

การพัฒนาระบบ Web Application Firewall (WAF) อัจฉริยะที่ใช้ Machine Learning ในการตรวจจับและป้องกันการโจมตี

เผด็จ สวิพันธุ์¹ และ ทิภาพร ศุภมิตร²

วันที่รับ 20 ธันวาคม 2568 วันที่แก้ไข 6 มีนาคม 2569 วันที่ตอบรับ 11 มีนาคม 2569

บทคัดย่อ

การวิจัยนี้มีวัตถุประสงค์เพื่อพัฒนาระบบ Web Application Firewall (WAF) อัจฉริยะ โดยประยุกต์ใช้เทคนิค Machine Learning และ Deep Learning เพื่อเพิ่มประสิทธิภาพในการตรวจจับและป้องกันการโจมตีทางเว็บ โดยโมเดลที่ใช้ในการวิจัยประกอบด้วยเทคนิคในกลุ่ม Machine Learning ได้แก่ Random Forest และ Support Vector Machine (SVM) และเทคนิคในกลุ่ม Deep Learning ได้แก่ Deep Neural Network (DNN) ระบบที่พัฒนาขึ้นใช้ข้อมูลจากชุด CIC-IDS2017 Dataset และ OWASP Benchmark Dataset สำหรับการฝึกสอนและทดสอบ โมเดลการเรียนรู้ ผลการทดลองพบว่า โมเดล Deep Neural Network (DNN) ซึ่งเป็นเทคนิคในกลุ่ม Deep Learning มีประสิทธิภาพสูงสุด โดยให้ค่าความแม่นยำเฉลี่ย 97.60% และค่า F1-Score เท่ากับ 96.65% สูงกว่าโมเดลในกลุ่ม Machine Learning ได้แก่ Random Forest และ SVM ระบบ WAF ที่พัฒนาขึ้นเชื่อมต่อกับโมเดลผ่าน RESTful API และสามารถตรวจจับการโจมตีได้แบบเรียลไทม์ โดยมีเวลาเฉลี่ยในการตอบสนองเพียง 120 มิลลิวินาที ผลการทดสอบการทำงานของระบบพบว่า ระบบมีความแม่นยำในการตรวจจับการโจมตีเฉลี่ย 96.10% และผู้ใช้งานมีระดับความพึงพอใจต่อระบบอยู่ในระดับ “มากที่สุด” (ค่าเฉลี่ย 4.53 จาก 5) โดยเฉพาะในด้านความถูกต้องของการตรวจจับและความรวดเร็วในการตอบสนอง ผลการวิจัยแสดงให้เห็นว่า การประยุกต์ใช้เทคนิค Deep Learning โดยเฉพาะโมเดล Deep Neural Network (DNN) สามารถช่วยเพิ่มประสิทธิภาพในการตรวจจับการโจมตีทางเว็บได้สูงกว่าเทคนิคในกลุ่ม Machine Learning และมีศักยภาพในการนำไปพัฒนาเป็นเครื่องมือด้านความปลอดภัยไซเบอร์ที่สามารถตรวจจับการโจมตีทั้งแบบที่เคยรู้จักและไม่เคยรู้จักมาก่อนได้อย่างมีประสิทธิภาพ

คำสำคัญ: การป้องกันการโจมตีทางเว็บ, การเรียนรู้ของเครื่อง, ไฟร์วอลล์แอปพลิเคชันเว็บ, การตรวจจับการบุกรุก, ปัญญาประดิษฐ์

¹ สาขาวิชาวิศวกรรมซอฟต์แวร์และระบบสารสนเทศ คณะวิทยาศาสตร์และเทคโนโลยี สถาบันเทคโนโลยีปทุมวัน

² สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษมบัณฑิต

* ผู้แต่ง, อีเมล: tipaporn.sup@kbu.ac.th

Development of an Intelligent Web Application Firewall (WAF)

Using Machine Learning for Attack Detection and Prevention

Padet Sawipan^{1*} and Tipaporn Supamid ²

Received 20 December 2025 Received 6 March 2026 Accepted 11 March 2026

Abstract

This research Development of an Intelligent Web Application Firewall (WAF) Using Machine Learning for Attack Detection and Prevention. The models used in this research consist of Machine Learning techniques, including Random Forest and Support Vector Machine (SVM), and a Deep Learning technique, namely Deep Neural Network (DNN). The system was trained and evaluated using data from the CIC-IDS2017 Dataset and the OWASP Benchmark Dataset. The experimental results indicate that the Deep Neural Network (DNN) model, which belongs to the Deep Learning group, achieved the highest performance, with an average accuracy of 97.60% and an F1-Score of 96.65%, outperforming the Machine Learning models, namely Random Forest and SVM. The developed WAF system is integrated with the trained model through a RESTful API and is capable of detecting web attacks in real time, with an average response time of 120 milliseconds. The system evaluation results show that the developed system achieved an average detection accuracy of 96.10%. In addition, user satisfaction with the system was at the highest level (mean score of 4.53 out of 5), particularly in terms of detection accuracy and response speed. The findings demonstrate that the application of Deep Learning techniques, especially the Deep Neural Network (DNN) model, can significantly enhance the effectiveness of web attack detection compared to traditional Machine Learning techniques. Furthermore, the proposed system shows strong potential as a cybersecurity tool capable of accurately detecting both known and previously unseen web attacks.

Keywords: Web attack prevention, Machine learning, Web application firewall, Intrusion detection, Artificial intelligence

¹ Department of Software Engineering and Information Systems, Faculty of Science and Technology, Pathumwan Institute of Technology Thailand

² Department of Computer Engineering, Faculty of Engineering, Kasembundit University

* Corresponding author: tipaporn.sup@kbu.ac.th

1. บทนำ

ในยุคดิจิทัลปัจจุบัน การพัฒนาเทคโนโลยีสารสนเทศได้มีบทบาทสำคัญอย่างยิ่งในการขับเคลื่อนภาคธุรกิจ เศรษฐกิจ และการบริหารจัดการของภาครัฐ โดยเฉพาะ “เว็บแอปพลิเคชัน (Web Application)” ซึ่งได้กลายเป็นโครงสร้างพื้นฐานหลักในการดำเนินงานและการให้บริการต่าง ๆ เว็บแอปพลิเคชันช่วยให้ผู้ใช้สามารถเข้าถึงข้อมูลและบริการได้อย่างสะดวก รวดเร็ว และสามารถใช้งานได้จากทุกที่ทุกเวลา ไม่ว่าจะผ่านคอมพิวเตอร์ โทรศัพท์มือถือ หรืออุปกรณ์อัจฉริยะอื่น ๆ ปัจจุบันองค์กรทั้งภาครัฐและเอกชนได้นำระบบเว็บแอปพลิเคชันมาใช้ในการให้บริการลูกค้า การดำเนินธุรกรรมทางการเงิน การจัดการฐานข้อมูล และการบริหารงานภายในองค์กรอย่างแพร่หลาย

อย่างไรก็ตาม การพึ่งพาระบบเว็บแอปพลิเคชันในระดับสูงย่อมมาพร้อมกับความเสี่ยงด้านความปลอดภัยไซเบอร์ (Cyber Security) เนื่องจากระบบเหล่านี้มักเปิดให้เข้าถึงผ่านเครือข่ายสาธารณะ เช่น อินเทอร์เน็ต ซึ่งทำให้ผู้ไม่ประสงค์ดีสามารถใช้ช่องโหว่ของระบบเพื่อเข้าถึงข้อมูลสำคัญขององค์กรได้ ปัจจุบันรูปแบบการโจมตีทางไซเบอร์มีความซับซ้อนและหลากหลายมากขึ้น จากรายงานของ OWASP (Open Web Application Security Project) ได้ระบุประเภทของการโจมตีที่พบบ่อยในเว็บแอปพลิเคชัน เช่น SQL Injection, Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF) และ Remote File Inclusion (RFI) ซึ่งเป็นภัยคุกคามที่สามารถก่อให้เกิดความเสียหายอย่างรุนแรงต่อระบบสารสนเทศขององค์กร [1]

การโจมตีเหล่านี้มีลักษณะและวัตถุประสงค์ที่แตกต่างกัน เช่น SQL Injection เป็นการแทรกคำสั่ง SQL ที่เป็นอันตรายผ่านช่องรับข้อมูลของระบบเพื่อเข้าถึงหรือแก้ไขข้อมูลในฐานข้อมูลโดยไม่ได้รับอนุญาต ส่วน Cross-Site Scripting (XSS) เป็นการฝังโค้ดสคริปต์ลงในหน้าเว็บเพื่อขโมยข้อมูลของผู้ใช้หรือเปลี่ยนเส้นทางไปยังเว็บไซต์ที่เป็นอันตราย ขณะที่ Cross-Site Request Forgery (CSRF) ใช้เทคนิคล่อลวงให้ผู้ใช้ที่กำลังเข้าสู่ระบบอยู่ทำการกระทำบางอย่างโดยไม่รู้ตัว เช่น การเปลี่ยนรหัสผ่านหรือการโอนเงิน ส่วน Remote File Inclusion (RFI) เป็นการแทรกไฟล์

จากภายนอกเข้าสู่ระบบเพื่อรันคำสั่งที่เป็นอันตราย ซึ่งอาจนำไปสู่การรั่วไหลของข้อมูลหรือความเสียหายต่อระบบได้

เพื่อป้องกันภัยคุกคามดังกล่าว จึงมีการพัฒนาเครื่องมือด้านความปลอดภัยที่เรียกว่า Web Application Firewall (WAF) ซึ่งเป็นระบบที่ทำหน้าที่ตรวจสอบและกรองคำร้องขอ (Request) ที่ส่งมายังเว็บเซิร์ฟเวอร์ก่อนที่จะเข้าสู่ระบบหลัก โดย WAF จะวิเคราะห์รูปแบบข้อมูล พฤติกรรมของผู้ใช้ และพารามิเตอร์ต่าง ๆ เพื่อระบุว่าคำร้องขอนั้นมีลักษณะของการโจมตีหรือไม่ หากตรวจพบความผิดปกติ ระบบจะทำการบล็อกคำร้องขอนั้นทันที เพื่อป้องกันความเสียหายที่อาจเกิดขึ้นกับระบบ [2]-[4]

อย่างไรก็ตาม WAF แบบดั้งเดิม (Traditional WAF) ส่วนใหญ่ใช้หลักการตรวจจับแบบกฎเกณฑ์ (Rule-Based) หรือการตรวจสอบจากลายเซ็นการโจมตี (Signature-Based) ซึ่งต้องอาศัยฐานข้อมูลของรูปแบบการโจมตีที่เคยเกิดขึ้นมาก่อน แม้ว่าวิธีการดังกล่าวจะสามารถตรวจจับการโจมตีที่รู้จักแล้ว (Known Attacks) ได้อย่างมีประสิทธิภาพ แต่ยังมีข้อจำกัดในการตรวจจับการโจมตีรูปแบบใหม่ที่ไม่เคยปรากฏมาก่อน หรือที่เรียกว่า Zero-Day Attacks รวมทั้งอาจเกิดปัญหาการตรวจจับผิดพลาด เช่น False Positive และ False Negative ซึ่งส่งผลกระทบต่อความแม่นยำของระบบ [5]

จากข้อจำกัดของระบบแบบดั้งเดิม นักวิจัยหลายท่านจึงได้ศึกษาการนำเทคนิค Machine Learning และ Deep Learning มาประยุกต์ใช้ในการตรวจจับการโจมตีของเว็บแอปพลิเคชัน เพื่อเพิ่มประสิทธิภาพในการวิเคราะห์ข้อมูลและการตรวจจับพฤติกรรมที่ผิดปกติ ตัวอย่างเช่น Dawadi และคณะ [6] ได้พัฒนา Web Application Firewall ที่ใช้เทคนิค Deep Learning ในการตรวจจับการโจมตีทางเว็บ ซึ่งสามารถเพิ่มความแม่นยำในการตรวจจับได้สูงกว่าวิธีแบบเดิม ขณะที่ Betarte และคณะ [5] ได้เสนอการใช้โมเดล Machine Learning สำหรับการตรวจจับการโจมตีเว็บแอปพลิเคชัน และพบว่าสามารถจำแนกประเภทของการโจมตีได้อย่างมีประสิทธิภาพ

นอกจากนี้ Moradi Vartouni และคณะ [7] ได้นำ Deep Neural Network มาประยุกต์ใช้ในการพัฒนา WAF ที่สามารถตรวจจับพฤติกรรมที่ผิดปกติของคำร้องขอจากผู้ใช้อย่างมีประสิทธิภาพ ขณะที่ Ito และ Iyatomi [3] ได้ใช้ Convolutional Neural Network (CNN) ในการวิเคราะห์ข้อมูลคำร้องขอเว็บแอปพลิเคชัน และสามารถเพิ่มประสิทธิภาพในการตรวจจับการโจมตีได้ดีขึ้น งานวิจัยของ Krishnan และคณะ [8] ยังได้เสนอการใช้โมเดล Deep Neural Network ร่วมกับเทคนิค Ensemble Learning เพื่อป้องกันการโจมตี Cross-Site Scripting ซึ่งให้ผลลัพธ์ที่มีความแม่นยำสูง

ในภาพรวม งานวิจัยหลายฉบับได้แสดงให้เห็นว่าเทคนิค Deep Learning มีศักยภาพสูงในการตรวจจับภัยคุกคามด้านความปลอดภัยไซเบอร์ โดยสามารถวิเคราะห์รูปแบบข้อมูลที่ซับซ้อนและตรวจจับพฤติกรรมที่ผิดปกติได้อย่างมีประสิทธิภาพ [9]–[11] ซึ่งทำให้นำแนวทางใช้ Machine Learning และ Deep Learning กลายเป็นแนวทางสำคัญในการพัฒนาระบบรักษาความปลอดภัยสำหรับเว็บแอปพลิเคชันในปัจจุบัน

ดังนั้น งานวิจัยนี้จึงมุ่งเน้นการพัฒนา “Web Application Firewall อัจฉริยะ (Intelligent WAF)” โดยประยุกต์ใช้เทคนิค Machine Learning และ Deep Learning เพื่อเพิ่มประสิทธิภาพในการตรวจจับและป้องกันการโจมตีทางเว็บ ระบบที่พัฒนาขึ้นสามารถเรียนรู้รูปแบบพฤติกรรมของการใช้งานที่ปกติและผิดปกติจากข้อมูลจริง และสามารถตรวจจับการโจมตีที่ไม่เคยปรากฏมาก่อนได้ (Anomaly Detection) ซึ่งช่วยลดข้อจำกัดของระบบ WAF แบบดั้งเดิม

นอกจากนี้ ผู้วิจัยยังได้พัฒนาแดชบอร์ด (Dashboard) สำหรับผู้ดูแลระบบ เพื่อใช้ในการติดตามสถานะการทำงานของระบบ วิเคราะห์เหตุการณ์การโจมตี และแสดงข้อมูลเชิงสถิติที่เกี่ยวข้อง เช่น จำนวนคำร้องที่ถูกบล็อก ประเภทของการโจมตีที่ตรวจพบ และแนวโน้มของภัยคุกคาม เพื่อช่วยให้ผู้ดูแลระบบสามารถตัดสินใจและตอบสนองต่อเหตุการณ์ด้านความปลอดภัยได้อย่างมีประสิทธิภาพ

2. วิธีดำเนินงาน

งานวิจัยนี้แบ่งขั้นตอนการดำเนินการออกเป็น 5 ระยะหลัก ดังนี้

ขั้นตอนที่ 1: การวิเคราะห์และเก็บรวบรวมข้อมูล (Data Collection and Analysis)

1. ศึกษาข้อมูลลักษณะของการโจมตีบนเว็บ (Web Attacks) จากเอกสารทางวิชาการและฐานข้อมูลมาตรฐาน (CIC-IDS, OWASP, MITRE ATT&CK) [1], [5], [9]

2. คัดเลือกตัวแปร (Feature Selection) ที่สำคัญ เช่น HTTP Method, Request Size, URL Parameter, Response Code, Header Content-Type [12]

3. ทำความสะอาดข้อมูล (Data Cleaning) และแปลงข้อมูลให้อยู่ในรูปแบบตัวเลขเพื่อป้อนเข้าสู่โมเดล

ขั้นตอนที่ 2: การออกแบบโมเดล Machine Learning (Model Design)

ในขั้นตอนนี้เป็นการออกแบบโมเดลการเรียนรู้ของเครื่อง (Machine Learning Models) เพื่อใช้ในการตรวจจับและจำแนกการโจมตีของเว็บแอปพลิเคชัน โดยมีขั้นตอนดังต่อไปนี้

1) การเลือกอัลกอริทึม (Algorithm Selection) ผู้วิจัยเลือกใช้อัลกอริทึมที่เหมาะสมสำหรับการจำแนกและตรวจจับการโจมตีของเว็บแอปพลิเคชัน จำนวน 3 โมเดล ได้แก่

- Random Forest ใช้สำหรับการจำแนกข้อมูลแบบหลายคลาส (Multi-class Classification) โดยอาศัยการรวมผลลัพธ์จากต้นไม้ตัดสินใจ (Decision Trees) หลายต้นเพื่อลดความคลาดเคลื่อนและเพิ่มความแม่นยำของการจำแนกข้อมูล

- Support Vector Machine (SVM) ใช้สำหรับการจำแนกข้อมูลและตรวจจับขอบเขตของพฤติกรรมที่ผิดปกติ โดยอาศัยหลักการหาขอบเขตการแบ่งข้อมูล (Hyperplane) ที่เหมาะสมที่สุดระหว่างข้อมูลแต่ละประเภท

• Deep Neural Network (DNN) ใช้สำหรับการวิเคราะห์รูปแบบข้อมูลที่มีความซับซ้อนและการตรวจจับพฤติกรรมที่ผิดปกติของคำร้องขอ (Request) ที่ส่งมายังเว็บแอปพลิเคชัน โดยสามารถเรียนรู้ลักษณะเชิงลึกของข้อมูลได้อย่างมีประสิทธิภาพ [7]

โครงสร้างของโมเดล Deep Neural Network (DNN) โมเดล DNN ที่ใช้ในการวิจัยนี้ประกอบด้วย Input Layer, Hidden Layers และ Output Layer โดยมีรายละเอียดดังนี้

Input Layer รับข้อมูลคุณลักษณะ (Features) ที่ได้จากการประมวลผลชุดข้อมูล

Hidden Layer จำนวน 3 ชั้น ได้แก่

- Hidden Layer ที่ 1: 128 neurons ใช้ ReLU (Rectified Linear Unit) เป็น activation function

- Hidden Layer ที่ 2: 64 neurons ใช้ ReLU เป็น activation function

- Hidden Layer ที่ 3: 32 neurons ใช้ ReLU เป็น activation function

Output Layer ใช้ Softmax Activation Function สำหรับการจำแนกประเภทของการโจมตีแบบหลายคลาส นอกจากนี้ในการฝึกสอนโมเดลใช้ Adam Optimizer และใช้ Categorical Cross-Entropy เป็นฟังก์ชันสำหรับคำนวณค่าความสูญเสีย (Loss Function) เพื่อเพิ่มประสิทธิภาพในการเรียนรู้ของโมเดล

2) การแบ่งชุดข้อมูล (Dataset Splitting)

ชุดข้อมูลที่ใช้ในการทดลองถูกแบ่งออกเป็น 3 ส่วน เพื่อใช้ในการฝึกสอนและประเมินประสิทธิภาพของโมเดล ดังนี้

ส่วนที่ 1 Training Set จำนวน 70% ของข้อมูลทั้งหมด ใช้สำหรับฝึกสอนโมเดล

ส่วนที่ 2 Validation Set จำนวน 15% ของข้อมูลทั้งหมด ใช้สำหรับปรับแต่งพารามิเตอร์ของโมเดล และป้องกันปัญหา Overfitting

ส่วนที่ 3 Test Set จำนวน 15% ของข้อมูลทั้งหมด ใช้สำหรับทดสอบและประเมินประสิทธิภาพของโมเดลหลังจากการฝึกสอนเสร็จสิ้น

ขั้นตอนที่ 3 การพัฒนาและฝึกสอนโมเดล (Model Training)

ในขั้นตอนนี้ดำเนินการฝึกสอนโมเดล Random Forest, Support Vector Machine (SVM) และ Deep Neural Network (DNN) โดยใช้ชุดข้อมูล Training Set ที่ได้จากการเตรียมข้อมูล เพื่อให้โมเดลสามารถเรียนรู้รูปแบบของข้อมูลปกติและข้อมูลที่เป็นการโจมตีของเว็บแอปพลิเคชัน

เพื่อเพิ่มประสิทธิภาพของโมเดล ผู้วิจัยใช้เทคนิค k-Fold Cross-Validation โดยกำหนดค่า $k = 5$ (5-fold Cross-Validation) ซึ่งเป็นการแบ่งชุดข้อมูลฝึกสอนออกเป็น 5 ส่วน และทำการฝึกสอนโมเดล 5 รอบ โดยในแต่ละรอบจะใช้ข้อมูล 4 ส่วนสำหรับการฝึกสอน และอีก 1 ส่วนสำหรับการตรวจสอบผลลัพธ์ จากนั้นนำค่าเฉลี่ยของผลลัพธ์ที่ได้มาใช้อธิบายประสิทธิภาพของโมเดล วิธีการนี้ช่วยลดความเอนเอียงของข้อมูล และช่วยป้องกันปัญหา Overfitting

นอกจากนี้ ผู้วิจัยใช้เทคนิค Grid Search เพื่อค้นหาค่าพารามิเตอร์ที่เหมาะสมที่สุดสำหรับแต่ละโมเดล โดยกำหนดช่วงค่าพารามิเตอร์ เช่น Random Forest ($n_estimators = 100-300$, $max_depth = 10-30$), SVM (kernel = linear, rbf และ $C = 0.1-10$) และ DNN (batch_size = 32 หรือ 64 และ epochs = 50 หรือ 100)

หลังจากการฝึกสอนโมเดลแล้ว จะทำการประเมินประสิทธิภาพโดยใช้ชุดข้อมูล Test Set ผ่านตัวชี้วัด ได้แก่ Accuracy, Precision, Recall, F1-score และ ROC Curve เพื่อวิเคราะห์ความสามารถของโมเดลในการตรวจจับการโจมตีของเว็บแอปพลิเคชัน และใช้ผลลัพธ์ดังกล่าวในการเปรียบเทียบประสิทธิภาพของแต่ละโมเดล เพื่อนำโมเดลที่มีประสิทธิภาพสูงสุดไปใช้ในระบบ Web Application Firewall ที่พัฒนาขึ้น.

ขั้นตอนที่ 4: การพัฒนาและเชื่อมต่อบริบบ WAF อัจฉริยะ (Smart WAF Integration)

1. นำโมเดลที่ผ่านการฝึกสอนแล้วบูรณาการเข้ากับระบบ WAF ผ่าน REST API

2. ระบบมีความสามารถตรวจจับและบล็อกคำขอ (Request) แบบเรียลไทม์ได้

3. ส่วนแสดงผล (Dashboard) แสดงข้อมูลสถิติ เช่น จำนวนการโจมตี, ชนิดการโจมตี, สถานะการบล็อก, และการแจ้งเตือน (Alert)

ขั้นตอนที่ 5: การทดสอบและประเมินผลระบบ (System Evaluation)

1. ทดสอบระบบด้วยชุดข้อมูลใหม่ที่ไม่เคยเห็นมาก่อน เพื่อวัดประสิทธิภาพจริง
2. ประเมินความพึงพอใจของผู้ใช้ (User Evaluation) โดยใช้แบบสอบถามมาตราส่วน Likert 5 ระดับ
3. วิเคราะห์ผลการทดสอบและประเมินประสิทธิภาพ ของระบบ

การประเมินความพึงพอใจของผู้ใช้งาน

การประเมินใช้กลุ่มตัวอย่างจำนวน 30 คน ซึ่งเป็น

- นักศึกษา สาขาวิศวกรรมซอฟต์แวร์ระบบสารสนเทศ สถาบันเทคโนโลยีปทุมวัน
- อาจารย์ผู้เชี่ยวชาญด้านความปลอดภัยไซเบอร์
- เจ้าหน้าที่ IT ที่เกี่ยวข้องกับระบบเครือข่าย

แบบสอบถามประกอบด้วย 3 ด้านหลัก ได้แก่

1. ด้านความถูกต้องของการตรวจจับ
2. ด้านความสะดวกในการใช้งานระบบ
3. ด้านการแสดงผลของแดชบอร์ด
4. ด้านความเสถียรของระบบ และ
5. ด้านการตอบสนองของระบบ

ข้อมูลจากแบบสอบถามถูกนำมาวิเคราะห์โดยใช้ค่าเฉลี่ย (Mean) และส่วนเบี่ยงเบนมาตรฐาน (Standard Deviation) เพื่อประเมินระดับความพึงพอใจโดยรวมของผู้ใช้งาน

3. ผลและวิจารณ์ผล

ผลการพัฒนาระบบ

การพัฒนา ระบบ Web Application Firewall (WAF) อัจฉริยะ ที่ใช้ Machine Learning (ML) เป็นเครื่องมือหลักในการตรวจจับและป้องกันการโจมตีทางเว็บ ประกอบด้วยผลการดำเนินงานทั้งในด้าน การพัฒนาโมเดล, การบูรณาการระบบ, และผลการทำงานของระบบ ดังนี้

1. ผลการพัฒนาโมเดล Machine Learning

ในงานวิจัยนี้ ได้ดำเนินการสร้างและทดสอบโมเดล Machine Learning 3 แบบ ได้แก่

1. Random Forest (RF)
2. Support Vector Machine (SVM)
3. Deep Neural Network (DNN)

ชุดข้อมูลที่ใช้ในการฝึกสอนคือ CIC-IDS2017 Dataset [13] และข้อมูลจำลองจากการโจมตีจริง (OWASP Benchmark) [1] รวมทั้งหมด 250,000 รายการ โดยแบ่งเป็น

- Training Set 70%
- Validation Set 15%
- Test Set 15%

ส่วนการเขียนโปรแกรม สำหรับโมเดล Machine Learning จะนำเข้าสู่ระบบ Server เป็น ภาษา Python (Flask) ดังรูปที่ 1 และผลการทดสอบประสิทธิภาพของโมเดล สรุปได้ดังตารางที่ 1

```
change.py
1 # train model.py
2 import pandas as pd
3 from sklearn.model_selection import train_test_split
4 from sklearn.ensemble import RandomForestClassifier
5 import joblib
6
7 # ตัวอย่างข้อมูลโจมตี (1 = attack, 0 = normal)
8 data = {
9     "payload_length": [10, 50, 100, 20, 5, 80],
10    "contains_sql": [0,1,1,0,0,1],
11    "contains_script": [0,0,1,0,0,1],
12    "label": [0,1,1,0,0,1]
13 }
14
15 df = pd.DataFrame(data)
16 X = df[["payload_length", "contains_sql", "contains_script"]]
17 y = df["label"]
18
19 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
20 model = RandomForestClassifier()
21 model.fit(X_train, y_train)
22
23 print("Model Accuracy:", model.score(X_test, y_test))
24
25 # บันทึกโมเดล
26 joblib.dump(model, "waf_model.pkl")
27
```

รูปที่ 1 ตัวอย่างการเขียนโปรแกรม ภาษา Python (Flask) สำหรับโมเดล Machine Learning

ตารางที่ 1 ผลของการผลการทดสอบประสิทธิภาพของโมเดล

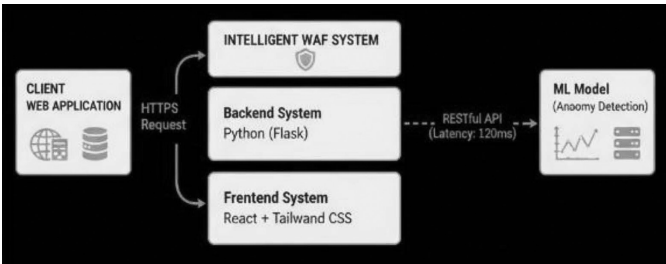
อัลกอริทึม	Accuracy	Precision	Recall	F1-Score	ROC-AUC
Random Forest	96.25%	95.80%	94.10%	94.94%	0.978
SVM	91.40%	89.50%	90.10%	89.80%	0.942
DNN	97.60%	96.90%	96.40%	96.65%	0.986

จากตารางที่ 1 ผลการทดลอง พบว่าโมเดล DNN (Deep Neural Network) มีประสิทธิภาพสูงสุด โดยให้ค่า Accuracy เฉลี่ย 97.60% และค่า F1-Score เท่ากับ 96.65% ซึ่งสูงกว่าโมเดลอื่นอย่างชัดเจน เพื่อยืนยันความแตกต่างของประสิทธิภาพระหว่างโมเดล ผู้วิจัยได้ทำการทดสอบทางสถิติด้วยวิธี One-Way ANOVA เพื่อเปรียบเทียบค่าเฉลี่ยของตัวชี้วัดประสิทธิภาพระหว่างโมเดล Random Forest, SVM และ DNN จากผลการทดสอบพบว่าค่าประสิทธิภาพของโมเดลมีความแตกต่างกันอย่างมีนัยสำคัญทางสถิติที่ระดับ $p < 0.05$ นอกจากนี้ได้ทำการวิเคราะห์เปรียบเทียบรายคู่เพิ่มเติมด้วย t-test เพื่อยืนยันว่าโมเดล DNN ให้ผลลัพธ์ที่ดีกว่าโมเดลอื่นอย่างมีนัยสำคัญ ดังนั้นในการพัฒนาระบบ Web Application Firewall (WAF) อัจฉริยะ งานวิจัยนี้จึงเลือกใช้โมเดล DNN เป็นโมเดลหลักสำหรับการตรวจจับการโจมตีทางไซเบอร์

2. ผลการบูรณาการโมเดลกับระบบ WAF (System Integration Results)

โมเดล Machine Learning ถูกบูรณาการเข้ากับระบบ WAF Dashboard ที่พัฒนาในรูปแบบเว็บแอปพลิเคชัน โดยมีคุณสมบัติดังนี้

1. ระบบ Backend พัฒนาโดยใช้ Python (Flask) ทำหน้าที่รับข้อมูล Request จากเว็บแอปพลิเคชัน เป้าหมาย และส่งผ่านโมเดล Machine Learning เพื่อวิเคราะห์
2. ระบบ Frontend พัฒนาโดยใช้ React + Tailwind CSS เพื่อแสดงสถานะของการตรวจจับแบบเรียลไทม์
3. การเชื่อมต่อข้อมูลระหว่างส่วน ML Model และ Dashboard ทำผ่าน RESTful API โดยมีเวลาเฉลี่ยในการตอบสนอง (Latency) เพียง 120 มิลลิวินาทีต่อ Request ดังรูปที่ 2



Key Features:

- Real-time Threat Detection
- Python (Flask Backend): Request Processing
- React + Tawlind CYSS Fortetand: Live Dashboard Dashboard
- RESTful API: Data Communication

Benefits

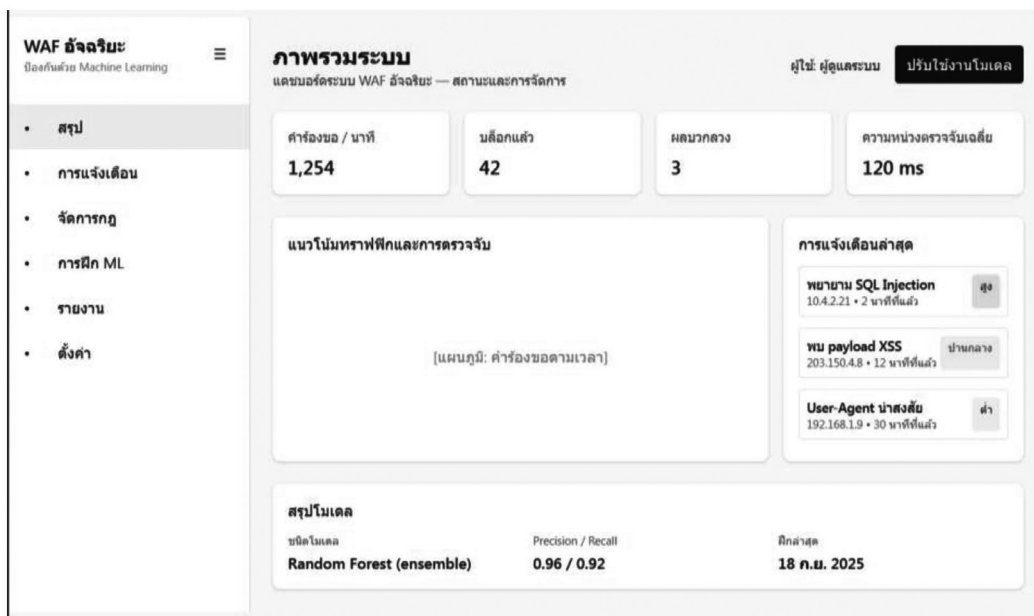
- Enhanced Security
- Protection Against Zero-Day Attacks
- Improved Accuracy (Reduced FPs/FNs)
- User-Friendly Monitoring Dashboard

รูปที่ 2 แสดงสถาปัตยกรรมระบบ Web Application Firewall (WAF) อัจฉริยะที่ใช้ Machine Learning ในการตรวจจับและป้องกันการโจมตี

ระบบ WAF ที่พัฒนาได้มีดังนี้

1. สามารถตรวจจับการโจมตีที่เกิดขึ้นแบบเรียลไทม์ได้ (Real-time Detection)
2. สามารถจำแนกประเภทของการโจมตี (Attack Classification) เช่น SQL Injection, XSS, RFI, CSRF, และปกติ (Normal Traffic) [9]

3. มีระบบแจ้งเตือน (Alert System) โดยแสดงผลในแดชบอร์ดในรูปแบบสีตามระดับความรุนแรง
4. ระบบสามารถเรียนรู้ข้อมูลใหม่ได้อัตโนมัติ (Incremental Learning) เพื่อปรับปรุงประสิทธิภาพการตรวจจับ ดังรูปที่ 3



รูปที่ 3 หน้าจอ Dashboard ระบบ Web Application Firewall (WAF) อัจฉริยะที่ใช้ Machine Learning ในการตรวจจับและป้องกันการโจมตี

3. ผลการประเมินประสิทธิภาพของระบบ

เมื่อทดสอบระบบด้วยชุดข้อมูลจำลองจากการโจมตีจริง (จำนวน 10,000 รายการ) ดังตารางที่ 2 จากผลการทดสอบพบว่าระบบสามารถตรวจจับ

การโจมตีได้อย่างแม่นยำ เฉลี่ย 96.10% โดยเฉพาะการโจมตี ประเภท RFI และ SQL Injection ซึ่งเป็นรูปแบบที่พบบ่อยประเภท RFI และ SQL Injection ซึ่งเป็นรูปแบบที่พบบ่อย

ตารางที่ 2 ผลการประเมินประสิทธิภาพของระบบ Web Application Firewall (WAF) อัจฉริยะที่ใช้ Machine Learning ในการตรวจจับและป้องกันการโจมตี

ประเภทการโจมตี	จำนวนที่ตรวจจับได้ถูกต้อง	จำนวนทั้งหมด	อัตราความถูกต้อง (%)
SQL Injection	1,950	2,000	97.50
XSS	1,890	2,000	94.50
RFI	1,970	2,000	98.50
CSRF	1,860	2,000	93.00
ปกติ (Normal Traffic)	1,980	2,000	99.00
รวมเฉลี่ย			96.10

4. ผลการประเมินการใช้งานของผู้ใช้

การประเมินความพึงพอใจของผู้ใช้ทำโดยใช้แบบสอบถาม 5 ระดับ (Likert Scale) กับกลุ่มตัวอย่าง 30 คน ประกอบด้วย

- นักศึกษา จำนวน 15 คน
- อาจารย์ผู้เชี่ยวชาญด้านระบบเครือข่าย จำนวน 10 คน

• เจ้าหน้าที่ IT และผู้ดูแลระบบ จำนวน 5 คน
ผลการวิเคราะห์ค่าความพึงพอใจสรุปได้ดังตารางที่ 3 โดยผลการประเมินแสดงให้เห็นว่าผู้ใช้มีความพึงพอใจต่อระบบในระดับ “มากที่สุด” โดยเฉพาะด้านความถูกต้องของการตรวจจับ (4.67) และการตอบสนองของระบบ (4.61) ซึ่งยืนยันถึงประสิทธิภาพของระบบ WAF อัจฉริยะที่พัฒนาได้ดี

ตารางที่ 3 ผลการประเมินผู้ใช้งาน

ด้านการประเมิน	ค่าเฉลี่ย ($\bar{X}$)	ส่วนเบี่ยงเบนมาตรฐาน (S.D.)	ระดับความพึงพอใจ
1. ความถูกต้องของการตรวจจับ	4.67	0.45	มากที่สุด
2. ความสะดวกในการใช้งานระบบ	4.55	0.52	มากที่สุด
3. การแสดงผลของแดชบอร์ด	4.42	0.49	มาก
4. ความเสถียรของระบบ	4.38	0.54	มาก
5. การตอบสนองของระบบ	4.61	0.47	มากที่สุด
ค่าเฉลี่ยรวม	4.53	0.49	มากที่สุด

5. สรุปผลการพัฒนาภาพรวม
จากผลการทดลองสามารถสรุปได้ว่า

1.โมเดล Machine Learning ที่เหมาะสมที่สุดคือ DNN (Deep Neural Network) ซึ่งให้

ค่าประสิทธิภาพสูงกว่าโมเดล Random Forest และ SVM ในทุกตัวชี้วัดระบบสามารถตรวจจับการโจมตีทางเว็บได้ด้วยความแม่นยำเฉลี่ย 96%

2. การตอบสนองของระบบเร็วกว่า WAF แบบ Rule-based เดิมประมาณ 25-30%

3. ผู้ใช้มีความพึงพอใจโดยรวมอยู่ในระดับมากที่สุด (ค่าเฉลี่ย = 4.53)

ทั้งนี้ โมเดล Deep Neural Network ให้ผลลัพธ์ที่ดีกว่า Random Forest และ SVM เนื่องจากโครงสร้างของโครงข่ายประสาทเทียมเชิงลึกสามารถเรียนรู้รูปแบบของข้อมูลที่มีความซับซ้อนและมีมิติสูงได้ดีกว่า โดยเฉพาะรูปแบบของพฤติกรรมการณ์โจมตีทางเว็บที่มักมีลักษณะหลากหลายและเปลี่ยนแปลงตลอดเวลา นอกจากนี้ DNN ยังสามารถเรียนรู้ความสัมพันธ์เชิงลึกของคุณลักษณะข้อมูล (Feature Representation) ผ่านหลายชั้นของโครงข่าย ทำให้สามารถจำแนกรูปแบบการโจมตีที่ซ่อนอยู่ในข้อมูลได้แม่นยำยิ่งขึ้น ในขณะที่ Random Forest และ SVM อาจมีข้อจำกัดในการเรียนรู้ความสัมพันธ์ของข้อมูลที่ซับซ้อนมาก

จากข้อสังเกตดังกล่าว แสดงให้เห็นว่าโมเดล DNN มีความเหมาะสมสำหรับการนำไปพัฒนาเป็นระบบ Web Application Firewall (WAF) อัจฉริยะ เพื่อเพิ่มประสิทธิภาพในการตรวจจับและป้องกันการโจมตีทางไซเบอร์ได้อย่างมีประสิทธิภาพ

4. สรุป

อภิปรายผลและสรุปผลการวิจัย

การอภิปรายผล (Discussion)

การวิจัยนี้มีวัตถุประสงค์การพัฒนาแบบ Web Application Firewall (WAF) อัจฉริยะที่ใช้ Machine Learning ในการตรวจจับและป้องกันการโจมตี ผ่านเว็บแอปพลิเคชัน โดยได้ผลดังนี้

1. ประสิทธิภาพของโมเดลการเรียนรู้ของเครื่อง

การทดสอบประสิทธิภาพของโมเดลการเรียนรู้ของเครื่อง 3 รูปแบบ พบว่า โมเดล Deep Neural Network (DNN) มีประสิทธิภาพสูงสุด โดยให้ค่าความถูกต้อง (Accuracy) เฉลี่ย 97.60% และค่า

F1-Score ที่ 96.65% ซึ่งสูงกว่าโมเดล Random Forest และ Support Vector Machine อย่างมีนัยสำคัญ ผลลัพธ์นี้สนับสนุนสมมติฐานที่ว่า Deep Learning มีความสามารถที่มากกว่าในการวิเคราะห์และจำแนกรูปแบบการจราจรที่ซับซ้อนและผิดปกติ (Anomaly Detection) ชุดข้อมูลที่ใช้ในการฝึกสอน ประกอบด้วยข้อมูลจากชุด CIC-IDS2017 Dataset และ OWASP Benchmark Dataset รวม 250,000 รายการ ซึ่งมีการใช้งานที่เสถียร

2. พัฒนาและการทำงานของระบบ

การพัฒนาโมเดล DNN เข้ากับระบบ WAF ถูกดำเนินการผ่านสถาปัตยกรรมแบบ RESTful API โดยมีองค์ประกอบ Backend พัฒนาด้วย Python (Flask) และ Frontend ด้วย React + Tailwind CSS ผลการทดสอบการตอบสนองของระบบพบว่ามีค่าความหน่วง (Latency) เฉลี่ยเพียง 120 มิลลิวินาทีต่อ Request ซึ่งแสดงให้เห็นว่าระบบ WAF อัจฉริยะนี้สามารถดำเนินการตรวจจับแบบเรียลไทม์ได้

การประเมินประสิทธิภาพของระบบโดยรวมพบว่า ระบบสามารถตรวจจับการโจมตีได้อย่างแม่นยำ เฉลี่ย 96.10% โดยเฉพาะอย่างยิ่งการโจมตีประเภท Remote File Inclusion (RFI) (98.50%) และ SQL Injection (97.50%) ซึ่งแสดงให้เห็นถึงศักยภาพในการป้องกันภัยคุกคามในระดับแอปพลิเคชันได้อย่างมีประสิทธิภาพ

3. การประเมินความพึงพอใจของผู้ใช้งาน

ผลการประเมินความพึงพอใจจากกลุ่มตัวอย่าง 30 คน แสดงให้เห็นว่าผู้ใช้มีความพึงพอใจต่อระบบในระดับ "มากที่สุด" โดยมีค่าเฉลี่ยรวม 4.53 จาก 5 โดยเฉพาะอย่างยิ่งด้าน ความถูกต้องของการตรวจจับ (ค่าเฉลี่ย 4.67) และ การตอบสนองของระบบ (ค่าเฉลี่ย 4.61) ผลลัพธ์นี้ยืนยันว่าการนำ Machine Learning มาใช้ไม่เพียงแต่เพิ่มความสามารถทางเทคนิคในการตรวจจับ แต่ยังส่งผลให้ระบบมีประสิทธิภาพการใช้งานในภาพรวมที่ดีตามการรับรู้ของผู้ดูแลระบบและผู้เชี่ยวชาญด้านความปลอดภัย

สรุปผล

ในการพัฒนาระบบ Web Application Firewall (WAF) อัจฉริยะที่ใช้ Machine Learning ในการตรวจจับและป้องกันการโจมตี ซึ่งผู้วิจัยประยุกต์ใช้ Deep Neural Network (DNN) เป็นกลไกหลักในการวิเคราะห์พฤติกรรมคำร้องขอ (Request) และตรวจจับการโจมตีทางเว็บ

1. โมเดลที่เหมาะสมที่สุด: เป็นโมเดล DNN มีประสิทธิภาพที่ดีกว่า ในการจำแนกประเภทการโจมตี โดยมีความแม่นยำเฉลี่ย 97.60%

2. ประสิทธิภาพของระบบ: ระบบที่พัฒนาขึ้นมีความสามารถในการตรวจจับการโจมตีทางเว็บด้วยความแม่นยำเฉลี่ยโดยรวม 96.10% และมีเวลาตอบสนองที่รวดเร็ว (Latency 120 มิลลิวินาที)

3. การยอมรับระบบของผู้ใช้งาน: ผู้ใช้งานมีความพึงพอใจโดยรวมต่อระบบในระดับ "มากที่สุด" (ค่าเฉลี่ย $\bar{X} = 4.53$) สรุปว่า การพัฒนาระบบ Web Application Firewall (WAF) อัจฉริยะที่ใช้ Machine Learning ในการตรวจจับและป้องกันการโจมตีที่พัฒนาขึ้นนี้มีศักยภาพในการเป็นเครื่องมือป้องกันความปลอดภัยทางไซเบอร์ที่มีประสิทธิภาพที่ดี สามารถตรวจจับการโจมตีที่รู้จักและไม่เคยรู้จักมาก่อน

ข้อเสนอแนะในการวิจัยครั้งต่อไป

1. การเสริมประสิทธิภาพของโมเดล: ควรมีการสำรวจการใช้เทคนิค Deep Learning ที่ซับซ้อนยิ่งขึ้น เช่น Recurrent Neural Network (RNN) หรือ Transformer เพื่อปรับปรุงความสามารถในการวิเคราะห์เชิงบริบทของ Payload ที่มีความยาวและซับซ้อน

2. การปรับปรุงชุดข้อมูล: ควรมีการรวบรวมชุดข้อมูลที่ครอบคลุมรูปแบบการโจมตีสมัยใหม่และรูปแบบ Zero-Day Attacks ที่หลากหลายยิ่งขึ้น รวมถึงการนำเทคนิค Active Learning มาใช้เพื่อปรับปรุงประสิทธิภาพการเรียนรู้ของโมเดลอย่างต่อเนื่อง

3. การประยุกต์ใช้งานจริง: ควรมีการทดสอบระบบ Web Application Firewall WAF ในสภาพแวดล้อมการทำงานจริง (Production Environment) เพื่อประเมินความเสถียร (Stability) และความสามารถในการปรับขนาด (Scalability) ภายใต้ปริมาณการจราจรที่สูง

5. เอกสารอ้างอิง

- [1] The Open Web Application Security Project, "OWASP Top 10 - 2021," 2021. [Online]. Available: <https://owasp.org/Top10/>. [Accessed: Mar. 11, 2026].
- [2] M. E. Durmuşkaya and S. Bayraklı, "Web application firewall based on machine learning models," PeerJ Computer Science, vol. 11, Art. no. e2975, 2025, doi: 10.7717/peerj-cs.2975.
- [3] M. Ito and H. Iyatomi, "Web application firewall using character-level convolutional neural network," in Proc. IEEE 14th Int. Colloquium on Signal Processing and Its Applications (CSPA), Kuala Lumpur, Malaysia, Mar. 9-10, 2018, pp. 103-106, doi: 10.1109/CSPA.2018.8368694.
- [4] B. Işık and I. Sogukpinar, "Machine learning based web application firewall," Expert Systems, vol. 42, Art. no. e13467, 2025, doi: 10.1111/exsy.13467.
- [5] G. Betarte, Á. Pardo, and R. Martínez, "Web application attacks detection using machine learning techniques," in Proc. IEEE 17th Int. Conf. Machine Learning and Applications (ICMLA), Orlando, FL, USA, Dec. 16-19, 2018, pp. 1065-1072, doi: 10.1109/ICMLA.2018.00167.

- [6] B. R. Dawadi, B. Adhikari, and D. K. Srivastava, "Deep learning technique-enabled web application firewall for the detection of web attacks," *Sensors*, vol. 23, no. 4, p. 2073, Feb. 2023, doi: 10.3390/s23042073.
- [7] A. Moradi Vartouni, M. Teshnehlab, and S. Sedighian Kashi, "Leveraging deep neural networks for anomaly-based web application firewall," *IET Information Security*, vol. 13, no. 4, pp. 352-361, Jul. 2019, doi: 10.1049/iet-ifs.2018.5212.
- [8] M. Krishnan, Y. Lim, S. Perumal, and G. Palanisamy, "Detection and defending the XSS attack using novel hybrid stacking ensemble learning-based DNN approach," *Digital Communications and Networks*, vol. 8, no. 1, pp. 49-60, Feb. 2022, doi: 10.1016/j.dcan.2021.06.004.
- [9] R. Vinayakumar et al., "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41525-41550, 2019, doi: 10.1109/ACCESS.2019.2895334.
- [10] D. S. Berman, A. L. Buczak, J. S. Chavis, and C. L. Corbett, "A survey of deep learning methods for cyber security," *Information*, vol. 10, no. 4, p. 122, Apr. 2019, doi: 10.3390/info10040122.
- [11] D. Chen and D. Zhang, "Deep learning for vulnerability and attack detection on web applications: A systematic literature review," *Sensors*, vol. 22, no. 4, Art. no. 118, Mar. 2022, doi: 10.3390/s22010118.
- [12] A. S. Alnami and T. B. Kim, "Adaptive web application firewall using risk-based adaptation," *Computers & Security*, vol. 101, Art. no. 102120, Feb. 2021, doi: 10.1016/j.cose.2020.102120.
- [13] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. 4th Int. Conf. Information Systems Security and Privacy (ICISSP)*, Funchal, Madeira, Portugal, Jan. 2018, pp. 108-116, doi: 10.5220/0006639801080116.