28	2,299	62.35	65.43	98.63	1.37	63.85	
	P2P	324	1	7	2,080	99.69	97.89	99.95	0.05	98.78	
	VIDEO	12	250	0	2,150	4.58	100	89.58	10.42	8.76	
	VOIP	687	0	254	1,471	100	73	100	0	84.39	

(*Continued*)

Table 8. (Continued).

Classifier		TP	FP	FN	TN	Precision	Detection Rate	Specificity	FPR	f-Measure	Accuracy
MLP	AUDIO	190	26	144	2,052	87.96	56.89	98.75	1.25	69.09	89.97
	BROWSING	332	149	24	1,907	69.02	93.26	92.75	7.25	79.33	
	CHAT	97	0	0	2,315	100	100	100	0	100.00	
	FILE-TRANSFER	259	0	4	2,149	100	98	100	0	98.99	
	MAIL	79	6	2	2,325	92.94	97.53	99.74	0.26	95.18	
	P2P	323	2	19	2,068	99.38	94.44	99.9	0.1	96.85	
	VIDEO	208	54	5	2,145	79.39	97.65	97.54	2.46	87.58	
	VOIP	682	5	44	1,681	99.27	93.94	99.7	0.3	96.53	
SVM	AUDIO	216	0	324	1,872	100	40	100	0	57.14	81.97
	BROWSING	167	314	28	1,903	34.72	85.64	85.84	14.16	49.41	
	CHAT	97	0	3	2,312	100	97	100	0	98.48	
	FILE-TRANSFER	235	24	23	2,130	90.73	91.09	98.89	1.11	90.91	
	MAIL	5	80	3	2,324	5.88	62.5	96.67	3.33	10.75	
	P2P	322	3	2	2,085	99.08	99.38	99.86	0.14	99.23	
	VIDEO	260	2	12	2,138	99.24	95.59	99.91	0.09	97.38	
	VOIP	675	12	40	1,685	98.25	94.41	99.29	0.71	96.29	
Our Approach	**AUDIO**	**216**	**0**	**0**	**2,196**	**100**	**100**	**100**	**0**	**100.00**	**100**
	BROWSING	**481**	**0**	**0**	**1,931**	**100**	**100**	**100**	**0**	**100.00**	
	CHAT	**97**	**0**	**0**	**2,315**	**100**	**100**	**100**	**0**	**100.00**	
	FILE-TRANSFER	**259**	**0**	**0**	**2,153**	**100**	**100**	**100**	**0**	**100.00**	
	MAIL	**85**	**0**	**0**	**2,327**	**100**	**100**	**100**	**0**	**100.00**	
	P2P	**325**	**0**	**0**	**2,087**	**100**	**100**	**100**	**0**	**100.00**	
	VIDEO	**262**	**0**	**0**	**2,150**	**100**	**100**	**100**	**0**	**100.00**	
	VOIP	**687**	**0**	**0**	**1,725**	**100**	**100**	**100**	**0**	**100.00**	

Table 9. The performance of the proposed algorithm compared to doing Classification with other methods of Scenario A.

Classifier	Intrusion type	Detection Rate	False Positive Rate	Accuracy
k-NN	Tor	15.64	1.13	37.36
	Non Tor	98.87	84.36	
C4.5	Tor	11.87	0	11.95
	Non Tor	100.00	88.13	
LDA	Tor	17.43	0	43.85
	Non Tor	100.00	82.57	
MLP	Tor	43.07	0	84.32
	Non Tor	100.00	56.93	
SVM	Tor	8.62	12.02	84.13
	Non Tor	87.98	91.38	
Our Approach	Tor	**99.97**	**0**	**99.98**
	Non Tor	**100**	**0.03**	

2015) in Table 12, NSL-KDD (UNB 2018) and KDD Cup'99 (KDD Cup 99 1999) Table 13 shows that the algorithms we present are effective in classification. In the case of the UNSW-NB15 dataset, the NSL-KDD dataset, and the KDD Cup'99 dataset, we will use the database for training and for testing the original file created by the developer.

In Table 11, classification with the website phishing dataset showed that the proposed algorithm had the highest efficiency of 97.54%, with a detection rate of phishing class of 100% as well as the UNSW-NB15 dataset. As shown in Table 12, even with the classification of many classes and invasions were different, the way we present still gives the highest performance in the NSL-KDD and KDD Cup'99, there are five types of invasions that are similar, even NSL-KDD is a modified version of the KDD Cup'99as shown in Table 13. The algorithm we are presenting can also detect both dataset intrusion and capture well on every dataset presented.

Furthermore, we tested the procedure with the ensemble method to test 50 models by applying C4.5 as a weak learner; it was found that Adaboost.M1 had an accuracy of 76%. According to the analysis, it was found that our proposed method was regarded as an incorporation of effective procedures, but it required only 1 model to vote with the method of Adaboost.M1. This resulted in the most effective experimental result, and it was substantially different from other methods.

Conclusions

In this paper, the new hybrid machine learning for cybersecurity threat detection is proposed. This new hybrid classifier is the combination of C4.5, MLP, SVM and LDA based on adaptive boosting. The UNB-CIC Tor Network Traffic datasets are used in the experiments for evaluating the performance of the proposed model. In addition, the experiments, correlation-based feature selection method is applied to all datasets in order to reduce redundant features.

Table 10. The performance of the proposed algorithm compared to doing classification with other methods of Scenario B.

Classifier	Intrusion type	Detection Rate	False Positive Rate	Accuracy
k-NN	AUDIO	66.24	2.8	66.79
	BROWSING	57.84	10.72	
	CHAT	16.05	3.6	
	FILE-TRANSFER	42.42	8.41	
	MAIL	27.69	2.85	
	P2P	87.22	0.54	
	VIDEO	43.42	7.46	
	VOIP	82.90	1.12	
C4.5	AUDIO	57.41	3.02	73.51
	BROWSING	72.40	14.65	
	CHAT	53.53	0.27	
	FILE-TRANSFER	100.00	7.99	
	MAIL	100.00	0.47	
	P2P	86.79	0.15	
	VIDEO	49.55	2.23	
	VOIP	85.52	0.12	
MLP	AUDIO	67.02	6.37	60.98
	BROWSING	52.00	1.89	
	CHAT	6.45	3.99	
	FILE-TRANSFER	61.46	8.65	
	MAIL	53.85	2.8	
	P2P	80.60	0.05	
	VIDEO	36.28	3.86	
	VOIP	98.91	16.09	
SVM	AUDIO	32.90	3.24	74.17
	BROWSING	83.28	10.15	
	CHAT	39.53	2.71	
	FILE-TRANSFER	100.00	8.15	
	MAIL	97.44	1.98	
	P2P	99.38	0.19	
	VIDEO	99.13	1.56	
	VOIP	77.22	0.59	
LDA	AUDIO	43.45	6.75	26.87
	BROWSING	57.99	15.17	
	CHAT	59.32	2.63	
	FILE-TRANSFER	28.27	8.83	
	MAIL	21.43	3.42	
	P2P	23.98	0.09	
	VIDEO	0.00	11.65	
	VOIP	0.00	30.68	
Our Approach	AUDIO	**100**	**0**	100
	BROWSING	**100**	**0**	
	CHAT	**100**	**0**	
	FILE-TRANSFER	**100**	**0**	
	MAIL	**100**	**0**	
	P2P	**100**	**0**	
	VIDEO	**100**	**0**	
	VOIP	**100**	**0**	

Tables 5–6 show the confusion matrix and Tables 7–8 show the efficiency of our proposed model including precision, detection rate, false positive rate, f-measure, and accuracy. It was found that the algorithm we offer has a high

Table 11. The efficiency of all classifiers when training with Website Phishing Dataset in case of weak learner before vote and after vote model.

Classifier		Detection Rate	FPR	f-Measure	Accuracy	Classifier		Detection Rate	FPR	f-Measure	Accuracy
k-NN	Normal	87.23	0	93.18	80.3	MLP	Normal	85.99	11.65	84.11	78.08
	Suspicious	25.33	3.63	35.85			Suspicious	7.14	7.71	9.19	
	Phishing	100	25.86	80.79			Phishing	92.23	15.49	88.12	
C4.5	Normal	89.53	4.27	91.66	89.66	SVM	Normal	85.64	1.38	91.48	82.02
	Suspicious	61.11	5.15	44.89			Suspicious	26.67	4.34	35.17	
	Phishing	92.13	6.32	93.21			Phishing	98.73	22.18	84.55	
LDA	Normal	80.9	8.77	84.21	82.51	**Our Approach**	**Normal**	**94.25**	**0**	**97.04**	**97.54**
	Suspicious	0	7.71	0.00			**Suspicious**	**100**	**1.06**	**93.11**	
	Phishing	85.27	10.99	87.82			**Phishing**	**100**	**2.99**	**98.56**	

Table 12. The efficiency of all classifiers when training with UNSW-NB15 dataset in case of weak learner before vote and after the vote model.

Classifier		Detection Rate	FPR	f-Measure	Accuracy	Classifier		Detection Rate	FPR	f-Measure	Accuracy
k-NN	Analysis	14.02	0.2	23.71	90.97	MLP	Analysis	0	0.82	0.00	50.19
	Backdoor	34.19	0.61	19.58			Backdoor	0	0.71	0.00	
	DOS	63.03	2.28	59.42			DOS	35.71	4.96	0.48	
	Exploit	94.31	3.82	83.42			Exploit	46.31	9.47	41.55	
	Fuzzers	76.94	1.65	78.08			Fuzzers	2.19	7.87	2.41	
	Generic	99.96	0.06	99.87			Generic	39.25	3.38	55.25	
	Normal	99.69	1.27	99.06			Normal	93.47	28.64	67.05	
	Reconnaissance	82.52	0.38	86.81			Reconnaissance	1.76	4.25	0.17	
	Shellcode	98.95	0	99.21			Shellcode	0	0.46	0.00	
	Worms	100	0	98.85			Worms	0	0.05	0.00	
C4.5	Analysis	85.25	0.76	14.09	89.28	SVM	Analysis	0	0.82	0.00	27.59
	Backdoor	57.45	0.68	8.57			Backdoor	0	0.71	0.00	
	DOS	51.32	1.53	59.70			DOS	0	4.97	0.00	
	Exploit	73.98	3.63	75.41			Exploit	0	13.52	0.00	
	Fuzzers	75.6	2.64	70.75			Fuzzers	16.57	1.11	28.03	
	Generic	99.52	0.51	98.90			Generic	90.59	21.65	13.49	
	Normal	96.18	1.95	96.90			Normal	90.71	32.67	58.04	
	Reconnaissance	96.35	0.65	90.44			Reconnaissance	0	4.26	0.00	
	Shellcode	77.22	0.16	70.32			Shellcode	0	0.46	0.00	
	Worms	75	0.02	63.16			Worms	0.11	0.02	0.22	
LDA	Analysis	0	0.82	0.00	66.51	**Our Approach**	**Analysis**	**27.53**	**0**	**43.17**	**97.84**
	Backdoor	0	0.71	0.00			**Backdoor**	**100**	**0.61**	**24.13**	
	DOS	13.86	4.33	15.92			**DOS**	**100**	**0.58**	**94.07**	
	Exploit	47.6	9.38	42.36			**Exploit**	**100**	**0.65**	**97.85**	
	Fuzzers	27.36	7.29	2.55			**Fuzzers**	**100**	**0.04**	**99.77**	
	Generic	92.57	4.11	89.10			**Generic**	**100**	**0.04**	**99.92**	
	Normal	66.84	10.97	76.87			**Normal**	**100**	**0.02**	**99.98**	
	Reconnaissance	0	4.25	0.00			**Reconnaissance**	**100**	**0.36**	**95.72**	
	Shellcode	0	0.46	0.00			**Shellcode**	**100**	**0**	**100.00**	
	Worms	0	0.05	0.00			**Worms**	**100**	**0**	**100.00**	

Table 13. The efficiency of all classifiers when training with NSL-KDD dataset and KDD Cup'99 in case of weak learner before vote and after vote model.

Dataset	Classifier		Detection Rate	FPR	f-Measure	Accuracy	Dataset	Classifier	Precision	Specificity	FPR	f-Measure	Accuracy
KDD-Cup'99	k-NN	Normal	79.96	1.84	85.85	91.13	NSL-KDD	k-NN	98.71	99.03	0.97	99.06	99.17
		DoS	99.96	17.36	96.13				99.79	99.89	0.11	99.85	
		Probe	29.74	0.09	45.10				99.71	99.97	0.03	99.83	
		R2L	100	0	93.94				98.72	99.81	0.19	97.30	
		U2R	71.19	1.97	67.54				98.51	100	0	99.25	
	C4.5	Normal	96.01	1.9	94.00	97.64		C4.5	99.23	99.41	0.59	98.77	98.8
		DoS	99.93	0.05	99.95				99.71	99.85	0.15	99.79	
		Probe	97.99	0.14	93.82				98.76	99.85	0.15	98.82	
		R2L	80	0.01	77.04				95.36	99.32	0.68	96.58	
		U2R	74.95	0.7	80.71				86.57	99.96	0.04	87.22	
	LDA	Normal	78.58	6.63	75.17	86.21		LDA	89.01	89.52	10.48	78.32	72.91
		DoS	88.57	11.43	92.62				68.89	86.08	13.92	77.13	
		Probe	39.95	0.93	35.13				70.22	96.48	3.52	76.27	
		R2L	45.16	0.01	42.42				32.1	90.36	9.64	36.25	
		U2R	19.34	5.24	0.57				41.79	99.83	0.17	45.90	
	MLP	Normal	94.18	6.27	82.03	86.53		MLP	77.14	84.96	15.04	85.51	71.72
		DoS	98	9.25	97.29				89.26	94.01	5.99	80.21	
		Probe	8.46	0.4	15.17				83.4	97.63	2.37	50.45	
		R2L	0.38	0.02	0.73				0	87.19	12.81	0.00	
		U2R	100	5.26	0.02				0	99.7	0.3	0.00	
	SVM	Normal	81.27	0.23	89.30	94.46		SVM	94.72	95.88	4.12	92.91	66.02
		DoS	99.69	4.61	99.00				15.38	70.48	29.52	26.63	
		Probe	78.27	0.33	77.08				97.6	99.71	0.29	92.00	
		R2L	3.55	0.01	6.64				74.73	96.35	3.65	79.02	
		U2R	84.65	3.84	42.32				29.85	99.72	0.28	0.65	
	Our Approach	**Normal**	**99.78**	**0**	**99.89**	**99.96**		**Our Approach**	**100**	**100**	**0**	**99.99**	**99.99**
		DoS	**100**	**0**	**100.00**				**100**	**100**	**0**	**100.00**	
		Probe	**100**	**0.01**	**99.66**				**100**	**100**	**0**	**100.00**	
		R2L	**100**	**0**	**95.52**				**99.97**	**99.99**	**0.01**	**99.98**	
		U2R	**100**	**0.03**	**99.69**				**98.51**	**100**	**0**	**99.25**	

performance for classifying different types of Tor in scenario B dataset results up to 100%.

However, the overall efficiency was satisfactorily high. The experimental result, compared with efficiency between machine learning as weak learner five methods: k-NN, C4.5, MLP, SVM, was LDA and our approach before voting and after voting with adaboost.M1 found that our proposed model had the highest efficiency. This means that high detection accuracy and low false positives are ideal for further development for real-time intrusion detection. In addition, compared with other intrusion databases such as Phishing website dataset UNSW-NB15 dataset NSL-KDD dataset and KDD Cup'99 dataset, it was found that our proposed model still had the highest efficiency in detecting errors compared with other methods. Additionally, it was compared with other research (Hodo et al. 2014) studies that employed UNB-CIC Tor Network Traffic datasets and the result found was that our research had higher efficiency.

Efficiency compared with the experimental work presented in the report, CFS-ANN was used for 99.8% accuracy. However, the research was 100% for Scenario B dataset.

According to all experimental results, it could be confirmed that our proposed model not only had higher efficiency in detecting intrusion than other methods, but it also had efficiency in detecting new intrusions that have never been found in the system. It is suitable for detecting abnormalities in the current situations where new abnormalities are hidden in the network and they are harmful to implementation.

Funding

This project is supported by the Thailand Research Fund (TRF) under grant number RTA6080013.

References

Abdelhamid, N., A. Ayesh, and F. Habtahb. 2014. Phishing detection based associative classification data mining. *Expert Systems with Applications* 41 (13):5948–59. doi:10.1016/j.eswa.2014.03.019.

Abdelhamid, N., A. Ayesh, and F. Thabtah. 2014. Phishing detection based associative classification data mining. *Expert Systems With Applications (ESWA)* 41 (13):5948–59. doi:10.1016/j.eswa.2014.03.019.

AbdElrahman, S. M., and A. Abraham. 2014. Intrusion detection using error correcting output code based ensemble. In 14th International Conference on Hybrid Intelligent System, 181–86. Kuwait: IEEE.

Aburomman, A. A., and M. B. I. Reaz. 2016. A novel SVM-kNN-PSO ensemble method for intrusion detection system. *Applied Soft Computing* 38 (C):360–72. doi:10.1016/j.asoc.2015.10.011.

Aburomman, A. A., and M. B. I. Reaz. 2017. A survey of intrusion detection systems based on ensemble and hybrid classifiers. *Computers & Security* 65:135–52. doi:10.1016/j.cose.2016.11.004.

Amer, M., M. Goldstein, and S. Abdennadher. 2013. Enhancing one-class support vector machines for unsupervised anomaly detection. In Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description, 8–15. Chicago, Illinois: ACM. doi:10.1177/1753193412444401

Anand, A., and B. Patel. 2012. An overview on intrusion detection system and types of attacks it can detect considering different protocols. *International Journal of Advanced Research in Computer Science and Software Engineering* 38 (1):94–98.

Bahl, S., and S. K. Sharma. 2015. Detection rate analysis for user to root attack class using correlation feature selection. In International Conference on Computing, Communication & Automation, 66–71. Noida, India: IEEE. doi:10.3389/fmed.2015.00066.

Dingledine, R., N. Mathewson, and P. Syverson. 2004. Tor: The second-generation onion router. In Proceedings of the 13th USENIX Security Symposium, 21–21. San Diego, CA: USENIX Association. doi:10.1186/1476-0711-3-21.

Eid, H. F., A. E. Hassanien, T. Kim, and S. Banerjee. 2013. Linear correlation-based feature selection for network intrusion detection model. In International Conference on Security of Information and Communication Networks, 240-248. Berlin, Heidelberg: Springer.

Freund, Y., and R. E. Schapire. 1996. Experiments with a new boosting algorithm, machine learning. In Proceedings of the Thirteenth International Conference on International Conference on Machine Learning, 148–56. San Francisco: Morgan Kaufmann Publishers Inc.

Galar, M., A. Fernandez, E. Barrenechea, and H. Bustince. 2014. A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 42 (4):463–84. doi:10.1109/TSMCC.2011.2161285.

Ghafir, I., J. Svoboda, and V. Prenosil. 2014b. Tor-based malware and tor connection detection. In International Conference on Frontiers of Communications, Networks and Applications, 1–6. Malaysia: IEEE Xplore Digital Library.

Hall, M. A. 1999. Correlation-based feature selection for machine learning. Phd. Diss., University of Waikato.

Hamid, I. R. A., J. Abawajy 1, and T. H. Kim. 2013. Using feature selection and classification scheme for automating phishing email detection. *Studies in Informatics and Control* 22 (1):61–70. doi:10.24846/v22i2y101307.

Hodo, E., X. Bellekens, E. Iorkyase, A. Hamilton, C. Tachtatzis, and R. Atkinson. 2014. Machine learning approach for detection of nontor traffic. In Proceedings of the 12th International Conference on Availability, Reliability and Security, 85: 1–85:6. Reggio Calabria, Italy: ACM.

Hussain, J., and S. Lalmuanawma. 2014. A hybrid approach for determining the efficient network intrusion detection system. *The IUP Journal of Computer Sciences* 8 (3):34–36.

KDD Cup 1999. 1999. *UCI machine learning repository*. Irvine: University of California, School of Information and Computer Science. Accessed February 2018. https://archive.ics.uci.edu/ml/machine-learning-databases/kddcup99-mld/.

Kumar, P. A. R., and S. Selvakumar. 2013. Detection of distributed denial of service attacks using an ensemble of adaptive and hybrid neuro-fuzzy systems. *Computer Communications* 36 (3):303–19. doi:10.1016/j.comcom.2012.09.010.

Kuncheva, L. I., and J. J. Rodríguez. 2014. A weighted voting framework for classifiers ensembles. *Knowledge and Information Systems* 38 (2):259–75. doi:10.1007/s10115-012-0586-6.

Lashkari, A. H., G. D. Gil, M. S. I. Mamun, and A. A. Ghorbani. 2017. Characterization of tor traffic using time based features. In Proceedings of the 3rd International Conference on Information Systems Security and Privacy, 253–62. Porto, Portugal: SCITEPRESS.

Mascaro, S., A. E. Nicholson, and K. B. Korb. 2014. Anomaly detection in vessel tracks using Bayesian networks. *International Journal of Approximate Reasoning* 55 (1):84–98. doi:10.1016/j.ijar.2013.03.012.

Michael, G., A. Kumaravel, and A. Chandrasekar. 2015. Detection of malicious attacks by meta classification algorithms. *International Journal of Advanced Networking and Applications* 6 (5):2455.

Moustafa, N., and J. Slay. 2015. UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In MilCIS-IEEE Stream, Military Communications and Information Systems Conference, 1–6. Canberra, ACT, Australia: IEEE.

Nejad, T. R., and M. S. A. Abadi. 2014. Intrusion detection in computer networks through a hybrid approach of data mining and decision trees. *Walia Journal* 30 (S1):233–37.

Nguyen, H. Y., K. Franke, and S. Petrovic. 2014. Improving effectiveness of intrusion detection by correlation feature selection. In 2010 International Conference on Availability, Reliability and Security, 17–24. Krakow, Poland: IEEE.

Ren, Y. 2014. An integrated intrusion detection system by combining SVM with adaboost. *Journal of Software Engineering and Applications* 7 (12):1031–38. doi:10.4236/jsea.2014.712090.

Sagha, H., H. Bayati, J. R. Millán, and R. Chavarriaga. 2013. On-line anomaly detection and resilience in classifier ensembles. *Pattern Recognition Letters* 34 (15):1916–27. doi:10.1016/j.patrec.2013.02.014.

Shahbaz, M. B., X. Wang, A. Behnad, and J. Samarabandu. 2016. On efficiency enhancement of the correlation-based feature selection for intrusion detection systems. In Proceedings of the 2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference, 1–7. Vancouver, BC, Canada: IEEE.

Sheykhkanloo, N. M. 2014. Employing neural networks for the detection of sql injection attack. In Proceedings of the 7th International Conference on Security of Information and Networks, 318–23. Glasgow, Scotland, UK: ACM. doi:10.1177/1753193413517839.

UNB, University of New Brunswick. 2018. http://www.unb.ca/cic/datasets/tor.html.

University of New Brunswick (UNB). 2017. http://www.unb.ca/cic/datasets/tor.html.

Wahba, Y., E. ElSalamouny, and G. ElTaweel. 2015. Improving the performance of multi-class intrusion detection systems using feature reduction. *IJCSI International Journal of Computer Science* 12 (3):255–62.

Zhang, H., Z. Xie, Y. Yang, Y. Zhao, B. Zhang, and J. Fang. 2017. The correlation-base-selection algorithm for diagnostic schizophrenia based on blood-based gene expression signatures. *BioMed Research International* 2017:7860506.

Zhang, Z., and X. Xie. 2010. Research on adaboost. m1 with random forest. In Proceedings of the 2nd International Conference on Computer Engineering and Technology, 647–52. Chengdu, China: IEEE.

Zhou, Z.-H. 2012. *Ensemble methods: Foundations and algorithms*. Boca Raton, FL: CRC Press.

ภาคผนวก ข
ผลงานวิจัยตีพิมพ์ที่ Engineering Journal

Journal homepage: https://engj.org/index.php/ej

Sornsuwit, P., & Jaiyen, S. 2019. "A New Incremental Decision Tree Learning for Cyber Security based on ILDA and Mahalanobis Distance". *Engineering Journal*, 23(5).

หมายเหตุ ผลงานนี้ได้รับการตอบรับในวันที่ 11 มิถุนายน 2562 และรอการเผยแพร่ในเดือนสิงหาคม 2562

Engineering Journal issn: 0125-8281

Article

A New Incremental Decision Tree Learning for Cyber Security based on ILDA and Mahalanobis Distance

Ployphan Sornsuwit[a*] and Saichon Jaiyen[b]

Department of Computer Science, Faculty of Science King Mongkut's Institute of Technology Ladkrabang, Bangkok 10520, Thailand
E-mail: [a*]ployphan.en@gmail.com (Corresponding author), [b]saichon.ja@kmitl.ac.th

Abstract. A cyber-attack detection is currently essential for computer network protection. The fundamentals of protection are to detect cyber-attack effectively with the ability to combat it in various ways and with constant data learning such as internet traffic. With these functions, each cyber-attack can be memorized and protected effectively any time. This research will present procedures for a cyber-attack detection system Incremental Decision Tree Learning (IDTL) that use the principle through Incremental Linear Discriminant Analysis (ILDA) together with Mahalanobis distance for classification of the hierarchical tree by reducing data features that enhance classification of a variety of malicious data. The proposed model can learn a new incoming datum without involving the previous learned data and discard this datum after being learned. The results of the experiments revealed that the proposed method can improve classification accuracy as compare with other methods. They showed the highest accuracy when compared to other methods. If comparing with the effectiveness of each class, it was found that the proposed method can classify both intrusion datasets and other datasets efficiently.

Keywords: Cybersecurity, IDTL, incremental learning.

ENGINEERING JOURNAL Volume # Issue #
Received Date Month Year
Accepted Date Month Year
Published Date Month Year
Online at http://www.engj.org/
DOI:10.4186/ej.20xx.xx.x.xx

DOI:10.4186/ej.year.vol.issue.pp

1. Introduction

At this current time, the internet has become an important part of people's routines and is utilized for business communication, online social activity, education, medicine, public sector support, etc. Once any of these aspects becomes significant, they are always prone to malicious activities and data theft because any important information stored in organizational networks can be a target of ill-intended individuals attempting to access and misuse these pieces of information. Cyber- attack detection systems are designed differently depending on system vulnerabilities and intrusion intentions such as phishing website, password guessing, spam, distributed denial of service (ddos) attacks, access to observe activities of a target, Elevating Privileges Access Attacks and so on. Currently, new technology is planned and designed to monitor network attacks, for example, cloud computing [1] or Internet of Things (IoT) [2], etc. Especially in current intrusion to communication by secure traffic there is an inadequate transmission of data to the communication, such as chat, file transfer protocol (ftp), p2p or tor traffic etc. It is difficult to detect when sending data over the network. And the intruders are now trying to develop a way to overcome the security of data traffic for commercial purposes, or maybe in an attempt to scramble and test the system's capabilities.

They can be classified into 2 major types – Anomaly-based IDS and Misuse-based IDS [3]. These 2 features have both pros and cons. Misuse-based IDS can detect intrusion accurately by memorizing patterns or dataset rules. Despite highly accurate detection, it is ineffective against newer intrusion due to its limited dataset or unfamiliarity to some systems. Anomaly-based IDS can detect intrusion through analyzing statistics of the normal behavior of users. In the case of any significantly unfamiliar activities contrasting from normal behavior, it can immediately detect them. Therefore, this feature can detect newer intrusions but holds a higher rate of false alarm.

Many previous studies attempted to use algorithms in order to function as anomaly-based IDS detection by presenting procedures to enhance detection features while lowering mistakes and speeding up the processor. In fact, it is compulsory to immediately notify of intrusion [3]–[7]. Linear Discriminant Analysis (LDA) is another important method of feature reduction that has been practiced for many years. This method is utilized effectively in IDS by combining preprocesses and classification of intrusion [8]–[12]. Many previous studies had applied LDA with other machine learning to enhance IDS together with newer intrusion datasets. Likewise, some studies developed and improved LDA algorithms [11], [13] to make the application more effective and more suitable such as Incremental Linear Discriminant Analysis (ILDA) [14] This method used incremental loading for processing. Similar to LDA generalization that models specific processes when finishing without reusing, it is considered as a method suitable for modern IDS. Some studies might classify data by using distance function for clustering as well [15]–[17] that benefits detection of a newer intrusion when a large number are hidden in other intrusion datasets. It's because of using distance function without forming a model resulting in newer intrusion detection.

And cyber attack detection on the current network within the system should be able to incrementally learn behavior of the normal user and then continuously learn the types of invasion that can be detected immediately. This is a different point from traditional machine learning in traditional Anomaly-based IDS, which uses the model to detect abnormalities. Over time, the model will be updated to the new version. The learning model can't be detected immediately. It is a challenge to develop a network intrusion detection system that is currently in use. It can be incrementally learned behavior with usage and risk of invasion at any time through the data into the system and can be classified as the invasion types that are found in the current system effectively.

In relation to these previous studies, this study aimed to develop intrusion detection procedures through Incremental Decision Tree Learning (IDTL) to classify data in order to enhance the effectiveness and the suitability of the framework for detection on network traffic. To incrementally function, a framework is provided to build models and classify intrusion datasets of the classified structure in a binary hierarchy. Development and improvement of Incremental Linear Discriminant Analysis (ILDA) with using mahalanobis distance to measure distance to classify. Similar to the combination of supervised learning and unsupervised learning, it can detect intrusions that the model is familiar with and other unfamiliar intrusions. The objectives of this study are as follows.

1. Some features, out of the high number of features on network traffic, were selected through Pearson Correlation, as many features may not benefit the overall classification measurement.

2. Provide an IDTL structure for binary hierarchical cyber-attack detection and development and improvement from ILDA using mahalanobis distance. The procedure launches increment learning of the

data to form a constant, one-time, and immediate model without restoring the data to re-calculate with other data.

3. Effectiveness of other procedures of intrusion detection were also compared and analyzed.

This study, therefore, is broken down as follows: Section 2: The review of related studies, Section 3: Theory, Section 4: Materials and Methods, Section 5: Results, Section 6: Discussion and Section 7: Conclusions.

2. Relate Studies

Many researchers have tried to study machine learning used in a variety of algorithm-based intrusion detection. They may come in the form of unsupervised learning [15]–[17] clustering without target specification and supervised learning that is used for training to model in estimation before new data estimation. Semi-supervised learning [5], [13], [18]–[20] is another type involved in function estimation on labeled and unlabeled data, falling between unsupervised learning and supervised learning, and Ensemble Learning [21]–[23] which uses many classification models to vote on an estimation. However, even if the ensemble has to build multiple models for high-quality voting, it may not be suitable for intrusion detection on an always-available network.

Many mentioned procedures attempted to enhance detection by increasing accuracy and reducing false alarm rates. Therefore, many studies applied many methods called Hybrids [24]–[26] which are a combination of methods to improve maximum effectiveness of intrusion detection.

The study [10] used LDA to implement feature reduction and NDL-KDD dataset before classification on neutral networks. It was found that it could reduce features resulting in lower training time and highly effective classification of intrusions. Likewise, the study [25] functions both PCA and LDA to run feature extraction by finding class-pair that both PCA and LDA could pinpoint the best feature value. The feature was later classified through SVM. The result of the test found that it could improve efficiency. In addition, there were more ideas to improve LDA effectiveness. As shown in the study [26], Direct LDA was developed by removing S_b eigenvectors, corresponding to the eigenvalues that were equal to zero or close to zero and kept the null space of S_w, to enhance effective detection rates and lower false alarm rates. Not only did LDA prevent DoS but also black hole attacks of self-driving communication and semi self-driving vehicles in VANETs [27]. LDA was more efficient than QDA. It has been stated that feature selection and dataset intrusion could enhance effective classifications [9], [28]. Many researchers emphasized this point, as it could lower data insignificant for calculation, enhancing effectiveness and reducing time consumption. To function in real life applications, a lot of data on networks running through IDS should be significantly feature-selected so that it could detect an intrusion immediately as it occurs.

Currently, the procedure suitable for intrusion detection that is similar to real life application conditions is incremental learning because it can learn from large-scale dynamic stream data and build up a knowledge base over time to benefit future learning and decision-making processes [29]. While, intrusion on incoming and outgoing network traffic is essential to have statistical calculation of constant timely changes to make a decision at a certain moment ensuring if it is a detected intrusion [30]. The study [31] presented Weight ILDA (WILDA) to function with online hand-written Chinese character recognition. WILDA is a method to recognize the issue of an uncertain number of incremental data through methods of weight of S_w and S_b calculation to reduce the problem of lower accuracy found in a small proportion of increment new samples. In the test, WILDA was found to solve this problem by increasing accuracy and holding higher efficiency than ILDA.

Besides this, there is another study [32] developing online system FNTAE that can detect real-time intrusion through FIncLDA as a learning model and k-NN as a decision agent to make decisions regarding intrusion detection. This system can utilize chunk LDA for online learning and can be applied to increase the effectiveness of intrusion detection. To increase effectiveness of intrusion detection, preprocessing is important. Many studies tested feature extraction or feature selection through different methods as found in the study [33] using Discrete Wavelet Transform (DWT) for effective enhancement and iPCA in conducting an interactive factor analysis. This study used them for visual comparison of various features and the researchers comparatively experimented data NSL-KDD projection during DWT and non-DWT usage. It was found that using DWT made a clear separation among the attack categories in some classes, for example, R2L which is unidentifiable with raw features. However, R2L can be identified with DWT and is effective with a machine learning test. Additionally, another study [14], [34] used Chi Squared Attribute Evaluator to select relevant features for classification through LDA and logistic regression (LR). It revealed that both methods could perform well on multiclass and binary classification. Despite higher accuracy than Naïve Bayes,

it cannot be higher than SVM and C4.5 having a low computational overhead that is higher than SVM is considered more appropriate to the development of real-time network monitoring. Adding Tor dataset indicates the current significance [37] and discussed Pearson [35].

As shown in the previous studies, this study is interested in developing each procedure of cyber-attack detection to be effective through IDTL based on Sequential Incremental LDA (ILDA) and mahalanobis distance integrated with classification of Tor traffic datasets, which is an invasion of hidden services and other cyber-attack datasets. It aims to be more effective and function in Incremental Learning that can increment numbers of new datasets for modeling constantly as well as efficiently classify each attack type. The model is extended by Incremental learning which will probably be applied with real cyber-attack detection on a real network in the future.

3. Theory

3.1. Pearson Correlation

Pearson correlation coefficient is to determine and measure of the strength of the association between the variable X and variable Y based on the method of covariance of these two values, divided by the product of their standard deviations the following values are calculated in EQ.(1). [35]

$$\text{Let } X = \{x_1, x_2, \ldots, x_n\} \text{ and } Y = \{y_1, y_2, \ldots, y_n\}$$

$$r_{XY} = \sum_{i=1}^{n} \frac{(X_i - \bar{X})}{\sqrt{\sum_{i=1}^{n}(X_i - X)^2}} \cdot \frac{Y_i - \bar{Y}}{\sqrt{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}} \quad (1)$$

r_{XY} is correlation coefficient whose value is between -1 and 1.

3.2. Sequential Incremental Linear Discriminant (Sequential ILDA)

Let $X = \{x_1, x_2, \ldots, x_N\}$ be a set of training samples with M classes and N be the number of training samples. Let y be the new incoming datum with the class label k, the new eigenspace model, $\Omega' = (Sw', Sb', \bar{x}, N+1)$, must be updated by using only the old Ω and the new incoming datum y. For the new mean parameter $\bar{x}'$, it can be calculated as EQ.(2). [14]:

$$\bar{x}' = \frac{(N\bar{x}+y)}{(N+1)} \quad (2)$$

For between-class scatter matrix Sb', if $k = M+1$ representing a newly introduced class, are shown in Eq. (3) and Eq. (4) [14]

$$Sb' = \sum_{c=1}^{M} n_c(\overline{x_c} - \bar{x}')(\overline{x_c} - \bar{x}')^T + (y - \bar{x}')(y - \bar{x}')^T \quad (3)$$

$$Sb' = \sum_{c=1}^{M+1} n_c'(\overline{x_c} - \bar{x}')(\overline{x_c} - \bar{x}')^T \quad (4)$$

where n_c' is the number of samples in class c after having data y appear, $n_c' = n_c$ when $1 \le c \le M$, $n_c' = 1$ when $c = M+1$, and $\overline{x_c} = y$ when $c = M+1$.

When $1 \le c \le M$ then Sb' is updated using the Eq. (5) [14]

$$Sb' = \sum_{c=1}^{M} n_c'(\bar{x}_c - \bar{x}')(\bar{x}_c - \bar{x}')^T \quad (5)$$

where $\overline{x_c} = (1/(n_c+1))(n_c\overline{x_c} + y)$ and $n_c' = n_c + 1$ if y equals class c; else $\overline{x_c}' = \overline{x_c}$ and $n_c' = n_c'$

For within-class scatter matrix Sw, if y is a new class which means k is the $(M+1)$ class. Therefore, updating within-class scatter matrix is not changing as in the Eq. (6).

$$Sw' = \sum_{c=1}^{M} \sum_{c} + \sum_{k} = \sum_{c=1}^{M+1} \sum_{c} = \sum_{c=1}^{M} \sum_{c} \tag{6}$$

In case that $1 \leq c \leq M$ will update s_w as in the equation shown at the proof in the Appendix.

$$Sw' = \sum_{c=1,c\neq k}^{M} \sum_{c} + \sum_{k}{}' \tag{7}$$

$$\Sigma_k' = \Sigma_k + \frac{n_k}{n_k+1}(y - \bar{x}_k)(y - \bar{x}_k)^T \tag{8}$$

3.3. Mahalanobis Distance

Mahalanobis distance is another interesting measure of the distance between two points in multivariate space as defined in equation (9) where d is mahalanobis distance, x is the observation and μ is the mean of samples. S is the covariance, it can be displayed as EQ.(9). [36].

$$d(x,\mu) = \sqrt{(x-\mu)^T S^{-1}(x-\mu)} \tag{9}$$

4. Materials and Methods

4.1 The proposed method.

This research proposes IDTL which is a new incremental hierarchical learning based on Incremental LDA and mahalanobis distance. The proposed method adopts an Incremental LDA method that can learn several attack types through binary classification. This method is different from traditional LDA algorithms and intrusion detection procedures from other studies. Incremental LDA learns to perform intrusion detection through tree-diagram node forming. Each node can classify different classes of the attack types. IDTL can specify new data through calculation as a one-time process. Briefly, each new data is calculated to form a model once before being completely discarded. Therefore, the learned data can be discarded after being learned. There is no need to store the old data to learn the new incoming data. Additionally, classification is enhanced by mahalanobis distance to increase accuracy. This proposed method is suitable for a modern model of cyber attack detection on an online computer network that is prone to cyber attack all the time without attack type identification and damage protection during the application. The process of doing this research is presented as Figure.1.

We used Tor dataset [37] which is a dataset that has hidden services in traffic network. It has developed a tool to use tor widely and is difficult to detect because it uses multiple protocols. Tor will protect or obscure the personal privacy of its users, as well as their freedom and ability from Internet activities. Therefore, it is a vulnerability for attackers to use Tor as a channel to avoid detection when attacking the network as show in figure 1.

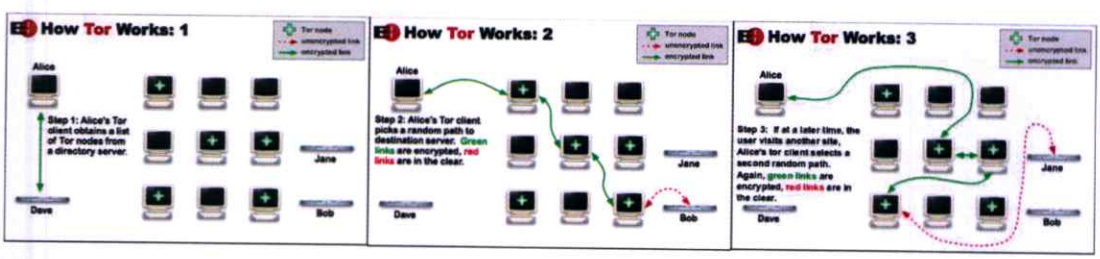


Figure 1. Attack process of Tor [37]

Our research was conducted with the Tor dataset in two scenarios: scenario A represents of the implementation classification binary class and scenario B represent the implementation of multiclass classification. It is also tested with other datasets that looks like an invasion, for example, NSL-KDD as a dataset revised from KDD Cup'99 [38], spam dataset [39], phishing dataset [40] and SAME is the dataset of Android system invasion [41]. All stages of the experiment are shown in Figure 1. They consists of the stages of classification with prior stage of data preprocessing for data availability. Then, features were selected to obtain only related features. Next stages are training and testing along with performance measurement. All stages of the experiment can be discussed as follows:

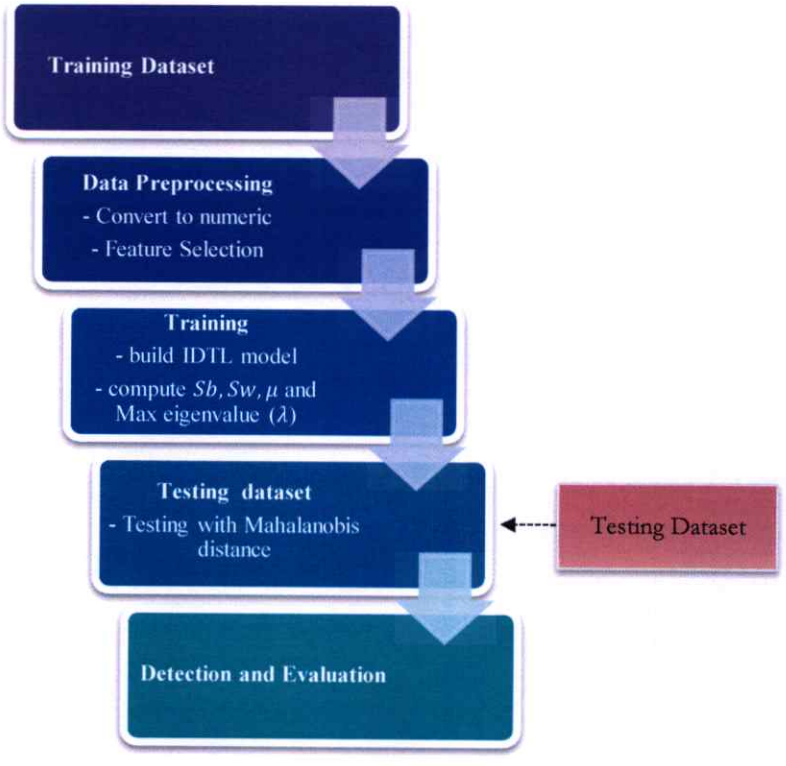


Fig. 2. Proposed cyber-attack detection model based on IDTL

4.1.1 Data Preprocessing

At this stage, we will convert every text feature into numeric features and select a feature using Pearson Correlation. Because of the algorithm that selects the relationship of each feature, the research uses a variety of fields [42][43] efficiently. Data is divided into two parts. The first part is for training stage and the second one is for testing stage. By doing so, data for testing stage is not found in data for training stage. This way is similar to authentic intrusion detection in the network.

4.1.2 Training Stage

After completing the preprocessing of the training, datasets of the training procedure can be divided into two main structures. First, training structure for the binary class and this is for the case normal and abnormal classification is required. Second, multiclass structure is for classifying multi classes that are both normal and several attack types. Both structures use similar IDTL method.

The binary structure is shown in Figure 3. This demonstrates a learning process of hierarchical visualization of the IDTL, which is classified as normal and abnormal intrusion detection.

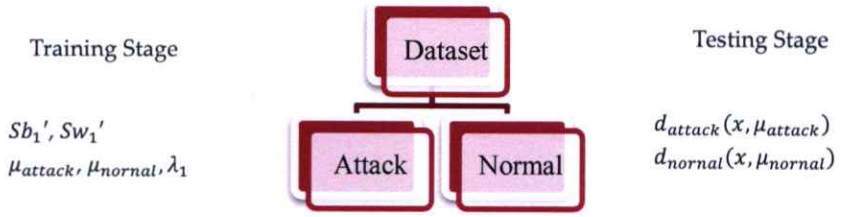


Fig. 3. Procedures of training the proposed model based on binary classes.

Figure.4 demonstrates a learning process of hierarchical visualization of the IDTL which is incremental learning structures: the multiclass classifier structure classifies intrusion detection. The procedure launches increment learning of the data to form a constant, one-time, and immediate model without restoring the data to re-calculate with other data.

All stages of training process of hierarchical structure based on IDTL in figure 4 are described as follows:

First, data are divided into Normal class and Attack class as the first stage of classification. When cyber attack occurs, detection operates to analyze whether data are Normal class or Attack class. At this stage, all data are incremented in the learning of Sb_1', Sw_1' between Normal Class and Attack Class, and the mean (μ) of two classes. Then, the first maximum eigenvalue λ_1 is solved to prepare testing stages.

Next, Attack type 1 and Other attack#1 types are classified. Attack type 1 has cyber attack behaviors that can be clearly distinguished from other attack types of intrusion due to its various attempts to cause attack in services that can be found in network disturbance.

It processes through the learning of Sb_2', Sw_2', mean of two classes (μ) and the second maximum eigenvalue λ_2. Next, Sb_3', Sw_3', mean of two classes (μ) and the third maximum eigen value λ_3. IDTL will increment learn in attack type classes in sequence up to the last stage, Attack type n-1 and Attack type n are classified as separate from each other as shown in Figure 4.

Our research has improved the equation of the Sequential ILDA, which is multiclass into a binary class equation for binary learning in each tree hierarchy.

$$Sb' = \sum_{c=1}^{2} (\bar{x}'_1 - \bar{x}'_2)(\bar{x}'_1 - \bar{x}'_2)^T \quad (10)$$

In the training process, we will read the sequential data one record at a time to calculate and update the virtual model to read the internet traffic to calculate one record. After calculating and updating the model, we do not take that information back to calculations.

That means that any data will be calculated only once, then the model will represent all data. This research has focused on data stream reading, which is used for outlier detection [44].

The incremental learning algorithm is highly effective and is consistent with outlier detection with network-based intrusion detection data streams.

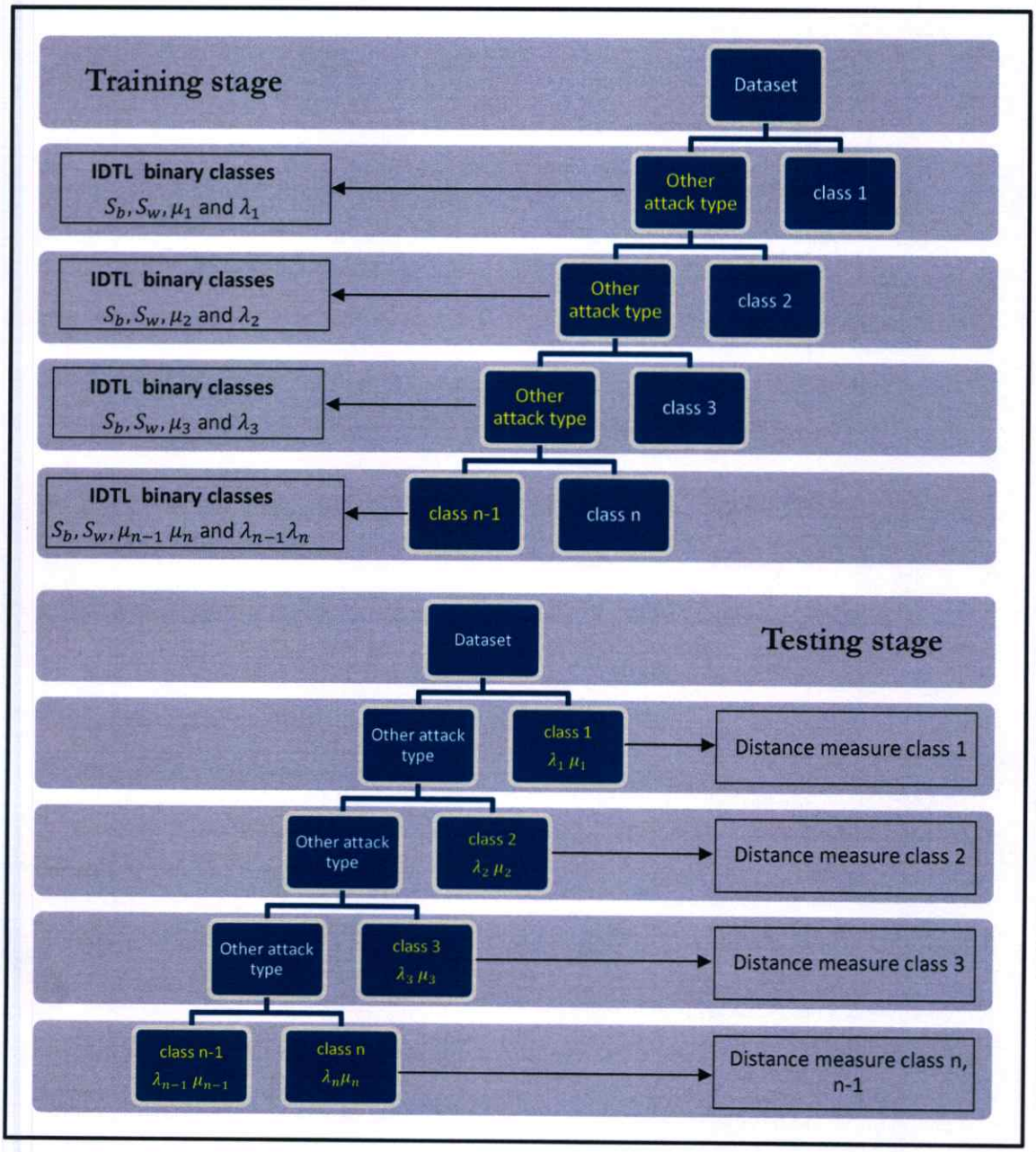


Fig. 4. Procedures of training the proposed model based on multiclass classes.

In the training stage, every learning n class hierarchy of IDTL, the eigenspace is updated in every 1 record that is currently taining as shown in Figure 5.

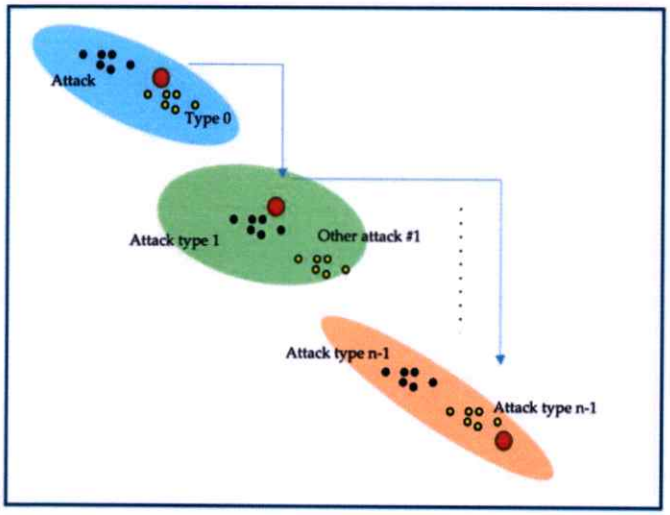


Fig. 4. The eigenspace update in each IDTL hierarchy.

4.1.3 Testing stage

During the testing stage, mahalanobis distance is used to find the distance for identifying the class of test data in the same sense as real network detection. If the test data is classified to be attack type, these data must be submitted to calculate the distance until they can be identified into the class they belong. The procedure repeats eventually till all data classes have been identified.

$$d(x,\mu) = \sqrt{(x-\mu)^T \lambda^{-1} (x-\mu)} \tag{11}$$

To calculate the distance, we use the Max eigenvalue (λ) and mean (μ) obtained by increment learning in each class of the class to calculate in the equation.

4.2 Evaluation

This research used various methods of performance measurement to ensure analysis accuracy. In classification, class could be predicted through all test data that underwent performance measurement of values as shown in Table 1. Then, the following values' performance was measured.

Table 1. Confusion Matrix

Predict Value	Actual Value	
	Positive	Negative
Positive	True Positive: TP	False Positive: FP
Negative	False Negative: FN	True Negative: TN

Precision: The amount of data predicted from the prediction of considering class as shown in the EQ.(10).

$$Precision = \frac{TP}{TP+FP} \tag{10}$$

Recall or Sensitivity or Detection Rate is a proportion of True Positive cases that are correctly predicted as positive as shown in the EQ.(11).

$$Recall = \frac{TP}{TP+FN} \tag{11}$$

F-measure is an overall measure of Precision and Recall as shown in the EQ.(12).

$$f - Measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{12}$$

Accuracy is the number of correct data prediction from classes as shown in the EQ.(13).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{13}$$

4.3 Implementation

This research was conducted by using a personal computer intel core, i5-4258U CPU @2.4GHz, 8 GB memory without GPU acceleration and our algorithm was implemented in MATLAB R2017. In the step of preprocess it was used to select important features before the training and testing of data that was previously presented. In classification, class of attack types training and testing was done to measure efficiency.

Next, this study's method was compared with other methods of various machine's learning. For example, Naïve Bayes, Decision Tree, k-NN, Multi-layer Perceptron (MLP) and SVM. It was also tested with other attack types that the dataset currently represents an intruder to confirm the effectiveness of our approach.

4.4 Algorithm

The proposed incremental decision tree learning (IDTL) algorithm can be described as follows.

Algorithm IDTL Algorithm

Input: Training set $= \{(x_1, y_1), (x_2, y_2), \ldots, (x_n y_n)\}$, with class label $y_i \in \{1, .., k\}$ of training dataset
Output: IDTL Model
For $i = 1, .., n$ // n is number of samples
  For $y = 1, \ldots, k$ // k is number of class label
    If class label belong to any k class. Then
      Update $\vec{\mu}'$, Sb' and Sw'
    Else
      Update $\vec{\mu}'$, Sb' and Sw' for other k class
      Calculate max eigenvalue for any k class // for binary classification
      If class label belong to any $k - 1$ class
        Update $\vec{\mu}'$, Sb' and Sw'
      Else
        Update $\vec{\mu}'$, Sb' and Sw' for k class
        Calculate max eigenvalue for any k class // for binary classification
      End
    End
  End
  IDTL model with max eigenvalue of every class label for hierarchical distance measure

$$d(x, \mu) = \sqrt{(x - \mu)^T \lambda^{-1} (x - \mu)}$$

End

5. Results and Discussion

5.1. Results

We will test with Tor the data stored in a real-world model. The view is divided into 2 scenarios. Scenario A tests the binary classes Tor and non –Tor. In Scenario B, it tests multiclass include Browsing, Email, Chat, Audio, Video, FTP, VoIP and P2P. Each class has different numbers and types of hidden services. We have similar and smaller group classes as follows, Browsing's label as class 1, FTP + P2P label as class 2, Audio + VOIP label as class 3 and Chat + Email + Video label as class 4

The experiment will use all data from all developers, divided into 70 percent for training datasets and 30 percent for testing datasets. Data for testing will never appear in the training dataset as with cyber attack detection in computer networks.

5.1.1. Feature Selection

In scenario A: 9 features were selected out of a total of 28 features using as shown in Table 2. And scenario B: 11 features were selected out of a total of 28 features as shown in Table 3.

Table 2. Selected features from the feature selection stage and its explanation of scenario A.

Feature No.	Feature Name
8	Flow Packets/s
7	Flow Bytes/s
6	Flow Duration
5	Protocol
19	Bwd IAT Max
1	Source IP
2	Source Port
12	Flow IAT Min
9	Flow IAT Mean

Table 3. Selected features from the feature selection stage and its explanation of scenario B.

Feature No.	Feature Name
2	Source Port
11	Flow IAT Max
15	Fwd IAT Max
19	Bwd IAT Max
18	Bwd IAT Std
14	Fwd IAT Std
4	Destination Port
10	Flow IAT Std
6	Flow Duration
1	Source IP
9	Flow IAT Mean

5.1.2. Results for IDTL

When tested, IDTL is most effective when compared to traditional ILDA algorithms, and better than without feature selection is shown in Table 4 - 5. When we compared the IDTL with feature selection, we proposed the choice of feature, or no feature being chosen. In addition, compared with traditional ILDA with mahalanobis the distance would have a classification procedure similar to the one we proposed. The

difference is based on the original Sb'value of the traditional ILDA algorithm and without the feature selection.

The results show that the methods we proposed are most effective overall. And to determine the f-measure and accuracy, the sub-class was found to be the most effective as well.

In the case of a binary class, it is classified separately between Tor traffic and non Tor, which is classified as normal and abnormal. In the case of a multi class, it distinguishes difficult and some illegal services in some countries. Like P2P. Other services are also unobtrusive services, such as submitting malware via ftp. The experiments show that IDTL can detect these services and performs better than algorithms others have compared.

Table 4 - Table 5. Compare the efficiency between the algorithms we proposed in the feature selection and without the feature selection and the traditional ILDA approach with the mahalanobis distance of scenario A and scenario B.

Table 4. Comparison of performance between algorithms we proposed in feature selection and not feature selection and conventional traditional ILDA with mahalanobis distance of scenario A

Traditional ILDA with mahalanobis distance	Tor	Non-Tor
Precision	14.27	100
Sensitivity	100	19.15
Specificity	19.15	100
f-Measure	24.98	32.14
Accuracy	28.74	
IDTL without features selection	Tor	Non-Tor
Precision	22.3	99.99
Sensitivity	99.96	53.16
Specificity	53.16	99.96
f-Measure	36.47	69.42
Accuracy	58.71	
IDTL with feature selection	Tor	Non-Tor
Precision	44.68	97.42
Sensitivity	82.52	86.62
Specificity	86.62	82.52
f-Measure	57.97	91.70
Accuracy	86.14	

Table 5. Comparison of performance between algorithms we proposed in feature selection and not feature selection and conventional traditional ILDA with mahalanobis distance of scenario B

Methods	Class 1	Class 2	Class 3	Class 4
Traditional ILDA with mahalanobis distance				
Precision	50.31	87.58	70.21	45.32
Sensitivity	51.35	66.44	77.52	49.1
Specificity	87.36	96.99	80.32	86.64
f-Measure	50.82	75.56	73.68	47.13
Accuracy	64.39			
IDTL with full features	Class 1	Class 2	Class 3	Class 4
Precision	50.31	85.49	71.4	48.13
Sensitivity	51.35	66.61	77.96	52.03
Specificity	87.36	96.39	81.31	87.35
f-Measure	50.82	74.88	74.54	50.00

Methods	Class 1	Class 2	Class 3	Class 4
Accuracy		65.13		
IDTL with feature selection	Class 1	Class 2	Class 3	Class 4
Precision	55.6	98.55	97.72	75.37
Sensitivity	78.38	93.32	75.97	81.31
Specificity	84.41	99.56	98.94	94
f-Measure	65.05	95.86	85.48	78.23
Accuracy		81.63		

Then, when comparing the IDTL with other machine learning methods, the results are shown in Table 6 and Table 7. Even though IDTL is an incremental learning course, the overall classification performance is far superior to any other learning machine.

Table 6. IDTL performance vs. machine learning of scenario A

Method	Efficiency	Tor	Non-Tor
IDTL	Precision	44.68	97.42
	Sensitivity	82.52	86.62
	Specificity	86.62	82.52
	f-Measure	57.97	91.70
	Accuracy	86.14	
Tree	Precision	11.86	100
	Sensitivity	100	0.07
	Specificity	0.07	100
	f-Measure	21.21	0.14
	Accuracy	11.92	
Naïve Bayes	Precision	10.86	87.49
	Sensitivity	36.3	59.9
	Specificity	59.9	36.3
	f-Measure	16.72	71.11
	Accuracy	57.11	
k-NN	Precision	15.9	98.5
	Sensitivity	96.44	31.4
	Specificity	31.4	96.44
	f-Measure	27.30	47.62
	Accuracy	39.11	
MLP	Precision	16.02	89.48
	Sensitivity	32.78	76.87
	Specificity	76.87	32.78
	f-Measure	21.52	82.70
	Accuracy	71.65	
SVM	Precision	15.1	100
	Sensitivity	100	24.36
	Specificity	24.36	100
	f-Measure	26.24	39.18
	Accuracy	33.33	

Considering Table 6, it was found that when tested scenario A: IDTL had an accuracy of 74.66 and had the highest class of f-measure. When considering the class, it was found that IDTL could best classify two classes. As in the Table 7 scenario B: the highest accuracy is 81.63 and this classification is most effective.

Table 7. IDTL performance compared with machine learning of scenario B

Method	Efficiency	Class 1	Class 2	Class 3	Class 4
IDTL	Precision	55.6	98.55	97.72	75.37
	Sensitivity	78.38	93.32	75.97	81.31
	Specificity	84.41	99.56	98.94	94
	f-Measure	65.05	95.86	85.48	78.23
	Accuracy		81.63		
Tree	Precision	60.53	72.48	78.44	74.4
	Sensitivity	76.51	67.64	80.18	56.31
	Specificity	87.57	91.79	86.81	95.63
	f-Measure	67.59	69.98	79.30	64.10
	Accuracy		72.01		
Naïve Bayes	Precision	60	73.41	78.13	73.14
	Sensitivity	74.84	67.12	80.29	57.66
	Specificity	87.57	92.23	86.55	95.22
	f-Measure	66.60	70.12	79.20	64.48
	Accuracy		71.85		
k-NN	Precision	55.13	66.87	75.98	51.14
	Sensitivity	59.25	57.02	87.93	40.54
	Specificity	87.99	90.97	83.37	91.26
	f-Measure	57.12	61.55	81.52	45.23
	Accuracy		66.00		
MLP	Precision	52.17	42.92	66.89	0
	Sensitivity	2.49	99.66	76.52	0
	Specificity	99.43	57.66	77.34	100
	f-Measure	4.75	60.00	71.38	0
	Accuracy		53.28		
SVM	Precision	60.23	97.7	8.33	26.34
	Sensitivity	33.06	94.35	0.11	93.24
	Specificity	94.56	99.29	99.27	41.16
	f-Measure	42.69	96.00	0.22	41.08
	Accuracy		46.64		

5.1.3. Other Datasets

When testing a cyber-attack dataset set, it was found that, initially, the IDTL method we proposed was highly effective at classifying different attacking behaviors. And many systems such as NSL-KDD are the popular datasets for detecting abnormalities. There are 5 classes including Normal, Dos, Probe, R2L, and U2R. Next is the Phishing website, a dataset that used phishing sites.

There are two classes, Phishing and Non-Phishing. Next, SAME which is an invasion on the Android operating system. There are two classes of smartphone applications: benign and malicious. Next, the spam base which is a spam-infested dataset. There are two classes, spam and non-spam. The dataset is still present, and IDTL is tested. In the case of NSL-KDD, we used the KDDTrain + _20Percent dataset for the training

dataset and KDDTest + for testing datasets. It's 100% used by developer's other datasets and uses 70% for training and 30% for testing. Testing data is new in the training process the results of the experiment show the efficiency as shown in Table 8.

Table 8. The comparison of accuracy machines learning between IDTL with other datasets.

Dataset	Method/ Accuracy					
	IDTL	C4.5	Naïve Bayes	k-NN	MLP	SVM
NSL-KDD	75.71	74.73	68.12	71.38	70.16	70.86
SAME	96.04	95.38	93.23	95.38	52.15	92.57
Phishing	91.05	89.93	87.13	83.54	49.23	71.30
Spambase	85	82.61	43.36	72.83	34.28	84.86

5.2 Discussion

The results show that IDTL is highly effective in classifying cyber-attacks. The structure of the IDTL is an incremental learning model that updates the model in sequential training. The equation we think is that the value of $Sb'$is computed in binary class in each class to identify any two nodes, thus obtaining the appropriate value of λ to find the distance to classify for the two nodes. If using ILDA's traditional Sb', the equation is to find the value between any class by the number of classes, for example, to classify 4 classes. Each class of the binary tree computes the other classes by taking the mean of all $\bar{x}'$ together with this there are some distortions in the calculation.

The IDTL focuses on only one layer, two layers, as a layered layer. The final class is the traditional, incremental learning model that is being updated to detect other types of cyber-attacks. And IDTL classified with distance in the testing phase by recognizing the values of the training phase, similar to the research of Aborujilah and Musa [45] which has the same high efficiency.

6. Conclusion

In this paper, IDTL developed a cyber- attack detection algorithm. Based on the ILDA algorithm, our research has tested on the main dataset, the Tor dataset, which is a dataset of hidden services. Since the current invasion is difficult to detect on the computer network, the goal of this research is to develop algorithms that can incrementally learn in hierarchical order. The proposed algorithms have feature selection to select only the most important features and classify them by using Pearson correlation as the algorithm for selecting the feature and developing the IDTL. Some enhancements have been made to optimize the IDTL structure. The results showed that IDTL was the most effective when compared to other methods, both binary and multiclass, as well as when tested with other cyber-attack datasets. It's high performance as well in a variety of ways regarding system intrusion. Future research will develop an incremental learning system based on current research. IoT or smart devices must be able to detect real-time intrusions at all times, such as in a factory or smart farmer.

7. Appendix

When new sample y in the kth class; $k\epsilon[1, M]$ as $\Sigma_{x\epsilon\{x_k\}}(x - \bar{x}_k) = 0$ Then, covariance matrix is equally updated as in the equation.

$$\Sigma_k' = \Sigma_k + \frac{{n_k}^2 + n_k}{n_k + 1^2}(y - \bar{x}_k)(y - \bar{x}_k)^T$$

$$= \Sigma_k + \frac{n_k}{n_k + 1}(y - \bar{x}_k)(y - \bar{x}_k)^T$$

References

[1] P. Mishra, E. S. Pilli, V. Varadharajan, and U. Tupakula, "Intrusion detection techniques in cloud environment: A survey," *Journal of Network and Computer Applications*, vol. 77, pp. 18–47, 2017.

[2] B. B. Zarpelão, R. S. Miani, C. T. Kawakani, and S. C. de Alvarenga, "A survey of intrusion detection in Internet of Things," *Journal of Network and Computer Applications*, vol. 84, pp. 25–37, 2017.

[3] A. L. Buczak and E. Guven, "A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection," *IEEE Communications Surveys Tutorials*, vol. 18, no. 2, pp. 1153–1176, Secondquarter 2016.

[4] G. Kim, S. Lee, and S. Kim, "A novel hybrid intrusion detection method integrating anomaly detection with misuse detection," *Expert Systems with Applications*, vol. 41, no. 4, Part 2, pp. 1690–1700, Mar. 2014.

[5] R. A. R. Ashfaq, X.-Z. Wang, J. Z. Huang, H. Abbas, and Y.-L. He, "Fuzziness based semi-supervised learning approach for intrusion detection system," *Information Sciences*, vol. 378, pp. 484–497, Feb. 2017.

[6] C. Yin, Y. Zhu, J. Fei, and X. He, "A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks," *IEEE Access*, vol. 5, pp. 21954–21961, 2017.

[7] D. J. Weller-Fahy, B. J. Borghetti, and A. A. Sodemann, "A Survey of Distance and Similarity Measures Used Within Network Intrusion Anomaly Detection," *IEEE Communications Surveys and Tutorials*, vol. 17, no. 1, pp. 70–91, Jan. 2015.

[8] Z. Tan, A. Jamdagni, X. He, and P. Nanda, "Network intrusion detection based on LDA for payload feature selection," *2010 IEEE Globecom Workshop on Web and Pervasive Security: 6-10 December 2010, Miami, Florida*, pp. 1545–1549, 2010.

[9] D. S. S. Sathya, "Discriminant Analysis based Feature Selection in KDD Intrusion Dataset," *International Journal of Computer Applications*, vol. 31, p. 7.

[10] R. Datti and B. Verma, "B.: Feature Reduction for Intrusion Detection Using Linear Discriminant Analysis," *International Journal on Engineering Science and Technology*, pp. 1072–1078.

[11] S. Singh and S. Silakari, "Generalized Discriminant Analysis algorithm for feature reduction in Cyber Attack Detection System," vol. 6, no. 1, p. 9, 2009.

[12] P. G. Jeya, M. Ravichandran, and C. S. Ravichandran, *Efficient Classifier for R2L and U2R Attacks.*.

[13] Q. Gao, Y. Huang, X. Gao, W. Shen, and H. Zhang, "A novel semi-supervised learning for face recognition," *Neurocomputing*, vol. 152, pp. 69–76, Mar. 2015.

[14] S. Pang, S. Ozawa, and N. Kasabov, "Incremental Linear Discriminant Analysis for Classification of Data Streams," p. 17.

[15] I. Syarif, A. Prugel-Bennett, and G. Wills, "Unsupervised Clustering Approach for Network Anomaly Detection," in *Networked Digital Technologies*, vol. 293, R. Benlamri, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 135–145.

[16] K. S. A. Kumar and A. M. M. O. Chacko, "Clustering Algorithms for Intrusion Detection: A Broad Visualization," in *Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies*, Udaipur, India, 2016, pp. 135:1–135:4.

[17] A. Bohara, U. Thakore, and W. H. Sanders, "Intrusion Detection in Enterprise Systems by Combining and Clustering Diverse Monitor Data," in *Proceedings of the Symposium and Bootcamp on the Science of Security*, Pittsburgh, Pennsylvania, 2016, pp. 7–16.

[18] M. Wurzenberger, F. Skopik, M. Landauer, P. Greitbauer, R. Fiedler, and W. Kastner, "Incremental Clustering for Semi-Supervised Anomaly Detection Applied on Log Data," in *Proceedings of the 12th International Conference on Availability, Reliability and Security*, Reggio Calabria, Italy, 2017, pp. 31:1–31:6.

[19] Z. Xue, Y. Shang, and A. Feng, "Semi-supervised outlier detection based on fuzzy rough C-means clustering," *Mathematics and Computers in Simulation*, vol. 80, no. 9, pp. 1911–1921, May 2010.

[20] Y. Yuan, G. Kaklamanos, and D. Hogrefe, "A Novel Semi-Supervised Adaboost Technique for Network Anomaly Detection," in *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, Malta, Malta, 2016, pp. 111–114.

[21] M. A. Jabbar, R. Aluvalu, and S. S. S. Reddy, "Cluster Based Ensemble Classification for Intrusion Detection System," in *Proceedings of the 9th International Conference on Machine Learning and Computing*, Singapore, Singapore, 2017, pp. 253–257.

[22] P. Arun Raj Kumar and S. Selvakumar, "Detection of distributed denial of service attacks using an ensemble of adaptive and hybrid neuro-fuzzy systems," *Computer Communications*, vol. 36, no. 3, pp. 303–319, Feb. 2013.

[23] S. T. Miller and C. Busby-Earle, "Multi-Perspective Machine Learning a Classifier Ensemble Method for Intrusion Detection," in *Proceedings of the 2017 International Conference on Machine Learning and Soft Computing*, Ho Chi Minh City, Vietnam, 2017, pp. 7–12.

[24] J.-G. Yang, J.-K. Kim, U.-G. Kang, and Y.-H. Lee, "Coronary Heart Disease Optimization System on Adaptive-network-based Fuzzy Inference System and Linear Discriminant Analysis (ANFIS—LDA)," *Personal Ubiquitous Comput.*, vol. 18, no. 6, pp. 1351–1362, Aug. 2014.

[25] A. A. Aburomman and M. B. I. Reaz, "Ensemble of binary SVM classifiers based on PCA and LDA feature extraction for intrusion detection," in *Proceedings of 2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference, IMCEC 2016*, Xi'an, China, 2017, pp. 636–640.

[26] A. Saad, C. Khalid, and J. Mohamed, "Network intrusion detection system based on Direct LDA," in *2015 Third World Conference on Complex Systems (WCCS)*, Marrakech, Morocco, 2015, pp. 1–6.

[27] K. M. A. Alheeti, A. Gruebler, and K. McDonald-Maier, "Using discriminant analysis to detect intrusions in external communication for self-driving vehicles," *Digital Communications and Networks*, vol. 3, no. 3, pp. 180–187, Aug. 2017.

[28] R. Datti and S. Lakhina, "Performance Comparison of Features Reduction Techniques for Intrusion Detection System," vol. 3, no. 1, p. 4, 2012.

[29] H. He, S. Chen, K. Li, and X. Xu, "Incremental Learning From Stream Data," *IEEE Transactions on Neural Networks*, vol. 22, no. 12, pp. 1901–1914, Dec. 2011.

[30] D. Bhosale and R. Ade, "Intrusion Detection Using Incremental Learning from Streaming Imbalanced Data," *International Journal of Managing Public Sector Information and Communication Technologies*, vol. 6, no. 1, pp. 09–20, Mar. 2015.

[31] L. Jin, K. Ding, and Z. Huang, "Incremental learning of LDA model for Chinese writer adaptation," *Neurocomputing*, vol. 73, no. 10, pp. 1614–1623, Jun. 2010.

[32] S. Pang, Y. Peng, T. Ban, D. Inoue, and A. Sarrafzadeh, "A federated network online network traffics analysis engine for cybersecurity," in *2015 International Joint Conference on Neural Networks (IJCNN)*, Killarney, Ireland, 2015, pp. 1–8.

[33] S.-Y. Ji, B.-K. Jeong, S. Choi, and D. H. Jeong, "A multi-level intrusion detection method for abnormal network behaviors," *Journal of Network and Computer Applications*, vol. 62, pp. 9–17, Feb. 2016.

[34] B. Subba, S. Biswas, and S. Karmakar, "Intrusion Detection Systems using Linear Discriminant Analysis and Logistic Regression," in *2015 Annual IEEE India Conference (INDICON)*, New Delhi, India, 2015, pp. 1–6.

[35] P. Di Lena and L. Margara, "Optimal Global Alignment of Signals by Maximization of Pearson Correlation," *Inf. Process. Lett.*, vol. 110, no. 16, pp. 679–686, Jul. 2010.

[36] R. De Maesschalck, D. Jouan-Rimbaud, and D. L. Massart, "The Mahalanobis distance," *Chemometrics and Intelligent Laboratory Systems*, vol. 50, no. 1, pp. 1–18, Jan. 2000.

[37] A. Habibi Lashkari, G. Draper Gil, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of Tor Traffic using Time based Features:," in *Proceedings of the 3rd International Conference on Information Systems Security and Privacy*, Porto, Portugal, 2017, pp. 253–262.

[38] L. Dhanabal and D. S. P. Shantharajah, "A Study on NSL-KDD Dataset for Intrusion Detection System Based on Classification Algorithms," vol. 4, no. 6, p. 7, 2015.

[39] M. Hopkins, E. Reeber, G. Forman and J. Suermondt. (1999). UCI Machine Learning Repository: Spambase Data Set [Online]. Available: https://archive.ics.uci.edu/ml/datasets/spambase. [Accessed: 20 Aug 2018].

[40] R. M. A. Mohammad and L. McCluskey. (2015). UCI Machine Learning Repository: Phishing Websites Data Set [Online]. Available: https://archive.ics.uci.edu/ml/datasets/phishing+websites. [Accessed: 20 Aug 2018].

[41] K. Demertzis and L. Iliadis, "SAME: An Intelligent Anti-malware Extension for Android ART Virtual Machine," in *Computational Collective Intelligence*, 2015, pp. 235–245.

[42] H. F. Eid, A. E. Hassanien, T. Kim, and S. Banerjee, "Linear Correlation-Based Feature Selection for Network Intrusion Detection Model," in *International Conference on Security of Information and Communication Networks*, Berlin, Heidelberg, 2013, pp. 240–248.

[43] D.-J. Chang, A. H. Desoky, M. Ouyang, and E. C. Rouchka, "Compute Pairwise Manhattan Distance and Pearson Correlation Coefficient of Data Points with GPU," in *2009 10th ACIS International Conference on*

Software Engineering, Artificial Intelligences, Networking and Parallel/ Distributed Computing, Daegu, Korea, 2009, pp. 501–506.
[44] H. Yao *et al.*, "An Incremental Local Outlier Detection Method in the Data Stream," *Applied Sciences*, vol. 8, no. 8, pp. 1248, Jul. 2018.
[45] A. Aborujilah and S. Musa, "Cloud-Based DDoS HTTP Attack Detection Using Covariance Matrix Approach," *Journal of Computer Networks and Communications.*, vol. 2017. pp.1-8, 2017

ประวัติผู้เขียน

ชื่อ	นางสาวพลอยพรรณ สอนสุวิทย์
วัน เดือน ปีเกิด	17 มกราคม 2526
ที่อยู่ปัจจุบัน	281/41 หมู่ 12 ตำบลหนองควาย อำเภอหางดง จังหวัดเชียงใหม่ 50230
ประวัติการศึกษา	(2548) วิทยาศาสตรบัณฑิต สาขาวิทยาศาสตร์และทคโนโลยีการอาหาร มหาวิทยาลัยแม่โจ้ (2552) วิศวกรรมศาสตร์มหาบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเชียงใหม่
ทุนการศึกษาที่ได้รับ	ทุนอุดหนุนการศึกษา มหาวิทยาลัยราชภัฏกำแพงเพชร
ผลงานวิชาการ	1. Sornsuwit, P., & Jaiyen, S. 2019. "A New Hybrid Machine Learning for Cybersecurity Threat Detection Based on Adaptive Boosting". *Engineering Journal*, 23(5). 2. Sornsuwit, P., & Jaiyen, S. 2019. "A New Hybrid Machine Learning for Cybersecurity Threat Detection Based on Adaptive Boosting". *Applied Artificial Intelligence*, 33(5), 462-482. 3. Sornsuwit, P., & Jaiyen, S. 2015. "Intrusion detection model based on ensemble learning for U2R and R2L attacks". 354-359 **In 2015 7th international conference on information technology and electrical engineering (ICITEE)**, Thailand: IEEE.