



การตรวจจับพฤติกรรมผิดปกติของเหตุการณ์การเข้าถึงไฟล์บนระบบคลาวด์แบบเรียลไทม์ โดยใช้เทคนิคการเรียนรู้ของเครื่องแบบสตรีม

Real-Time Anomaly Detection of Cloud File Access Events Using Streaming Machine Learning Techniques

พงษ์พิชญ์ เลิศเจริญวุฒา เสกสรรค์ ศิวาลัย และ ธงรบ อักษร*

สาขาวิชาเทคโนโลยีสารสนเทศ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏพิบูลสงคราม พิษณุโลก 65000

Phongphit Loescharoenwutha, Sakesan Sivilai and Thongrob Auxsorn*

Department of Information Technology, Faculty of Science and Technology,

Pibulsongkram Rajabhat University, Phitsanulok 65000

*Corresponding author: auxsorn@psru.ac.th

Received: 23 May 2026/ Revised: 21 June 2026/ Accepted: 23 June 2026

บทคัดย่อ

งานวิจัยนี้มีวัตถุประสงค์เพื่อพัฒนาแนวทางการตรวจจับพฤติกรรมผิดปกติของเหตุการณ์การเข้าถึงไฟล์บนระบบคลาวด์แบบเรียลไทม์ โดยใช้เทคนิคการเรียนรู้ของเครื่องแบบสตรีม งานวิจัยดำเนินการเป็น 5 ระยะ ได้แก่ การสร้างแบบจำลองพื้นฐาน การตรวจสอบเสถียรภาพของโมเดล การจัดอันดับความสำคัญของฟีเจอร์ การเปรียบเทียบผลภายใต้ชุดฟีเจอร์ที่ลดลง และการทดลองหลักบนข้อมูลแบบสตรีมที่จำลองขึ้นโดยการปรับโครงสร้างข้อมูลบันทึกเหตุการณ์ HDFS ให้สอดคล้องกับโครงสร้างเหตุการณ์การเข้าถึงไฟล์บนระบบคลาวด์ ซึ่งใช้ชุดข้อมูลรวมทั้งสิ้น 250,000 เหตุการณ์ (แบ่งเป็นเหตุการณ์ปกติ 225,000 เหตุการณ์ และเหตุการณ์ผิดปกติ 25,000 เหตุการณ์) โมเดลที่ใช้ในการศึกษา ได้แก่ Half-Space Trees (HST), Logistic Regression (LR) และ Adaptive Random Forest (ARF) โดยประเมินผลด้วยวิธีเพรีเควชันเซียล ซึ่งเหมาะสมกับข้อมูลที่ไหลเข้าต่อเนื่องตามเวลา ผลการศึกษาพบว่า ARF ให้ผลดีที่สุดเชิงตัวเลขเมื่อใช้ชุดฟีเจอร์หลักร่วมกับฟีเจอร์ที่สร้างเพิ่มเติม 11 ตัว โดยมี Accuracy เท่ากับ 0.849965 และ F1-score เท่ากับ 0.839121 อย่างไรก็ตามเมื่อพิจารณาพร้อมกับจำนวนฟีเจอร์และความซับซ้อนของระบบ พบว่าชุดฟีเจอร์หลักร่วมกับฟีเจอร์ที่สร้างเพิ่มเติม 5 ตัว มีความเหมาะสมมากกว่าสำหรับการนำไปใช้จริง โดยยังคงให้ Accuracy เท่ากับ 0.803863 และ F1-score เท่ากับ 0.782305 สรุปได้ว่า ARF เป็นโมเดลหลักที่เหมาะสมที่สุดสำหรับการตรวจจับความผิดปกติในข้อมูลสตรีมของระบบไฟล์บนคลาวด์ และผลที่ได้สามารถใช้เป็นฐานสำหรับการพัฒนาระบบตรวจจับและแจ้งเตือนความเสี่ยงในสภาพแวดล้อมจริงต่อไป

คำสำคัญ: การตรวจจับ ความผิดปกติ เหตุการณ์การเข้าถึงไฟล์ ระบบคลาวด์ การเรียนรู้ของเครื่องแบบสตรีม

Abstract

This study aimed to develop a real-time anomaly detection approach for cloud file access events using streaming machine learning techniques. The research was organized into five phases: baseline model construction, model stability analysis, feature importance ranking, reduced feature-set comparison, and the main experiment on a schema-adapted data stream transformed from HDFS logs to mimic cloud file access

events. The evaluation utilized a streaming dataset comprising a total of 250,000 events (consisting of 225,000 normal events and 25,000 anomalous events). The models examined in this study were Half-Space Trees (HST), Logistic Regression (LR), and Adaptive Random Forest (ARF), and their performance was evaluated using the prequential approach, which is suitable for continuously arriving data. The results showed that ARF achieved the highest numerical performance when the core feature set was combined with 11 additional engineered features, with an Accuracy of 0.849965 and an F1-score of 0.839121. However, when the number of features and system complexity were also considered, the core feature set combined with 5 additional engineered features was more appropriate for practical use, while still maintaining an Accuracy of 0.803863 and an F1-score of 0.782305. These findings indicate that ARF is the most suitable primary model for anomaly detection in cloud file-system streams, and that the selected feature configuration can serve as a basis for developing a risk detection and alerting system in real-world cloud environments.

Keywords: detection, anomaly, cloud, file access events, streaming machine learning

บทนำ

ในปัจจุบันคลาวด์คอมพิวเตอร์เป็นโครงสร้างพื้นฐานที่องค์กรจำนวนมากใช้สำหรับการจัดเก็บข้อมูล การประมวลผล และให้บริการระบบสารสนเทศ โดยเฉพาะการให้บริการโครงสร้างพื้นฐานบนคลาวด์หรือ Infrastructure as a Service (IaaS) ที่เปิดให้ผู้ใช้เข้าถึงทรัพยากรได้อย่างยืดหยุ่นมากขึ้น [1] อย่างไรก็ตามเมื่อข้อมูลและภาระงานถูกย้ายเข้าสู่สภาพแวดล้อมคลาวด์มากขึ้น ความเสี่ยงด้านความมั่นคงปลอดภัยของข้อมูลก็เพิ่มขึ้นตามไปด้วย โดยเฉพาะเหตุการณ์การเข้าถึงหรือเปลี่ยนแปลงไฟล์ที่อาจเกิดขึ้นโดยไม่ได้รับอนุญาต ซึ่งเป็นหนึ่งในจุดเริ่มต้นของการละเมิดข้อมูลในองค์กรได้ [2, 3]

งานวิทยานิพนธ์ก่อนหน้านี้ได้ศึกษาการวัดประสิทธิภาพของระบบบันทึกเหตุการณ์ในสภาพแวดล้อมคลาวด์ประเภท IaaS ซึ่งให้เห็นว่าระบบบันทึกเหตุการณ์มีความสำคัญต่อการเก็บหลักฐานของการเข้าถึงและการกระทำที่เกิดขึ้นกับไฟล์ โดยเฉพาะเมื่อต้องตรวจสอบย้อนหลังหรือระบุผู้ที่เกี่ยวข้องกับเหตุการณ์ผิดปกติ [4] งานดังกล่าวจึงเป็นฐานสำคัญที่ทำให้เห็นบทบาทของ log ในระบบคลาวด์อย่างเป็นรูปธรรม แต่ในอีกด้านหนึ่งยังเหลือช่องว่างที่ควรศึกษาต่อ คือ ข้อมูล log ยังถูกใช้ในลักษณะของการเก็บหลักฐานและตรวจสอบภายหลังเป็นหลัก มากกว่าการนำมาใช้วิเคราะห์พฤติกรรมผิดปกติในขณะที่เหตุการณ์กำลังเกิดขึ้น [5]

ช่องว่างนี้มีผลโดยตรงต่อการออกแบบระบบตรวจจับความเสี่ยง เพราะเหตุการณ์ของระบบไฟล์ในสภาพแวดล้อมคลาวด์ไม่ได้เกิดขึ้นเป็นชุดข้อมูลนิ่ง แต่เกิดขึ้นต่อเนื่องตามเวลา หากระบบรอให้เหตุการณ์ผ่านไปแล้วจึงตรวจสอบย้อนหลัง การตอบสนองต่อความเสี่ยงอาจช้าเกินไป โดยเฉพาะกรณีที่มีการเปลี่ยนสิทธิ์ไฟล์ การเข้าถึงไฟล์ที่มีความเสี่ยง หรือพฤติกรรมที่ค่อย ๆ เบี่ยงออกจากรูปแบบปกติ งานวิจัยนี้จึงนำช่องว่างดังกล่าวมาเป็นจุดตั้งต้นในการพัฒนาแนวทางตรวจจับพฤติกรรมผิดปกติของเหตุการณ์การเข้าถึงไฟล์ด้วยเทคนิคการเรียนรู้ของเครื่องแบบสตรีม

แม้ในช่วงหลังจะมีการนำเทคนิคการเรียนรู้ของเครื่องมาใช้กับข้อมูลบันทึกเหตุการณ์มากขึ้น แต่งานจำนวนหนึ่งยังประเมินผลบนชุดข้อมูลแบบคงที่ หรือเน้นโมเดลเพียงชนิดเดียว ทำให้ยังสะท้อนธรรมชาติของข้อมูลที่เปลี่ยนแปลงตามเวลาได้ไม่เต็มที่ [6-8] ด้วยเหตุนี้การประเมินผลแบบสตรีมจึงมีความสำคัญ เพราะช่วยจำลองลักษณะการทำงานของระบบตรวจจับที่ต้องรับข้อมูลใหม่เข้ามาอย่างต่อเนื่อง และต้องตัดสินใจจากข้อมูลที่มีอยู่ในขณะนั้น ไม่ใช่จากข้อมูลที่ถูกรวบรวมครบถ้วนไว้ล่วงหน้า

สำหรับงานวิจัยนี้ ข้อมูลตั้งต้นเป็นเหตุการณ์ปกติที่จัดเตรียมจากบันทึกเหตุการณ์ของระบบ HDFS/DataNode แล้วปรับให้อยู่ในรูปแบบเหตุการณ์ระบบไฟล์จำนวน 14 ไฟล์ [9] ข้อมูลส่วนนี้ถูกใช้เป็นฐานของพฤติกรรมปกติก่อนเข้าสู่การ



ทดลองหลัก ส่วนเหตุการณ์ผิดปกติในระยะที่ 5 ถูกสร้างเพิ่มเติมโดยอ้างอิงลักษณะพฤติกรรมจากกรอบ MITRE ATT&CK และปรับให้สอดคล้องกับฟีเจอร์ของข้อมูลระบบไฟล์ [10] วิธีนี้ช่วยให้การทดลองยังยึดอยู่กับโครงสร้างข้อมูลเดิม แต่มีบริบทของเหตุการณ์ผิดปกติที่เพียงพอต่อการประเมินความสามารถของโมเดล

จากเหตุผลดังกล่าว งานวิจัยนี้จึงมุ่งพัฒนาแนวทางการตรวจจับพฤติกรรมผิดปกติของเหตุการณ์การเข้าถึงไฟล์บนระบบคลาวด์แบบเรียลไทม์ โดยใช้เทคนิคการเรียนรู้ของเครื่องแบบสตรีม และประเมินผลด้วยวิธีเพริเคินเซียล กระบวนการดำเนินงานวิจัยถูกออกแบบแบ่งออกเป็น 5 ระยะ ตั้งแต่การสร้างแบบจำลองพื้นฐาน การทดสอบเสถียรภาพ การคัดเลือกฟีเจอร์ จนถึงการทดลองหลักบนข้อมูลสตรีมแบบผสม ในส่วนของแบบจำลอง ผู้วิจัยได้นำโมเดล Half-Space Trees (HST), Logistic Regression (LR) และ Adaptive Random Forest (ARF) มาใช้เป็นโมเดลหลักในการวิเคราะห์และเปรียบเทียบประสิทธิภาพร่วมกัน

การเลือกแนวทางสุดท้ายของงานวิจัยนี้ไม่ได้พิจารณาเพียงว่าโมเดลใดให้ค่าประสิทธิภาพสูงที่สุดเท่านั้น แต่พิจารณาพร้อมกับจำนวนฟีเจอร์ ความซับซ้อนของระบบ และความเป็นไปได้ในการนำผลไปต่อยอดเป็นระบบตรวจจับและแจ้งเตือนความเสี่ยงในสภาพแวดล้อมคลาวด์จริง

วิธีดำเนินการวิจัย

งานวิจัยนี้ออกแบบการทดลองเป็นการประเมินผลข้อมูลแบบสตรีม โดยแบ่งกระบวนการออกเป็น 5 ระยะ เพื่อค่อย ๆ ตรวจสอบตั้งแต่ข้อมูลตั้งต้น พฤติกรรมของโมเดล การคัดเลือกฟีเจอร์ ไปจนถึงการประเมินผลบนข้อมูลที่มีเหตุการณ์ผิดปกติร่วมด้วย การแยกระยะในลักษณะนี้ช่วยให้การตีความผลไม่ปะปนกันระหว่างการสร้างฐานพฤติกรรมปกติของระบบกับการตรวจจับ anomaly ในการทดลองหลัก

1. ข้อมูลและการจัดเตรียมข้อมูล

ข้อมูลตั้งต้นของงานวิจัยนี้จัดเตรียมจากชุดข้อมูล HDFS/DataNode logs ซึ่งเป็นบันทึกเหตุการณ์ของระบบไฟล์แบบกระจายที่ถูกใช้เป็นตัวอย่างสาธารณะในเอกสารของ Quickwit สำหรับการจัดทำและค้นคืนข้อมูล log เหตุผลที่เลือกใช้ข้อมูลกลุ่มนี้ เนื่องจากมีลักษณะใกล้เคียงกับเหตุการณ์ที่พบในระบบจัดเก็บไฟล์บนสภาพแวดล้อมคลาวด์ เช่น การอ้างอิงไฟล์ directory เวลา และบริบทของกระบวนการที่เกี่ยวข้องกับการทำงานของระบบไฟล์ แม้ข้อมูลต้นทางจะไม่ได้เป็น cloud production log โดยตรง แต่มีโครงสร้างพื้นฐานของระบบไฟล์กระจายตัว (Distributed File System) ที่มีคุณลักษณะร่วมสำคัญ เช่น การระบุสิทธิ์ผู้ใช้ เส้นทางไดเรกทอรี และพฤติกรรมการเข้าถึงไฟล์ในลักษณะเดียวกับสภาพแวดล้อมคลาวด์ประเภท IaaS งานวิจัยนี้จึงใช้ข้อมูลดังกล่าวในฐานะข้อมูลตัวแทน (Proxy Data) โดยนำมาผ่านกระบวนการปรับโครงสร้าง (Schema Mapping) เพื่อให้สอดคล้องกับพฤติกรรมการเข้าถึงไฟล์บนคลาวด์ก่อนนำไปทดสอบและใช้เป็นฐานพฤติกรรมปกติของระบบในการทดลองนี้

ก่อนนำเข้าสู่การทดลอง ข้อมูลต้นทางถูกปรับให้อยู่ใน schema ของเหตุการณ์การเข้าถึงไฟล์บนระบบคลาวด์ โดยกำหนดฟิลด์ดั้งเดิมจำนวน 14 ตัว ได้แก่ directory, file, Pid, PName, User, Mode, Link, Uid, Gid, Size, Inode, Atime, Mtime และ Ctime ฟิลด์เหล่านี้ถูกใช้เพื่อแทนองค์ประกอบสำคัญของเหตุการณ์ระบบไฟล์ ได้แก่ ผู้ใช้ กระบวนการ สิทธิ์ไฟล์ โครงสร้างไฟล์ ขนาดไฟล์ และเวลา การปรับข้อมูลในลักษณะนี้ทำให้ข้อมูลไม่ได้อยู่ในรูปข้อความ log ดิบอีกต่อไป แต่ถูกจัดรูปแบบให้เป็นเหตุการณ์หนึ่งรายการต่อหนึ่งระยะเบียน เพื่อให้เหมาะกับการประมวลผลแบบสตรีม โดยในการทดลองครั้งนี้ใช้ชุดข้อมูลรวมทั้งสิ้น 250,000 เหตุการณ์ แบ่งเป็นเหตุการณ์ปกติจำนวน 225,000 เหตุการณ์ (คิดเป็นร้อยละ 90) ซึ่งสกัดมาจาก HDFS logs และเหตุการณ์ผิดปกติจำนวน 25,000 เหตุการณ์ (คิดเป็นร้อยละ 10) ที่ถูกจำลองขึ้นตามกรอบพฤติกรรมโจมตีของ MITRE ATT&CK การกำหนดสัดส่วนในลักษณะนี้ทำขึ้นเพื่อให้สอดคล้องกับสภาพแวดล้อมจริงในระบบเครือข่ายที่มีลักษณะข้อมูลไม่สมดุล (Imbalanced Data Scenario) ซึ่งพฤติกรรมผิดปกติจะเกิดขึ้นเป็นอัตราส่วนที่น้อยมากเมื่อเทียบกับกิจกรรมปกติของระบบ



สำหรับเหตุการณ์ผิดปกติในการทดลองหลัก ผู้วิจัยสร้างเพิ่มเติมโดยอ้างอิงกรอบ MITRE ATT&CK แล้วปรับให้อยู่ใน schema เดียวกับข้อมูลปกติ [10, 11] เหตุการณ์ผิดปกติถูกออกแบบให้กระทบพีเจอร์ที่สัมพันธ์กับพฤติกรรมเสี่ยง เช่น การเปลี่ยนสิทธิ์ไฟล์ การเปลี่ยนแปลงเวลา การเข้าถึง directory และชนิดของไฟล์ แนวทางนี้ไม่ได้มีเป้าหมายเพื่อแทนเหตุการณ์โจมตีจริงทั้งหมดในระบบคลาวด์ แต่ใช้เพื่อสร้างสถานการณ์ทดสอบที่ควบคุมได้ และช่วยประเมินว่าโมเดลตอบสนองต่อพฤติกรรมที่เบี่ยงออกจากฐานปกติได้เพียงใด

ตารางที่ 1 สรุปการจัดเตรียมข้อมูล

แหล่งข้อมูลปกติ	รูปแบบที่ใช้จริง	พีเจอร์ดั้งเดิม	การสร้าง anomaly	รูปแบบประเมินผล
HDFS/DataNode logs	Schema-adapted events	14 file-system features	MITRE ATT&CK-based design	Prequential stream

ตารางที่ 1 สรุปที่มาและรูปแบบการจัดเตรียมข้อมูลที่ใช้ในการทดลอง เพื่อแยกให้ชัดเจนระหว่างข้อมูลปกติที่ปรับจากบริบท HDFS/DataNode logs กับเหตุการณ์ผิดปกติที่ออกแบบโดยอ้างอิงกรอบ MITRE ATT&CK

2. แบบจำลองที่ใช้ในการทดลอง

งานวิจัยนี้คัดเลือกแบบจำลองที่มีสถาปัตยกรรมและแนวคิดที่แตกต่างกัน 3 รูปแบบ เพื่อเปรียบเทียบประสิทธิภาพภายใต้สภาพแวดล้อมข้อมูลสตรีม โดยแบบจำลองทั้งหมดถูกกำหนดค่าพารามิเตอร์เริ่มต้น (Hyperparameters) ที่เหมาะสมตามมาตรฐานสากล ดังนี้

- Half-Space Trees (HST): เป็นแบบจำลองประเภท Unsupervised Anomaly Detection ที่ทำงานโดยการประเมินคะแนนความผิดปกติ (Anomaly Score) จากความหนาแน่นของพฤติกรรมข้อมูลโดยไม่ต้องพึ่งพายก่ากับ [12] งานวิจัยนี้กำหนดค่าพารามิเตอร์หลักได้แก่ จำนวนต้นไม้สุ่ม (n_trees) เท่ากับ 25, ความลึกสูงสุดของต้นไม้ (max_depth) เท่ากับ 15, และกำหนดค่าขนาดหน้าต่างรับข้อมูลเริ่มต้น ($window_size$) สำหรับการอัปเดตโครงสร้างโครงข่าย

- Logistic Regression (LR): เป็นแบบจำลองพื้นฐานประเภท Linear Classification ที่ประยุกต์ใช้ร่วมกับกลไก Stochastic Gradient Descent (SGD) เพื่อให้สามารถเรียนรู้และอัปเดตค่าน้ำหนัก (Weights) ที่ละระยะข้อมูลตามเป็นสตรีมได้ โดยกำหนดค่าพารามิเตอร์ให้อัตราการเรียนรู้ (learning_rate) เป็นแบบคงที่ที่ค่า 0.01 และใช้ฟังก์ชันการสูญเสียแบบ Log-loss ในส่วนของการประเมินผลเชิงเปรียบเทียบร่วมกันระหว่างแบบจำลองสายจำแนกประเภท (ARF, LR) และแบบจำลองคำนวณคะแนนความผิดปกติ (HST) ผู้วิจัยได้กำหนดจุดตัดสินใจตัดสินคลาส (Decision Threshold) สำหรับแปลงคะแนนของ HST ให้เป็นสถานะตรรกะ เพื่อให้สามารถวัดผลภายใต้เกณฑ์ชี้วัด ความแม่นยำ (Accuracy) และ ค่าเฉลี่ยความสมดุล (F1-score) เดียวกันได้อย่างเท่าเทียม

- Adaptive Random Forest (ARF): เป็นแบบจำลองประเภท Supervised Ensemble สำหรับข้อมูลสตรีมที่มีกลไกการรับมือกับการเปลี่ยนแปลงโครงสร้างข้อมูล (Concept Drift) [13, 14] งานวิจัยนี้กำหนดค่าพารามิเตอร์หลักได้แก่ จำนวนต้นไม้ในสตรีม (n_models) เท่ากับ 15, ระยะเวลาผ่อนปรนในการประเมินซ้ำ (grace_period) เท่ากับ 20, และใช้เกณฑ์การแยกโหนดแบบ Hoeffding Tree โดยมีการปรับใช้งานตัวตรวจสอบการดริฟต์ข้อมูลย่อยภายใน

3. ขั้นตอนการทดลอง

การทดลองในงานวิจัยนี้แบ่งออกเป็น 5 ระยะ เพื่อให้สามารถตรวจสอบข้อมูล พีเจอร์ และพฤติกรรมของแบบจำลองได้เป็นลำดับก่อนที่จะเข้าสู่การทดลองหลัก ระยะที่ 1 ใช้สำหรับสร้างแบบจำลองพื้นฐานและตรวจสอบความถูกต้องของกระบวนการเตรียมข้อมูล ระยะที่ 2 ใช้ตรวจสอบเสถียรภาพของโมเดลภายใต้ข้อมูลปกติแบบสตรีม ระยะที่ 3 ใช้จัดอันดับ



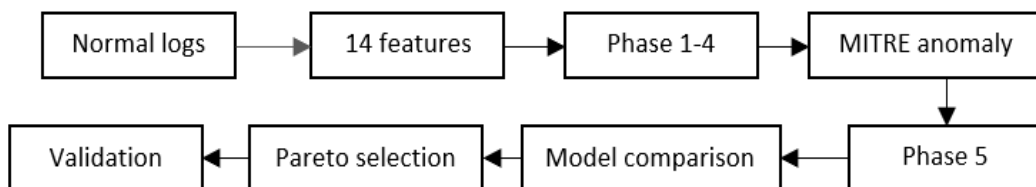
ความสำคัญของพีเจอรเพื่อคัดเลือกพีเจอรหลัก ระยะที่ 4 ใช้เปรียบเทียบผลของโมเดลภายใต้ชุดพีเจอรที่ลดลง และระยะที่ 5 เป็นการทดลองหลักบนข้อมูลแบบสตรีมที่ประกอบด้วยทั้งเหตุการณ์ปกติและเหตุการณ์ผิดปกติ

การแยกระยะการทดลองเช่นนี้ช่วยให้ระยะที่ 1-4 ทำหน้าที่เป็นฐานสำหรับตรวจสอบพฤติกรรมปกติของระบบก่อน ส่วนระยะที่ 5 จึงเป็นจุดที่นำเหตุการณ์ผิดปกติเข้ามาประเมินจริง วิธีนี้ช่วยลดความสับสนระหว่างกรณีที่โมเดลยังไม่นิ่งกับกรณีที่โมเดลตรวจจับ anomaly ได้ไม่ดี ซึ่งเป็นปัญหาที่พบได้ง่ายในงานที่ใช้ข้อมูลแบบสตรีม

ตารางที่ 2 แผนการทดลอง 5 ระยะ

ระยะ	จุดประสงค์หลัก	เงื่อนไขข้อมูล	ผลลัพธ์
1	Baseline	Normal-only	Baseline metrics
2	Stability	Normal-only	Model stability result
3	Feature ranking	Normal-only	Top5 core features ranked
4	Feature comparison	Normal-only	Baseline performance with reduced features
5	Main evaluation	ed stream	Selected config.

ในตารางที่ 2 คำว่า Mixed stream หมายถึง ข้อมูลแบบสตรีมที่ประกอบด้วยเหตุการณ์ปกติและเหตุการณ์ผิดปกติ ซึ่งใช้เฉพาะในการทดลองหลักระยะที่ 5



ภาพที่ 1 กรอบการดำเนินงานวิจัยแบบ 5 ระยะ

จากภาพที่ 1 แสดงให้เห็นภาพรวมของการทดลองตั้งแต่ต้นจนจบ งานวิจัยเริ่มต้นจากข้อมูลปกติที่รับจากบริบท HDFS/DataNode logs แล้วใช้ระยะที่ 1-4 ตรวจสอบฐานของข้อมูล เสถียรภาพของโมเดล และพีเจอรหลักก่อน จากนั้นจึงสร้างเหตุการณ์ผิดปกติโดยอ้างอิงกรอบ MITRE ATT&CK และนำเข้าสู่การประเมินในระยะที่ 5 ด้วยโมเดล HST, LR และ ARF ภาพนี้ช่วยแยกให้เห็นบทบาทของข้อมูลปกติในระยะที่ 1-4 เพื่อคัดเลือกพีเจอรหลัก (Top5 Core Features) จากนั้นในระยะที่ 5 จึงนำเหตุการณ์ผิดปกติเข้ามาประเมินร่วมกับการเพิ่มพีเจอรประยุกต์ แล้วใช้การวิเคราะห์แนวหน้าพาเรโต (Pareto Frontier Analysis) เพื่อเลือกคอนฟิกูเรชันสุดท้ายที่สมดุลที่สุดระหว่างประสิทธิภาพและภาระของระบบ

เมื่อกำหนดลำดับการทดลองและบทบาทของข้อมูลในแต่ละระยะแล้ว ขั้นตอนถัดไปคือ การกำหนดวิธีประเมินผลให้สอดคล้องกับลักษณะข้อมูลแบบสตรีม เพื่อให้ผลที่ได้สะท้อนการตัดสินใจของโมเดลตามลำดับเหตุการณ์จริงมากที่สุด

4. การประเมินผล

การประเมินผลในงานวิจัยนี้ใช้แนวทางแบบพริควินเชียน โดยให้แบบจำลองทำนายผลจากข้อมูลที่เข้ามาก่อน แล้วจึงอัปเดตแบบจำลองด้วยข้อมูลใหม่ในลำดับถัดไป วิธีนี้เหมาะกับข้อมูลแบบสตรีม เนื่องจากช่วยสะท้อนสถานการณ์ที่ระบบต้องตัดสินใจจากข้อมูลที่ไหลเข้ามาอย่างต่อเนื่อง ไม่ใช่จากชุดข้อมูลที่ถูกรวบรวมไว้ครบถ้วนล่วงหน้าแบบคงที่ ซึ่งสอดคล้องกับแนวทางการประเมินประสิทธิภาพในระบบประมวลผลสตรีมข้อมูลเชิงอุตสาหกรรมยุคใหม่ [15]

ตัวชี้วัดที่ใช้ในการประเมินผลประกอบด้วย Accuracy, Precision, Recall และ F1-score โดยใช้เปรียบเทียบผลของ HST, LR และ ARF ภายใต้เงื่อนไขการทดลองเดียวกัน อย่างไรก็ตามในการอภิปรายผลหลักของบทความนี้ ผู้วิจัยเน้น



Accuracy และ F1-score เป็นหลัก เนื่องจาก Accuracy ช่วยสะท้อนความถูกต้องโดยรวมของการจำแนก ส่วน F1-score ช่วยพิจารณาสมดุลระหว่างการตรวจจับเหตุการณ์ผิดปกติและความผิดพลาดของการจำแนกข้อมูล ซึ่งเหมาะกับลักษณะของงานตรวจจับ anomaly มากกว่าการพิจารณาค่าความถูกต้องเพียงตัวเดียว

ในการทดลองหลักระยะที่ 5 มีการเปรียบเทียบชุดฟีเจอร์ 3 รูปแบบ ได้แก่ ชุดฟีเจอร์หลัก 5 ตัว ชุดฟีเจอร์หลักร่วมกับฟีเจอร์ที่สร้างเพิ่มเติม 5 ตัว และชุดฟีเจอร์หลักร่วมกับฟีเจอร์ที่สร้างเพิ่มเติม 11 ตัว เพื่อพิจารณาว่าการเพิ่มฟีเจอร์เชิงพฤติกรรมและเชิงโครงสร้างช่วยให้แบบจำลองตรวจจับเหตุการณ์ผิดปกติได้ดีขึ้นเพียงใด

ตารางที่ 3 ชุดฟีเจอร์ที่ใช้ในการทดลองระยะที่ 5

ชุดฟีเจอร์	รายละเอียดฟีเจอร์ที่เพิ่ม	บทบาทในการทดลอง
Top5	ใช้ฟีเจอร์หลักดั้งเดิม 5 ตัว ที่ผ่านการจัดอันดับความสำคัญจากระยะที่ 3	ชุดเปรียบเทียบพื้นฐาน
Top5+5	เพิ่มฟีเจอร์ที่สร้างขึ้น 5 ตัว	ชุดที่เน้นความเหมาะสมต่อการใช้งานจริง
Top5+11	เพิ่มฟีเจอร์ที่สร้างขึ้น 11 ตัว	ชุดที่ใช้ดูขีดความสามารถสูงสุดเชิงตัวเลข

หมายเหตุ: ชุด Top5+5 ประกอบด้วยฟีเจอร์ที่สร้างเพิ่มเติม ได้แก่

unique_dir_count_user_w, perm_writebit_ratio_user_w, size_delta_abs, is_hidden_path และ file_ext_group

ฟีเจอร์ที่สร้างเพิ่มเติมในระยะที่ 5 ถูกคำนวณจากข้อมูลเหตุการณ์ปัจจุบันร่วมกับสถานะสะสมของเหตุการณ์ก่อนหน้าในระดับผู้ใช้ ไฟล์ และหน้าต่างข้อมูลล่าสุด เพื่อสะท้อนพฤติกรรมด้านเวลา ความถี่ของการเข้าถึง directory สิทธิการเขียนไฟล์ การเปลี่ยนแปลงขนาดไฟล์ โครงสร้าง path ประเภทไฟล์ และการใช้ process id ซ้ำ แนวทางนี้ช่วยให้ชุดฟีเจอร์ไม่ได้พิจารณาเฉพาะค่าของเหตุการณ์เดี่ยว แต่รวมสัญญาณเชิงพฤติกรรมที่เกิดขึ้นต่อเนื่องในข้อมูลแบบสตรีมด้วย [7, 8, 16]

ผลการวิจัย

ผลการทดลองในระยะที่ 1-4 ทำหน้าที่เป็นฐานสนับสนุนก่อนเข้าสู่การทดลองหลัก โดยระยะที่ 1 ใช้ตรวจสอบความถูกต้องของกระบวนการเตรียมข้อมูล ระยะที่ 2 ใช้ดูเสถียรภาพของโมเดลภายใต้ข้อมูลปกติแบบสตรีม ระยะที่ 3 ใช้จัดอันดับความสำคัญของฟีเจอร์ และระยะที่ 4 ใช้เปรียบเทียบผลภายใต้ชุดฟีเจอร์ที่ลดลง ผลจากระยะเหล่านี้ช่วยยืนยันว่าการทดลองหลักในระยะที่ 5 ไม่ได้เริ่มจากข้อมูลที่ยังไม่ผ่านการตรวจสอบ แต่มีฐานของข้อมูลปกติและชุดฟีเจอร์หลักรองรับก่อนแล้ว

ในระยะที่ 3 ฟีเจอร์ที่มีความสำคัญสูงสุด 5 อันดับแรก ได้แก่ User, Mtime, Ctime, Link และ Atime ฟีเจอร์กลุ่มนี้จึงถูกใช้เป็นชุดฟีเจอร์หลักสำหรับสร้าง configuration ในระยะที่ 5 โดยเปรียบเทียบระหว่าง Top5, Top5+5 และ Top5+11 เพื่อดูผลของการเพิ่มฟีเจอร์เชิงพฤติกรรมและเชิงโครงสร้างต่อประสิทธิภาพของโมเดล

จากตารางที่ 4 เมื่อพิจารณาเปรียบเทียบประสิทธิภาพระหว่างแบบจำลองทั้ง 3 รูปแบบ ภายใต้เงื่อนไขข้อมูลสตรีมผสม (Mixed Stream) พบว่าแบบจำลองโครงสร้างกลุ่มต้นไม้แบบปรับตัว Adaptive Random Forest (ARF) ให้ประสิทธิภาพทั้งในแง่ Accuracy และ F1-score สูงกว่าแบบจำลองอื่นอย่างมีนัยสำคัญในทุกชุดฟีเจอร์ โดยขีดความสามารถสูงสุดเชิงตัวเลขเกิดขึ้นในชุดฟีเจอร์ Top5+11 (Accuracy = 0.849965, F1-score = 0.839121) อย่างไรก็ตามเมื่อพิจารณาบนเกณฑ์ประสิทธิภาพขั้นต่ำที่กำหนดไว้ (Accuracy $\geq$ 0.80 และ F1-score $\geq$ 0.75) พบว่า ARF ภายใต้ชุดฟีเจอร์ Top5+5 เป็นคอนฟิกรูชันเดียวที่สามารถทำคะแนนผ่านเกณฑ์ดังกล่าวได้สำเร็จ โดยให้ค่า Accuracy เท่ากับ 0.803863 และ F1-score เท่ากับ 0.782305

ในทางกลับกันแบบจำลองพื้นฐานอย่าง Logistic Regression (LR) มีพฤติกรรมการเรียนรู้ที่ค่อนข้างจำกัดบนข้อมูลสตรีมที่มีมิติสูง แม้จะมีการเพิ่มฟีเจอร์ประยุกต์เข้ามาถึง 11 ตัว แต่ค่า F1-score สูงสุดยังคงอยู่ที่ 0.556210 ซึ่งไม่ผ่านเกณฑ์



ขั้นต่ำ ซึ่ให้เห็นว่าโครงสร้างการคำนวณเชิงเส้น (Linear Configuration) ไม่สามารถจับความสัมพันธ์ที่ซับซ้อนของรูปแบบพฤติกรรมการเข้าถึงไฟล์ที่จำลองขึ้นมาได้ดีเท่าที่ควร สำหรับแบบจำลอง Half-Space Trees (HST) มีค่า Accuracy เชิงตัวเลขต่ำที่สุด (เกาะกลุ่มอยู่ที่ประมาณ 0.48 - 0.50) เนื่องจากธรรมชาติของโมเดลเป็นประเภท Unsupervised Anomaly Detection ที่ประเมินคะแนนจากความหนาแน่นโดยไม่มีป้ายกำกับช่วยสอน แต่เมื่อพิจารณาค่า F1-score พบว่ามีแนวโน้มพัฒนาขึ้นอย่างชัดเจนเมื่อเพิ่มพีเจอร์เชิงพฤติกรรมเข้ามา โดยขยับจาก 0.512450 ขึ้นเป็น 0.674539 ในชุด Top5+11 ผลการทดลองเปรียบเทียบรวมในตารางนี้จึงช่วยยืนยันเป็นรูปธรรมว่า ARF เหมาะสมที่สุดสำหรับใช้เป็นโมเดลแกนหลักในการจำแนกประเภท ในขณะที่ HST ยังคงรักษาคุณค่าในฐานะโมเดลเสริม (Complementary Model) สำหรับดักจับสัญญาณผิดปกติในแง่ของความหนาแน่นที่โมเดลหลักอาจละเลยไป

ตารางที่ 4 ผลการเปรียบเทียบประสิทธิภาพระหว่างแบบจำลองในระยะที่ 5 ภายใต้ชุดพีเจอร์รูปแบบต่าง ๆ

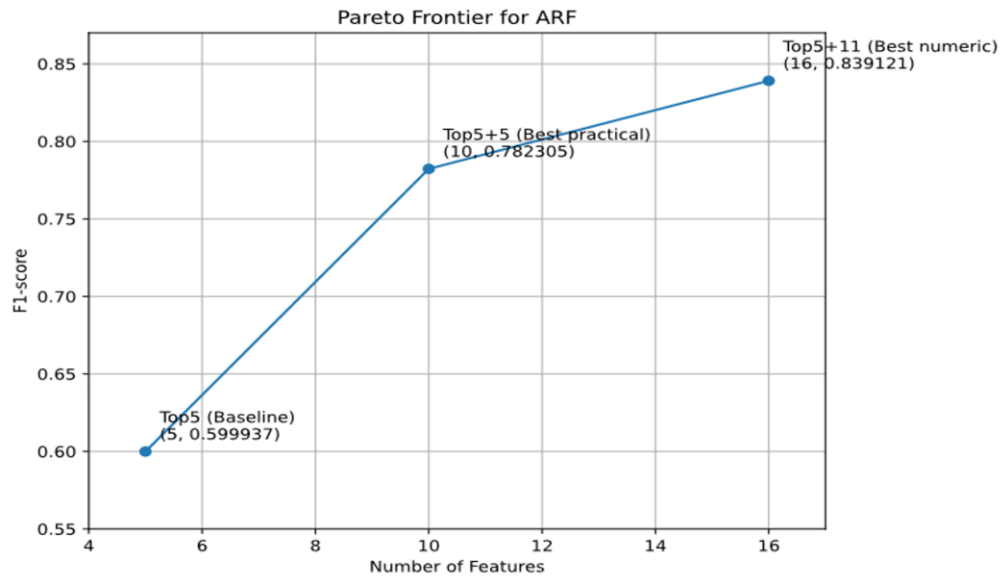
แบบจำลอง	ชุดพีเจอร์	จำนวนพีเจอร์	Accuracy	F1-score	บทบาท
ARF	Top5	5	0.703315	0.599937	โมเดลหลัก (Baseline)
	Top5+5	10	0.803863	0.782305	โมเดลหลักที่เหมาะสมจริง
	Top5+11	16	0.849965	0.839121	ขีดความสามารถสูงสุด
LR	Top5	5	0.612450	0.451230	โมเดลเชิงเส้นอ้างอิง
	Top5+5	10	0.654120	0.523410	โมเดลเชิงเส้นอ้างอิง
	Top5+11	16	0.678910	0.556210	โมเดลเชิงเส้นอ้างอิง
HST	Top5	5	0.485120	0.512450	โมเดลเสริมสแตนด์บาย
	Top5+5	10	0.502130	0.621450	โมเดลเสริมสแตนด์บาย
	Top5+11	16	0.508909	0.674539	โมเดลเสริม (Unsupervised)

สำหรับการเลือกชุดพีเจอร์เพื่อต่อยอดเชิงปฏิบัติ งานวิจัยนี้ใช้เกณฑ์ขั้นต่ำร่วมกับจำนวนพีเจอร์ที่ต้องประมวลผล โดยพิจารณาชุดที่ให้ Accuracy ไม่น้อยกว่า 0.80 และ F1-score ไม่น้อยกว่า 0.75 เป็นฐานก่อน จากนั้นจึงพิจารณาความคุ้มค่าของจำนวนพีเจอร์ที่เพิ่มขึ้น ภายใต้เกณฑ์นี้ Top5+5 เป็นชุดที่ยังผ่านระดับประสิทธิภาพที่กำหนดไว้ และใช้พีเจอร์น้อยกว่า Top5+11 จึงเหมาะกว่าในมุมมองระบบที่ต้องประมวลผลต่อเนื่องตามเวลา

เมื่อพิจารณาโมเดลอื่นร่วมด้วย พบว่า LR และ HST ยังไม่ให้ผลคงที่เท่ากับ ARF ในการทดลองหลัก โดยเฉพาะเมื่อเพิ่มพีเจอร์เชิงพฤติกรรมเข้ามา ARF ได้ประโยชน์จากพีเจอร์ที่สร้างเพิ่มเติมมากกว่าโมเดลอื่นอย่างเห็นได้ชัด จึงเหมาะสำหรับใช้เป็นโมเดลหลักของแนวทางที่เสนอ ส่วน HST ยังมีบทบาทในฐานะโมเดลเสริมที่ช่วยสะท้อนสัญญาณผิดปกติจากมุมมองที่แตกต่างออกไป

เพื่อให้การเลือกชุดพีเจอร์ไม่ยึดเพียงค่าประสิทธิภาพสูงสุด งานวิจัยนี้จึงพิจารณาความสัมพันธ์ระหว่างจำนวนพีเจอร์กับค่า F1-score เพิ่มเติมผ่าน Pareto frontier ดังภาพที่ 2 เพื่อช่วยในการเลือกชุดพีเจอร์ที่มีความสมดุล ซึ่งสอดคล้องกับแนวทางการประยุกต์ใช้แนวคิดแบบหลายวัตถุประสงค์ (Multi-objective optimization) ในกระบวนการเลือกพีเจอร์ เพื่อค้นหาจุดลงตัวสูงสุดระหว่างการลดมิติข้อมูลและการรักษาระดับประสิทธิภาพของแบบจำลอง [17] การพิจารณาในลักษณะนี้ช่วยให้เห็นได้ชัดขึ้นว่า ชุดพีเจอร์ใดให้ผลคุ้มค่าเมื่อเทียบกับภาระของระบบที่เพิ่มขึ้น

จากภาพที่ 2 การวิเคราะห์ Pareto frontier ช่วยให้เห็นความสัมพันธ์ระหว่างจำนวนพีเจอร์กับค่า F1-score ได้ชัดขึ้น ชุด Top5+11 แสดงขีดความสามารถสูงสุดของ ARF ในเชิงตัวเลข ส่วนชุด Top5+5 ให้ความสมดุลที่ดีกว่าระหว่างประสิทธิภาพกับจำนวนพีเจอร์ที่ใช้ ขณะที่ชุด Top5 แม้ใช้พีเจอร์น้อยที่สุด แต่ยังไม่ให้ผลต่ำเกินไปสำหรับเลือกเป็นแนวทางหลัก



ภาพที่ 2 Pareto frontier ของแบบจำลอง ARF

นอกจากผลการทดลองหลักแล้ว งานวิจัยนี้ยังตรวจสอบองค์ประกอบที่เกี่ยวข้องกับการนำผลไปต่อยอด ได้แก่ การทวนสอบค่าพารามิเตอร์ของ ARF การปรับเทียบค่า threshold และการทวนสอบ HST ในฐานะโมเดลเสริม ผลการตรวจสอบแสดงในตารางที่ 5

ตารางที่ 5 ผลการยืนยันเชิงเทคนิคในระยะที่ 5

รายการตรวจสอบ	Accuracy	F1-score	การตีความ
ARF re-validation	0.814367	0.796728	โมเดลหลัก
Threshold calibration	0.815418	0.815837	จุดตัดสินใจ
HST validation	0.508909	0.674539	โมเดลเสริม

การตรวจสอบในตารางที่ 5 ใช้ค่าพารามิเตอร์ที่กำหนดไว้เฉพาะสำหรับการทวนสอบระยะที่ 5 โดย ARF ใช้ค่า $n_models=15$ และ $grace_period=20$ ส่วนการปรับเทียบจุดตัดสินใจใช้ค่า $threshold=0.41$ ขณะที่ HST ใช้ค่า $n_trees=25$, $height=10$, $window_size=300$ และ $threshold_q=0.85$ การระบุค่าเหล่านี้ช่วยให้ผลการตรวจสอบเชื่อมโยงกลับไปยังเงื่อนไขของโมเดลที่ใช้จริง ไม่ใช่เป็นเพียงการรายงานค่าประสิทธิภาพปลายทางเท่านั้น

จากผลการทวนสอบ ARF ภายใต้ค่าพารามิเตอร์ที่ปรับแล้ว พบว่า Accuracy เท่ากับ 0.814367 และ F1-score เท่ากับ 0.796728 ซึ่งยังสนับสนุนข้อสรุปเดิมว่า ARF เป็นโมเดลหลักที่เหมาะสม ส่วนการปรับเทียบ threshold ให้ผลดีขึ้นอีก โดยมี Accuracy เท่ากับ 0.815418 และ F1-score เท่ากับ 0.815837 แสดงให้เห็นว่าจุดตัดสินใจของระบบมีผลต่อคุณภาพการแจ้งเตือนโดยตรง ขณะที่ HST ยังไม่เหมาะสมสำหรับเป็นโมเดลหลัก แต่สามารถใช้เป็นโมเดลเสริมเพื่อสะท้อนสัญญาณผิดปกติอีกมุมหนึ่งได้

อภิปรายและสรุปผลการวิจัย

ผลการทดลองของงานนี้สอดคล้องกับแนวคิดของงานด้าน Anomaly Detection บนข้อมูลสตรีมที่มองว่าโมเดลควรถูกประเมินภายใต้ลำดับข้อมูลที่เปลี่ยนแปลงตามเวลา ไม่ใช่เฉพาะชุดข้อมูลคงที่เพียงครั้งเดียว [6-8] ซึ่งธรรมชาติของข้อมูลที่มีลักษณะแปรผันตามเวลา (Concept Drift) ย่อมส่งผลกระทบต่อเสถียรภาพของแบบจำลอง [18] ในบริบทนี้ ARF



แสดงพฤติกรรมที่เหมาะสมกับข้อมูลแบบสตรีมมากกว่า LR และ HST โดยเฉพาะเมื่อมีการเพิ่มฟีเจอร์เชิงพฤติกรรมและเชิงโครงสร้างเข้ามาช่วยอธิบายเหตุการณ์ ผลที่ได้จึงชี้ว่าความเหมาะสมของโมเดลไม่ได้ขึ้นอยู่กับชนิดของโมเดลเพียงอย่างเดียว แต่ขึ้นอยู่กับวิธีแทนพฤติกรรมของเหตุการณ์ในข้อมูลสตรีมด้วย

ผลการวิจัยแสดงให้เห็นว่าชุดฟีเจอร์ Top5+5 มีความเหมาะสมที่สุดสำหรับการนำไปต่อยอดเชิงปฏิบัติในระบบจริง เนื่องจากการวิจัยนี้ใช้เกณฑ์ประสิทธิภาพขั้นต่ำ (Accuracy ไม่น้อยกว่า 0.80 และ F1-score ไม่น้อยกว่า 0.75) ร่วมกับจำนวนฟีเจอร์ที่ต้องประมวลผลเป็นหลักในการพิจารณา แม้ชุด Top5+11 จะให้ค่าประสิทธิภาพสูงสุดเชิงตัวเลข (Accuracy = 0.849965, F1-score = 0.839121) แต่การเพิ่มฟีเจอร์ขึ้นอีก 6 ตัว จะสร้างภาระในการคำนวณ (Computation Overhead) และเพิ่มความซับซ้อนในการจัดเก็บสถานะสะสมของข้อมูลในสภาพแวดล้อมข้อมูลสตรีมแบบเวลาจริง (Real-time Data Stream) ซึ่งข้อจำกัดด้านมิติข้อมูล (Dimensionality) บนระบบสตรีมนั้น การลดจำนวนฟีเจอร์ลงเป็นสิ่งจำเป็นเพื่อควบคุมทรัพยากรระบบ [19] การรักษาสสมดุลระหว่างประสิทธิภาพและความเร็วในการทำงานจึงเป็นสิ่งสำคัญยิ่ง [20, 21] การเลือกชุด Top5+5 จึงไม่ได้เป็นการเลือกผลที่ดีที่สุดบนตารางเพียงอย่างเดียว แต่เป็นจุดลงตัว (Trade-off) ที่ช่วยลดมิติข้อมูลแต่ยังคงรักษาระดับประสิทธิภาพของแบบจำลองไว้ได้อย่างเพียงพอต่อการใช้งานเชิงทดลอง และเหมาะสมกว่าในมุมมองของระบบที่ต้องประมวลผลต่อเนื่องตามเวลา

เมื่อพิจารณาเปรียบเทียบระหว่างแบบจำลอง พบว่า LR และ HST ยังไม่ให้ผลคงที่เท่ากับ ARF ในการทดลองหลัก โดยเฉพาะเมื่อเพิ่มฟีเจอร์เชิงพฤติกรรมเข้ามา ARF ได้ประโยชน์จากฟีเจอร์ที่สร้างเพิ่มเติมมากกว่าโมเดลอื่นอย่างเห็นได้ชัด จึงเหมาะสำหรับใช้เป็นโมเดลหลัก สำหรับโมเดล HST แม้จะให้ค่าประสิทธิภาพเชิงตัวเลขที่ต่ำกว่า ARF อย่างเห็นได้ชัด แต่นั่นเป็นผลมาจากธรรมชาติของโมเดลที่เป็นแบบ Unsupervised ซึ่งไม่ได้ปรับแตงน้ำหนักตามป้ายกำกับข้อมูลโจมตีโดยตรงเหมือนโมเดลอื่น อย่างไรก็ตามผลการทดลองนี้ไม่ได้หมายความว่า HST ไม่มีประสิทธิภาพ แต่ชี้ให้เห็นว่าในสถานการณ์ที่มีข้อมูลสตรีมแบบผสม (Mixed Stream) และต้องการความแม่นยำสูงในระดับจำแนกคลาสทันที โมเดลสาย Supervised Ensemble อย่าง ARF จะตอบโต้ภัยได้ดีกว่า ในขณะที่ HST ยังคงมีคุณค่าในฐานะโมเดลเสริมสแตนด์บาย (Complementary Model) ที่ช่วยสะท้อนและตรวจจับสัญญาณผิดปกติจากมุมมองที่แตกต่างออกไป เพื่อช่วยมองสัญญาณแปลกใหม่ที่ระบบจำแนกประเภททั่วไปอาจตรวจไม่พบ (Zero-day anomalies) ซึ่งระบบสามารถนำไปประยุกต์ใช้ร่วมกันในลักษณะการแจ้งเตือนแบบหลายระดับ (Multi-tiered Alerting System) ในอนาคตได้

อย่างไรก็ตามผลการทดลองนี้ยังควรถูกตีความภายใต้ขอบเขตของชุดข้อมูลผ่านการปรับ Schema และการสร้างเหตุการณ์ผิดปกติตามกรอบที่กำหนดไว้ งานวิจัยนี้จึงไม่ควรสรุปว่าแนวทางดังกล่าวแทนพฤติกรรมของระบบคลาวด์จริงได้ครบทุกกรณี แต่ควรมองว่าเป็นฐานการทดลองที่ช่วยชี้ให้เห็นแนวโน้มของโมเดลและชุดฟีเจอร์ที่เหมาะสม ก่อนนำไปทดสอบกับข้อมูลจากสภาพแวดล้อมจริงที่หลากหลายมากขึ้น ข้อจำกัดสำคัญของงานนี้อยู่ที่ลักษณะของข้อมูลที่ใช้ในการทดลอง ข้อมูลปกติถูกปรับจากบริบท HDFS/DataNode logs ให้เป็น Schema ของเหตุการณ์ระบบไฟล์บนคลาวด์ ส่วนเหตุการณ์ผิดปกติถูกออกแบบโดยอ้างอิงกรอบ MITRE ATT&CK ไม่ใช่ข้อมูลโจมตีจริงจากระบบคลาวด์ที่ใช้งานจริงทั้งหมด ดังนั้นผลการทดลองควรถูกตีความในฐานะการประเมินแนวทางและพฤติกรรมของโมเดลภายใต้ข้อมูลที่ควบคุมได้ มากกว่าการยืนยันว่าระบบพร้อมใช้งานจริงในทุกสภาพแวดล้อมทันที

โดยสรุปงานวิจัยนี้เสนอแนวทางการตรวจจับพฤติกรรมผิดปกติของเหตุการณ์การเข้าถึงไฟล์บนระบบคลาวด์ด้วยเทคนิคการเรียนรู้ของเครื่องแบบสตรีม ผลการประเมินสรุปได้ว่าแบบจำลอง ARF ร่วมกับชุดฟีเจอร์ประยุกต์ Top5+5 เป็นแนวทางที่มีความเหมาะสมและสมดุลที่สุดสำหรับการนำไปต่อยอดเป็นระบบตรวจจับและแจ้งเตือนความเสี่ยงในอนาคต การพัฒนาต่อไปควรทดลองติดตั้งใช้งานร่วมกับระบบจัดเก็บไฟล์บนคลาวด์จริง เพื่อประเมินค่าความหน่วง (Latency) และภาระการทำงานของระบบ (Resource Overhead) ภายใต้ปริมาณข้อมูลสตรีมที่แท้จริง ซึ่งอาจมีความหนาแน่นสูงกว่าชุดข้อมูลทดลอง



กิตติกรรมประกาศ

ผู้วิจัยขอขอบพระคุณอาจารย์ที่ปรึกษาและคณาจารย์ที่ให้คำแนะนำในการดำเนินงานวิจัย ตลอดจนข้อเสนอแนะที่ช่วยให้งานวิจัยนี้มีความสมบูรณ์มากยิ่งขึ้น ขอขอบคุณแหล่งข้อมูล งานวิจัย และกรอบอ้างอิงด้านความมั่นคงปลอดภัยไซเบอร์ที่ใช้เป็นพื้นฐานในการออกแบบการทดลองและการวิเคราะห์ผล รวมถึงผู้มีส่วนเกี่ยวข้องทุกท่านที่ให้การสนับสนุนงานวิจัยนี้สำเร็จลุล่วงได้

เอกสารอ้างอิง

1. Attou H, Guezzaz A, Benkirane S, Azrour M, Farhaoui Y. Cloud-based intrusion detection approach using machine learning techniques. *Big Data Min Anal* 2023;6(3):311-20.
2. Al-Ghuwairi AR, Sharrab Y, Al-Fraihat D, AlElaimat M, Alsarhan A, Algarni A. Intrusion detection in cloud computing based on time series anomalies utilizing machine learning. *J Cloud Comput* 2023;12(1):127.
3. Nassif AB, Talib MA, Nasir Q, Albadani H, Dakalbab FM. Machine learning for cloud security: a systematic review. *IEEE Access* 2021;9:20717-35.
4. Auxsorn T. A performance measurement framework for IaaS logging system with threat mitigation. Ph.D. Thesis, Naresuan University. Phitsanulok; 2022.
5. Padhiar S, Patel R. Outside the Closed World: On Using Machine Learning for Network Intrusion. in: *Proceedings of Information and Communication Technology for Intelligent Systems (ICTIS)*, September 8, 2023; Singapore; 2023. p. 265-70.
6. Lu T, Wang L, Zhao X. Review of anomaly detection algorithms for data streams. *Appl Sci* 2023;13(10):6353.
7. Cao Y, Ma Y, Zhu Y, Ting KM. Revisiting streaming anomaly detection: Benchmark and evaluation. *Artif Intell Rev* 2025;58(1):8.
8. Suarez-Cetrulo AL, Quintana D, Cervantes A. A survey on machine learning for recurring concept drifting data streams. *Expert Syst Appl* 2023;213:118934.
9. Quickwit. Index a logging dataset locally. Quickwit Documentation [Internet]. 2026 [cited 2026 May 13]. Available from: <https://quickwit.io/docs/0.7.1/get-started/tutorials/tutorial-hdfs-logs>
10. MITRE. ATT&CK Matrix for Enterprise. The MITRE Corporation [Internet]. 2026 [cited 2026 May 22]. Available from: <https://attack.mitre.org/>
11. Al-Sada B, Sadighian A, Oligeri G. MITRE ATT&CK: State of the art and way forward. *ACM Comput Surv* 2024;57(1):1-37.
12. Liu JJ, Cassales GW, Liu FT, Pfahringer B, Bifet A. Adaptive Isolation Forest. In: *International Conference on Discovery Science*, September 22, 2025; Slovenia; 2025. p. 363-78.
13. AlQabbany AO, Azmi AM. Measuring the effectiveness of adaptive random forest for handling concept drift in big data streams. *Entropy* 2021;23(7):859.
14. Cano A, Krawczyk B. ROSE: robust online self-adjusting ensemble for continual learning on imbalanced drifting data streams. *Mach Learn* 2022;111(7):2561-99.



15. Garcia-Mendez S, de Arriba-Perez F, Leal F, Veloso B, Malheiro B, Burguillo-Rial JC. An explainable machine learning framework for railway predictive maintenance using data streams from the metro operator of Portugal. *Sci Rep* 2025;15(1):27495.
16. Xu J, Lin C, Liu F, Wang Y, Xiong W, Li Z, et al. StreamAD: A cloud platform metrics-oriented benchmark for unsupervised online anomaly detection. *BenchCouncil Trans Benchmarks Stand Eval* 2023;3(2):100121.
17. Dhal P, Azad C. A fine-tuning deep learning with multi-objective-based feature selection approach for the classification of text. *Neural Comput Appl* 2024;36(7):3525–53.
18. Gama J, Zliobaite I, Bifet A, Pechenizkiy M, Bouchachia A. A survey on concept drift adaptation. *ACM Comput Surv* 2014;46(4):1-37.
19. Souiden I, Omri MN, Brahmi Z. A survey of outlier detection in high dimensional data streams. *Comput Sci Rev* 2022;44:100463.
20. Saiyed AM. Experimental Evaluation of Real-Time Data Streaming Analytics Using Reinforcement Learning. *J Integr Des Res* 2024;5(4):51-61.
21. Bifet A, Gavalda R, Holmes G, Pfahringer B. Machine learning for data streams: with practical examples in MOA. Massachusetts: The MIT Press; 2017.