Setif 1 university Ferhat Abbass
Faculty of sciences
Computer sciences department



Master Thesis

Presented for the award of the MASTER degree in Computer Science by

BENKHEROUF Nassim and MOHAMEDI Imane

Delivered by

Setif 1 university - Ferhat Abbass

An AI-Driven Framework for Real-Time Anomaly Detection and Self-Mitigation in Distributed Networks

---

Research Unit:

Laboratory LRSD

Defended on **June 21, 2026** in front of the committee composed of:

President	**Nasri Khaled**	Associate Professor at UFAS1	Algeria
Reviewer	**Hamdi Skander**	Assistant Professor at UFAS1	Algeria
Supervisor	**Mohammed AISSAOUI**	Assistant Professor at UFAS1	Algeria

Abstract

Traditional Intrusion Detection Systems (IDS) face a critical limitation: the detection-response gap. This temporal delay between identifying an attack and executing a mitigation action creates a vulnerability window that modern automated threats can easily exploit. Centralized architectures, reliance on static rules, and human intervention further exacerbate this problem, particularly in large-scale or high-speed network environments.

To address this challenge, this thesis proposes an AI-driven framework for real-time anomaly detection and self-mitigation deployed at the network edge. The proposed system integrates three main components: a hybrid detection mechanism combining signature-based and anomaly-based characteristics using supervised machine learning, an inline edge-based architecture that processes traffic locally without relying on centralized infrastructure, and an automated response mechanism that enforces immediate countermeasures upon detection.

The system was implemented and evaluated within a virtualized network environment using PNETLab. Two ensemble learning algorithms, Random Forest and XGBoost, were compared using the CIC-IDS2017 dataset. Experimental results show that XGBoost achieves an overall accuracy of 99.65% and a macro F1-score of 88.53%, outperforming Random Forest in precision while maintaining competitive recall. The system demonstrates a detection-to-block latency of approximately 15.38 milliseconds, effectively closing the attack window.

These findings confirm that deploying an AI-based IDS with automated response at the network edge is a feasible and effective strategy for minimizing the detection-response gap, offering a practical solution for enhancing network security in real-time environments.

Keywords: Intrusion Detection System, Edge Computing, Machine Learning, Anomaly Detection, Automated Response, XGBoost, Detection-Response Gap, Network Security

Résumé

Les systèmes traditionnels de détection d'intrusion (Intrusion Detection Systems, IDS) présentent une limitation majeure : le délai entre la détection et la réponse (*detection-response gap*). Ce délai temporel, qui sépare l'identification d'une attaque de l'exécution d'une action de mitigation, crée une fenêtre de vulnérabilité que les menaces automatisées modernes peuvent facilement exploiter. Les architectures centralisées, la dépendance aux règles statiques ainsi que l'intervention humaine accentuent davantage ce problème, en particulier dans les environnements réseau à grande échelle ou à haut débit.

Afin de répondre à cette problématique, ce mémoire propose un cadre basé sur l'intelligence artificielle pour la détection d'anomalies en temps réel et l'auto-mitigation, déployé à la périphérie du réseau (*network edge*). Le système proposé intègre trois composants principaux : un mécanisme hybride de détection combinant les caractéristiques des approches basées sur les signatures et sur les anomalies à l'aide de l'apprentissage supervisé, une architecture *edge* en ligne (*inline*) qui traite le trafic localement sans dépendre d'une infrastructure centralisée, ainsi qu'un mécanisme de réponse automatisée capable d'appliquer immédiatement des contre-mesures dès qu'une intrusion est détectée.

Le système a été implémenté et évalué dans un environnement réseau virtualisé utilisant PNETLab. Deux algorithmes d'apprentissage automatique basés sur des méthodes d'ensemble, à savoir Random Forest et XGBoost, ont été comparés à l'aide du jeu de données CIC-IDS2017. Les résultats expérimentaux montrent que XGBoost atteint une précision globale (*Accuracy*) de 99,65% ainsi qu'un macro F1-score de 88,53%, surpassant Random Forest en termes de précision tout en conservant un taux de rappel (*Recall*) compétitif. Le système présente également une latence moyenne d'environ 15,38 millisecondes entre la détection d'une attaque et son blocage effectif, réduisant ainsi de manière significative la fenêtre d'attaque.

Ces résultats confirment que le déploiement d'un système de détection d'intrusion basé sur l'intelligence artificielle, doté d'un mécanisme de réponse automatisée et positionné à la périphérie du réseau, constitue une approche à la fois réalisable et efficace pour minimiser le délai entre la détection et la réponse. Cette approche offre ainsi une solution pratique permettant de renforcer la sécurité des réseaux dans des environnements exigeant une protection en temps réel.

Mots-clés : Système de détection d'intrusion, Informatique en périphérie (*Edge Computing*), Apprentissage automatique, Détection d'anomalies, Réponse automatisée, XGBoost, Délai entre détection et réponse, Sécurité des réseaux.

ملخص

تعاني أنظمة كشف التسلل التقليدية (Intrusion Detection Systems - IDS) من محدودية جوهرية تتمثل في الفجوة الزمنية بين اكتشاف الهجوم والاستجابة له. ويؤدي هذا التأخير، الذي يفصل بين التعرف على الهجوم وتنفيذ إجراءات تخفيف حدة الهجمات، إلى إنشاء نافذة زمنية يمكن للتهديدات السيبرانية الحديثة استغلالها بسهولة. كما أن الاعتماد على البنية المركزية، والقواعد الثابتة، والتدخل البشري يزيد من حدة هذه المشكلة، خاصةً في بيئات الشبكات واسعة النطاق أو عالية السرعة.

ولمعالجة هذه الإشكالية، يقترح هذا البحث إطارًا يعتمد على الذكاء الاصطناعي للكشف عن الشذوذ في الوقت الفعلي والاستجابة الذاتية للهجمات، مع نشر النظام عند حافة الشبكة (Edge). ويعتمد النظام المقترح على ثلاثة مكونات رئيسية، وهي: آلية كشف هجينة تجمع بين الكشف القائم على التواقيع والكشف القائم على الشذوذ باستخدام التعلم الآلي ، بنية طرفية تعمل في وضعية *Inline* لمعالجة حركة المرور محليًا دون الاعتماد على بنية مركزية، بالإضافة إلى آلية استجابة ، آلية تقوم بتطبيق إجراءات الحماية مباشرةً بمجرد اكتشاف الهجمة.

تم تنفيذ النظام وتقييمه داخل بيئة شبكية افتراضية باستخدام منصة PNETLab. كما تمت مقارنة خوارزميتي التعلم الآلي Random Forest وXGBoost باستخدام مجموعة البيانات CIC-IDS2017. وأظهرت النتائج التجريبية أن خوارزمية XGBoost حققت دقة تصنيف بلغت 99.65% , F1 Macro بلغت 88.53%، متفوقةً على Random Forest من حيث الدقة مع الحفاظ على معدل استدعاء مرتفع. كذلك أظهر النظام زمن استجابة يقارب 15.38 مللي ثانية بين اكتشاف الهجوم وتفعيل عملية الحظر، مما ساهم في تقليص نافذة الهجوم بشكل ملحوظ.

وتؤكد هذه النتائج أن نشر نظام كشف ومنع التسلل المعتمد على الذكاء الاصطناعي عند حافة الشبكة، والمزود بآلية استجابة آلية، يُعد حلاً عملياً وفعالًا لتقليل الفجوة الزمنية بين الكشف والاستجابة، مما يعزز أمن الشبكات ويوفر حماية فورية ضد التهديدات السيبرانية الحديثة.

الكلمات المفتاحية: نظام كشف التسلل، الحوسبة الطرفية، الذكاء الاصطناعي، التعلم الآلي، كشف الشذوذ، الاستجابة الآلية، XGBoost , الفجوة الزمنية بين الكشف والاستجابة، أمن الشبكات.

Contents

General introduction **1**

1 State of the art **4**

1.1 Introduction 4

1.2 Fundamentals of Network Security and Emerging Threats 4

1.3 Taxonomy of Modern Attacks 6

1.3.1 Volumetric (DDoS) 6

1.3.2 Infiltration (Lateral Movement) 10

1.3.3 Adversarial AI 12

1.4 Intrusion Detection Systems (IDS) 14

1.4.1 Definition of IDS 14

1.4.2 Importance of IDS 14

1.4.3 How IDS Works 14

1.4.4 IDS Placement in Network Architecture 15

1.4.5 Classification of IDS 16

1.4.6 Detection Methods Used in IDS 16

1.4.7 Benefits of IDS 17

1.4.8 Comparison between IDS and Firewall 17

1.4.9 Comparison between IDS and IPS 17

1.5 State of the Art of Attack Detection and Analysis of Existing Solutions and Their Limitations 19

1.5.1 Centralized Intrusion Detection Systems 19

1.5.2 Examples of Centralized Solutions 19

1.5.3 Discussion on Limitations and the Response Gap 21
1.6 The Problem of The Detection-Response Gap 21
1.6.1 Centralized Security Architecture 21
1.6.2 Detection–Response Gap 22
1.6.3 Network and Processing Latency 24
1.6.4 Attack Window 25
1.6.5 High-Speed and Automated Attacks 27
1.6.6 Limitations of Rule-Based Detection 29
1.6.7 Lateral Movement and Insider Threats 30
1.6.8 Scalability and Single Point of Failure 32
1.6.9 Human Intervention Delay 33
1.7 Conclusion 34

2 The proposed solution 35
2.1 Introduction 35
2.2 Towards a Low-Latency AI-Based Edge Intrusion Detection and Response System 35
2.2.1 Design Requirements of the Proposed System 37
2.2.2 Minimizing the Response Delay 38
2.2.3 Role of Artificial Intelligence in Intrusion Detection 42
2.2.4 Selection of the Machine Learning Algorithms 43
2.2.5 Threat Model and Security Assumptions 44
2.2.6 Complete System Workflow 45
2.3 Capabilities and Deployment Scenarios of the Proposed Framework 46
2.3.1 Capabilities of the Proposed Framework 46
2.3.2 Potential Deployment Scenarios 48
2.3.3 Operational Scope 50
2.4 Dataset and Data Preparation 50
2.4.1 Evaluation Environment 52
2.4.2 Summary of System Characteristics 54

2.5 Conclusion 55

3 Implementation and Experimental Evaluation 56

3.1 Introduction 56

3.1.1 Development Environment and Tools 56

3.1.2 PNETLab-Based Virtual Network Topology 58

3.1.3 Network Configuration and Traffic Flow 59

3.1.4 Attack Simulation Scenarios 62

3.2 Data Processing and Feature Engineering 63

3.2.1 Dataset Preparation 63

3.2.2 Feature Selection 63

3.2.3 Class Balancing Using SMOTE 64

3.2.4 Feature Scaling and Encoding 66

3.2.5 Dataset Partitioning and Scaling 66

3.3 Machine Learning-Based Detection Module 66

3.3.1 Model Training Configuration 69

3.3.2 Feature Importance Analysis 70

3.4 Real-Time Detection Mechanism 71

3.4.1 Detection Workflow 71

3.5 Automated Real-Time Response Mechanism 73

3.5.1 Response Workflow 73

3.6 Experimental Results and Evaluation 74

3.6.1 Evaluation Metrics 76

3.6.2 Model Comparison Results 78

3.6.3 Response Time Evaluation 82

3.6.4 Discussion of Results 84

3.7 Conclusion 85

Conclusion and perspectives 87

Bibliography 91

List of Figures

1 State of the art **4**

1.1 Illustration of a volumetric DDoS attack [1] 8

1.2 General Architecture of an Intrusion Detection System (IDS)[2] 15

1.3 Centralized Security Architecture [3]. 22

1.4 Detection–response timeline in intrusion detection systems [4]. 23

1.5 Detection–response workflow showing IDS stages and processing latency [4]. 23

1.6 Operational pipeline and sources of delay in centralized IDS architectures [4]. 25

1.7 Detection latency and the resulting attack window in centralized IDS [5, 4]. 26

1.8 Architecture comparison: Latency reduction in Edge-based IDS versus Centralized Cloud analysis [6]. 27

1.9 Typical botnet architecture used to coordinate automated cyber-attacks.[7]. 28

1.10 Sudden increase in network traffic during a high-speed DDoS attack. [8]. . 28

1.11 Rule-based IDS operation using predefined attack signatures [9, 10]. 29

1.12 Comparison of rule-based and AI/anomaly-based IDS showing adaptability and reduced false negatives [11, 12]. 31

1.13 Structural comparison: Centralized architecture (Single Point of Failure) vs. Distributed architecture (Scalability and Fault Tolerance) [13, 14]. . . . 33

2 The proposed solution **35**

2.1 Centralized IDS vs edge-based IDS 36

2.2 Proposed network architecture 37

2.3 traditional IDS VS Proposed solution 38
2.4 Hybrid intrusion detection approach 40
2.5 Human intervention vs Automated intervention 41
2.6 The role of AI in our IDS 42
2.7 Complete system workflow 47
2.8 A PC with a HIDS 48
2.9 A Network with a NIDS 49
2.10 Internal Network Topology 53

3 Implementation and Experimental Evaluation 56

3.1 PNETLab-Based Virtual Network Topology 58
3.2 Traffic Flow Through the Implemented IDS/IPS Node 60
3.3 An IDS enforcing IP forwarding to block malicious traffic and protect the target server using iptables 61
3.4 Workflow of dataset preparation and transformation prior to model training 67
3.5 Machine Learning Pipeline 68
3.6 XGBoost Feature Importance Graph 70
3.7 Detection Workflow Visualization 72
3.8 Response Workflow Visualization 75
3.9 Example of Confusion Matrix(XGBoost results) 79
3.10 XGBoost results 82

List of Tables

1 State of the art **4**

1.1 Comparison between IDS and Firewall 18

1.2 Comparison between IDS and IPS 18

2 The proposed solution **35**

2.1 Comparison between centralized and edge-based IDS deployment 36

2.2 Representative Feature Set from CIC-IDS2017 Dataset 51

2.3 Attack Categories in CIC-IDS2017 Dataset 51

2.4 Summary of Proposed System Characteristics 55

3 Implementation and Experimental Evaluation **56**

3.1 Experimental Host Machine Specifications 57

3.2 Experimental Environment Components 59

3.3 Network Configuration Summary 59

3.4 IDS Network Configuration 62

3.5 Attack Generation Tools 63

3.6 Final set of features selected from the CIC-IDS2017 dataset for machine learning-based intrusion detection. 64

3.7 Performance Comparison: Imbalanced vs. SMOTE-Balanced Model 64

3.8 Per-Class Performance: Before SMOTE vs After SMOTE 65

3.9 XGBoost Training Configuration 70

3.10 Real-Time Detection Components 73

3.11 Evaluation Metrics Used in the Experiments 78
3.12 Performance Comparison Between Random Forest and XGBoost 79
3.13 Per-class classification results for Random Forest (balanced model) 81
3.14 Per-class classification results for XGBoost (SMOTE balanced model) . . . 81
3.15 Centralized IDS vs Proposed Edge-Based IDS 83