UNIVERSITE GASTON BERGER
L'excellence au service du développement

Ecole **D**octorale des **S**ciences et des **T**echnologies
UFR de **S**ciences **A**ppliquées et de **T**echnologie

THÈSE

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ GASTON BERGER DE SAINT LOUIS

Mention : Informatique

Spécialité : Cybersécurité & Intelligence Artificielle

Présentée et soutenue publiquement par

Mouhamadou Lamine DIAKHAME

le 22 avril 2026

Système Intelligent de Détection Précoce d'Attaques en Cybersécurité

Thèse dirigée par Pr. Chérif DIALLO

JURY

Moussa	LO	Professeur Titulaire, Université Gaston Berger de Saint-Louis	Président
Jacques	KLEIN	Professeur Titulaire, Université du Luxembourg, Luxembourg	Rapporteur
Cheikh	SARR	Professeur Titulaire, Université Iba Der Thiam de Thiès	Rapporteur
Abdoul A.	CISS	Maître de Conférences (CAMES), École Polytechnique de Thiès	Rapporteur
Mohamed	MEJRI	Professeur Titulaire, Université Laval, Québec, Canada	Examinateur
Cheikh M. B.	DIONE	Assistant, Université Gaston Berger de Saint-Louis	Examinateur
Chérif	DIALLO	Professeur Titulaire, Université Gaston Berger de Saint-Louis	Directeur de thèse
Fehmi	JAAFAR	Professeur Agrégé, Université du Québec à Chicoutimi, Canada	Invité

Thèse Numéro EDST016-2026

Préparée au sein du Laboratoire d'Algèbre, de Cryptographie, de Codes et Applications (LACCA)

Remerciements

Je tiens tout d'abord à exprimer ma profonde gratitude à mon directeur de thèse, le Professeur **Chérif Diallo**, pour sa confiance, sa disponibilité, ses conseils avisés et son accompagnement constant tout au long de ce travail de recherche. Son encadrement scientifique, sa rigueur méthodologique et sa bienveillance ont été déterminants dans l'aboutissement de cette thèse.

Je remercie également le Professeur **Mohamed Mejri** pour ses orientations scientifiques, ses encouragements et l'intérêt qu'il a porté à mes travaux. Ses remarques pertinentes ont grandement contribué à la qualité de ce travail.

Mes sincères remerciements s'adressent à chacun des membres du jury qui ont accepté d'évaluer ce travail et de participer à sa soutenance.

Je remercie avec respect le Professeur **Moussa Lo**, Professeur Titulaire à l'Université Gaston Berger de Saint-Louis, pour avoir accepté de présider ce jury. Son autorité scientifique et son engagement pour la recherche en informatique au Sénégal sont une source d'inspiration.

J'exprime ma profonde reconnaissance au Professeur **Jacques Klein**, Professeur Titulaire à l'Université du Luxembourg, pour avoir accepté d'être rapporteur de cette thèse. La qualité de son rapport a grandement contribué à l'amélioration de ce travail.

Je remercie sincèrement le Professeur **Cheikh Sarr**, Professeur Titulaire à l'Université Iba Der Thiam de Thiès, d'avoir accepté la charge de rapporteur. Ses remarques précises et constructives ont permis d'enrichir et d'affiner la qualité scientifique de ce manuscrit.

J'adresse mes vifs remerciements au Professeur **Abdoul Aziz Ciss**, Maître de Conférences à l'École Polytechnique de Thiès, pour avoir accepté d'être rapporteur de cette thèse. Son expertise en cybersécurité et la pertinence de ses observations ont été particulièrement précieuses pour la consolidation de mes travaux.

Je remercie le Docteur **Cheikh Mouhamadou Bamba Dione**, Assistant à l'Université Gaston Berger de Saint-Louis, pour avoir accepté de faire partie de ce jury en qualité d'examinateur. Son expertise en intelligence artificielle a apporté un éclairage précieux et complémentaire à l'évaluation de cette thèse.

Je remercie également le Professeur **Fehmi Jaafar**, Professeur Agrégé à l'Université du Québec à Chicoutimi, pour avoir accepté de participer à ce jury en qualité d'invité.

Je tiens aussi à remercier l'ensemble des enseignants-chercheurs, du personnel admi-

nistratif et technique de mon établissement pour leur soutien, leur disponibilité et les conditions favorables dans lesquelles cette thèse a pu être réalisée.

Je n'oublie pas mes collègues, amis et collaborateurs pour les échanges scientifiques, les discussions enrichissantes et le soutien moral tout au long de ce parcours doctoral.

Enfin, je remercie du fond du cœur ma famille pour son amour, son soutien indéfectible, sa patience et ses encouragements permanents. Cette thèse est aussi le fruit de leurs sacrifices.

À toutes et à tous, je dis sincèrement merci.

Résumé

Avec la transformation numérique accélérée des services, la généralisation du cloud computing, de l'Internet des objets et du télétravail, les systèmes d'information sont devenus des cibles privilégiées des cyberattaques. Ces dernières sont aujourd'hui plus fréquentes, plus complexes et souvent structurées en plusieurs étapes, rendant leur détection précoce particulièrement difficile. Les systèmes classiques de détection d'intrusion reposent majoritairement sur des mécanismes réactifs, ce qui limite leur efficacité face aux menaces modernes.

Cette thèse s'intéresse à la conception d'un système intelligent d'alerte précoce pour la cybersécurité, basé sur des techniques avancées d'intelligence artificielle. L'objectif principal est de corréler automatiquement des alertes de sécurité hétérogènes, de reconstruire des scénarios d'attaque cohérents et d'anticiper les étapes futures d'attaques en cours, afin d'améliorer les capacités de détection et de réaction des équipes de sécurité.

Cinq contributions scientifiques majeures sont proposées. La première porte sur un nouveau cadre de corrélation d'alertes basé sur le traitement automatique du langage naturel, la reconnaissance d'entités nommées et la similarité sémantique pour la reconstruction de scénarios d'attaque. La deuxième contribution introduit un modèle de langage de grande taille spécialisé pour la détection de menaces en temps réel par corrélation d'événements de sécurité. La troisième propose une approche adaptative de projection d'attaques combinant des réseaux de neurones sur graphes et l'apprentissage par renforcement. La quatrième développe une approche d'apprentissage automatique multi-classe pour améliorer la classification et la priorisation des alertes de sécurité. Enfin, une cinquième contribution propose un pipeline opérationnel intégrant SIEM, SOAR et un petit modèle de langage (TinyLlama-1.1B) pour la corrélation légère d'alertes et la réponse automatisée aux incidents sur infrastructure légère, sans GPU.

Les résultats expérimentaux montrent que les approches proposées permettent d'améliorer significativement la précision de la corrélation d'alertes, la capacité de détection précoce et les performances de prédiction des attaques. Ces travaux ouvrent ainsi des perspectives prometteuses pour le développement de systèmes d'alerte précoce intelligents et autonomes en cybersécurité.

Mots-clés : Cybersécurité, Système d'Alerte Précoce, Corrélation d'Alertes, Apprentis-

sage Automatique, Deep Learning, Modèles de Langage, Prédiction d'Attaques, SIEM, SOAR, Réponse Automatisée aux Incidents.

Abstract

With the rapid digitalization of services, the expansion of cloud computing, the Internet of Things, and remote working environments, modern information systems have become increasingly exposed to cyber threats. Cyberattacks are now more frequent, more sophisticated, and often multi-stage, making their early detection a major challenge for security operation centers. Traditional intrusion detection systems mainly rely on reactive mechanisms, which often detect attacks at an advanced stage, when significant damage has already occurred.

This PhD thesis addresses the problem of early cyber-attack detection through the design of an intelligent Early Warning System based on advanced artificial intelligence techniques. The main objective is to automatically correlate heterogeneous security alerts, reconstruct coherent attack scenarios, and anticipate the possible future steps of ongoing attacks in order to improve decision-making and response capabilities.

Five main scientific contributions are proposed in this thesis. First, a novel alert correlation framework based on Natural Language Processing, Named Entity Recognition, and semantic similarity is introduced for automated cyber-attack scenario reconstruction. Second, a fine-tuned Large Language Model is proposed for real-time threat detection through security event correlation. Third, an adaptive cyber-attack projection approach combining Graph Attention Networks and Reinforcement Learning is developed to anticipate future attack paths. Fourth, a machine learning-based multi-class alert correlation framework is proposed to enhance the classification and prioritization of security alerts. Fifth, a lightweight operational pipeline integrating SIEM, a small CPU-only Large Language Model (TinyLlama-1.1B), and SOAR automation is proposed for automated incident response on commodity infrastructure, achieving an end-to-end response time of 11.7 seconds with no GPU requirement.

Experimental evaluations conducted on different cybersecurity datasets demonstrate the effectiveness of the proposed approaches in improving alert correlation accuracy, early detection capability, and attack prediction performance. The results highlight the relevance of combining semantic analysis, deep learning, and adaptive learning mechanisms for the development of next-generation intelligent cybersecurity early warning systems.

Keywords : Cybersecurity, Early Warning System, Alert Correlation, Machine Learning,

Deep Learning, Large Language Models, Attack Prediction, SIEM, SOAR, Automated Incident Response.

Table des matières

Remerciements i

Résumé iii

Abstract v

Liste des Acronymes xxiii

Introduction générale 1

0.1 Contexte . . . 1
0.2 Problématique . . . 2
0.3 Objectifs de la Thèse . . . 3
0.3.1 Objectif général . . . 3
0.3.2 Objectifs spécifiques . . . 3
0.4 Contributions de la Thèse . . . 3
0.5 Liste des publications . . . 4
0.6 Organisation de la Thèse . . . 5

1 Cybersécurité, Corrélation d'Alertes et Systèmes d'Alerte Précoce 7

1.1 Introduction . . . 7
1.2 Cybersécurité et menaces modernes . . . 7
1.2.1 Évolution du paysage des menaces . . . 8
1.2.2 Typologie des cyberattaques modernes . . . 9
1.3 Mécanismes classiques de défense . . . 14
1.3.1 Dispositifs périmétriques et filtrage . . . 14
1.3.2 Systèmes de détection et de prévention d'intrusion . . . 14
1.3.3 Plates-formes de gestion des événements de sécurité (SIEM) . . . 15
1.4 Corrélation d'alertes de sécurité . . . 15
1.4.1 Objectifs et enjeux . . . 16
1.4.2 Approches basées sur des règles . . . 17
1.4.3 Approches basées sur des graphes . . . 17
1.4.4 Approches par apprentissage automatique . . . 17

1.4.5 Corrélation cross-domaines pour les systèmes industriels 17
1.5 Cadres de modélisation des attaques 19
1.5.1 Cyber Kill Chain 19
1.5.2 MITRE ATT&CK Framework 20
1.6 Systèmes d'alerte précoce en cybersécurité 20
1.6.1 Définition et principes 21
1.6.2 Architecture des EWS modernes 22
1.6.3 Évolution vers une sécurité proactive 22
1.6.4 Technologies de déception 22
1.6.5 Applications nationales : programme EINSTEIN 23
1.7 Prédiction et projection d'attaques 24
1.7.1 Prédiction basée sur les données 24
1.7.2 Projection d'attaques multi-étapes 24
1.7.3 Intelligence artificielle pour la prédiction 25
1.8 Conclusion 26
1.8.1 État des lieux 26
1.8.2 Positionnement des contributions 26
1.8.3 Alignement avec les tendances actuelles 26
1.8.4 Analyse comparative des solutions récentes 27

2 Intelligence artificielle, NLP et modèles de langage pour la cybersécurité 29
2.1 Introduction 29
2.2 Apprentissage automatique en cybersécurité 29
2.2.1 Principes généraux 30
2.2.2 Algorithmes classiques 30
2.2.3 Défis spécifiques en cybersécurité 32
2.3 Apprentissage profond (Deep Learning) 32
2.3.1 Fondements 32
2.3.2 Réseaux convolutifs et récurrents 33
2.4 Traitement du langage naturel pour la cybersécurité 34
2.4.1 Principes du NLP 34
2.4.2 Reconnaissance d'entités nommées 35
2.4.3 Similarité sémantique 35
2.5 Modèles de langage de grande taille 35
2.5.1 Principe des modèles de langage de grande taille (LLM) 36
2.5.2 Adaptation des LLM à la cybersécurité 37
2.5.3 Applications pratiques des LLM en cybersécurité 38
2.6 Réseaux de neurones sur graphes 38

2.6.1 Principe des Graph Neural Networks . . . 39
2.6.2 Applications des GNN en opérations cyber-défensives . . . 39
2.6.3 Graph Attention Networks . . . 40
2.6.4 Guide pratique pour l'utilisation des GNN . . . 41
2.7 Apprentissage par renforcement pour la cybersécurité . . . 41
2.7.1 Principes de l'apprentissage par renforcement . . . 41
2.7.2 Applications à la défense et à la projection d'attaques . . . 43
2.7.3 Deep Q-Networks et variantes . . . 44
2.8 Conclusion . . . 44
2.8.1 Modèles spécifiques mobilisés dans les contributions . . . 45
2.8.2 Analyse critique et limites opérationnelles des techniques mobilisées 47

3 Méthodologie générale et architecture du système d'alerte précoce 49
3.1 Introduction . . . 49
3.2 Formulation du problème scientifique général . . . 49
3.2.1 Problématique centrale . . . 49
3.2.2 Hypothèses de recherche . . . 50
3.3 Architecture globale et positionnement des contributions . . . 51
3.3.1 Vue d'ensemble de l'architecture intégrée . . . 51
3.3.2 Positionnement par rapport aux lacunes identifiées . . . 53
3.4 Contribution 1 : MCM-CASR - Reconstruction de scénarios par corrélation sémantique . . . 53
3.4.1 Objectif et problématique spécifique . . . 54
3.4.2 Architecture et méthodologie MCM-CASR . . . 54
3.4.3 Datasets et protocole expérimental . . . 56
3.4.4 Résultats principaux . . . 56
3.5 Contribution 2 : MCM-Llama - Corrélation par LLM fine-tuné . . . 57
3.5.1 Motivation et amélioration par rapport à MCM-CASR . . . 57
3.5.2 Méthodologie de fine-tuning . . . 58
3.5.3 Architecture de déploiement . . . 59
3.5.4 Résultats et amélioration . . . 59
3.6 Contribution 3 : GAT-DQN - Projection adaptative par sélection contextuelle 59
3.6.1 Objectif et problématique spécifique . . . 59
3.6.2 Innovation principale : Méta-modèle de sélection contextuelle . . . 60
3.6.3 Architecture globale du cadre . . . 60
3.6.4 Dataset et protocole expérimental . . . 64
3.6.5 Résultats expérimentaux . . . 64
3.6.6 Avantages de l'approche GAT-DQN . . . 65
3.6.7 Limitations et perspectives . . . 66

3.6.8 Positionnement dans l'architecture globale . . . 67
3.7 Contribution 4 : Multi-Classification - Classification multi-granularité . . . 67
3.7.1 Les trois approches de classification . . . 68
3.7.2 Ingénierie des caractéristiques . . . 68
3.7.3 Algorithmes évalués . . . 69
3.7.4 Dataset et protocole expérimental . . . 69
3.7.5 Résultats et analyse comparative . . . 69
3.8 Contribution 5 : TinyLlama-SOAR - Pipeline opérationnel de réponse automatisée . . . 70
3.8.1 Objectif et problématique spécifique . . . 70
3.8.2 Architecture en trois couches . . . 70
3.8.3 Protocole expérimental . . . 71
3.8.4 Résultats . . . 72
3.8.5 Positionnement dans l'architecture globale . . . 72
3.9 Approche expérimentale transversale . . . 72
3.9.1 Datasets utilisés . . . 72
3.9.2 Métriques communes . . . 74
3.9.3 Comparaisons avec l'état de l'art . . . 74
3.10 Intégration et synergies entre contributions . . . 74
3.10.1 Complémentarité des approches . . . 74
3.10.2 Pipeline opérationnel intégré . . . 76
3.11 Considérations de déploiement . . . 76
3.11.1 Infrastructure requise . . . 77
3.11.2 Contraintes opérationnelles . . . 77
3.11.3 Interopérabilité des modules et flux de données . . . 77
3.12 Conclusion . . . 78

4 MCM-CASR : Corrélation d'Alertes et Reconstruction de Scénarios 80
4.1 Introduction . . . 80
4.1.1 Contributions . . . 81
4.1.2 Organisation du chapitre . . . 81
4.2 Travaux connexes . . . 81
4.2.1 Évolution historique des approches . . . 81
4.2.2 Approches basées sur l'ontologie et la sémantique . . . 82
4.2.3 Approches basées sur les graphes d'attaque . . . 83
4.2.4 Approches par Apprentissage Profond . . . 83
4.2.5 Approches bayésiennes . . . 84
4.2.6 Synthèse et positionnement . . . 84
4.3 Jeu de données . . . 85

4.3.1 Caractéristiques du jeu de données . . . 85
4.3.2 Catégories d'attaques . . . 86
4.3.3 Justification du choix . . . 86
4.3.4 Limitations . . . 87
4.4 Fondements théoriques du TALN pour la cybersécurité . . . 87
4.4.1 Introduction au traitement automatique du langage naturel . . . 87
4.4.2 Reconnaissance d'entités nommées (REN) . . . 88
4.4.3 Similarité sémantique . . . 88
4.5 Cadre MCM-CASR . . . 89
4.5.1 Architecture générale . . . 89
4.5.2 Module de réduction des alertes . . . 90
4.5.3 Module de corrélation des alertes . . . 91
4.6 Analyse comparative . . . 93
4.6.1 Comparaison méthodologique . . . 93
4.6.2 Taxonomie des paradigmes . . . 94
4.6.3 Analyse des forces et limites . . . 95
4.6.4 Comparaison quantitative . . . 95
4.6.5 Comparaison fonctionnelle . . . 96
4.6.6 Analyse des techniques utilisées . . . 96
4.6.7 Profil comparatif multi-dimensionnel . . . 97
4.6.8 Synthèse visuelle . . . 98
4.6.9 Discussion et perspectives . . . 98
4.7 Conclusion . . . 100
4.7.1 Synthèse des contributions . . . 100
4.7.2 Positionnement dans l'état de l'art . . . 100
4.7.3 Limites et perspectives . . . 101

5 MCM-Llama : Détection de Menaces en Temps Réel 102
5.1 Introduction . . . 102
5.1.1 Contexte et motivation . . . 102
5.1.2 Problématique et questions de recherche . . . 103
5.1.3 Contributions . . . 103
5.1.4 Organisation du chapitre . . . 103
5.2 Travaux connexes . . . 104
5.2.1 Techniques d'IA pour la corrélation d'événements de sécurité . . . 104
5.2.2 Approches basées sur la similarité . . . 104
5.2.3 Détection des menaces persistantes avancées . . . 104
5.2.4 Positionnement de notre approche . . . 105
5.3 Fondements . . . 105

5.3.1 Grands Modèles de Langage (LLM) . . . 105
5.3.2 La Famille de Modèles LLaMA . . . 107
5.3.3 Mécanisme LoRA pour l'adaptation efficiente . . . 108
5.4 Méthodologie MCM-Llama . . . 108
5.4.1 Vue d'ensemble . . . 108
5.4.2 Jeu de données CTU-13 . . . 109
5.4.3 Processus de fine-tuning . . . 110
5.4.4 Architecture de déploiement . . . 113
5.5 Expérimentations et résultats . . . 114
5.5.1 Protocole expérimental . . . 114
5.5.2 Résultats du fine-tuning . . . 114
5.5.3 Matrice de confusion . . . 115
5.5.4 Métriques de performance . . . 116
5.5.5 Analyse des résultats . . . 116
5.6 Analyse comparative approfondie . . . 117
5.6.1 Méthodes de référence . . . 117
5.6.2 Limitations de la comparaison inter-études . . . 117
5.6.3 Comparaison quantitative . . . 118
5.6.4 Comparaison qualitative . . . 118
5.6.5 Analyse des critères d'évaluation . . . 119
5.6.6 Discussion et positionnement . . . 119
5.7 Conclusion . . . 120
5.7.1 Synthèse des contributions . . . 120
5.7.2 Implications pratiques . . . 120
5.7.3 Limitations . . . 120
5.7.4 Perspectives de recherche . . . 121

6 Projection Adaptative de Cyberattaques par GAT et RL 122
6.1 Introduction . . . 122
6.1.1 Définition et importance de la projection d'attaque . . . 122
6.1.2 État actuel de la cybersécurité proactive . . . 123
6.1.3 Problématique et limitations des approches existantes . . . 123
6.1.4 Questions de recherche . . . 123
6.1.5 Contributions . . . 124
6.1.6 Organisation du chapitre . . . 124
6.2 Travaux connexes . . . 124
6.2.1 Approches par apprentissage par renforcement profond . . . 125
6.2.2 Approches par modèles de Markov cachés . . . 125
6.2.3 Approches de sélection d'algorithmes . . . 126

6.2.4 Sélection dynamique de modèles . . . 126
6.2.5 Réseaux d'attention sur graphes pour la détection de malware . . . 126
6.2.6 Apprentissage par renforcement profond pour la détection d'intrusion 127
6.2.7 Synthèse et positionnement . . . 127
6.3 Fondements théoriques . . . 127
6.3.1 Réseaux de neurones sur graphes (GNN) . . . 127
6.3.2 Réseaux d'attention sur graphes (GAT) . . . 128
6.3.3 Apprentissage par renforcement profond (DRL) . . . 129
6.3.4 Jeu de données CTU-13 . . . 130
6.4 Méthodologie proposée . . . 130
6.4.1 Architecture globale du système . . . 130
6.4.2 Modèle GAT avec encodeur Transformer . . . 132
6.4.3 Modèle DQN avec architecture Transformer . . . 132
6.4.4 Approche de méta-apprentissage pour la sélection dynamique . . . 133
6.5 Expérimentations et résultats . . . 134
6.5.1 Configuration expérimentale . . . 134
6.5.2 Résultats du modèle GAT . . . 134
6.5.3 Résultats du modèle DQN . . . 135
6.5.4 Validation complémentaire sur alertes synthétiques . . . 138
6.6 Analyse comparative avec l'état de l'art . . . 139
6.6.1 Méthodes de référence . . . 139
6.6.2 Limitations de la comparaison inter-études . . . 139
6.6.3 Comparaison qualitative . . . 139
6.6.4 Comparaison quantitative . . . 140
6.6.5 Matrice de support des fonctionnalités . . . 141
6.6.6 Discussion du positionnement . . . 141
6.7 Discussion . . . 142
6.7.1 Forces . . . 142
6.7.2 Limitations . . . 142
6.7.3 Discussion approfondie : déséquilibre de classes et compromis de conception . . . 143
6.7.4 Implications pratiques . . . 143
6.7.5 Directions futures . . . 144
6.8 Conclusion . . . 144
6.8.1 Synthèse des contributions . . . 144
6.8.2 Perspectives . . . 144

7 Corrélation d'Alertes par Approche Multi-Classification 146
7.1 Introduction . . . 146

7.1.1 Contexte et motivation . . . 146
7.1.2 Problématique . . . 147
7.1.3 Questions de recherche . . . 147
7.1.4 Contributions . . . 147
7.1.5 Organisation du chapitre . . . 148
7.2 Travaux connexes . . . 148
7.2.1 Évolution des systèmes de corrélation d'alertes . . . 148
7.2.2 Cadres complets de corrélation d'alertes . . . 149
7.2.3 Applications de l'apprentissage automatique en cybersécurité . . . 149
7.2.4 Jeux de données et méthodologies d'évaluation . . . 149
7.2.5 Détection d'attaques coordonnées . . . 150
7.2.6 Positionnement de notre contribution . . . 150
7.3 Méthodologie . . . 150
7.3.1 Génération et prétraitement des données d'alertes . . . 150
7.3.2 Vue d'ensemble du cadre multi-classification . . . 152
7.3.3 Tâche de classification 1 : détection binaire de corrélation . . . 152
7.3.4 Tâche de classification 2 : classification des types de corrélation . . . 154
7.3.5 Tâche de classification 3 : classification des incidents . . . 154
7.3.6 Algorithmes d'apprentissage automatique . . . 155
7.3.7 Méthodologie d'évaluation . . . 163
7.4 Résultats expérimentaux . . . 163
7.4.1 Caractéristiques du jeu de données . . . 164
7.4.2 Résultats de la détection binaire de corrélation . . . 164
7.4.3 Résultats de la classification des types de corrélation . . . 164
7.4.4 Résultats de la classification des incidents . . . 164
7.4.5 Comparaison globale des performances . . . 165
7.4.6 Meilleurs modèles par tâche . . . 166
7.4.7 Visualisation des résultats . . . 166
7.5 Discussion . . . 168
7.5.1 Analyse des performances des modèles . . . 168
7.5.2 Efficacité des tâches de classification . . . 169
7.5.3 Considérations d'implémentation pratique . . . 169
7.5.4 Limitations . . . 169
7.5.5 Discussion sur la validité des performances élevées . . . 170
7.5.6 Perspectives d'interprétabilité . . . 170
7.6 Conclusion . . . 171
7.6.1 Synthèse des contributions . . . 171
7.6.2 Résultats clés . . . 171
7.6.3 Implications pratiques . . . 171

7.6.4 Perspectives de recherche 171

8 TinyLlama-SOAR : Corrélation d'Alertes par LLM et Réponse Automatisée 173

8.1 Introduction 173
8.1.1 Contributions du chapitre 174
8.2 Travaux connexes 174
8.2.1 Corrélation d'alertes et détection par apprentissage automatique . . 174
8.2.2 LLM et approches SOAR 175
8.2.3 Positionnement 175
8.3 Architecture proposée 175
8.3.1 Vue d'ensemble 175
8.3.2 Couche SIEM : Collecte et normalisation 176
8.3.3 Couche IA : Corrélation sémantique 176
8.3.4 Couche SOAR : Orchestration et réponse 178
8.3.5 Flux d'intégration bout en bout 178
8.4 Implémentation 178
8.4.1 Environnement de déploiement 178
8.4.2 Service de corrélation LLM 180
8.4.3 Intégration SOAR 180
8.5 Évaluation expérimentale 180
8.5.1 Protocole expérimental 180
8.5.2 Performances de corrélation 180
8.5.3 Analyse du temps de réponse 181
8.5.4 Utilisation des ressources 181
8.5.5 Analyse comparative 181
8.6 Discussion 182
8.6.1 Interprétation des résultats 182
8.6.2 Menaces à la validité 183
8.6.3 Limitations connues 183
8.7 Conclusion 183

Conclusion générale 185

9.8 Discussion générale et analyse transversale 185
9.8.1 Analyse comparative des mécanismes de corrélation 185
9.8.2 Vers une défense proactive : L'apport de la projection adaptative . . 186
9.8.3 Performance et robustesse à grande échelle 186
9.8.4 Intégration et synergies au sein du système d'alerte précoce 186
9.8.5 De la corrélation à la réponse automatisée : apport de TinyLlama-SOAR 187

9.8.6 Synthèse des limitations et défis restants 187
9.9 Synthèse des travaux 187
9.10 Récapitulation des contributions 188
9.11 Perspectives de recherche 189

Table des figures

1.1 Vue d'ensemble du paysage des cybermenaces 2024-2025 avec statistiques clés des rapports Microsoft, Google et WEF. 8
1.2 Taxonomie complète des cyberattaques modernes selon les propriétés de sécurité et catégories de menaces. 10
1.3 Évolution des techniques de ransomware de la simple extorsion au RaaS (Ransomware-as-a-Service) alimenté par l'IA. 11
1.4 Cartographie des principaux groupes APT actifs en 2024-2025 avec leurs attributions, cibles et techniques caractéristiques. 12
1.5 Évolution du nombre de vulnérabilités zero-day exploitées dans la nature de 2021 à 2024, avec analyse détaillée des catégories ciblées. 13
1.6 Architecture en profondeur (Defense in Depth) des mécanismes de défense classiques avec couches périmètre, réseau, endpoints et applications. 16
1.7 Trois approches principales de corrélation d'alertes : basées sur règles, basées sur graphes et machine learning, avec leurs avantages et limitations respectives. 18
1.8 Architecture de corrélation d'alertes cross-domaines pour les systèmes de contrôle industriels avec enrichissement bidirectionnel et corrélateur adaptatif. 19
1.9 Comparaison détaillée entre Cyber Kill Chain et MITRE ATT&CK Framework avec tableau comparatif des critères clés. 21
1.10 Architecture complète d'un système d'alerte précoce moderne intégrant threat intelligence, SIEM/SOAR, network analysis, deception technology et moteur d'analyse IA/ML. Source : adapté de [1]. 23
1.11 Méthodes de prédiction et projection d'attaques : approches traditionnelles (Bayésiennes, Markov) et techniques modernes d'IA/ML (RNN/LSTM, GNN, Reinforcement Learning) avec workflow de projection. 25
1.12 Positionnement des quatre contributions de la thèse (MCM-CASR, MCM-Llama, GAT-DQN, Multi-Classification) par rapport aux lacunes identifiées et aux tendances actuelles en cybersécurité. 27

2.1 Vue d'ensemble de l'intelligence artificielle appliquée à la cybersécurité : de l'apprentissage automatique classique aux modèles de langage de grande taille . . . 30
2.2 Catégories d'apprentissage automatique et leurs applications en cybersécurité. Pour les approches NLP et LLM, les tâches incluent également la reclassification d'alertes, la génération de playbooks de réponse et l'automatisation de l'analyse d'incidents. . . . 31
2.3 Architecture typique d'un réseau de neurones profond pour la détection d'intrusions . . . 33
2.4 Pipeline de traitement NLP pour l'analyse d'alertes de sécurité . . . 34
2.5 Architecture Transformer et mécanisme d'attention multi-têtes. *Haut gauche* : bloc encodeur avec auto-attention et connexions résiduelles. *Haut droite* : bloc décodeur avec attention masquée, attention croisée (K, V de l'encodeur) et projection finale vers les probabilités de tokens. *Bas* : détail du mécanisme d'attention multi-têtes (matrices Q, K, V et concaténation des têtes). . . . 36
2.6 Processus de fine-tuning d'un LLM pour la cybersécurité . . . 37
2.7 Architecture d'un réseau de neurones sur graphe pour l'analyse de graphes d'attaque . . . 39
2.8 Exemple d'application des GNN pour l'analyse d'un graphe d'attaque multi-étapes . . . 40
2.9 Cadre général de l'apprentissage par renforcement appliqué à la cybersécurité 42
2.10 Scénario d'application du RL pour la défense adaptative contre les APT . . 43
2.11 Intégration des différentes techniques d'IA dans l'architecture globale du système d'alerte précoce . . . 45

3.1 Architecture globale du système intelligent d'alerte précoce montrant l'intégration des cinq contributions . . . 52
3.2 Architecture du cadre MCM-CASR comprenant les modules de réduction et de corrélation . . . 55
3.3 Architecture de MCM-Llama montrant le processus de fine-tuning et de déploiement . . . 57
3.4 Architecture GAT-DQN : sélection contextuelle entre deux modèles par méta-modèle k-NN . . . 61
3.5 Vue d'ensemble des jeux de données et protocoles expérimentaux utilisés dans les cinq contributions . . . 73
3.6 Intégration des cinq contributions dans une architecture unifiée . . . 75

4.1 Évolution des méthodes de reconstruction de scénarios d'attaque (2018-2025). La chronologie illustre la progression depuis les approches ontologiques et bayésiennes vers les réseaux neuronaux graphiques et les architectures de transformateurs. 82
4.2 Architecture du cadre MCM-CASR. Le pipeline comprend : (1) l'ingestion des alertes IDS, (2) la réduction par critères multi-attributs, (3) l'extraction d'entités par REN, (4) le calcul de similarité par BERT, et (5) la reconstruction des scénarios d'attaque. 90
4.3 Utilisation des techniques par approche. Le diagramme illustre l'intensité d'utilisation de chaque famille de techniques : TALN/Texte, Graphes, Apprentissage profond, Apprentissage par renforcement et Similarité. 97
4.4 Profil comparatif multi-dimensionnel. MCM-CASR (rouge) présente des forces en analyse sémantique et indépendance vis-à-vis des connaissances, tandis que GRAIN excelle en découverte de causalité. 98
4.5 Synthèse visuelle de la comparaison des méthodes de reconstruction de scénarios d'attaque. Ce tableau récapitule la technique principale, la dépendance à l'expertise, les capacités de détection, les taux de réduction des alertes, les jeux de données utilisés, la précision de corrélation, les scénarios reconstruits, la capacité temps réel et l'innovation clé pour chaque approche. 99

5.1 Mécanismes de fonctionnement des Grands Modèles de Langage. Le processus comprend l'encodage du prompt, le traitement par le modèle Transformer, et la génération de la complétion. 106
5.2 Scénarios et attaques correspondantes dans le jeu de données CTU-13. Chaque scénario représente un échantillon de botnet distinct avec des comportements malveillants spécifiques. 110
5.3 Exemple d'alerte IDS générée par Suricata. L'alerte contient les métadonnées essentielles : horodatage, adresses IP source/destination, signature, et catégorie de menace. 111
5.4 Architecture proposée pour l'implémentation de MCM-Llama. Le pipeline comprend : (1) IDS pour la génération d'alertes, (2) base de données pour le stockage, (3) framework MCM-Llama pour l'agrégation et la corrélation, (4) interface de visualisation (Kibana ou web). 113
5.5 Progression de la perte d'entraînement. La courbe montre une convergence rapide du modèle vers un minimum stable. 115
5.6 Réponse du modèle pour deux alertes corrélées. Le modèle identifie correctement la corrélation et répond "YES". 115
5.7 Réponse du modèle pour deux alertes non corrélées. Le modèle identifie correctement l'absence de corrélation et répond "NO". 115

6.1 Architecture globale du système avec sélection dynamique entre les modèles GAT et DQN. 131
6.2 Métriques de performance globales du modèle GAT. 135
6.3 Matrice de confusion du modèle GAT + Transformer. 136
6.4 Métriques de performance du modèle DQN. 137
6.5 Matrice de confusion du modèle DQN. 137

7.1 Comparaison des performances des modèles à travers les tâches de classification. 153
7.2 Comparaison des temps d'entraînement à travers tous les modèles. 166
7.3 Comparaison des scores F1 à travers les modèles et les tâches. 167
7.4 Matrice de confusion - Régression Logistique (Classification Binaire). . . . 167
7.5 Carte de chaleur de l'exactitude à travers les modèles et les tâches de classification. 168

8.1 Architecture trois couches SIEM–IA–SOAR. La flèche en pointillés représente l'interrogation REST périodique (toutes les 15 min) ; la double flèche représente l'échange bidirectionnel prompt/réponse avec le LLM. 176
8.2 Flux de corrélation d'alertes par LLM et de réponse SOAR (Algorithme 8.3.3). 179

Liste des tableaux

1.1 Principaux groupes APT actifs en 2024-2025 et leurs cibles 11
1.2 Synthèse des lacunes identifiées dans les approches existantes 26
1.3 Comparaison des familles d'approches selon les critères opérationnels clés . 28

2.1 Synthèse des techniques d'IA et leurs applications en cybersécurité dans cette thèse 44

3.1 Comparaison des performances MCM-CASR vs MCM-Llama 59
3.2 Performances GAT vs DQN sur CTU-13 64
3.3 Comparaison avec les approches de référence 65
3.4 Performances des algorithmes sur les trois tâches de classification 70
3.5 Performances de corrélation du pipeline TinyLlama-SOAR (500 alertes) . . 72
3.6 Synergies entre les cinq contributions 76
3.7 Interfaces d'échange entre les modules du système intégré 78

4.1 Synthèse comparative des approches de reconstruction de scénarios d'attaque 85
4.2 Caractéristiques du jeu de données UNSW-NB15 86
4.3 Résultats de la réduction des alertes par catégorie 91
4.4 Exemple de scénario de cyberattaque reconstruit 92
4.5 Résultats comparatifs de la corrélation des alertes 93
4.6 Comparaison méthodologique des approches de reconstruction de scénarios d'attaque 94
4.7 Forces de MCM-CASR 95
4.8 Comparaison quantitative des performances 95
4.9 Comparaison fonctionnelle des approches 96

5.1 Positionnement de MCM-Llama par rapport aux travaux connexes 105
5.2 Caractéristiques du jeu de données CTU-13 [2] 109
5.3 Paramètres de configuration pour le fine-tuning de MCM-Llama 112
5.4 Matrice de confusion de MCM-Llama 116
5.5 Métriques de performance de MCM-Llama 116
5.6 Comparaison quantitative des performances 118

5.7 Comparaison qualitative des approches 118
5.8 Évaluation selon les critères opérationnels 119

6.1 Synthèse comparative des approches de projection d'attaque 127
6.2 Métriques de performance globales du modèle GAT 134
6.3 Détails de la matrice de confusion pour le modèle GAT 135
6.4 Métriques de performance du modèle DQN 136
6.5 Détails de la matrice de confusion pour le modèle DQN 136
6.6 Performance des modèles sur 20 000 alertes synthétiques 138
6.7 Comparaison qualitative des approches 140
6.8 Comparaison quantitative des performances 140
6.9 Matrice de support des fonctionnalités 141

7.1 Positionnement de notre approche par rapport aux travaux connexes . . . 150
7.2 Distribution des alertes générées par type et par niveau de sévérité 151
7.3 Performance de la détection binaire de corrélation avec validation croisée . 164
7.4 Performance de la classification des types de corrélation avec validation croisée 165
7.5 Performance de la classification des incidents avec validation croisée 165
7.6 Comparaison complète des performances à travers toutes les tâches 165

8.1 Positionnement comparatif du framework proposé par rapport à l'état de l'art 175
8.2 Métriques de performance de corrélation (500 alertes) 181
8.3 Décomposition du temps de réponse bout en bout 181
8.4 Utilisation des ressources lors du traitement au pic de charge 181
8.5 Comparaison quantitative avec les approches connexes 182

Liste des Acronymes

ANN	Artificial Neural Networks (Réseaux de Neurones Artificiels)
APT	Advanced Persistent Threat (Menace Persistante Avancée)
BERT	Bidirectional Encoder Representations from Transformers
C2	Command and Control
CNN	Convolutional Neural Network (Réseau de Neurones Convolutif)
CVE	Common Vulnerabilities and Exposures
DDoS	Distributed Denial of Service (Déni de Service Distribué)
DoS	Denial of Service (Déni de Service)
DQN	Deep Q-Network (Réseau Q Profond)
DRL	Deep Reinforcement Learning (Apprentissage par Renforcement Profond)
EWS	Early Warning System (Système d'Alerte Précoce)
FN	Faux Négatifs (False Negatives)
FP	Faux Positifs (False Positives)
FPR	False Positive Rate (Taux de Faux Positifs)
GAT	Graph Attention Network (Réseau d'Attention sur Graphes)
GNN	Graph Neural Network (Réseau de Neurones sur Graphes)
GRU	Gated Recurrent Unit
HMM	Hidden Markov Model (Modèle de Markov Caché)
ICS	Industrial Control Systems (Systèmes de Contrôle Industriels)
IDS	Intrusion Detection System (Système de Détection d'Intrusion)
IPS	Intrusion Prevention System (Système de Prévention d'Intrusion)
LLM	Large Language Model (Modèle de Langage de Grande Taille)
LoRA	Low-Rank Adaptation (Adaptation de Faible Rang)
LSTM	Long Short-Term Memory
MDP	Markov Decision Process (Processus de Décision de Markov)

ML	Machine Learning (Apprentissage Automatique)
NER	Named Entity Recognition (Reconnaissance d'Entités Nommées)
NLP	Natural Language Processing (Traitement Automatique du Langage Naturel)
PEFT	Parameter-Efficient Fine-Tuning
RaaS	Ransomware-as-a-Service
REN	Reconnaissance d'Entités Nommées
RL	Reinforcement Learning (Apprentissage par Renforcement)
RLHF	Reinforcement Learning from Human Feedback
RNN	Recurrent Neural Network (Réseau de Neurones Récurrent)
EDR	Endpoint Detection and Response (Détection et Réponse sur les Points de Termina
MTTR	Mean Time to Respond (Temps Moyen de Réponse)
SIEM	Security Information and Event Management
SOAR	Security Orchestration, Automation and Response
SOC	Security Operations Center (Centre d'Opérations de Sécurité)
SVM	Support Vector Machine (Machine à Vecteurs de Support)
TALN	Traitement Automatique du Langage Naturel
TPR	True Positive Rate (Taux de Vrais Positifs)
TTPs	Tactics, Techniques, and Procedures
VN	Vrais Négatifs (True Negatives)
VP	Vrais Positifs (True Positives)
VPN	Virtual Private Network (Réseau Privé Virtuel)

Introduction générale

La cybersécurité est devenue un enjeu majeur de notre époque, tant pour les gouvernements que pour les entreprises et les citoyens. Avec l'expansion continue des technologies numériques, l'interconnexion croissante des systèmes d'information et la multiplication des cybermenaces, la protection des infrastructures critiques et des données sensibles représente un défi de première importance. Les attaques informatiques ne cessent de croître en nombre, en complexité et en sophistication, causant des dommages économiques considérables et menaçant la sécurité nationale de nombreux pays [3, 4, 5].

Face à cette évolution des menaces, les mécanismes traditionnels de détection et de prévention des intrusions montrent leurs limites. Les systèmes de sécurité actuels génèrent un volume massif d'alertes dont la corrélation et l'analyse dépassent souvent les capacités humaines. De plus, ces systèmes adoptent majoritairement une posture réactive, intervenant après qu'une intrusion ait déjà commencé, plutôt qu'une approche proactive permettant d'anticiper et de prévenir les attaques.

C'est dans ce contexte que s'inscrit cette thèse, qui propose de concevoir un système intelligent d'alerte précoce pour la cybersécurité, exploitant les avancées récentes en intelligence artificielle, en apprentissage automatique et en traitement du langage naturel. L'objectif est de développer des mécanismes capables non seulement de détecter les attaques en cours, mais également d'anticiper leur évolution et de prédire les étapes futures, permettant ainsi une réponse proactive et adaptée. Ces travaux ont donné lieu à cinq publications scientifiques [6, 7, 8, 9, 10] et une communication à un colloque international [11].

0.1 Contexte

Au cours des dernières décennies, les systèmes d'information ont pris une place centrale dans le fonctionnement des organisations, qu'il s'agisse d'administrations publiques, d'entreprises privées ou d'infrastructures critiques. La généralisation de la numérisation des services, l'essor du télétravail, l'utilisation massive d'objets connectés et la croissance continue des volumes de données échangées ont profondément transformé la surface d'attaque non seulement des réseaux informatiques, mais aussi des applications, des services en ligne et des environnements cloud [5, 12].

Dans ce contexte, les attaques informatiques sont devenues plus fréquentes, plus sophistiquées et plus ciblées. Les attaquants exploitent aujourd'hui des techniques variées, allant des attaques par déni de service aux campagnes d'attaques multi-étapes, en passant par l'exploitation de vulnérabilités logicielles et l'ingénierie sociale [5]. Les conséquences de ces attaques peuvent être majeures : interruption de service, compromission de données sensibles, pertes financières, atteinte à la réputation et perturbation d'infrastructures critiques.

Face à ces menaces, les organisations ont déployé de nombreux mécanismes de sécurité tels que les pare-feu, les antivirus, les systèmes de prévention d'intrusion, les réseaux privés virtuels et les plateformes de supervision. Toutefois, ces dispositifs génèrent une quantité massive d'événements et d'alertes hétérogènes, dont l'analyse manuelle devient de plus en plus complexe.

Les systèmes d'alerte précoce (Early Warning Systems) visent à détecter le plus tôt possible les signaux précurseurs d'un événement critique afin de permettre des réactions rapides et adaptées. Ce concept est largement utilisé dans des domaines tels que la météorologie, la santé publique ou la gestion des risques environnementaux [13].

En cybersécurité, l'objectif est d'exploiter les données historiques, les flux d'alertes et les mécanismes de corrélation afin d'identifier les prémices d'une attaque et d'en anticiper l'évolution [14]. L'intelligence artificielle (IA), notamment l'apprentissage automatique et le traitement du langage naturel, offre des perspectives intéressantes pour relever ce défi [15, 16].

0.2 Problématique

Les systèmes de détection d'intrusion permettent d'identifier des comportements suspects à partir de signatures ou d'anomalies. Cependant, ils restent majoritairement fondés sur une logique réactive, intervenant souvent après le début effectif de l'attaque [17]. De plus, les attaques modernes, en particulier les menaces persistantes avancées, se caractérisent par une succession d'étapes discrètes réparties dans le temps.

Plusieurs difficultés majeures se posent :

— le volume extrêmement élevé d'alertes ;

— la présence de nombreux faux positifs ;

— la fragmentation des informations de sécurité ;

— l'incapacité à anticiper les étapes futures d'une attaque.

Les approches classiques de corrélation, souvent basées sur des règles statiques, montrent rapidement leurs limites face à l'évolution constante des menaces [18, 19].

0.3 Objectifs de la Thèse

0.3.1 Objectif général

L'objectif principal de cette thèse est de concevoir un système intelligent de détection précoce d'attaques basé sur des techniques avancées d'intelligence artificielle. Il s'agit de développer des mécanismes capables non seulement de détecter les attaques en cours, mais également d'anticiper leur évolution et de prédire les étapes futures, permettant ainsi une réponse proactive et adaptée.

0.3.2 Objectifs spécifiques

Pour atteindre l'objectif général, cinq axes de recherche complémentaires ont été définis. Le premier consiste à développer des mécanismes de corrélation d'alertes fondés sur le traitement du langage naturel et la similarité sémantique, afin de reconstruire automatiquement des scénarios d'attaque cohérents. Le deuxième vise à exploiter les modèles de langage de grande taille pour améliorer la corrélation et la détection de menaces en temps réel. Le troisième propose une approche de projection d'attaques combinant réseaux de neurones sur graphes et apprentissage par renforcement, permettant d'anticiper les étapes futures d'une attaque en cours. Le quatrième objectif consiste à améliorer la classification multi-classe des alertes de sécurité par apprentissage automatique, afin de prioriser les incidents selon différentes granularités décisionnelles. Enfin, le cinquième axe vise à opérationnaliser l'ensemble du système en intégrant SIEM, SOAR et un petit modèle de langage léger (TinyLlama-1.1B) dans un pipeline de réponse automatisée aux incidents, déployable sur infrastructure contrainte sans GPU.

0.4 Contributions de la Thèse

Cette thèse a donné les contributions suivantes :

— proposition d'un cadre méthodologique de corrélation d'alertes (MCM-CASR) exploitant le traitement automatique du langage naturel, la reconnaissance d'entités nommées et la similarité sémantique pour la reconstruction automatisée de scénarios d'attaque [6].

— introduction d'un modèle de langage de grande taille spécialisé (MCM-Llama) pour la corrélation et la détection de menaces en temps réel, exploitant les architectures de transformateurs adaptées au domaine de la sécurité informatique [7].

— développement d'une approche adaptative de projection d'attaques combinant les réseaux de neurones sur graphes (GAT) et l'apprentissage par renforcement (DQN) pour la sélection dynamique de modèles en fonction du contexte [8].

— conception d'un modèle de corrélation multi-classe par apprentissage automatique pour améliorer la classification et la priorisation des alertes de sécurité [9].

— développement d'un pipeline opérationnel intégrant SIEM (Elastic Stack), SOAR (TheHive/Cortex/SentinelOne) et un modèle de langage léger (TinyLlama-1.1B) pour la corrélation sémantique d'alertes et la réponse automatisée aux incidents, déployable sur infrastructure ordinaire sans GPU [10].

L'ensemble de ces contributions constitue un système intégré d'alerte précoce intelligent pour la cybersécurité, combinant plusieurs approches complémentaires de l'intelligence artificielle pour améliorer significativement la détection, la corrélation et la prédiction des cyberattaques.

0.5 Liste des publications

Les travaux de cette thèse ont donné lieu aux publications suivantes :

[6] Diakhame, M.L., Diallo, C., Mejri, M. *"MCM-CASR : Novel Alert Correlation Framework for Cyber Attack Scenario Reconstruction Based on NLP, NER, and Semantic Similarity"*. In *2023 7th Cyber Security in Networking Conference (CSNet)*, pp. 27–31. IEEE, 2023.

[7] Diakhame, M.L., Diallo, C., Mejri, M. *"MCM-Llama : A Fine-Tuned Large Language Model for Real-Time Threat Detection through Security Event Correlation"*. In *Proceedings of the 2024 International Conference on Electrical, Computer and Energy Technologies (ICECET)*, pp. 1–6. IEEE, 2024.

[8] Diakhame, M.L., Diallo, C., Mejri, M. *"Adaptive Cyber Attack Projection through Context-Driven Model Selection : Combining Graph Attention Networks and Reinforcement Learning"*. In *2025 5th Intelligent Cybersecurity Conference (ICSC)*, pp. 28–34. IEEE, 2025.

[9] Diakhame, M.L., Diallo, C., Mejri, M. *"Machine Learning-Based Alert Correlation for Enhanced Cybersecurity : A Multi-Classification Approach"*. In *2025 9th Cyber Security in Networking Conference (CSNet)*, pp. 1–8. IEEE, 2025.

[10] Diakhame, M.L., Diallo, C. *"LLM-Driven Alert Correlation for Automated Incident Response"*. Soumis à *2025 IEEE International Conference on Cyber Security and Resilience (CSR)*, IEEE, 2025. *(Article soumis à une conférence internationale avec comité de lecture — en cours d'évaluation.)*

[20] Diakhame, M.L., Dieng, D.B., Nougaret, J., Mwaura, J. *"Soil Drainage Classification Using Machine Learning Models : A Comparative Study"*. In *Pan-African Artificial Intelligence and Smart Systems Conference*, pp. 213–226. Springer, 2024.

En complément, une communication orale a été présentée au colloque suivant :

[11] Diakhame, M.L., Diallo, C., Mejri, M. *"Adaptive Cyber Attack Projection through Context-Driven Model Selection : Combining Graph Attention Networks and Reinforcement Learning"*. Communication orale au *Colloque International 2025 : Sciences et Technologies comme moteur du Bien-Être, de la Transition Énergétique et Agroécologique*, organisé par l'École Doctorale des Sciences et des Technologies (ED-ST) et le CEA-MITIC, Université Gaston Berger de Saint-Louis, Sénégal, 20–22 mai 2025.

0.6 Organisation de la Thèse

Ce manuscrit de thèse est organisé en huit chapitres, encadrés par une introduction générale et une conclusion générale.

Chapitre 1 – État de l'art sur la cybersécurité, la corrélation d'alertes et les systèmes d'alerte précoce. Ce chapitre dresse un panorama approfondi des travaux existants en matière de détection d'intrusion, de corrélation d'alertes et de systèmes d'alerte précoce. Il permet de positionner nos travaux par rapport aux approches existantes et d'identifier les lacunes que cette thèse vise à combler.

Chapitre 2 – Intelligence artificielle, NLP et modèles de langage pour la cybersécurité. Ce chapitre est consacré aux fondements en intelligence artificielle mobilisés dans cette thèse. Il couvre l'apprentissage automatique, le traitement du langage naturel, les réseaux de neurones sur graphes et les modèles de langage de grande taille, en les situant dans le contexte de la cybersécurité.

Chapitre 3 – Méthodologie générale et architecture du système d'alerte précoce. Ce chapitre détaille la méthodologie générale et l'architecture globale du système d'alerte précoce proposé. Il établit le cadre intégrateur des différentes contributions et présente la vision d'ensemble du système.

Chapitre 4 – MCM-CASR : corrélation d'alertes et reconstruction de scénarios d'attaque. Ce chapitre présente MCM-CASR, notre cadre de corrélation d'alertes basé sur le traitement automatique du langage naturel, la reconnaissance d'entités nommées et la similarité sémantique. Il propose une approche de reconstruction automatisée de scénarios de cyberattaques à partir de flux d'alertes IDS.

Chapitre 5 – MCM-Llama : détection de menaces en temps réel par modèle de langage. Ce chapitre introduit MCM-Llama, un modèle de langage de grande taille

affiné pour la corrélation d'événements de sécurité en temps réel. Il démontre l'apport des LLM par rapport aux méthodes classiques de TALN pour la détection de menaces.

Chapitre 6 – Projection adaptative d'attaques par GAT et apprentissage par renforcement. Ce chapitre détaille l'approche de projection adaptative d'attaques combinant les réseaux d'attention sur graphes (GAT) et l'apprentissage par renforcement profond (DQN). Il propose un méta-modèle de sélection dynamique pour la prédiction des phases futures d'une cyberattaque.

Chapitre 7 – Corrélation d'alertes par apprentissage automatique multi-classe. Ce chapitre expose le modèle de corrélation multi-classe qui transforme la corrélation d'alertes en trois problèmes de classification complémentaires. Il évalue huit algorithmes d'apprentissage automatique sur des données réalistes issues du jeu de données CIC-IDS-2017.

Chapitre 8 – Pipeline opérationnel SIEM-SOAR-LLM pour la corrélation d'alertes et la réponse automatisée aux incidents. Ce chapitre présente un pipeline opérationnel intégrant un SIEM (Elastic Stack), une plateforme SOAR (TheHive/Cortex/SentinelOne) et un modèle de langage léger (TinyLlama-1.1B) pour la corrélation sémantique d'alertes et la réponse automatisée aux incidents. Il démontre la faisabilité d'un déploiement sur infrastructure ordinaire sans GPU, en réduisant significativement le temps de réponse aux incidents de sécurité.

Conclusion générale. La conclusion générale propose une synthèse des travaux, résume les apports scientifiques, discute les limites des approches proposées et présente les perspectives de recherche futures.

Chapitre 1

Cybersécurité, Corrélation d'Alertes et Systèmes d'Alerte Précoce

1.1 Introduction

Ce chapitre présente un état de l'art des principaux concepts liés à la cybersécurité, à la corrélation d'alertes et aux systèmes d'alerte précoce. Il vise à situer les travaux de cette thèse par rapport aux approches existantes, en mettant en évidence leurs apports, mais aussi leurs limites. Une attention particulière est portée aux mécanismes de corrélation, de prédiction et de projection d'attaques, qui constituent le socle conceptuel des contributions proposées dans la suite de ce manuscrit.

Le paysage des menaces cybernétiques a connu une transformation majeure ces dernières années, marquée par l'émergence de menaces sophistiquées telles que les ransomwares, les attaques de la chaîne d'approvisionnement et les menaces persistantes avancées (APT). Selon le rapport Global Cybersecurity Outlook 2025 du World Economic Forum [3], la complexité du paysage cyber est exacerbée par les tensions géopolitiques, l'intégration croissante des chaînes d'approvisionnement et l'adoption rapide de technologies émergentes contribuant à de nouvelles vulnérabilités.

1.2 Cybersécurité et menaces modernes

Cette section présente le paysage actuel des cybermenaces, leur évolution et les principales catégories d'attaques auxquelles sont confrontées les organisations. Elle constitue le socle contextuel nécessaire à la compréhension des enjeux de détection et de corrélation traités dans cette thèse.

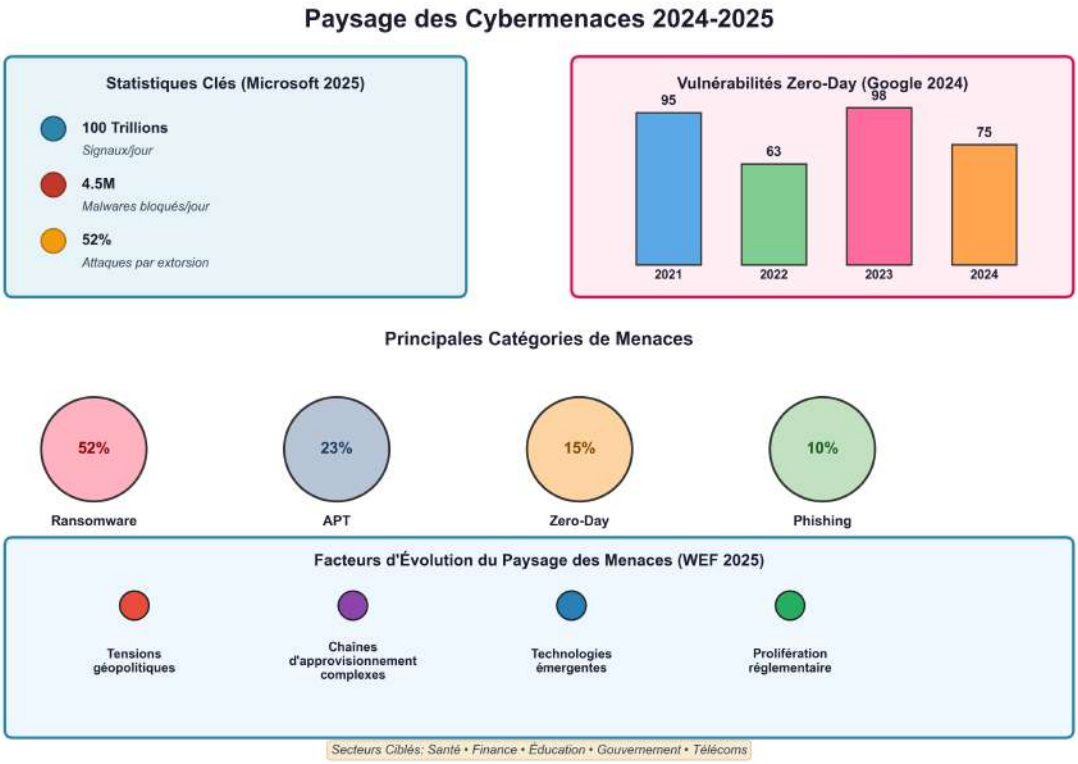


FIGURE 1.1 – Vue d'ensemble du paysage des cybermenaces 2024-2025 avec statistiques clés des rapports Microsoft, Google et WEF.

1.2.1 Évolution du paysage des menaces

Le paysage des menaces en cybersécurité a connu une évolution rapide au cours des dernières années, marquée par une diversification des vecteurs d'attaque, une sophistication accrue des outils utilisés par les attaquants et une augmentation significative du nombre d'incidents [5]. La généralisation des services en ligne, du cloud computing et de l'Internet des objets a considérablement élargi la surface d'attaque des systèmes d'information.

Statistiques clés 2024-2025

Les rapports récents révèlent des tendances préoccupantes. Selon le Microsoft Digital Defense Report 2025 [4], plus de 52% des cyberattaques avec des motivations connues sont désormais motivées par l'extorsion ou les ransomwares, tandis que l'espionnage ne représente que 4%.Microsoft traite quotidiennement plus de 100 000 milliards de signaux, bloque environ 4,5 millions de nouvelles tentatives de malware et analyse 38 millions de détections de risques d'identité.

Le rapport Google Threat Intelligence Group (GTIG) de 2024 [21] a identifié 75 vulnérabilités zero-day exploitées dans la nature, contre 98 en 2023. Bien que ce chiffre soit en baisse, il représente une augmentation par rapport aux 63 vulnérabilités de 2022, confirmant une tendance à la hausse sur le long terme. De manière significative, 44% de ces

zero-days ciblaient des technologies d'entreprise, avec 60% d'entre eux visant des logiciels et appliances de sécurité.

Facteurs d'évolution

Les attaquants poursuivent des objectifs variés, allant du vol de données sensibles à l'extorsion financière, en passant par l'espionnage et le sabotage. Les campagnes d'attaques les plus avancées sont généralement structurées en plusieurs étapes et s'inscrivent dans la durée [22]. Le rapport du World Economic Forum [3] identifie plusieurs facteurs contribuant à cette complexité croissante :

— les tensions géopolitiques contribuant à un environnement plus incertain ;

— L'intégration et la dépendance accrues envers des chaînes d'approvisionnement plus complexes ;

— L'adoption rapide de technologies émergentes contribuant à de nouvelles vulnérabilités ;

— la prolifération des réglementations internationales créant des défis de conformité.

1.2.2 Typologie des cyberattaques modernes

Les cyberattaques peuvent être classées selon plusieurs critères, notamment en fonction de leur objectif et de leur mode opératoire. La taxonomie contemporaine distingue plusieurs catégories majeures présentées dans les sous-sections suivantes.

Attaques selon les propriétés de sécurité

— attaques contre la confidentialité : Exfiltration de données, interception de communications, vol de propriété intellectuelle. Selon le Microsoft Digital Defense Report [4], 80% des incidents investigués visaient le vol de données.

— attaques contre l'intégrité : Altération de données, compromission de configurations, injection de code malveillant, manipulation de transactions.

— attaques contre la disponibilité : Attaques par déni de service distribué (DDoS), ransomwares avec chiffrement, crypto mining non autorisé, sabotage de systèmes critiques.

Ransomwares et extorsion

Les attaques par ransomware représentent aujourd'hui la menace la plus significative pour les organisations. Le rapport Hornetsecurity 2026 [23] révèle que 24% des organisations ont signalé un incident ransomware en 2025, inversant plusieurs années de déclin.

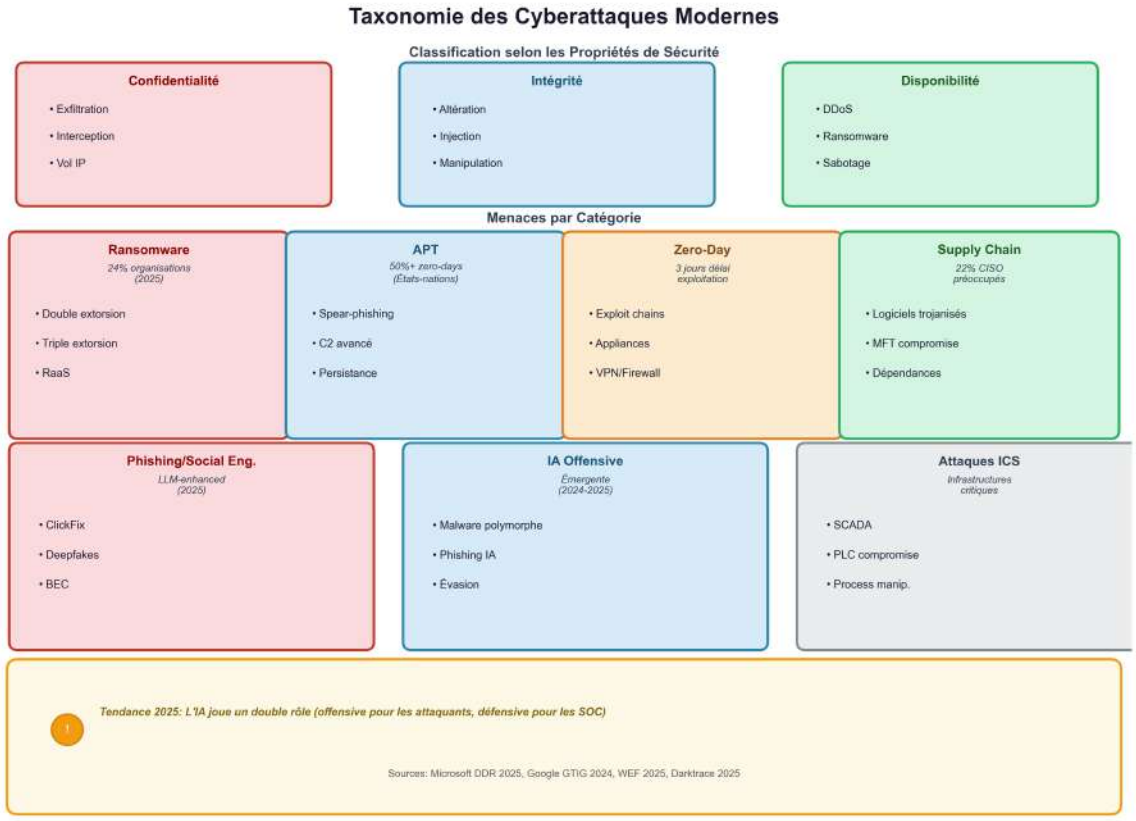


FIGURE 1.2 – Taxonomie complète des cyberattaques modernes selon les propriétés de sécurité et catégories de menaces.

Les acteurs du ransomware ont adapté leurs tactiques, passant du simple chiffrement à des stratégies de double et triple extorsion.

[Double extorsion] La double extorsion est une technique où les attaquants non seulement chiffrent les données de la victime, mais menacent également de les divulguer publiquement si la rançon n'est pas payée, augmentant ainsi la pression sur les organisations [24].

Les groupes de ransomware les plus actifs en 2025 incluent Cl0p, RansomHub et Akira [24]. Ces groupes ciblent particulièrement les secteurs de la santé, de l'éducation et des gouvernements locaux en raison de leurs données sensibles et de leurs ressources de cybersécurité limitées.

Menaces persistantes avancées (APT)

Les APT représentent une catégorie particulièrement sophistiquée de cybermenaces, caractérisées par leur discrétion, leur durée et leur forte capacité d'adaptation [25]. Selon le rapport Darktrace 2025 [26], les groupes APT continuent d'exploiter des techniques avancées, avec une augmentation notable de l'activité de groupes tels que BlindEagle (APT-C-36) ciblant l'Amérique latine et des campagnes liées à la Chine utilisant le réseau

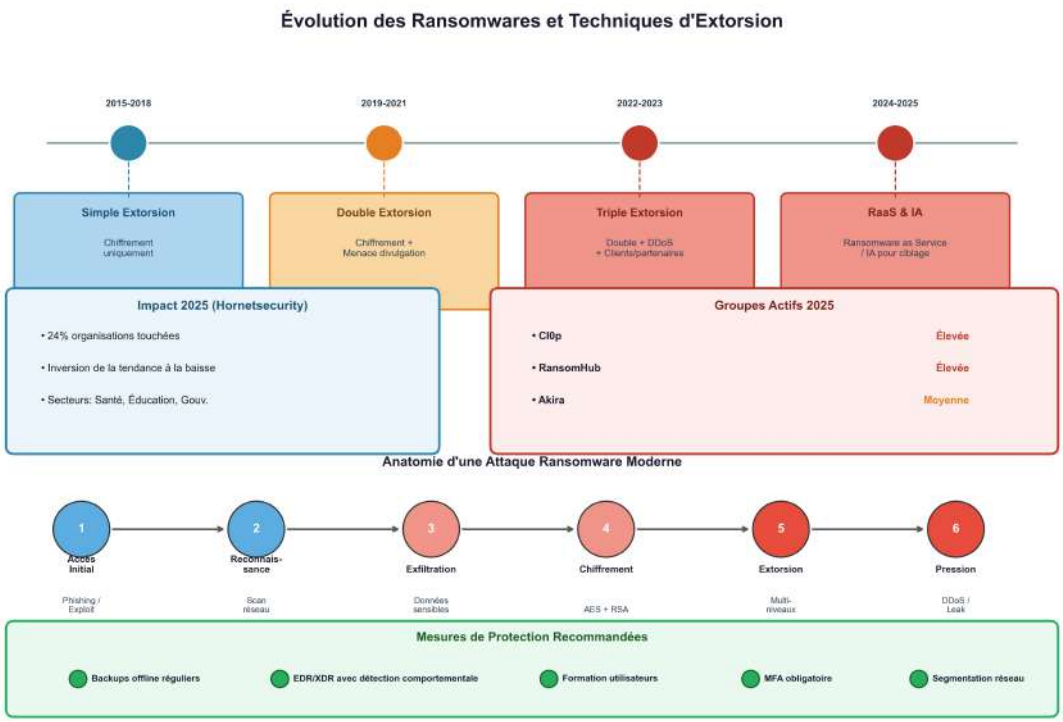


FIGURE 1.3 – Évolution des techniques de ransomware de la simple extorsion au RaaS (Ransomware-as-a-Service) alimenté par l'IA.

TABLE 1.1 – Principaux groupes APT actifs en 2024-2025 et leurs cibles

Groupe	**Attribution**	**Cibles principales**	**Techniques**
Salt Typhoon	Chine	Télécommunications	JumbledPath
APT29	Russie	Gouvernements, militaire	Spear-phishing
Gamaredon	Russie	Ukraine (état, militaire)	Malware
Earth Estries	Chine	Infrastructures critiques	Zero-days
BlindEagle	Amérique latine	Secteur financier	Phishing

ORB LapDogs.

Les groupes APT soutenus par des États (Chine, Russie, Corée du Nord) représentent plus de 50% des exploitations zero-day attribuées en 2024 [21]. Les groupes chinois et nord-coréens ont chacun exploité cinq zero-days, démontrant une parité croissante dans les capacités offensives.

Vulnérabilités zero-day

Les vulnérabilités zero-day constituent un vecteur d'attaque critique car elles sont exploitées avant qu'un correctif ne soit disponible. Le rapport GTIG [21] révèle plusieurs tendances importantes :

— ciblage des technologies d'entreprise : 44% des zero-days exploités en 2024 visaient des solutions d'entreprise, contre 37% en 2023.

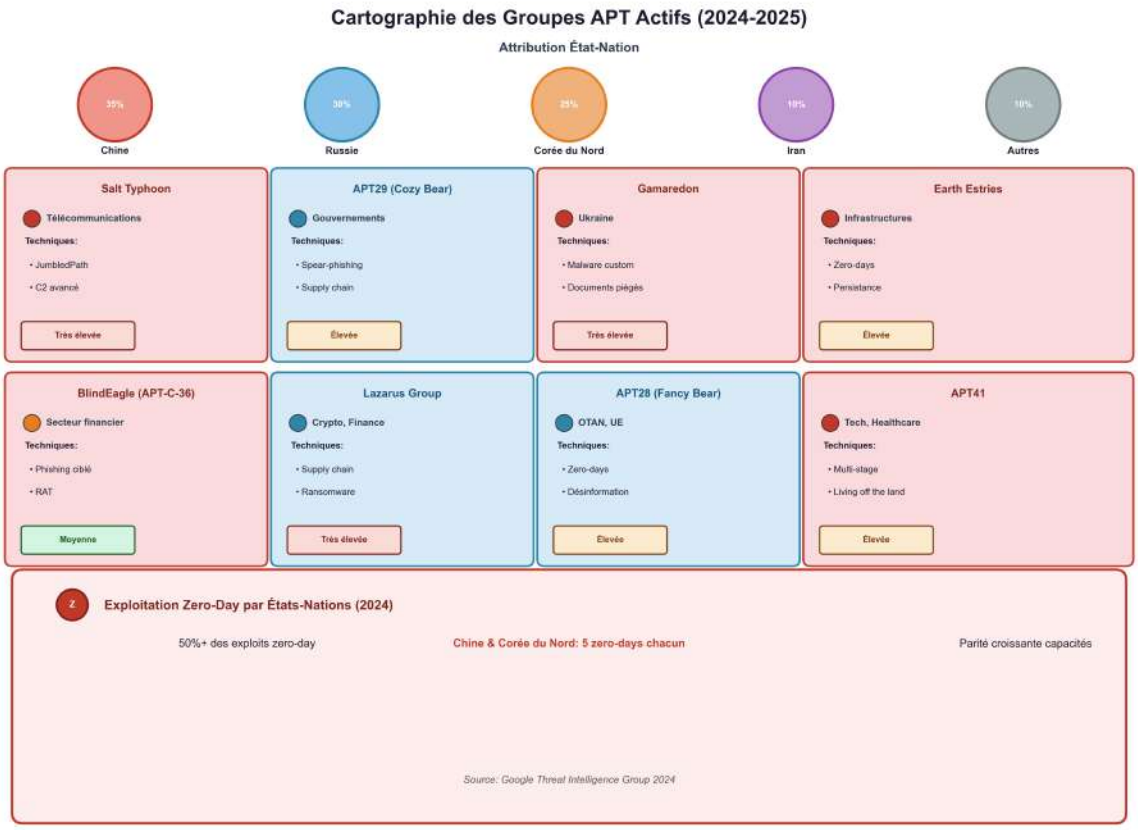


FIGURE 1.4 – Cartographie des principaux groupes APT actifs en 2024-2025 avec leurs attributions, cibles et techniques caractéristiques.

— focus sur la sécurité périmétrique : Les produits de sécurité et de réseau (VPN, pare-feu, passerelles) représentent 60% des exploitations de technologies d'entreprise.

— réduction du temps d'exploitation : Le délai moyen entre la divulgation d'une vulnérabilité et son exploitation active est passé à seulement 5 jours en 2024, contre 32 jours les années précédentes [27].

— déclin des exploitations mobile et navigateur : Les exploitations de navigateurs ont chuté de 17 à 11 et celles des mobiles de 17 à 9.

Attaques de la chaîne d'approvisionnement

Les attaques de la chaîne d'approvisionnement ont émergé comme un vecteur critique, permettant aux attaquants d'atteindre de multiples organisations via un seul point d'entrée. Le rapport du World Economic Forum [3] identifie les perturbations de la chaîne d'approvisionnement comme l'une des principales préoccupations des CISO (22%) et des CEO (26%).

Des incidents notables illustrent cette menace :

— L'exploitation de la vulnérabilité CVE-2025-8088 dans WinRAR démontrant comment les attaquants manipulent les distributions de logiciels de confiance ;

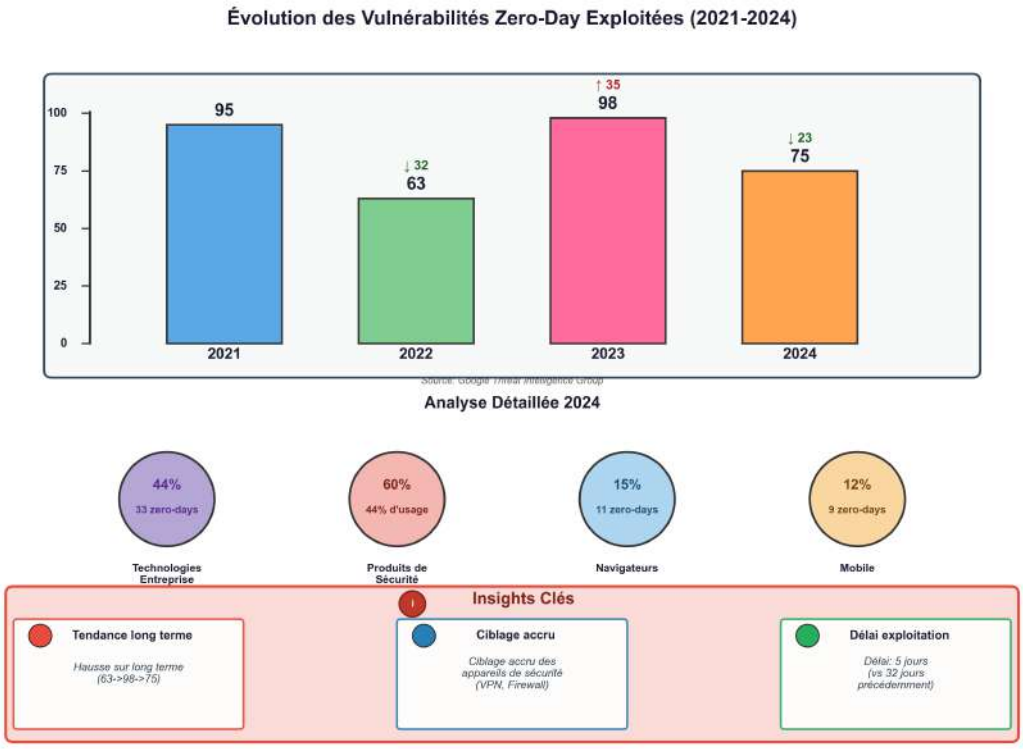


FIGURE 1.5 – Évolution du nombre de vulnérabilités zero-day exploitées dans la nature de 2021 à 2024, avec analyse détaillée des catégories ciblées.

— les campagnes ciblant les serveurs Microsoft SharePoint exposés sur Internet ;

— L'exploitation des produits de transfert de fichiers gérés (MFT) comme Cleo et MOVEit.

Phishing et ingénierie sociale

Le phishing reste l'un des principaux vecteurs d'accès initial. Selon le rapport Darktrace [26], la proportion d'e-mails de phishing avec un grand volume de texte a augmenté, suggérant que les acteurs malveillants exploitent les LLM (modèles de langage de grande taille) pour créer des e-mails de phishing plus crédibles et efficaces.

Une technique notable observée en 2025 est le *ClickFix social engineering*, qui exploite l'intersection entre les humains et la technologie pour tromper les utilisateurs et les amener à exécuter du code malveillant pour le compte de l'attaquant.

Rôle de l'intelligence artificielle

L'IA joue désormais un double rôle dans le paysage des cybermenaces :

IA Offensive Les acteurs malveillants exploitent l'IA pour :

— automatiser la création d'e-mails de phishing convaincants ;

— générer des deepfakes pour des escroqueries de type vishing ;

— développer des malwares polymorphes guidés par l'IA ;

— contourner les mécanismes de détection traditionnels.

IA Défensive Les défenseurs utilisent l'IA pour :

— détecter les menaces en analysant les schémas de comportement ;

— automatiser la réponse aux incidents ;

— réduire les faux positifs dans les systèmes de détection ;

— améliorer l'analyse des journaux et la corrélation d'événements.

1.3 Mécanismes classiques de défense

Face à la diversité des vecteurs d'attaque présentés dans la section précédente, les organisations ont développé un arsenal de mécanismes défensifs dont la maîtrise est indispensable pour situer les contributions de cette thèse. Cette section passe en revue les principaux dispositifs de défense, de la filtration périmétrique à la gestion centralisée des événements de sécurité.

1.3.1 Dispositifs périmétriques et filtrage

Les pare-feux et les mécanismes de filtrage constituent historiquement la première ligne de défense des réseaux. Ils permettent de contrôler les flux autorisés entre différentes zones du système d'information. Bien que ces dispositifs soient indispensables, ils restent insuffisants face à des attaques qui exploitent des failles applicatives ou des comportements légitimes détournés.

Le rapport CISA sur les systèmes de contrôle industriels [28] souligne que de nombreux environnements ICS fonctionnent avec des technologies « legacy » et des protocoles propriétaires en raison de leurs priorités de conception initiales, qui se concentraient sur l'opérabilité et la fiabilité plutôt que sur la cybersécurité.

1.3.2 Systèmes de détection et de prévention d'intrusion

Les systèmes de détection d'intrusion (IDS) et les systèmes de prévention d'intrusion (IPS) analysent le trafic réseau ou les activités des hôtes afin d'identifier des comportements suspects. Ils reposent principalement sur des approches par signatures ou par détection d'anomalies [17]. Ces systèmes sont largement déployés dans les environnements opérationnels, mais ils génèrent souvent un nombre important de faux positifs.

IDS à Signatures

Les IDS basés sur les signatures comparent le trafic réseau avec une base de données de signatures d'attaques connues. Bien qu'efficaces contre les menaces connues, ils sont inefficaces contre les attaques zero-day et nécessitent des mises à jour constantes.

IDS par Anomalies

Les IDS par anomalies établissent un profil du comportement normal du réseau et détectent les déviations. Ils peuvent identifier des attaques inconnues mais génèrent souvent plus de faux positifs. Ces deux approches sont complémentaires : les IDS à signatures offrent une haute précision sur les menaces connues, tandis que les IDS par anomalies couvrent les attaques zero-day, justifiant les approches hybrides de plus en plus répandues dans les déploiements opérationnels. À l'échelle nationale, ces principes de détection ont été intégrés dans des programmes institutionnels, dont le programme EINSTEIN en constitue l'exemple le plus représentatif.

Programme EINSTEIN

Le National Cybersecurity Protection System (NCPS) de la CISA [29] inclut EINSTEIN, une capacité de détection d'intrusion qui fournit au gouvernement fédéral américain un système d'alerte précoce, une conscience situationnelle améliorée des menaces d'intrusion et une identification en temps quasi réel des activités cyber malveillantes.

1.3.3 Plates-formes de gestion des événements de sécurité (SIEM)

Les plates-formes de gestion des événements et des informations de sécurité (SIEM) permettent d'agréger et de corréler les journaux provenant de sources multiples. Elles offrent une vue globale de l'état de sécurité du système d'information et facilitent l'investigation en cas d'incident. Toutefois, l'efficacité de ces plates-formes dépend fortement de la qualité des mécanismes de corrélation mis en œuvre [19].

Selon les données du marché [30], le segment SIEM a capturé plus de 34,4% de la part du marché du renseignement sur les menaces en 2024, avec des systèmes leaders tels que Splunk, IBM QRadar, Microsoft Sentinel et Palo Alto Cortex analysant les journaux et événements réseau pour anticiper les menaces en temps réel.

1.4 Corrélation d'alertes de sécurité

La corrélation d'alertes constitue l'une des pierres angulaires de la détection intelligente des cyberattaques. Après avoir décrit les mécanismes défensifs élémentaires, cette section

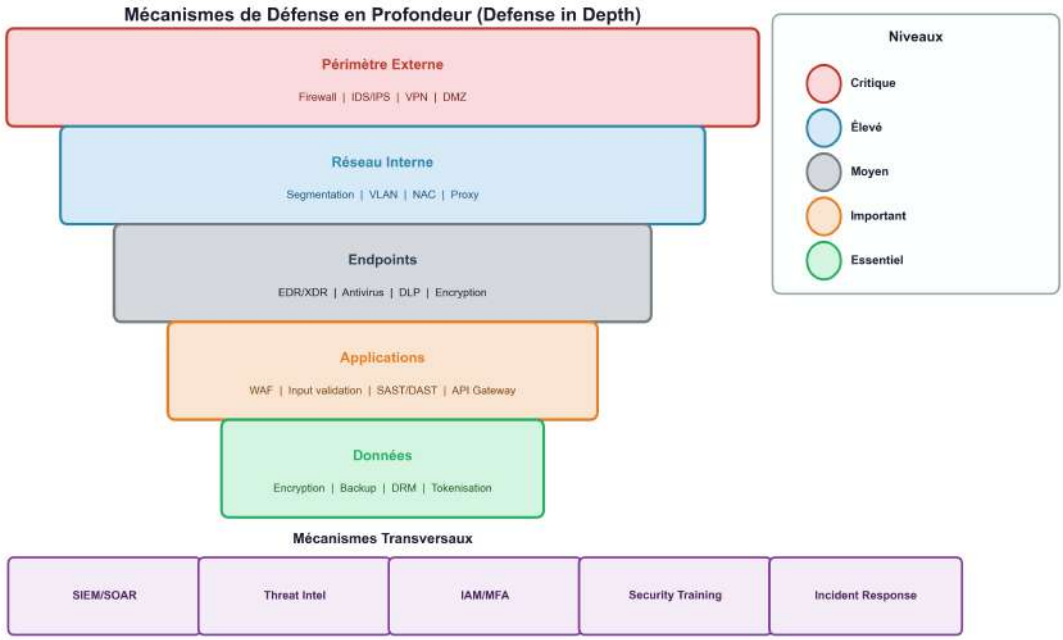


FIGURE 1.6 – Architecture en profondeur (Defense in Depth) des mécanismes de défense classiques avec couches périmètre, réseau, endpoints et applications.

approfondit les techniques permettant de relier entre eux des événements de sécurité disparates afin de reconstituer des scénarios d'attaque cohérents.

1.4.1 Objectifs et enjeux

La corrélation d'alertes vise à regrouper et à interpréter un ensemble d'événements de sécurité afin de reconstituer des scénarios d'attaque cohérents [18]. Elle permet de réduire le nombre d'alertes isolées à analyser et d'améliorer la compréhension globale des incidents.

[Corrélation d'alertes] La corrélation d'alertes est un ensemble de techniques utilisées pour traiter les alertes générées par divers systèmes de détection d'intrusion afin d'éliminer les alertes redondantes, réduire le nombre de fausses alertes et reconstruire les scénarios d'attaque [31].

Les objectifs principaux de la corrélation sont :

— L'identification des relations entre événements ;

— la reconstitution de chaînes d'attaque ;

— la priorisation des incidents ;

— L'aide à la décision pour les équipes de sécurité ;

— la réduction de la fatigue d'alerte des analystes.

Ces objectifs soulignent que la corrélation n'est pas une simple agrégation de données, mais un processus d'inférence structuré dont l'efficacité repose directement sur la qualité

des mécanismes sous-jacents. Les paragraphes suivants décrivent les principales familles d'approches adoptées dans la littérature.

1.4.2 Approches basées sur des règles

Les approches fondées sur des règles reposent sur la définition manuelle de motifs d'événements considérés comme significatifs. Bien qu'elles soient simples à interpréter, ces méthodes souffrent d'un manque de flexibilité et nécessitent une mise à jour continue pour rester pertinentes face à l'évolution des attaques [18].

1.4.3 Approches basées sur des graphes

Les graphes d'attaque permettent de représenter les vulnérabilités, les préconditions et les conséquences des actions adverses sous forme de nœuds et d'arêtes [32]. Les événements observés peuvent alors être projetés sur ces graphes afin d'identifier des trajectoires d'attaque plausibles.

1.4.4 Approches par apprentissage automatique

Plus récemment, les techniques d'apprentissage automatique ont été utilisées pour la corrélation d'alertes, notamment via des méthodes de classification, de clustering et d'apprentissage profond [16, 17]. Ces approches offrent une meilleure capacité d'adaptation, mais posent des défis en matière d'interprétabilité et de dépendance aux données annotées.

1.4.5 Corrélation cross-domaines pour les systèmes industriels

Une avancée significative dans le domaine de la corrélation d'alertes concerne les systèmes de contrôle industriels (ICS). Koucham et al. [31] ont développé un cadre de corrélation d'alertes spécifiquement adapté aux ICS, combinant les alertes de détection d'intrusion du domaine physique avec les alertes de détection d'intrusion du domaine cyber plus classiques.

Spécificités des ICS

Dans les ICS, la présence d'un processus physique et les menaces spécifiques associées ont conduit à l'hétérogénéité des alertes due au développement de techniques de détection multi-domaines :

— domaine cyber : Les IDS du domaine cyber analysent l'activité réseau à un niveau supérieur, surveillant les adresses IP, les ports TCP/UDP, les flux réseau ou les protocoles.

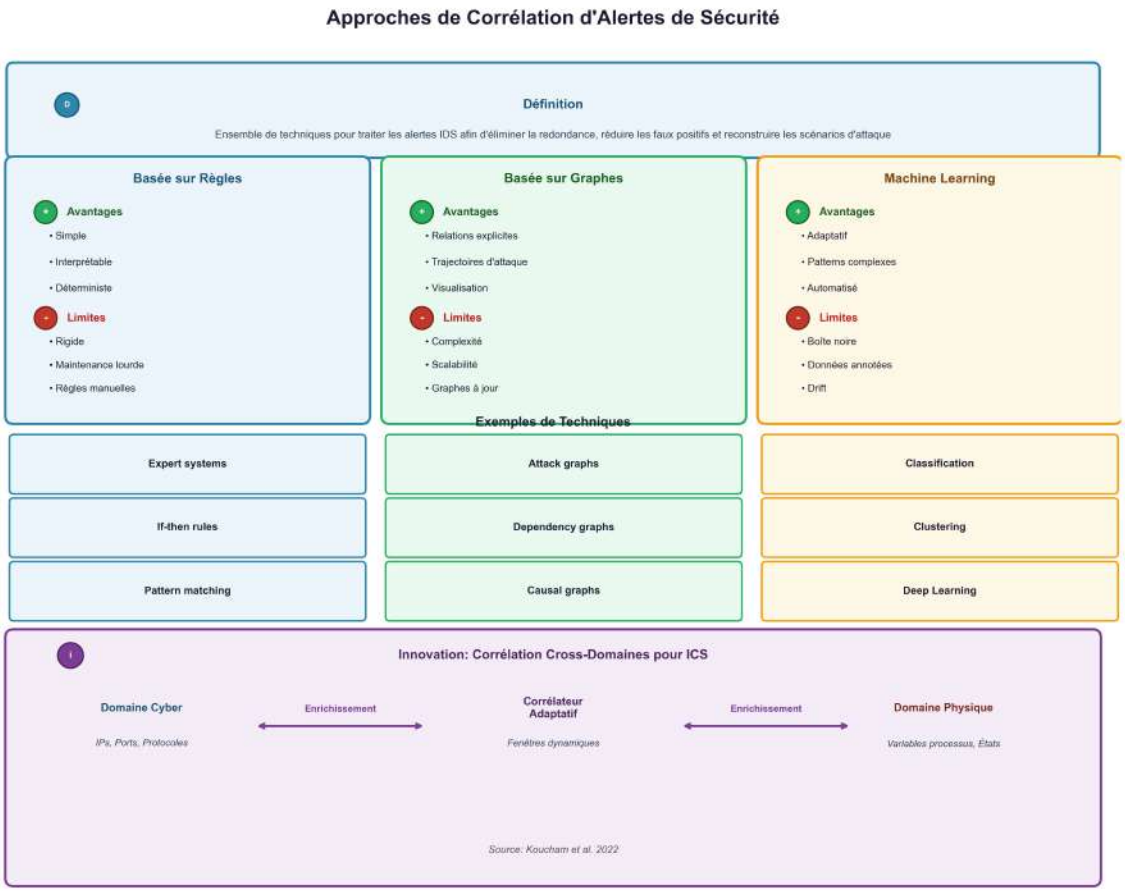


FIGURE 1.7 – Trois approches principales de corrélation d'alertes : basées sur règles, basées sur graphes et machine learning, avec leurs avantages et limitations respectives.

— domaine physique : Les IDS du domaine physique prennent en compte le processus physique que l'ICS contrôle, décidant si la valeur appliquée à une variable de processus peut mettre le processus physique dans un état critique.

Méthodologie d'enrichissement

L'approche de Koucham et al. [31] propose un enrichissement des alertes du domaine physique avec des attributs du domaine cyber, compte tenu de la connaissance des protocoles supportés par les contrôleurs et du mappage mémoire des variables de processus.

Politiques de sélection adaptatives

Contrairement aux approches classiques qui reposent sur des fenêtres temporelles fixes, cette méthodologie adopte des politiques de sélection qui s'adaptent à l'état du processus physique en ajustant dynamiquement la taille des fenêtres d'alertes en fonction de l'état des sous-processus.

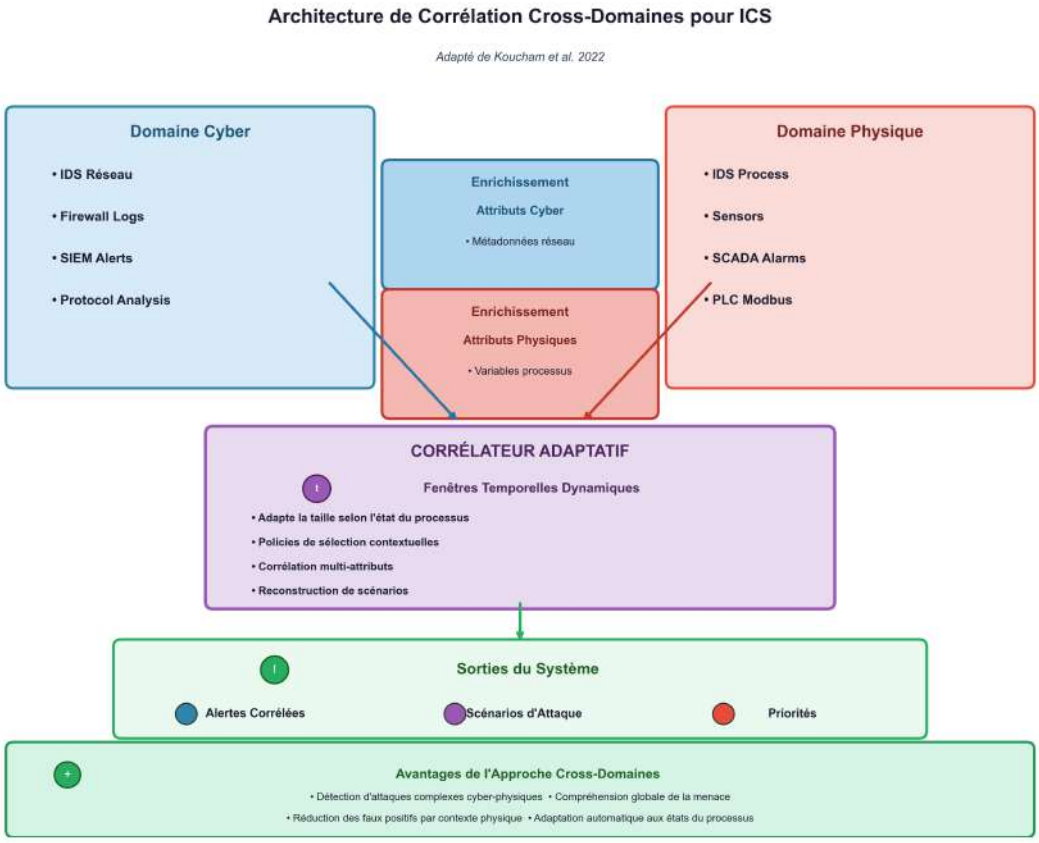


FIGURE 1.8 – Architecture de corrélation d'alertes cross-domaines pour les systèmes de contrôle industriels avec enrichissement bidirectionnel et corrélateur adaptatif.

1.5 Cadres de modélisation des attaques

La compréhension des mécanismes de corrélation ne peut se dissocier d'une connaissance approfondie des modèles conceptuels décrivant le déroulement des attaques. Cette section présente les deux cadres de référence les plus utilisés dans la communauté, qui servent également de grille d'analyse pour les contributions expérimentales de cette thèse.

1.5.1 Cyber Kill Chain

Le modèle Cyber Kill Chain, développé par Lockheed Martin en 2011 [33], décompose les cyberattaques en sept phases séquentielles, permettant aux défenseurs d'identifier et de perturber les attaques à chaque étape.

1. reconnaissance : Collecte d'informations sur la cible
2. armement (Weaponization) : Création de l'exploit et du payload
3. livraison (Delivery) : Transmission du payload à la cible
4. exploitation : Exécution du code malveillant
5. installation : Établissement de la persistance

6. commande et contrôle (C2) : Établissement d'un canal de communication
7. actions sur les objectifs : Réalisation de l'objectif final

Forces et limitations

Le Cyber Kill Chain offre un cadre systématique pour comprendre le cycle de vie d'une cyberattaque et fournit une approche structurée pour la communication avec les parties prenantes non techniques [34]. Cependant, son modèle linéaire et séquentiel peut ne pas refléter avec précision la nature complexe, itérative et souvent parallèle des cyberattaques modernes.

1.5.2 MITRE ATT&CK Framework

Le framework MITRE ATT&CK (Adversarial Tactics, Techniques, and Common Knowledge), développé par MITRE Corporation en 2013 [35], est devenu la norme de l'industrie pour comprendre et communiquer sur le fonctionnement des attaques.

Structure du framework

ATT&CK est organisé comme une matrice de tactiques et de techniques applicables à différentes plateformes (Enterprise, Mobile, ICS). Il couvre divers comportements adverses, y compris ceux utilisés après l'obtention de l'accès initial, et offre une vue détaillée de l'exécution des techniques adverses.

ATT&CK pour ICS

MITRE ATT&CK pour ICS, publié initialement en janvier 2020, est un framework séparé adapté aux environnements de systèmes de contrôle industriels, reconnaissant que les technologies employées et les motivations des attaquants diffèrent de ceux des attaquants d'entreprise traditionnels [36].

Complémentarité des frameworks

Selon la documentation officielle MITRE [35], ATT&CK et le Cyber Kill Chain sont complémentaires : « ATT&CK se situe à un niveau de définition plus bas pour décrire le comportement adversaire que le Cyber Kill Chain. Les tactiques ATT&CK ne sont pas ordonnées et peuvent ne pas toutes se produire dans une seule intrusion. »

1.6 Systèmes d'alerte précoce en cybersécurité

La maîtrise des cadres de modélisation des attaques prépare à la compréhension des systèmes d'alerte précoce, dont l'ambition est d'exploiter ces modèles pour anticiper les

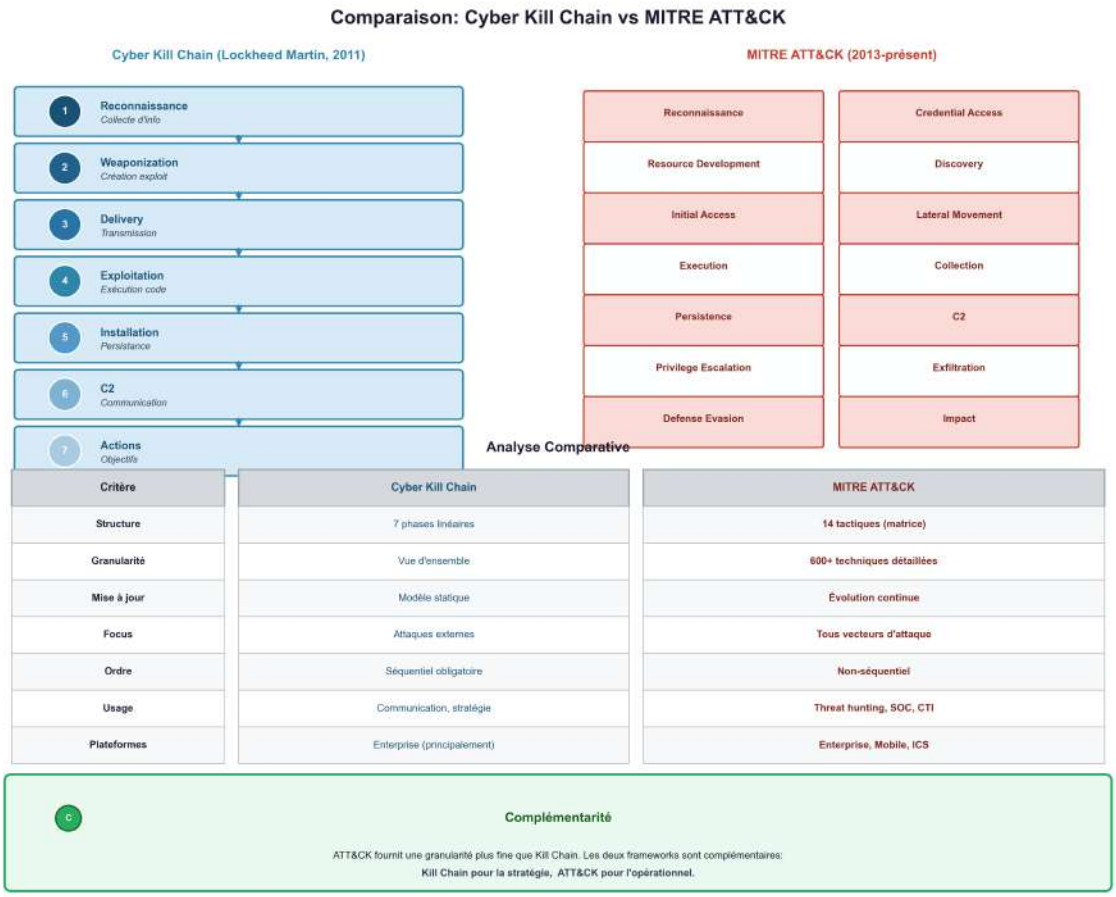


FIGURE 1.9 – Comparaison détaillée entre Cyber Kill Chain et MITRE ATT&CK Framework avec tableau comparatif des critères clés.

menaces avant qu'elles n'atteignent leur objectif. Cette section définit les principes fondateurs de ces systèmes et décrit les architectures modernes qui les caractérisent.

1.6.1 Définition et principes

Les systèmes d'alerte précoce (Early Warning Systems - EWS) ont pour objectif de détecter les signaux précurseurs d'un événement critique suffisamment tôt pour permettre la mise en place de mesures préventives [13]. Ils reposent généralement sur l'observation continue de données, l'analyse de tendances et la génération d'alertes synthétiques.

[Système d'alerte précoce cyber] Un système d'alerte précoce pour la cybersécurité nécessite un renseignement large et une détection ciblée d'indicateurs spécifiques pour inciter à une action préventive de la part des décideurs [1]. La valeur d'un tel système réside dans sa capacité à prédire les problèmes, identifier les indicateurs de menaces et offrir des preuves visuelles des tendances nécessitant une action.

1.6.2 Architecture des EWS modernes

Selon le rapport TAG Cyber [1], les systèmes d'alerte précoce modernes intègrent plusieurs composants :

1. plateformes de renseignement sur les menaces : Agrégation de signaux de menaces globaux pour prédire les attaques (ex : Arctic Security EWS, Recorded Future, Mandiant Threat Intelligence).
2. solutions SIEM et SOAR : Analyse des journaux et événements réseau pour anticiper les menaces en temps réel. Les plateformes SOAR (*Security Orchestration, Automation and Response*) complètent le SIEM en automatisant la réponse aux incidents et en orchestrant les actions correctives entre les différents outils de sécurité.
3. analyse du trafic réseau et IDS/IPS : Détection des anomalies aux périmètres réseau et segments internes.
4. technologie de déception : Déploiement de leurres (honeypots modernes) fournissant des alertes haute-fidélité.

1.6.3 Évolution vers une sécurité proactive

Le passage aux EWS marque un changement critique d'une cybersécurité réactive à proactive [30]. L'approche traditionnelle de « prévention, détection et réponse » est chronophage et inefficace face à l'évolution rapide des menaces et la volumétrie des alertes. Les EWS permettent aux organisations de :

— prédire la probabilité d'une attaque réelle ;

— utiliser des données en temps réel et historiques pour améliorer la préparation ;

— gérer proactivement les types de menaces généralement soumis au SOC ;

— réduire le temps de détection et de réponse ;

— gérer la volumétrie des alertes en priorisant celles à fort impact.

1.6.4 Technologies de déception

Les technologies de déception modernes ont évolué bien au-delà des simples serveurs honeypot [37]. Les plateformes de déception déploient de faux identifiants, fichiers et systèmes indiscernables des actifs réels, créant des « tripwires » que seul un acteur malveillant toucherait.

Lorsqu'un attaquant interagit avec un leurre (par exemple, tente d'utiliser un mot de passe honeytoken ou accède à une fausse base de données), une alerte haute-fidélité est déclenchée. Ce système d'alerte précoce peut accélérer considérablement la détection tout en détournant l'attaquant des véritables actifs critiques.

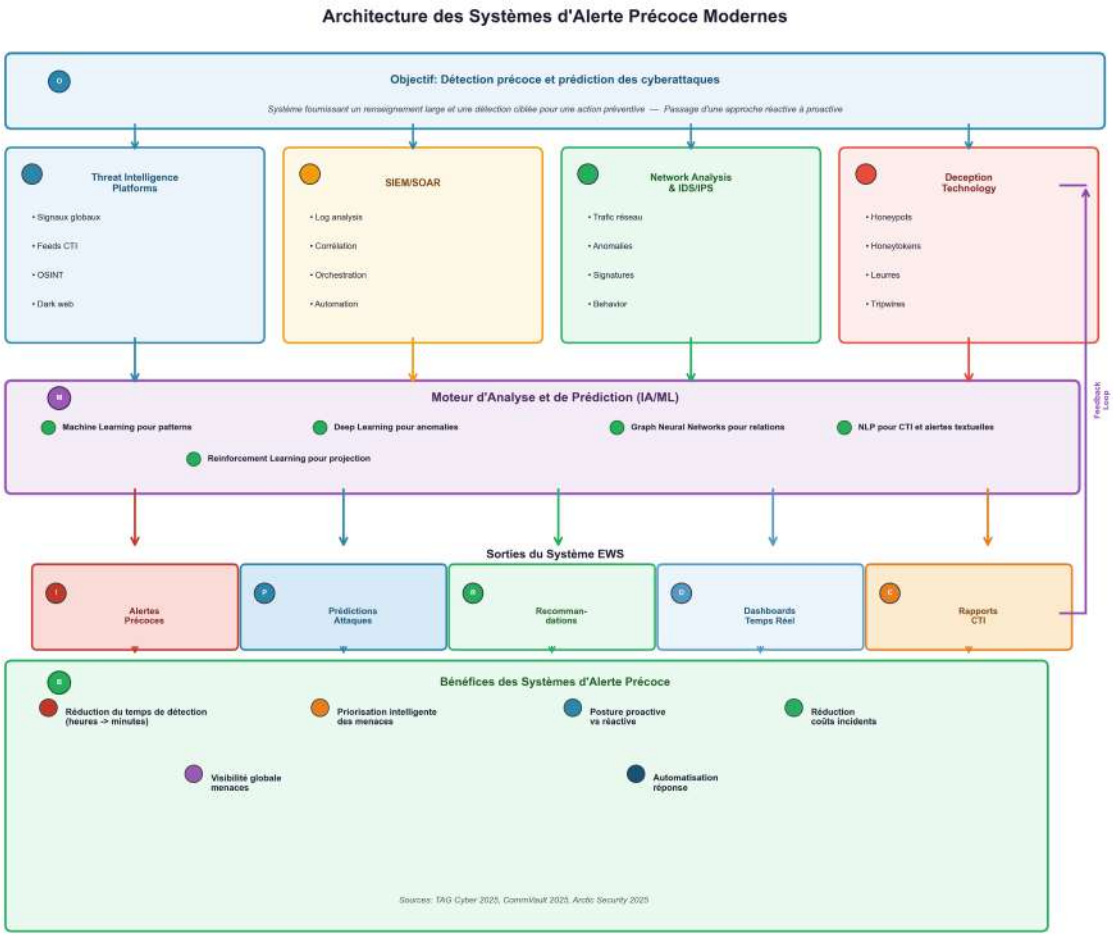


FIGURE 1.10 – Architecture complète d'un système d'alerte précoce moderne intégrant threat intelligence, SIEM/SOAR, network analysis, deception technology et moteur d'analyse IA/ML. Source : adapté de [1].

1.6.5 Applications nationales : programme EINSTEIN

Le programme EINSTEIN du National Cybersecurity Protection System (NCPS) de la CISA [29] représente une implémentation nationale d'un système d'alerte précoce. Ses objectifs incluent :

— fournir un système d'alerte précoce pour le gouvernement fédéral ;

— améliorer la conscience situationnelle des menaces d'intrusion sur les réseaux FCEB ;

— permettre l'identification en temps quasi réel des activités cyber malveillantes.

La capacité de détection d'intrusion du NCPS, délivrée via EINSTEIN, est une grille de capteurs qui surveille le trafic réseau pour détecter les activités malveillantes vers et depuis les départements et agences participants.

1.7 Prédiction et projection d'attaques

Les systèmes d'alerte précoce atteignent leur plein potentiel lorsqu'ils sont couplés à des mécanismes capables non seulement de détecter une attaque en cours, mais d'en anticiper les étapes futures. Cette section présente les approches de prédiction et de projection, qui forment le cœur algorithmique des contributions développées dans cette thèse.

1.7.1 Prédiction basée sur les données

La prédiction d'attaques repose sur l'analyse de données historiques pour estimer la probabilité d'occurrence d'événements futurs [38]. Ces approches utilisent des techniques statistiques et des modèles probabilistes.

Méthodes traditionnelles

— modèles bayésiens : Estimation des probabilités conditionnelles d'attaques basées sur des observations préalables.

— chaînes de Markov cachées (HMM) : Modélisation des transitions entre états d'attaque.

— analyse statistique : Identification de tendances et anomalies dans les données temporelles.

Méthodes par apprentissage automatique

Les approches récentes exploitent l'apprentissage automatique pour améliorer la précision des prédictions :

— réseaux de neurones récurrents (RNN/LSTM) : Capture des dépendances temporelles dans les séquences d'alertes.

— réseaux de neurones sur graphes (GNN) : Modélisation des relations structurelles entre entités du réseau.

— apprentissage par renforcement : Modélisation de la progression d'attaque comme un processus de décision.

1.7.2 Projection d'attaques multi-étapes

La projection d'attaques vise à anticiper les étapes futures d'un scénario d'attaque en fonction d'un état courant et d'un modèle du système [25, 32]. Elle joue un rôle clé dans les systèmes d'alerte précoce avancés. Contrairement à la simple détection d'anomalies, elle exploite la structure séquentielle des attaques multi-étapes pour raisonner sur leur progression probable, permettant aux équipes de sécurité d'agir de manière préventive plutôt que corrective.

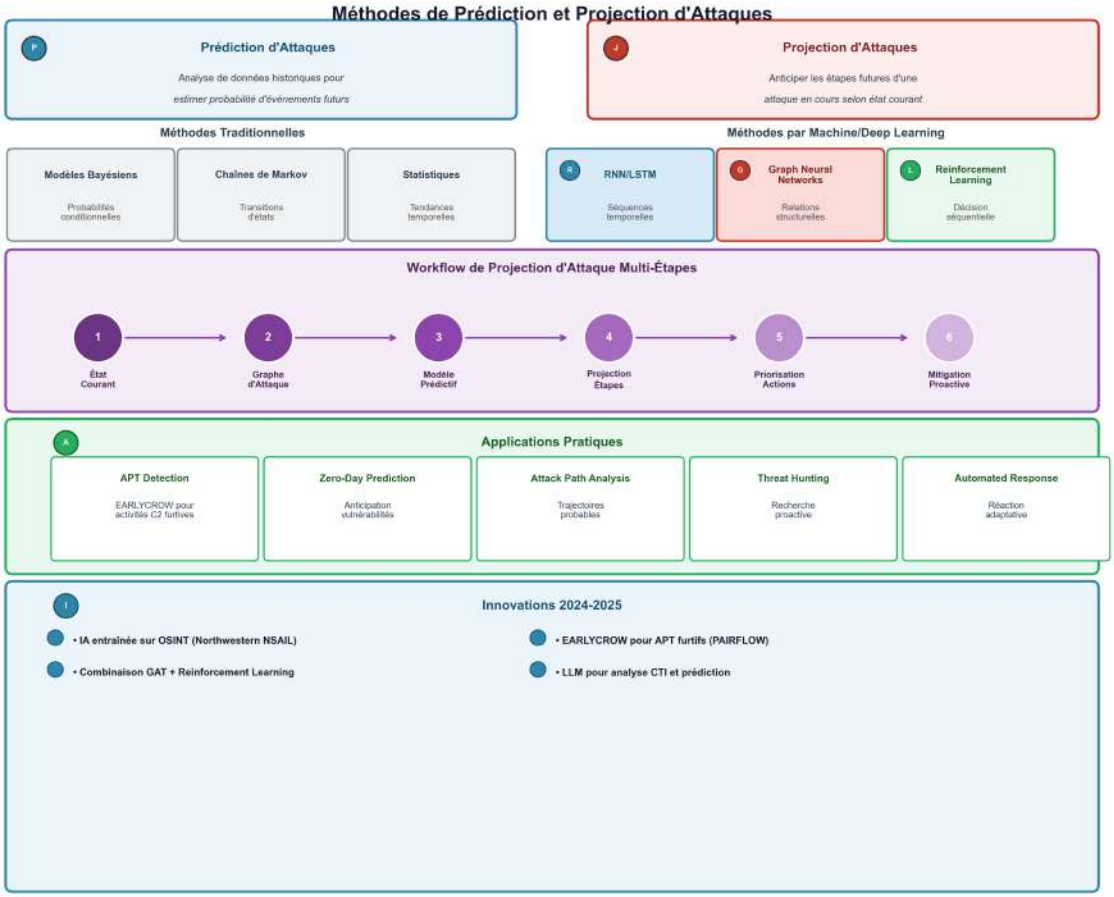


FIGURE 1.11 – Méthodes de prédiction et projection d'attaques : approches traditionnelles (Bayésiennes, Markov) et techniques modernes d'IA/ML (RNN/LSTM, GNN, Reinforcement Learning) avec workflow de projection.

Au-delà des approches probabilistes classiques, les avancées récentes en apprentissage profond ont ouvert de nouvelles voies pour la projection d'attaques, en particulier via les modèles capables de traiter les dépendances structurelles et temporelles entre événements de sécurité.

1.7.3 Intelligence artificielle pour la prédiction

Le Northwestern Security and AI Lab (NSAIL) [39] illustre l'application concrète de ces méthodes en développant des modèles d'IA entraînés sur des données open source non classifiées pour prédire les cyberattaques. Ces travaux démontrent que des modèles entraînés sur des flux hétérogènes sont capables d'identifier des précurseurs d'attaques terroristes et cyber, de détecter des deepfakes et des malwares, et de contrer des campagnes de phishing avancé en temps réel. Le système EARLYCROW, basé sur la méthode PAIRFLOW, étend ces capacités à la détection des activités APT furtives sur HTTP(S) [40]. Ces résultats confirment le potentiel de l'IA pour transformer des volumes massifs de données brutes en renseignements opérationnels actionnables.

TABLE 1.2 – Synthèse des lacunes identifiées dans les approches existantes

Domaine	**Lacunes identifiées**
Corrélation	Approches majoritairement mono-domaines, manque de prise en compte du contexte sémantique
Détection	Temps de réponse insuffisant face aux zero-days exploités en 5 jours en moyenne
Prédiction	Modèles peu adaptés aux menaces émergentes et aux attaques multi-étapes
Interprétabilité	Manque de transparence des modèles ML pour les analystes SOC

1.8 Conclusion

1.8.1 État des lieux

Les approches existantes en matière de détection, de corrélation et de prédiction d'attaques présentent des avancées significatives, mais restent insuffisantes pour faire face à la complexité croissante des cybermenaces. Le Tableau 1.2 résume les principales lacunes identifiées.

1.8.2 Positionnement des contributions

Les travaux présentés dans cette thèse s'inscrivent dans la continuité de ces recherches, en proposant des approches fondées sur :

1. MCM-CASR : Exploitation du traitement du langage naturel et de la reconnaissance d'entités nommées pour la corrélation sémantique des alertes, permettant une meilleure compréhension contextuelle des relations entre événements de sécurité.
2. MCM-Llama : Utilisation de modèles de langage spécialisés et affinés pour la détection en temps réel, offrant une capacité d'adaptation aux nouvelles menaces grâce au transfert d'apprentissage.
3. GAT-DQN : Combinaison de réseaux d'attention sur graphes et d'apprentissage par renforcement pour la projection adaptative d'attaques, permettant la sélection dynamique du meilleur modèle selon le contexte.
4. multi-Classification : Apprentissage multi-classe pour la classification avancée des alertes, adressant différentes granularités de décisions de corrélation.

1.8.3 Alignement avec les tendances actuelles

Nos contributions s'alignent avec les tendances identifiées dans les rapports récents :

— IA pour la sécurité : L'utilisation de LLM pour la corrélation d'alertes répond au besoin d'automatisation intelligente identifié par le WEF [3].

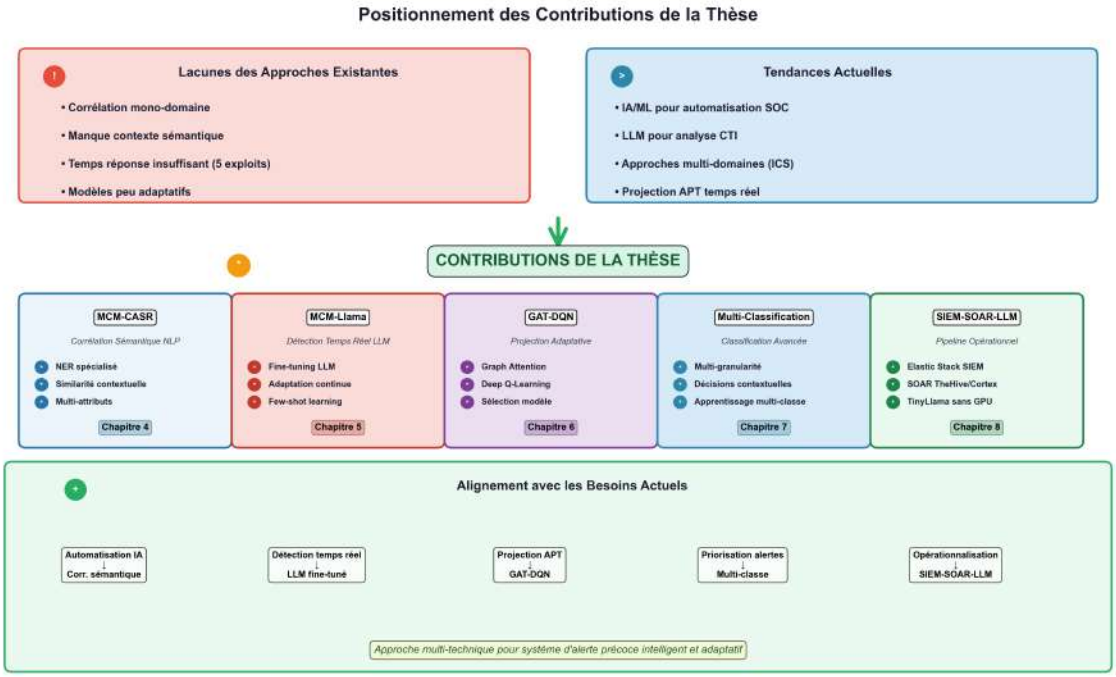


FIGURE 1.12 – Positionnement des quatre contributions de la thèse (MCM-CASR, MCM-Llama, GAT-DQN, Multi-Classification) par rapport aux lacunes identifiées et aux tendances actuelles en cybersécurité.

— temps de réponse réduit : La détection en temps réel avec MCM-Llama adresse le défi du délai d'exploitation réduit à 5 jours.

— approches multi-domaines : L'inspiration de la corrélation cross-domaines [31] informe notre approche de corrélation sémantique.

— projection d'attaques : Notre cadre GAT-DQN contribue à l'anticipation des menaces persistantes avancées.

Ces éléments constituent les fondements des contributions développées dans les chapitres suivants.

1.8.4 Analyse comparative des solutions récentes

Au-delà des lacunes génériques identifiées dans le Tableau 1.2, une analyse plus ciblée des solutions récentes permet de mieux positionner la séquence méthodologique NLP → LLM → GNN+RL adoptée dans cette thèse. Le Tableau 1.3 compare les familles d'approches représentatives de l'état de l'art selon cinq critères opérationnels critiques : la prise en compte du contexte sémantique, la capacité de traitement en temps réel, la modélisation structurelle des attaques, la projection adaptative, et la sélection dynamique de modèle.

Ce tableau illustre que les approches existantes adressent partiellement certains critères mais qu'aucune ne les couvre simultanément. Les méthodes basées sur des règles

TABLE 1.3 – Comparaison des familles d'approches selon les critères opérationnels clés

Approche	**Contexte sémantique**	**Temps réel**	**Structure graphique**	**Projection d'attaque**	**Adaptatif**
Règles / SIEM	×	∼	×	×	×
Bayésien / HMM [38]	×	∼	×	∼	×
ML classique [16]	∼	✓	×	×	×
NLP + BERT [12]	✓	∼	×	×	×
LLM fine-tuné [41]	✓	✓	×	×	×
GNN seul [42]	∼	✓	✓	∼	×
Notre système	✓	✓	✓	✓	✓

✓ = Pleinement supporté, ∼ = Support partiel, × = Non supporté

ou des modèles statistiques manquent de compréhension sémantique et ne permettent pas d'adaptation dynamique [38]. Les approches NLP avec BERT apportent la dimension sémantique mais restent limitées pour la projection structurelle des attaques [12]. Les GNN seuls capturent les relations structurelles mais n'intègrent pas de compréhension sémantique approfondie ni de mécanisme de sélection adaptative [42].

La séquence progressive proposée dans cette thèse — d'abord la reconstruction sémantique (MCM-CASR), puis l'amélioration par LLM (MCM-Llama), ensuite la projection structurelle adaptative (GAT-DQN), et enfin la classification multi-granularité — répond précisément à la nécessité d'un système couvrant l'ensemble de ces dimensions de manière intégrée. Chaque contribution surmonte les limites de la précédente, formant une chaîne de traitement cohérente dont aucune approche isolée de l'état de l'art ne propose l'équivalent.

Chapitre 2

Intelligence artificielle, NLP et modèles de langage pour la cybersécurité

2.1 Introduction

L'intelligence artificielle occupe aujourd'hui une place centrale dans le développement des technologies de cybersécurité [15, 16]. Face à la complexité croissante des attaques, à l'augmentation massive des volumes de données de sécurité et à l'exigence de réactions en temps réel, les approches classiques basées sur des règles statiques montrent rapidement leurs limites. Les techniques d'apprentissage automatique, d'apprentissage profond, de traitement du langage naturel et d'apprentissage par renforcement offrent des outils puissants pour améliorer la détection, la corrélation et la prédiction des cybermenaces [15, 17, 43].

Ce chapitre présente les principaux fondements théoriques et méthodologiques en intelligence artificielle mobilisés dans cette thèse. Il couvre successivement le machine learning, le deep learning, le traitement du langage naturel, les modèles de langage de grande taille, les réseaux de neurones sur graphes et l'apprentissage par renforcement. Une attention particulière est portée aux applications spécifiques en cybersécurité, notamment pour la détection d'intrusions, la corrélation d'alertes, l'analyse de vulnérabilités et la projection d'attaques [42, 41].

2.2 Apprentissage automatique en cybersécurité

L'apprentissage automatique constitue la première famille de techniques mobilisées dans cette thèse. Cette section en présente les principes fondamentaux, les principaux algorithmes et les défis spécifiques à leur application en cybersécurité, avant d'aborder les

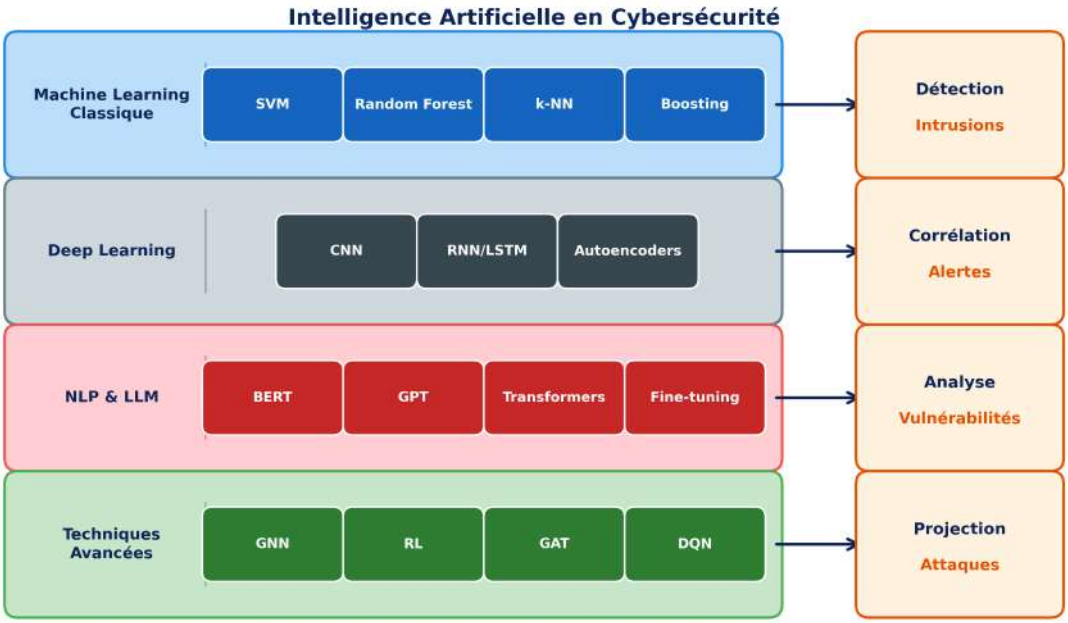


FIGURE 2.1 – Vue d'ensemble de l'intelligence artificielle appliquée à la cybersécurité : de l'apprentissage automatique classique aux modèles de langage de grande taille

approches plus avancées fondées sur l'apprentissage profond et les modèles de langage.

2.2.1 Principes généraux

L'apprentissage automatique (Machine Learning) regroupe un ensemble de méthodes permettant à un système d'extraire automatiquement des connaissances à partir de données, sans être explicitement programmé pour chaque tâche. En cybersécurité, ces méthodes sont largement utilisées pour la détection d'intrusions, la classification de comportements, l'analyse de journaux et la corrélation d'alertes [17, 16].

Les principales catégories de l'apprentissage automatique sont :

— l'apprentissage supervisé, fondé sur des données annotées ;

— l'apprentissage non supervisé, centré sur la découverte de structures latentes ;

— l'apprentissage semi-supervisé, combinant les deux approches ;

— l'apprentissage par renforcement, fondé sur l'interaction avec un environnement.

2.2.2 Algorithmes classiques

Plusieurs algorithmes de machine learning ont été utilisés pour la détection des intrusions et la classification des alertes, tels que les machines à vecteurs de support (SVM), les forêts aléatoires (Random Forest), les arbres de décision, les k-plus proches voisins (k-NN)

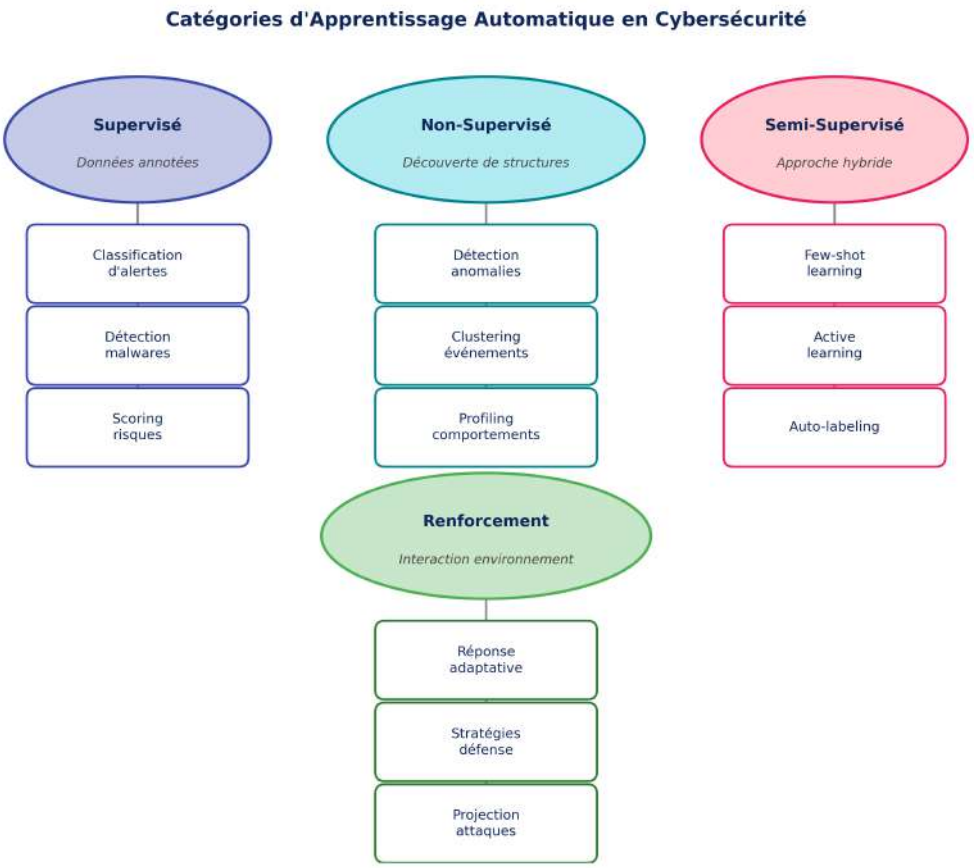


FIGURE 2.2 – Catégories d'apprentissage automatique et leurs applications en cybersécurité. Pour les approches NLP et LLM, les tâches incluent également la reclassification d'alertes, la génération de playbooks de réponse et l'automatisation de l'analyse d'incidents.

et les méthodes de boosting [17, 16]. Ces modèles offrent souvent de bonnes performances sur des données bien structurées, mais peuvent rencontrer des difficultés face à la forte hétérogénéité et au déséquilibre des classes caractéristiques des données de cybersécurité [16, 43].

2.2.3 Défis spécifiques en cybersécurité

L'application du machine learning en cybersécurité présente plusieurs défis particuliers :

— déséquilibre des classes : les événements malveillants sont généralement très minoritaires par rapport au trafic normal ;

— évolution constante des menaces : les attaquants adaptent continuellement leurs techniques pour contourner les défenses ;

— coût élevé des faux positifs : une alerte incorrecte peut mobiliser inutilement des ressources d'analyse ;

— données adversariales : les attaquants peuvent délibérément tenter de tromper les modèles de détection [15].

2.3 Apprentissage profond (Deep Learning)

Si l'apprentissage automatique classique offre des résultats solides sur des données structurées, il montre ses limites face aux représentations complexes et hétérogènes des données de sécurité. L'apprentissage profond répond à cette limite en apprenant des représentations hiérarchiques directement à partir des données brutes, ouvrant ainsi des perspectives plus puissantes pour la détection d'intrusions et la corrélation d'alertes.

2.3.1 Fondements

L'apprentissage profond repose sur l'utilisation de réseaux de neurones comportant plusieurs couches cachées, capables d'apprendre des représentations hiérarchiques des données [15]. Ces modèles ont montré des performances remarquables dans de nombreux domaines tels que la vision par ordinateur, la reconnaissance vocale et le traitement du langage naturel [44, 45].

En cybersécurité, les réseaux neuronaux profonds sont utilisés pour la détection d'anomalies, l'identification de logiciels malveillants, l'analyse de flux réseau et la corrélation avancée d'alertes [12, 16]. Leur capacité à extraire automatiquement des caractéristiques pertinentes à partir de données brutes en fait des outils particulièrement adaptés à l'analyse de grandes quantités de données de sécurité hétérogènes.

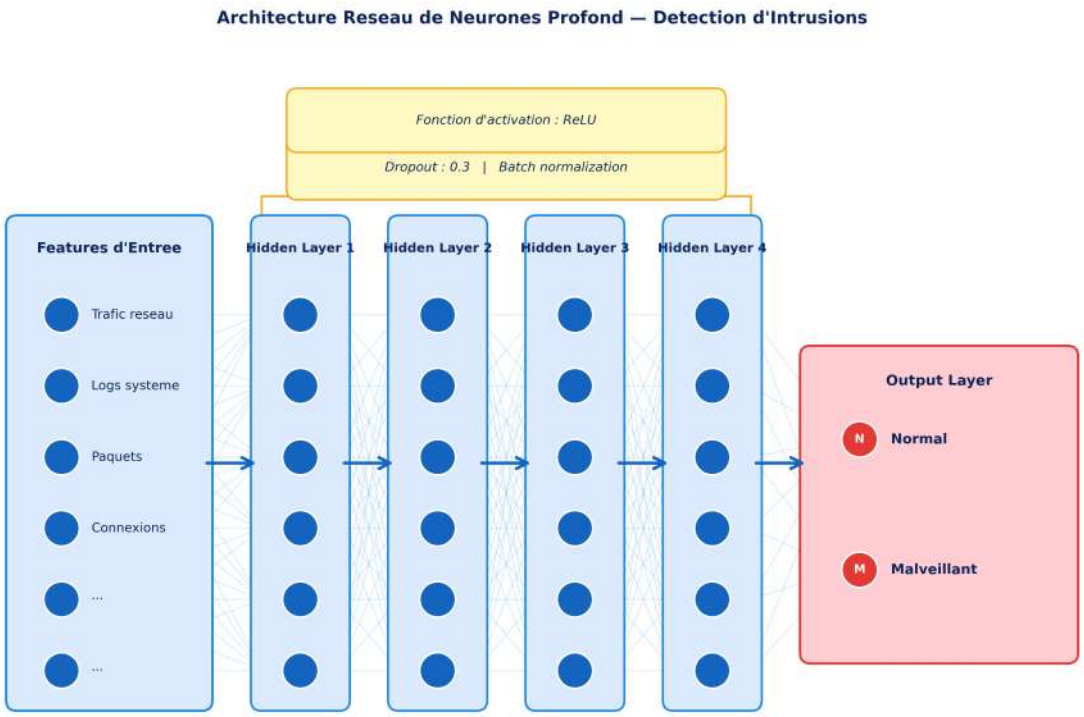


FIGURE 2.3 – Architecture typique d'un réseau de neurones profond pour la détection d'intrusions

Les architectures de deep learning se déclinent selon les propriétés des données à analyser. Selon que les données de sécurité présentent une structure spatiale, temporelle ou séquentielle, différentes familles de réseaux s'avèrent plus adaptées.

2.3.2 Réseaux convolutifs et récurrents

Les réseaux de neurones convolutifs (CNN) sont principalement utilisés pour extraire des motifs locaux dans des données structurées, tandis que les réseaux récurrents (RNN), notamment les LSTM (Long Short-Term Memory) et GRU (Gated Recurrent Unit), sont adaptés à la modélisation de séquences temporelles. Ces architectures sont particulièrement pertinentes pour analyser les journaux de sécurité, les flux réseau et les séries temporelles d'événements [16].

Les LSTM, en particulier, permettent de capturer des dépendances à long terme dans les séquences d'alertes, facilitant ainsi l'identification de campagnes d'attaques complexes qui se déploient sur plusieurs jours ou semaines [43].

2.4 Traitement du langage naturel pour la cybersécurité

Les données de cybersécurité — alertes IDS, rapports d'incidents, descriptions de vulnérabilités — sont majoritairement textuelles. Le traitement du langage naturel (NLP) constitue donc un axe central de cette thèse, permettant d'exploiter la sémantique de ces données pour améliorer la corrélation et la détection des menaces.

2.4.1 Principes du NLP

Le traitement automatique du langage naturel (Natural Language Processing – NLP) regroupe l'ensemble des techniques permettant à une machine d'analyser, comprendre et générer du texte. En cybersécurité, de nombreuses données (alertes, journaux, rapports d'incidents, descriptions de vulnérabilités) sont de nature textuelle, ce qui rend le NLP particulièrement pertinent [12, 41].

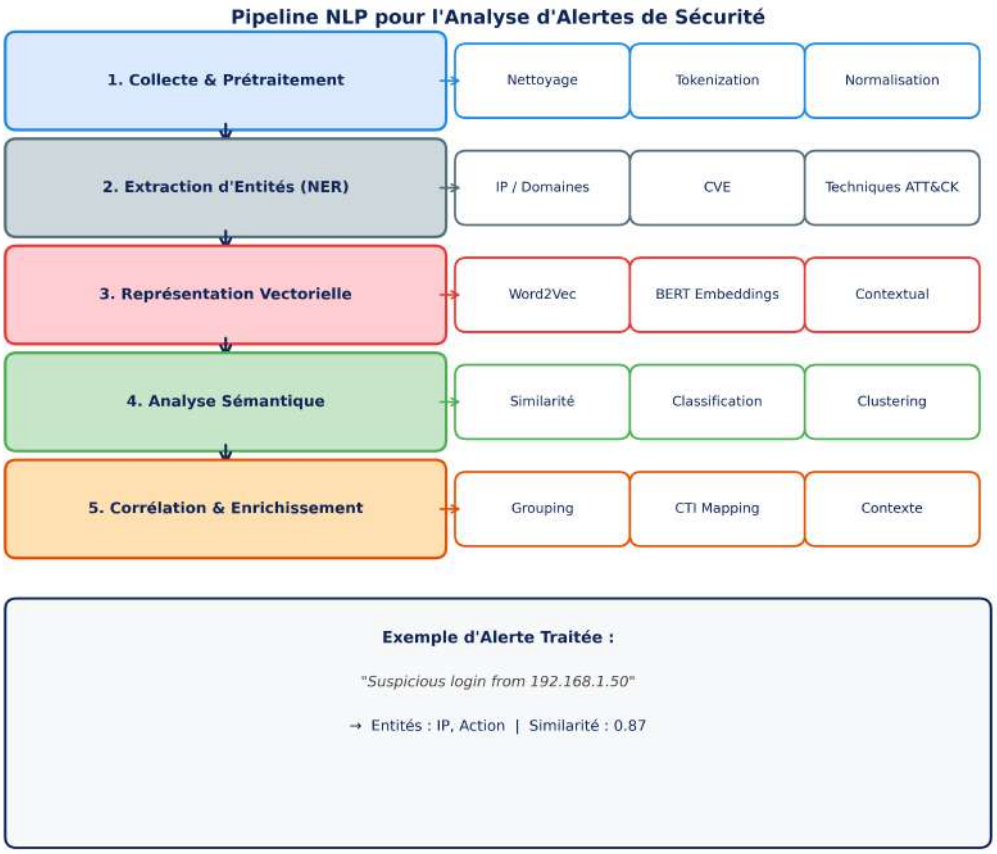


FIGURE 2.4 – Pipeline de traitement NLP pour l'analyse d'alertes de sécurité

Les techniques de NLP permettent notamment d'extraire automatiquement des entités d'intérêt telles que des adresses IP, des noms de domaine ou des types d'attaques, de classifier les alertes selon leur criticité, de regrouper des événements similaires pour

réduire la charge cognitive des analystes et d'enrichir les alertes avec des informations contextuelles provenant de bases de connaissances [12].

2.4.2 Reconnaissance d'entités nommées

La reconnaissance d'entités nommées (Named Entity Recognition – NER) permet d'identifier et de classifier automatiquement des éléments clés dans un texte, tels que des adresses IP, des noms de services, des types d'attaques ou des vulnérabilités. Cette étape est cruciale pour structurer l'information contenue dans les alertes et faciliter leur corrélation sémantique [46].

Dans le contexte cybersécurité, le NER doit être adapté pour reconnaître des entités spécifiques au domaine :

— indicateurs techniques : adresses IP, URLs, hashes de fichiers, noms de domaine ;

— vulnérabilités : identifiants CVE, versions de logiciels affectés ;

— **Tactiques et techniques** : références au framework MITRE ATT&CK ;

— acteurs malveillants : noms de groupes APT, familles de malwares.

2.4.3 Similarité sémantique

Les mesures de similarité sémantique permettent d'évaluer le degré de proximité entre deux textes en fonction de leur signification. Elles sont utilisées pour regrouper des alertes décrivant des événements similaires et pour reconstruire des scénarios d'attaque cohérents à partir de données textuelles hétérogènes [47].

Plusieurs approches existent pour calculer la similarité sémantique :

— approches basées sur les embeddings : utilisation de modèles pré-entraînés (Word2Vec, GloVe, BERT) pour représenter les textes dans un espace vectoriel ;

— mesures de distance : cosinus, distance euclidienne, distance de Jaccard ;

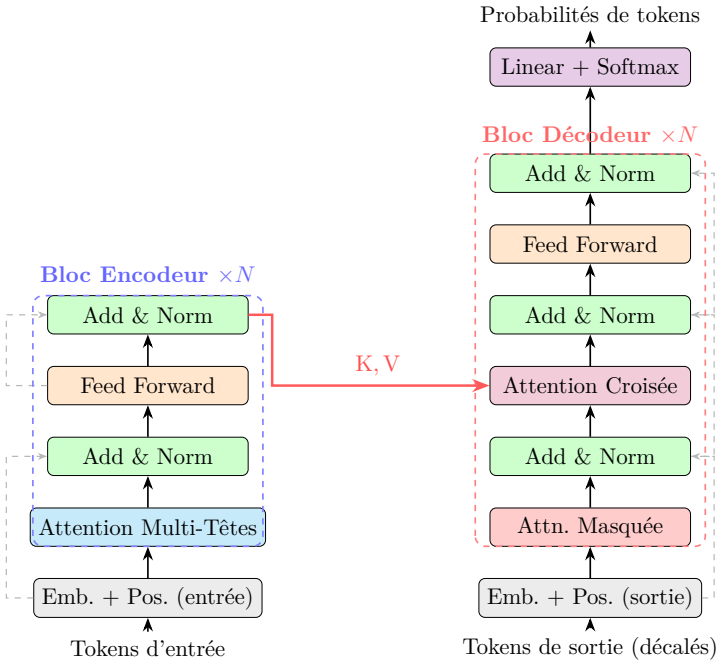
— méthodes contextuelles : prise en compte du contexte organisationnel et temporel des alertes.

2.5 Modèles de langage de grande taille

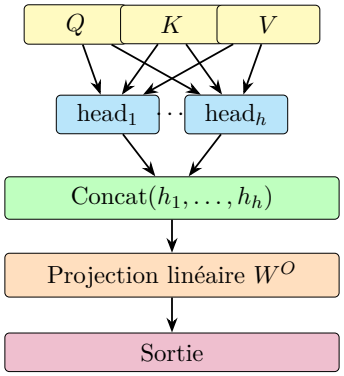
Les limitations des approches NLP classiques — notamment leur dépendance à des représentations figées et leur manque de compréhension contextuelle profonde — ont motivé l'émergence des modèles de langage de grande taille. Cette section présente leurs fondements, les stratégies d'adaptation au domaine cybersécurité, et les principales applications pratiques identifiées dans la littérature.

2.5.1 Principe des modèles de langage de grande taille (LLM)

Les modèles de langage de grande taille (LLM) sont des architectures de type Transformer entraînées sur de très grands corpus de données textuelles. Ils sont capables de comprendre le contexte, de générer du texte et de réaliser des tâches complexes comme la classification, le résumé ou la traduction [12, 45, 44].



Détail : Mécanisme d'Attention Multi-Têtes



$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

FIGURE 2.5 – Architecture Transformer et mécanisme d'attention multi-têtes. *Haut gauche* : bloc encodeur avec auto-attention et connexions résiduelles. *Haut droite* : bloc décodeur avec attention masquée, attention croisée (K, V de l'encodeur) et projection finale vers les probabilités de tokens. *Bas* : détail du mécanisme d'attention multi-têtes (matrices Q, K, V et concaténation des têtes).

Dans le domaine de la cybersécurité, ces modèles ouvrent de nouvelles perspectives pour l'analyse avancée d'alertes, la corrélation d'événements, l'assistance aux analystes et

la détection de menaces en temps réel [41, 45]. Les LLM présentent plusieurs avantages par rapport aux approches traditionnelles : leur compréhension contextuelle leur permet d'interpréter le sens global d'une alerte plutôt que de simples mots-clés ; leur capacité d'apprentissage few-shot facilite l'adaptation à de nouvelles tâches avec peu d'exemples [48] ; leur aptitude à générer du texte cohérent permet l'assistance à la documentation des incidents ; enfin, leurs capacités d'analyse sémantique sophistiquée les rendent particulièrement efficaces pour la détection de phishing et l'analyse d'emails suspects [41].

2.5.2 Adaptation des LLM à la cybersécurité

Les LLM généralistes peuvent être adaptés à la cybersécurité par des techniques de fine-tuning sur des données spécifiques (journaux, rapports d'incidents, bases de vulnérabilités). Cette spécialisation permet d'améliorer leur pertinence pour les tâches de détection de menaces et de corrélation d'alertes [49, 50, 51].

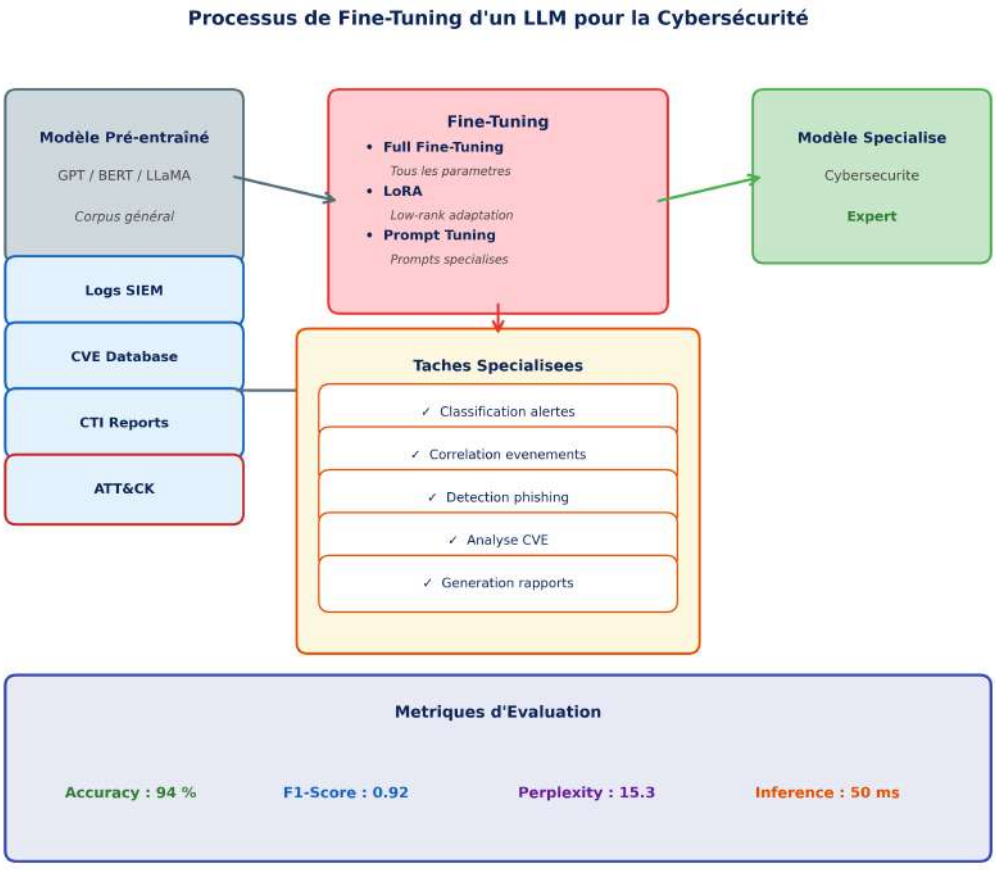


FIGURE 2.6 – Processus de fine-tuning d'un LLM pour la cybersécurité

Plusieurs stratégies de fine-tuning peuvent être employées :

— fine-tuning complet : ajustement de tous les paramètres du modèle sur un corpus cybersécurité ;

— loRA (Low-Rank Adaptation) : modification efficace de sous-ensembles de paramètres pour réduire les coûts computationnels [51] ;

— prompt engineering : conception de prompts spécialisés pour guider le modèle sans modification des poids ;

— instruction tuning : entraînement sur des instructions explicites pour améliorer la capacité à suivre des consignes complexes [52, 53] ;

— retrieval-Augmented Generation (RAG) : augmentation des capacités du modèle par récupération dynamique de connaissances pertinentes depuis des bases de données spécialisées, sans nécessiter de réentraînement [54].

Ces stratégies d'adaptation permettent de concilier la richesse des représentations apprises lors du pré-entraînement et les exigences spécifiques du domaine cybersécurité, selon les contraintes computationnelles et la quantité de données annotées disponibles.

2.5.3 Applications pratiques des LLM en cybersécurité

Les LLM trouvent de nombreuses applications concrètes en cybersécurité [41]. En matière d'analyse de vulnérabilités, ils permettent l'extraction et la synthèse automatique d'informations à partir de bulletins de sécurité, ainsi que l'identification de patterns de vulnérabilités [45, 15]. Pour la détection du phishing, ces modèles exploitent l'analyse sémantique des emails afin d'identifier les tentatives d'hameçonnage sophistiquées [26, 12]. Ils facilitent également la génération de règles SIEM en assistant la création de règles de détection à partir de descriptions en langage naturel [41]. Par ailleurs, leur capacité de compréhension sémantique est mise à profit pour la corrélation d'alertes, permettant un regroupement intelligent basé sur le sens des descriptions [6, 7]. Enfin, les LLM contribuent à l'assistance aux analystes SOC en générant automatiquement des recommandations et des procédures de réponse aux incidents [15]. Si les LLM offrent des capacités remarquables pour le traitement du langage et la corrélation sémantique, ils restent toutefois limités lorsque les données présentent une structure relationnelle complexe, telle que les graphes d'attaque ou les dépendances entre entités réseau. C'est pourquoi la section suivante s'intéresse aux réseaux de neurones sur graphes, mieux adaptés à l'exploitation de ce type de structure.

2.6 Réseaux de neurones sur graphes

Les réseaux informatiques, les graphes d'attaque et les relations entre alertes de sécurité présentent naturellement une structure de graphe. Les réseaux de neurones sur graphes permettent d'exploiter cette structure pour capturer des dépendances que les modèles séquentiels ou vectoriels ne peuvent pas représenter efficacement.

2.6.1 Principe des Graph Neural Networks

Les réseaux de neurones sur graphes (Graph Neural Networks – GNN) permettent de traiter des données structurées sous forme de graphes, en tenant compte des relations entre les entités. Ils sont particulièrement adaptés à la modélisation des réseaux de communication, des dépendances entre services et des graphes d'attaque [32, 55, 42].

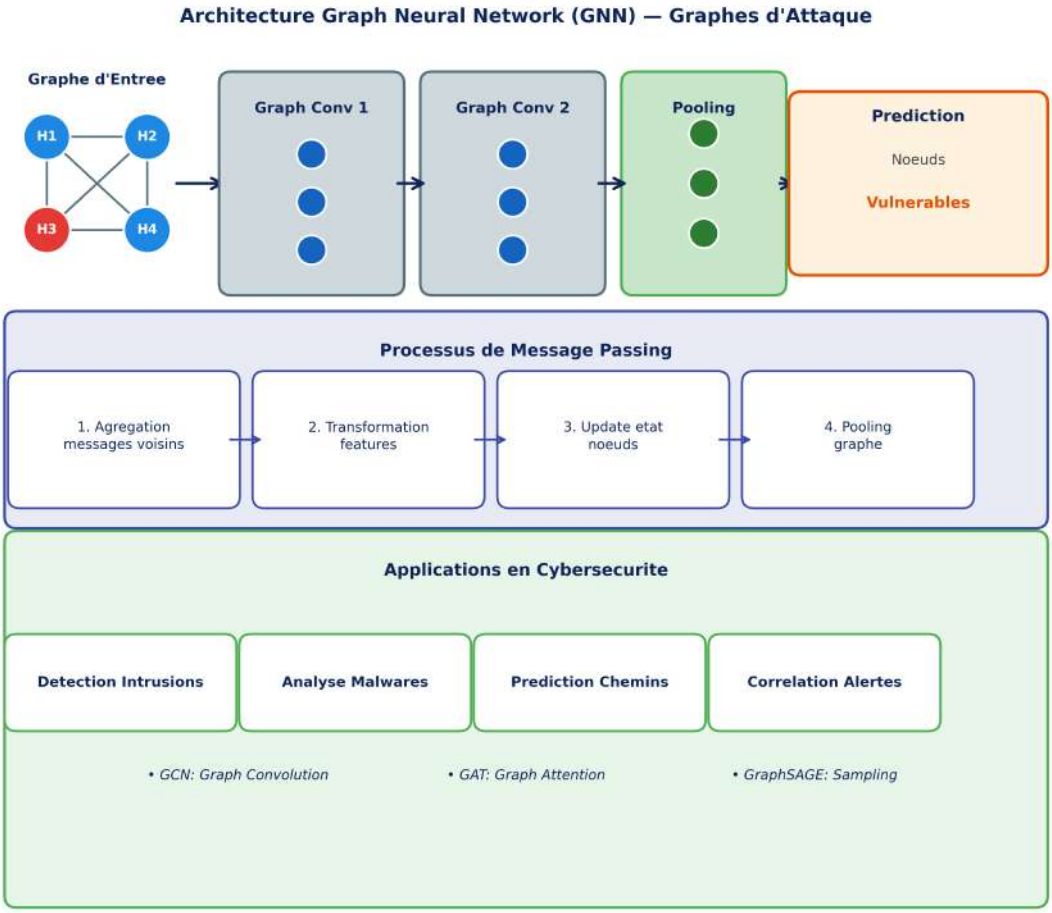


FIGURE 2.7 – Architecture d'un réseau de neurones sur graphe pour l'analyse de graphes d'attaque

Les GNN fonctionnent par propagation d'informations entre nœuds voisins, permettant ainsi de capturer les dépendances structurelles dans le graphe. Cette propriété est essentielle pour modéliser les relations complexes dans les systèmes cyber, où l'état d'un composant peut influencer l'état de ses voisins [56].

2.6.2 Applications des GNN en opérations cyber-défensives

Les GNN trouvent plusieurs applications en cybersécurité défensive [42, 56] :

— détection d'intrusions basée sur graphes : modélisation du trafic réseau comme un graphe pour identifier des patterns anormaux ;

— analyse de malwares : représentation des relations entre fonctions dans un exécutable pour classifier les logiciels malveillants [57] ;

— prédiction de chemins d'attaque : identification des trajectoires les plus probables qu'un attaquant pourrait emprunter dans un réseau ;

— corrélation d'alertes : construction de graphes d'événements pour reconstruire des scénarios d'attaque complets [58, 59].

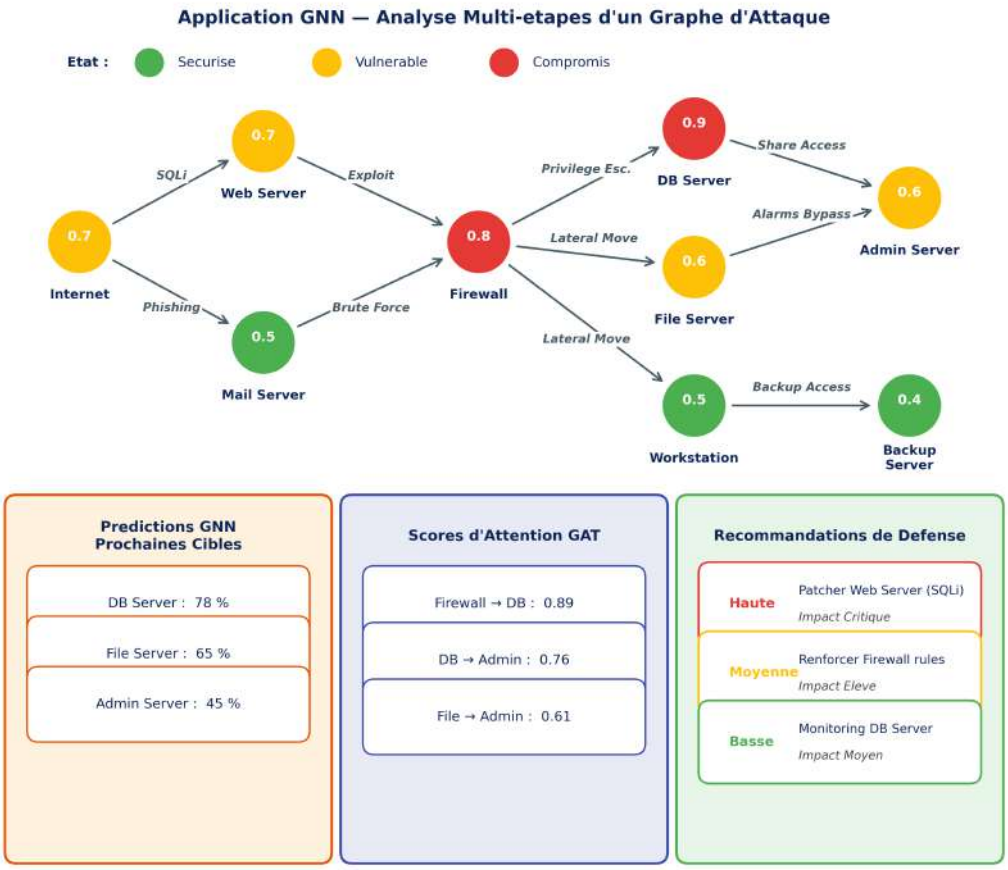


FIGURE 2.8 – Exemple d'application des GNN pour l'analyse d'un graphe d'attaque multi-étapes

2.6.3 Graph Attention Networks

Les Graph Attention Networks (GAT) introduisent des mécanismes d'attention pour pondérer l'influence des nœuds voisins lors de la propagation de l'information [60]. Cette propriété est utile pour mettre en évidence les chemins d'attaque les plus critiques dans un graphe d'attaque.

Le mécanisme d'attention permet au modèle d'apprendre quelles relations dans le graphe sont les plus pertinentes pour une tâche donnée. En cybersécurité, cela se traduit par l'identification automatique des nœuds les plus vulnérables dans un réseau, la priori-

sation des alertes selon leur position dans le graphe d'attaque et l'adaptation dynamique aux changements de topologie du réseau.

2.6.4 Guide pratique pour l'utilisation des GNN

L'implémentation de GNN pour la cybersécurité nécessite plusieurs étapes clés [56] :

1. représentation des données en graphe : définir les nœuds (hôtes, utilisateurs, services) et les arêtes (communications, dépendances) ;
2. extraction de features : associer à chaque nœud et arête des attributs pertinents (trafic, vulnérabilités, comportements) ;
3. choix de l'architecture : sélectionner le type de GNN adapté (GCN, GAT, GraphSAGE) selon la tâche ;
4. entraînement et validation : utiliser des graphes d'attaques connus pour l'apprentissage supervisé ;
5. déploiement et monitoring : intégrer le modèle dans le pipeline de détection et suivre ses performances.

Là où les GNN modélisent les relations structurelles statiques, l'apprentissage par renforcement permet de raisonner sur des séquences de décisions dans un environnement dynamique. Cette propriété est particulièrement précieuse pour la projection d'attaques multi-étapes, où l'objectif est de prédire les actions futures d'un attaquant en tenant compte de l'évolution du contexte.

2.7 Apprentissage par renforcement pour la cybersécurité

2.7.1 Principes de l'apprentissage par renforcement

L'apprentissage par renforcement (Reinforcement Learning – RL) repose sur l'interaction entre un agent et un environnement. L'agent apprend une politique d'action en cherchant à maximiser une récompense cumulée. Ce cadre est bien adapté aux problèmes de prise de décision séquentielle [61, 62].

Les composants principaux d'un système de RL sont :

— agent : le système de décision qui choisit les actions ;

— environnement : le système cyber à protéger ou à attaquer (dans le cas de simulations) ;

— état : la configuration actuelle du système (alertes actives, vulnérabilités connues, etc.) ;

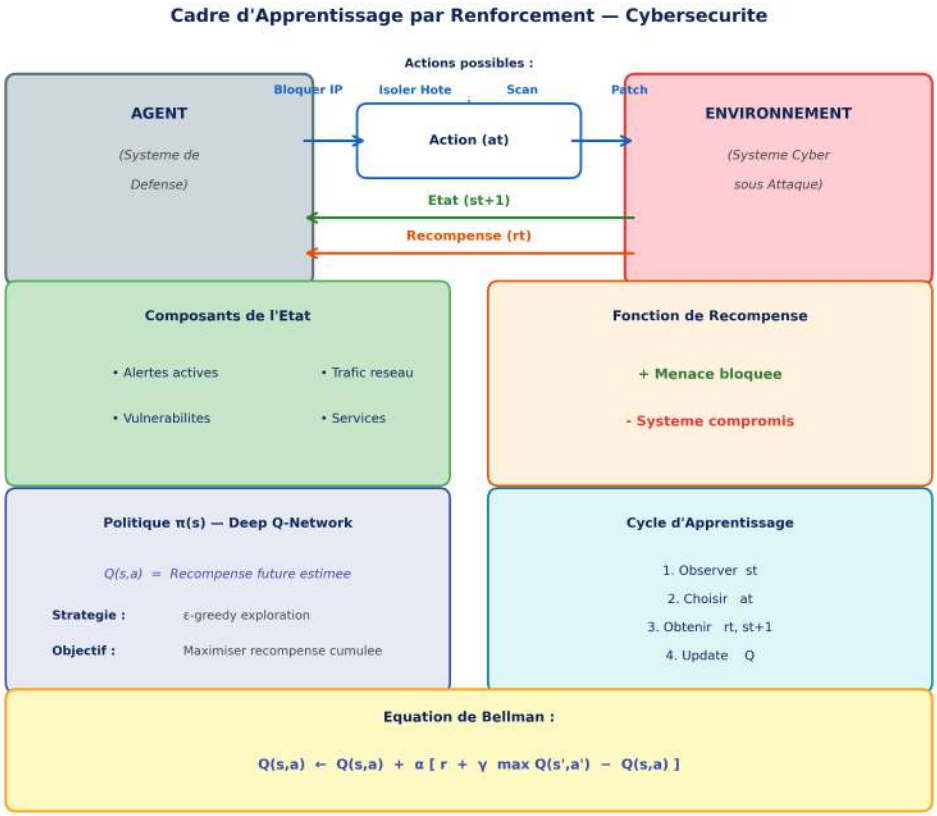


FIGURE 2.9 – Cadre général de l'apprentissage par renforcement appliqué à la cybersécurité

— action : les décisions possibles (bloquer un flux, isoler un hôte, lancer une analyse) ;

— récompense : signal indiquant la qualité de l'action (réduction du risque, impact opérationnel).

2.7.2 Applications à la défense et à la projection d'attaques

En cybersécurité, l'apprentissage par renforcement est utilisé pour la sélection dynamique de stratégies de défense, la réponse automatique aux incidents et la projection adaptative des attaques. Il permet de modéliser l'interaction entre l'attaquant et le défenseur dans un cadre formel et d'optimiser les décisions de protection en fonction du contexte [25, 63, 64].

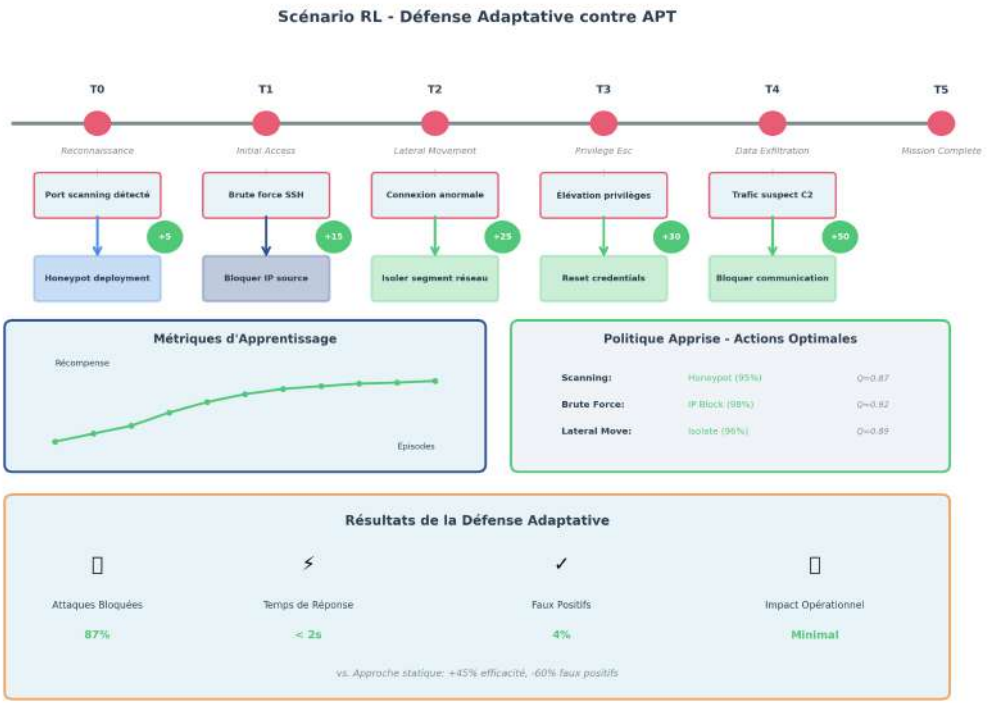


FIGURE 2.10 – Scénario d'application du RL pour la défense adaptative contre les APT

Les applications principales incluent :

— réponse automatique aux incidents : apprentissage de politiques de réaction optimales face à différents types d'attaques ;

— configuration adaptative de honeypots : ajustement dynamique des leurres pour maximiser la collecte d'informations sur les attaquants ;

— projection d'attaques APT : simulation du comportement d'attaquants sophistiqués pour anticiper leurs prochaines actions [63] ;

— optimisation de la posture de sécurité : allocation optimale des ressources de défense en fonction des menaces observées.

Les applications présentées ci-dessus reposent sur des algorithmes de RL relativement simples. Cependant, lorsque l'espace des états devient très grand — comme c'est le cas dans les environnements cyber complexes — des approches plus avancées combinant RL et réseaux de neurones profonds sont nécessaires, à l'image des deep Q-networks présentés dans la sous-section suivante.

2.7.3 Deep Q-Networks et variantes

Les deep Q-networks (DQN) combinent l'apprentissage par renforcement avec les réseaux de neurones profonds pour gérer des espaces d'états de grande dimension [62, 65]. Cette approche est particulièrement adaptée aux environnements cyber complexes où l'espace des états possibles est très vaste. Le DQN de base a donné lieu à plusieurs variantes améliorant sa stabilité et ses performances. Le Double DQN [65] réduit le biais d'estimation de la Q-valeur grâce à une séparation des réseaux d'évaluation et de sélection d'actions, produisant des politiques plus stables. Le Dueling DQN [66] décompose la Q-valeur en une valeur d'état et un avantage par action, ce qui améliore l'efficacité de l'apprentissage dans les états où le choix d'action a peu d'influence. La technique de Prioritized Experience Replay [67] complète ces améliorations en priorisant l'apprentissage sur les transitions les plus informatives.

2.8 Conclusion

Les différentes techniques présentées dans ce chapitre constituent le socle méthodologique des contributions développées dans cette thèse. Le tableau 2.1 résume les principales méthodes d'IA utilisées et leurs applications spécifiques.

TABLE 2.1 – Synthèse des techniques d'IA et leurs applications en cybersécurité dans cette thèse

Technique	**Application**	**Chapitre**
Machine Learning classique	Classification et clustering d'alertes	Chapitre 4
Deep Learning (LSTM)	Analyse séquentielle d'alertes	Chapitres 5, 6
NLP et NER	Extraction d'entités, similarité sémantique	Chapitre 5
LLM (fine-tuning)	Corrélation sémantique en temps réel	Chapitre 6
GNN (GAT)	Modélisation de graphes d'attaque	Chapitre 7
Apprentissage par renforcement (DQN)	Projection adaptative d'attaques	Chapitre 7

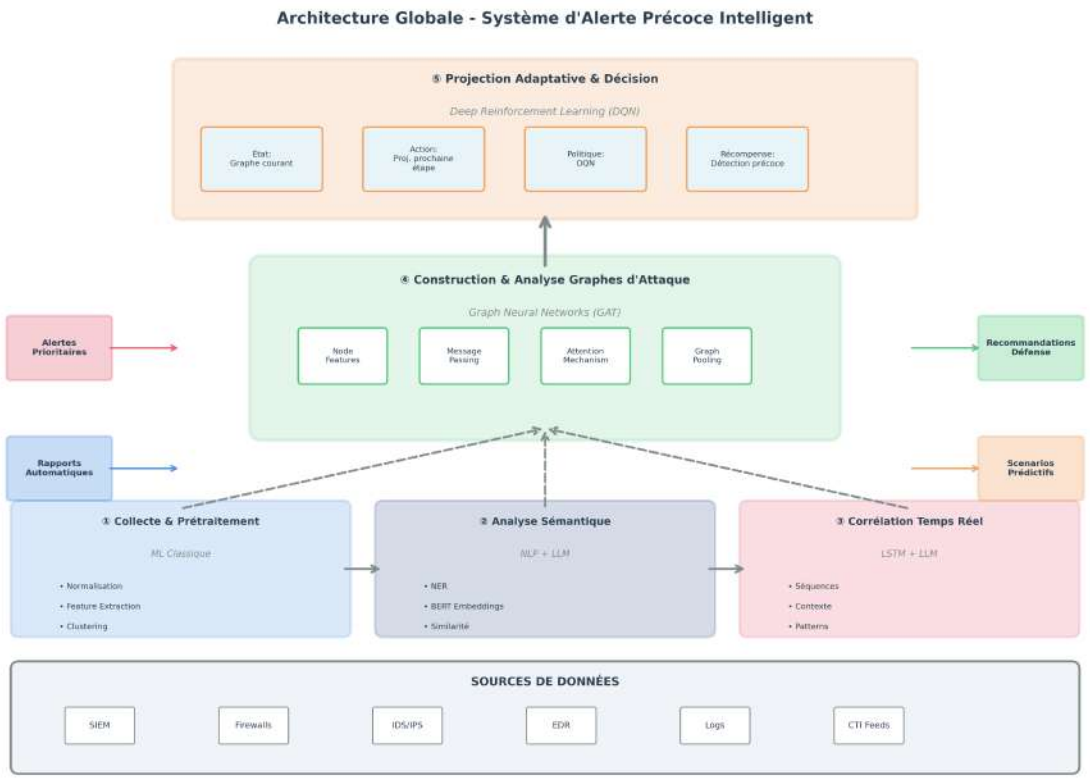


FIGURE 2.11 – Intégration des différentes techniques d'IA dans l'architecture globale du système d'alerte précoce

Le machine learning et l'apprentissage profond sont mobilisés pour la classification et la corrélation des alertes (Chapitres 4 et 5). Le traitement du langage naturel et les modèles de langage de grande taille sont utilisés pour l'analyse sémantique et la détection en temps réel (Chapitres 5 et 6) [6, 7]. Les réseaux de neurones sur graphes et l'apprentissage par renforcement sont exploités pour la projection adaptative des attaques (Chapitre 7).

Cette approche multi-technique permet de construire un système d'alerte précoce intelligent et adaptatif, capable de :

— traiter simultanément des données hétérogènes (numériques et textuelles) ;

— s'adapter automatiquement à l'évolution des menaces ;

— anticiper les prochaines étapes d'une campagne d'attaque ;

— assister les analystes SOC dans la prise de décision.

2.8.1 Modèles spécifiques mobilisés dans les contributions

Les sections précédentes ont présenté les grandes familles de techniques d'IA. Il convient ici de décrire le fonctionnement théorique des modèles précis utilisés dans les quatre contributions de cette thèse, afin d'en faciliter la compréhension lors de leur mise en œuvre détaillée.

BERT (Bidirectional Encoder Representations from Transformers). BERT [12] est un modèle de langage basé sur l'architecture Transformer entraîné de manière bidirectionnelle, c'est-à-dire en tenant compte du contexte à gauche et à droite de chaque token. Contrairement aux modèles auto-régressifs qui lisent le texte séquentiellement, BERT exploite un mécanisme d'attention multi-têtes sur l'intégralité de la séquence, produisant des représentations vectorielles contextuelles pour chaque token. La similarité entre deux textes est calculée via la distance cosinus entre leurs représentations dans l'espace d'embedding :

$$\text{sim}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\|\|\mathbf{v}\|} \tag{2.1}$$

Dans MCM-CASR, BERT est utilisé pour encoder les alertes IDS en vecteurs denses permettant le calcul de similarité sémantique entre événements de sécurité [12].

LLaMA et mécanisme LoRA. LLaMA (Large Language Model Meta AI) [49, 50] est une famille de modèles de type décodeur-Transformer, variant de 7B à 70B paramètres. Son adaptation à une tâche spécifique par fine-tuning complet est coûteuse. La technique LoRA (Low-Rank Adaptation) [51] résout ce problème en décomposant la mise à jour des poids $W \in \mathbb{R}^{d \times k}$ sous la forme d'un produit de deux matrices de faible rang :

$$W' = W + \Delta W = W + BA, \quad B \in \mathbb{R}^{d \times r},\ A \in \mathbb{R}^{r \times k},\ r \ll \min(d, k) \tag{2.2}$$

Seules les matrices A et B sont entraînées, ce qui réduit drastiquement le nombre de paramètres optimisés. Dans MCM-Llama, cette technique permet d'affiner LLaMA sur des paires d'alertes de sécurité annotées avec un coût computationnel acceptable.

Graph Attention Network (GAT). Le GAT [42] étend les GNN en pondérant les contributions des nœuds voisins par un coefficient d'attention appris. Pour un nœud i et ses voisins $\mathcal{N}(i)$, la représentation mise à jour est :

$$\mathbf{h}'_i = \sigma\left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij}\, W\mathbf{h}_j\right), \quad \alpha_{ij} = \frac{\exp(\text{LeakyReLU}(\mathbf{a}^T[W\mathbf{h}_i \| W\mathbf{h}_j]))}{\sum_{k \in \mathcal{N}(i)} \exp(\text{LeakyReLU}(\mathbf{a}^T[W\mathbf{h}_i \| W\mathbf{h}_k]))}, \tag{2.3}$$

où α_{ij} est le coefficient d'attention entre les nœuds i et j. Dans GAT-DQN, ce mécanisme capture les relations structurelles entre alertes corrélées, permettant de modéliser la progression d'une attaque comme un graphe évolutif.

Deep Q-Network (DQN). Le DQN [62] est un algorithme d'apprentissage par renforcement qui approxime la fonction valeur-action $Q(s, a)$ par un réseau de neurones profond.

La mise à jour des paramètres θ minimise l'erreur de Bellman :

$$\mathcal{L}(\theta) = \mathbb{E}_{(s,a,r,s')}\left[\left(r + \gamma \max_{a'} Q_{\text{target}}(s', a'; \theta^-) - Q(s, a; \theta)\right)^2\right], \tag{2.4}$$

où θ^- sont les paramètres du réseau cible mis à jour périodiquement. Dans GAT-DQN, le DQN est augmenté d'un encodeur Transformer pour capturer les dépendances temporelles entre les phases d'attaque, formalisant la projection comme un processus de décision séquentielle.

2.8.2 Analyse critique et limites opérationnelles des techniques mobilisées

Si les techniques présentées dans ce chapitre offrent des capacités remarquables pour la cybersécurité, une analyse critique de leurs performances relatives et de leurs limites spécifiques en contexte opérationnel est nécessaire pour justifier pleinement les choix méthodologiques de cette thèse.

Compromis entre paradigmes. Les méthodes d'apprentissage automatique classiques (Random Forest, SVM) présentent des avantages significatifs en termes d'interprétabilité et d'efficacité computationnelle, mais montrent leurs limites face à la complexité sémantique des alertes textuelles et aux dépendances structurelles entre événements [16]. L'apprentissage profond améliore la capacité de représentation hiérarchique, mais au prix d'une opacité accrue et de besoins en données annotées plus importants. Les LLM offrent une compréhension contextuelle inégalée grâce à leur pré-entraînement sur de larges corpus, mais leur inférence est coûteuse en ressources GPU et leur latence peut dépasser les exigences des SOC à haut volume d'alertes [41]. Les GNN sont particulièrement adaptés à la modélisation des relations structurelles dans les graphes d'attaque, mais supposent une représentation sous forme de graphe stable et explicite, ce qui peut être difficile à maintenir dans des environnements dynamiques [42].

Limites spécifiques des LLM en contexte cyber opérationnel. L'intégration des LLM dans des pipelines de cybersécurité opérationnels présente plusieurs défis pratiques. La latence d'inférence d'un LLM — même fine-tuné avec des techniques légères comme LoRA — reste supérieure à celle des modèles classiques, imposant des contraintes d'infrastructure (GPU dédiés, gestion de la mémoire). Le déséquilibre des classes, caractéristique des données de sécurité où les événements malveillants sont largement minoritaires par rapport au trafic normal, peut biaiser le fine-tuning si des précautions d'échantillonnage ne sont pas prises. Par ailleurs, les LLM peuvent présenter des comportements inattendus face à des données adversariales ou des patterns d'attaque non représentés dans les

données d'entraînement, ce qui soulève des questions sur leur robustesse en production [15].

Limites spécifiques des GNN en contexte cyber opérationnel. Les réseaux de neurones sur graphes supposent une topologie de graphe explicite, ce qui implique une étape préalable de construction du graphe d'alertes. Cette construction introduit des choix de modélisation (seuils de corrélation, définition des arêtes, granularité des nœuds) qui influencent directement les performances du GNN. De plus, la scalabilité des GNN sur des graphes de grande taille — typiques des environnements SOC traitant des millions d'alertes quotidiennes — reste un défi ouvert [42]. Enfin, comme pour les LLM, le déséquilibre de classes dans les données d'entraînement peut conduire à une sous-représentation des attaques rares dans les mécanismes d'attention du GAT.

Ces limites justifient précisément l'approche multi-technique adoptée dans cette thèse : plutôt que de reposer sur un seul paradigme, les contributions combinent les forces complémentaires de chaque technique — sémantique via NLP et LLM, structurelle via GNN, adaptative via RL — tout en adressant leurs limites respectives par des choix architecturaux délibérés (méta-modèle de sélection contextuelle pour GAT-DQN, fine-tuning ciblé sur données de sécurité pour MCM-Llama, ingénierie de caractéristiques spécialisée pour Multi-Classification).

Le chapitre suivant présente la méthodologie générale de recherche ainsi que l'architecture globale du système intelligent d'alerte précoce proposé.

Chapitre 3

Méthodologie générale et architecture du système d'alerte précoce

3.1 Introduction

Ce chapitre présente la méthodologie générale adoptée dans cette thèse ainsi que l'architecture globale du système intelligent d'alerte précoce proposé. L'objectif est de décrire de manière structurée le cadre scientifique dans lequel s'inscrivent les cinq contributions développées dans les chapitres suivants, depuis la formulation du problème jusqu'à l'évaluation expérimentale des solutions proposées.

La démarche adoptée repose sur une approche progressive, où chaque contribution adresse une problématique spécifique de la détection précoce d'attaques : (1) la reconstruction de scénarios d'attaque par corrélation sémantique (MCM-CASR), (2) l'amélioration de la corrélation par modèles de langage fine-tunés (MCM-Llama), (3) la projection adaptative d'attaques futures (GAT-DQN), (4) la classification multi-niveaux des alertes (Multi-Classification), et (5) l'opérationnalisation du système via un pipeline SIEM-SOAR-LLM léger pour la réponse automatisée aux incidents (TinyLlama-SOAR). Cette méthodologie intégrée combine des techniques avancées de traitement du langage naturel, d'apprentissage profond, de réseaux de neurones sur graphes, d'apprentissage par renforcement et d'orchestration de la sécurité [13, 43].

3.2 Formulation du problème scientifique général

3.2.1 Problématique centrale

La problématique centrale de cette thèse est la suivante : *Comment automatiser la reconstruction de scénarios de cyberattaques à partir d'un flux massif d'alertes IDS tout*

en permettant la détection en temps réel, la prédiction des étapes futures et la classification intelligente des alertes de sécurité ?

Cette problématique se décline en cinq sous-problèmes majeurs, chacun adressé par une contribution spécifique :

1. reconstruction de scénarios (MCM-CASR) : Comment réduire efficacement le volume d'alertes sans perdre d'information critique, identifier les relations sémantiques entre alertes hétérogènes, et reconstruire des scénarios d'attaque cohérents et interprétables ?
2. corrélation en temps réel (MCM-Llama) : Comment améliorer significativement la précision de corrélation par rapport aux approches basées sur la similarité sémantique classique, tout en permettant une compréhension contextuelle approfondie des alertes pour une détection en temps réel ?
3. projection d'attaques (GAT-DQN) : Comment anticiper les prochaines étapes d'une attaque en cours en modélisant les dépendances structurelles entre événements et en s'adaptant dynamiquement aux tactiques adverses ?
4. classification multi-granularité (Multi-Classification) : Comment concevoir un cadre de corrélation multi-classification capable d'adresser différentes granularités de décision (binaire, types de corrélation, incidents) tout en maintenant des performances exceptionnelles ?
5. opérationnalisation et réponse automatisée (TinyLlama-SOAR) : Comment intégrer la corrélation sémantique par LLM dans un pipeline opérationnel complet SIEM-SOAR déployable sur infrastructure contrainte, sans GPU, tout en automatisant la réponse aux incidents de bout en bout ?

Ces cinq sous-problèmes délimitent le périmètre scientifique de la thèse. Leur résolution repose sur un ensemble d'hypothèses fondamentales qui guident les choix méthodologiques et justifient le recours aux techniques d'intelligence artificielle mobilisées.

3.2.2 Hypothèses de recherche

Les travaux de cette thèse reposent sur cinq hypothèses scientifiques fondamentales :

1. hypothèse sémantique : L'exploitation conjointe du Traitement Automatique du Langage Naturel (TALN), de la Reconnaissance d'Entités Nommées (REN) et de la similarité sémantique via BERT permet d'améliorer significativement la corrélation d'alertes par rapport aux approches purement basées sur des règles.
2. hypothèse LLM : Le fine-tuning d'un modèle de langage de grande taille (LLM) sur des données de corrélation d'alertes spécifiquement structurées permet d'atteindre des performances supérieures aux méthodes de corrélation classiques en capturant le contexte profond des séquences d'alertes.

3. hypothèse hybride GNN-RL : La combinaison de Graph Attention Networks (GAT) pour modéliser les dépendances structurelles des graphes d'attaque et de Deep Q-Networks (DQN) pour l'apprentissage adaptatif permet une projection efficace des prochaines étapes d'attaque.
4. hypothèse multi-classification : La formulation de la corrélation d'alertes comme un ensemble de problèmes de classification complémentaires (binaire, multiclasse types, multiclasse incidents) permet d'adresser différentes granularités opérationnelles tout en maintenant des performances optimales.
5. hypothèse d'opérationnalisation légère : La réduction de la corrélation sémantique à un problème de classification binaire par paires (YES/NO) via un petit LLM (TinyLlama-1.1B) suffit à obtenir des performances comparables aux systèmes plus lourds, tout en permettant un déploiement opérationnel sur infrastructure ordinaire (CPU seul, 4 GB RAM) intégré à une chaîne SIEM-SOAR complète.

Ces hypothèses constituent le fil directeur des cinq contributions de la thèse. La section suivante présente l'architecture globale dans laquelle elles s'inscrivent, en montrant comment chaque contribution s'intègre dans un système cohérent d'alerte précoce.

3.3 Architecture globale et positionnement des contributions

Cette section décrit l'architecture d'ensemble du système intelligent d'alerte précoce proposé, en montrant comment les cinq contributions s'articulent en une chaîne de traitement cohérente, depuis la collecte des alertes brutes jusqu'à leur classification, la projection des phases d'attaque futures et la réponse automatisée aux incidents.

3.3.1 Vue d'ensemble de l'architecture intégrée

L'architecture globale (Figure 3.1) s'organise autour d'une chaîne de traitement progressive où chaque contribution adresse une phase spécifique du processus d'alerte précoce :

1. phase de collecte et normalisation : Ingestion des alertes IDS provenant de sources multiples (Snort, Suricata, Zeek pour MCM-CASR et MCM-Llama ; flux réseau pour Multi-Classification ; EVE-JSON via Filebeat/Elasticsearch pour TinyLlama-SOAR).
2. phase de corrélation et reconstruction : Application de MCM-CASR pour la corrélation sémantique offline ou de MCM-Llama pour la corrélation en temps réel, permettant la reconstruction de scénarios d'attaque.

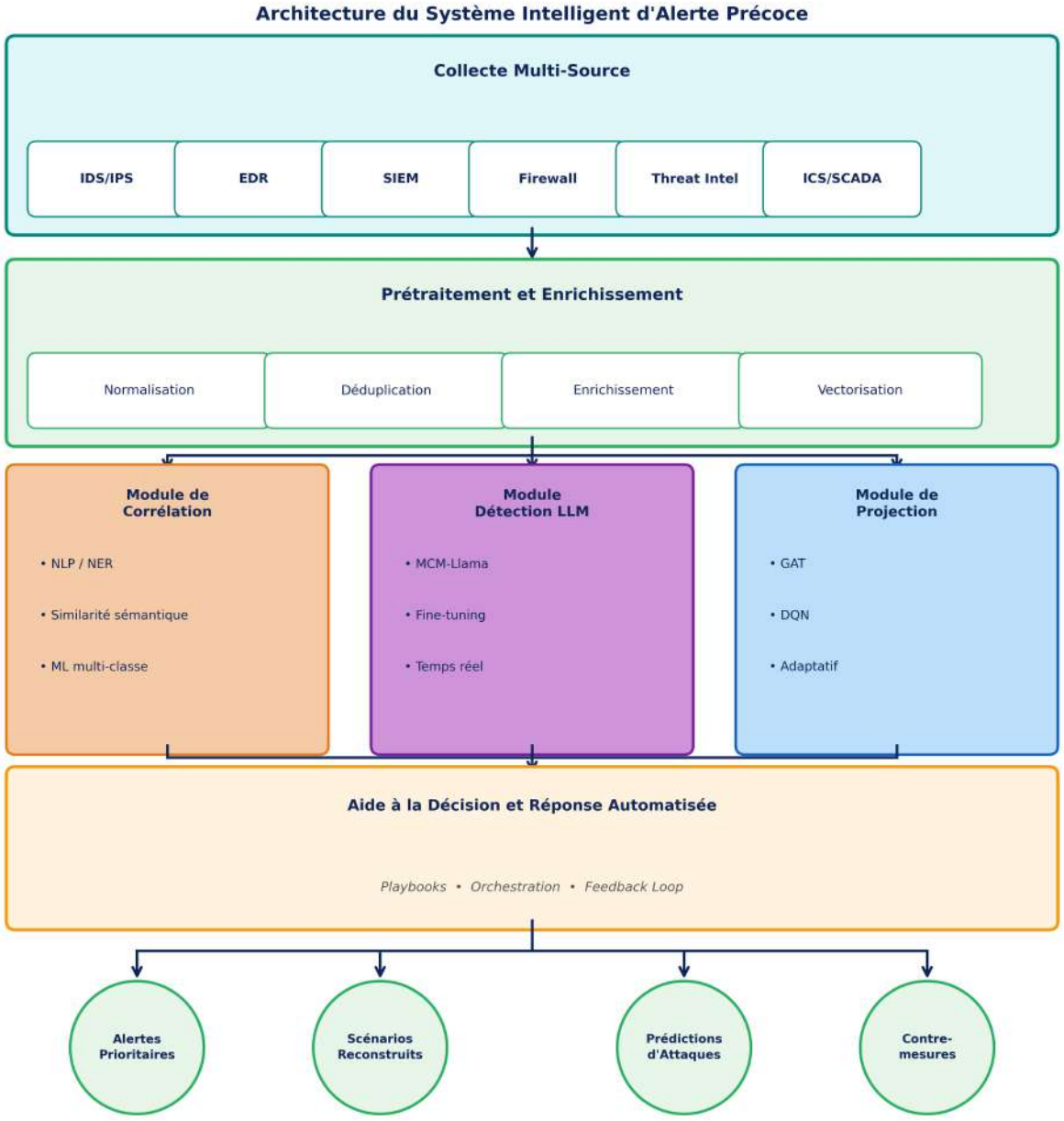


FIGURE 3.1 – Architecture globale du système intelligent d'alerte précoce montrant l'intégration des cinq contributions

3. phase de projection : Utilisation de GAT-DQN pour anticiper les prochaines étapes possibles de l'attaque en cours.
4. phase de classification et priorisation : Application du cadre Multi-Classification pour catégoriser les alertes selon différentes granularités et prioriser les incidents.
5. phase d'orchestration et réponse automatisée : Exploitation du pipeline TinyLlama-SOAR pour corréler les alertes par analyse par paires (YES/NO), créer des cas d'incident dans TheHive/Cortex et déclencher l'isolation automatique des hôtes compromis via SentinelOne EDR.

3.3.2 Positionnement par rapport aux lacunes identifiées

Les cinq contributions adressent des lacunes spécifiques identifiées dans l'état de l'art (Figure 3.12 de la thèse) :

— MCM-CASR comble le manque d'exploitation de la dimension sémantique profonde dans la corrélation d'alertes, en allant au-delà des simples attributs techniques.

— MCM-Llama répond au besoin d'automatisation intelligente via l'IA générative, aligné avec les tendances identifiées par le World Economic Forum [3].

— GAT-DQN adresse le défi de l'anticipation des menaces persistantes avancées (APT) par une approche adaptative combinant modélisation structurelle et apprentissage dynamique.

— multi-Classification résout le problème de granularité en proposant une approche unifiée couvrant différents niveaux de décision de corrélation.

— TinyLlama-SOAR comble le fossé entre la recherche et l'opérationnel : un LLM léger (1,1 milliard de paramètres, CPU seulement) couplé à une chaîne SOAR complète rend la corrélation sémantique accessible aux organisations sans infrastructure GPU.

Ces cinq sous-problèmes sont adressés séquentiellement dans les chapitres suivants, chaque contribution s'appuyant sur les résultats de la précédente pour construire un système de détection de plus en plus complet et précis.

3.4 Contribution 1 : MCM-CASR - Reconstruction de scénarios par corrélation sémantique

MCM-CASR constitue la première contribution de cette thèse. Elle adresse le problème fondamental de la reconstruction automatique de scénarios d'attaque à partir d'un flux massif d'alertes IDS, en exploitant les techniques de traitement automatique du langage naturel et de similarité sémantique.

3.4.1 Objectif et problématique spécifique

MCM-CASR vise à automatiser la reconstruction de scénarios d'attaque à partir d'alertes IDS hétérogènes en exploitant les techniques de TALN, REN et similarité sémantique. Les questions de recherche sont :

— Q1 : Comment réduire efficacement le volume d'alertes sans perdre d'information critique ?

— Q2 : Comment identifier et modéliser les relations sémantiques entre alertes hétérogènes ?

— Q3 : Comment reconstruire des scénarios d'attaque cohérents et interprétables ?

Ces trois questions structurent l'architecture de MCM-CASR, dont les composants sont présentés dans la sous-section suivante.

3.4.2 Architecture et méthodologie MCM-CASR

Le cadre MCM-CASR (Figure 3.2) s'articule autour de deux modules principaux :

Module 1 : Réduction des alertes

Ce module applique un mécanisme de filtrage multi-critères basé sur trois attributs clés :

— fenêtre temporelle : Regroupement des alertes dans des intervalles de temps configurables (par défaut 60 secondes).

— catégorie d'attaque : Agrégation des alertes partageant la même catégorie (ex : "Attempted User Privilege Gain").

— adresses IP source et destination : Groupement des événements impliquant les mêmes paires d'adresses.

Cette approche permet une réduction de 43,49% du volume d'alertes sur le jeu de données UNSW-NB15.

Module 2 : Corrélation sémantique

Le module de corrélation exploite trois techniques complémentaires :

1. reconnaissance d'Entités Nommées (REN) : Extraction automatique des entités cybersécuritaires (adresses IP, ports, protocoles, identifiants d'attaque) à partir des descriptions textuelles des alertes.
2. calcul de similarité sémantique : Utilisation du modèle BERT pré-entraîné pour générer des embeddings contextuels des descriptions d'alertes, suivie du calcul de similarité cosinus entre vecteurs.

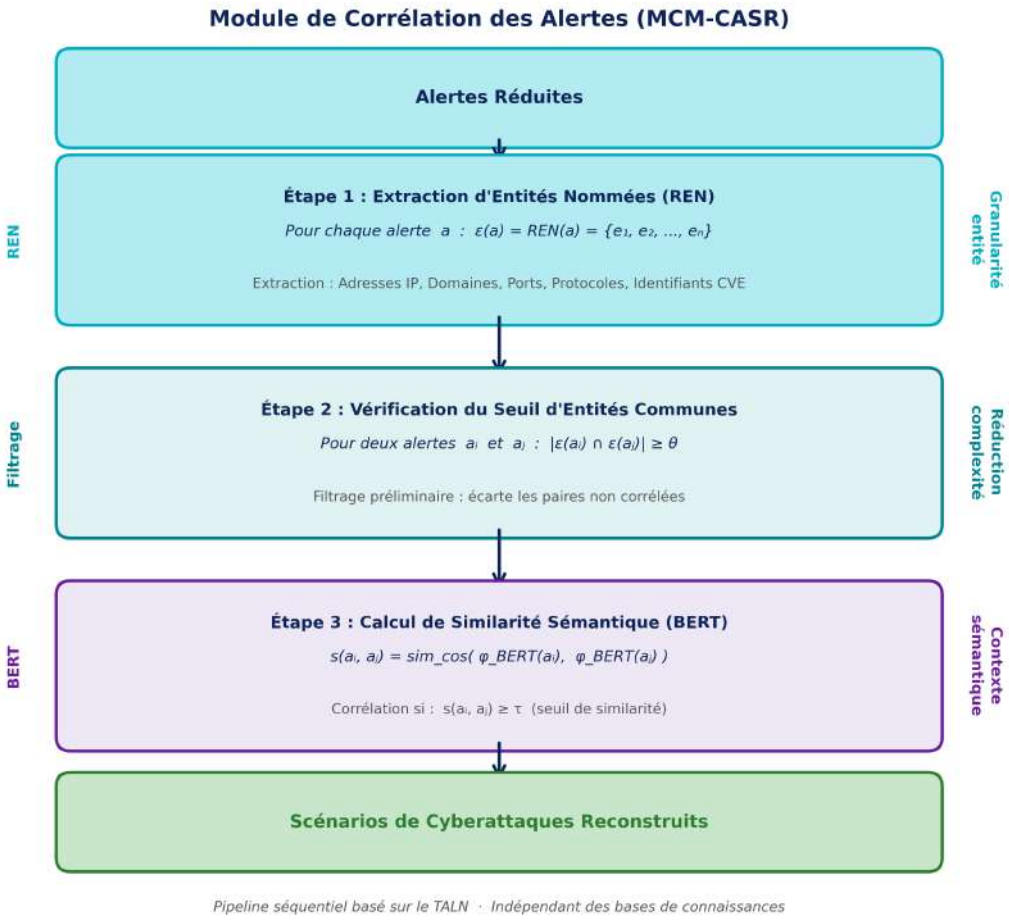


FIGURE 3.2 – Architecture du cadre MCM-CASR comprenant les modules de réduction et de corrélation

3. matching multi-critères : Fusion des scores de similarité sémantique avec les correspondances d'entités extraites par REN pour déterminer les corrélations finales.

3.4.3 Datasets et protocole expérimental

Afin de valider les choix architecturaux et les hypothèses de MCM-CASR, une évaluation expérimentale rigoureuse a été conduite sur un jeu de données de référence reconnu dans la communauté de la détection d'intrusions. Le protocole retenu repose sur des métriques adaptées aux spécificités de la corrélation d'alertes, permettant d'apprécier à la fois la capacité de réduction du bruit et la qualité des scénarios reconstruits.

Dataset UNSW-NB15

L'évaluation de MCM-CASR utilise le jeu de données UNSW-NB15 [68, 69], un ensemble de référence comprenant :

— 2,5 millions de flux réseau

— 49 features par flux

— 10 catégories d'attaques (Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, Worms, plus le trafic normal)

Métriques d'évaluation

Les performances de MCM-CASR sont évaluées selon :

— taux de réduction des alertes (ARR) : Pourcentage d'alertes éliminées par le processus de filtrage.

— précision de corrélation : Exactitude des paires d'alertes identifiées comme corrélées.

— cohérence des scénarios : Qualité sémantique et temporelle des scénarios reconstruits.

3.4.4 Résultats principaux

MCM-CASR atteint les performances suivantes sur UNSW-NB15 :

— réduction des alertes : 43,49%

— précision de corrélation : 53,91%

— scénarios reconstruits : Identification de séquences d'attaque multi-étapes cohérentes

Ces résultats démontrent l'efficacité de l'approche sémantique pour la reconstruction de scénarios, tout en identifiant la marge d'amélioration de la précision, ce qui motive la contribution suivante (MCM-Llama).

3.5 Contribution 2 : MCM-Llama - Corrélation par LLM fine-tuné

Les résultats de MCM-CASR, bien que prometteurs sur le plan de la réduction des alertes, ont mis en évidence une marge d'amélioration significative en termes de précision de corrélation. MCM-Llama constitue la réponse à cette limitation en substituant le module de corrélation sémantique classique par un grand modèle de langage spécialement affiné sur des données de sécurité.

3.5.1 Motivation et amélioration par rapport à MCM-CASR

MCM-Llama vise à surmonter la limitation principale de MCM-CASR (précision de 53,91%) en remplaçant le module de corrélation basé sur REN et similarité sémantique par un modèle de langage de grande taille (LLM) spécifiquement fine-tuné pour la tâche de corrélation binaire d'alertes.

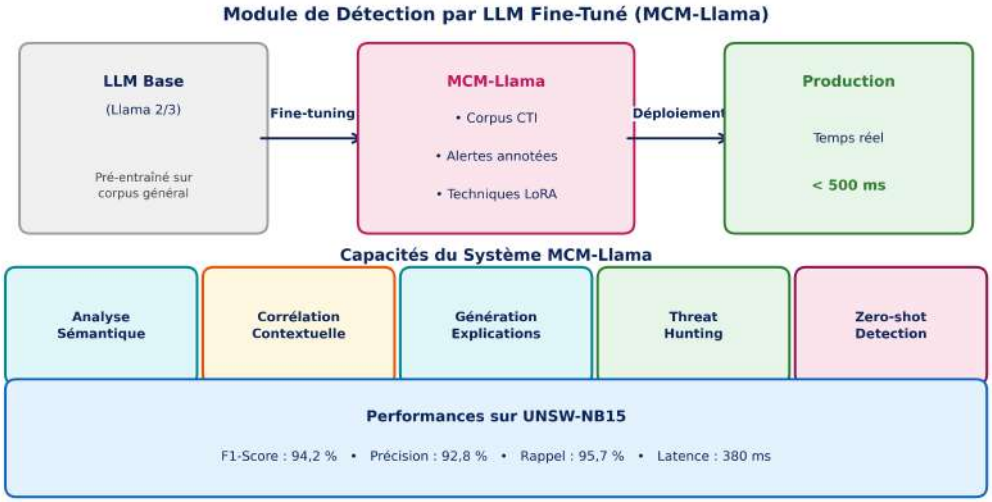


FIGURE 3.3 – Architecture de MCM-Llama montrant le processus de fine-tuning et de déploiement

3.5.2 Méthodologie de fine-tuning

Choix du modèle de base

MCM-Llama utilise Llama-2-7B comme modèle de base, un LLM open-source de 7 milliards de paramètres développé par Meta [50].

Préparation du jeu de données CTU-13

Le fine-tuning s'appuie sur le jeu de données CTU-13 [2], qui comprend 13 scénarios de botnets différents avec des données NetFlow capturées sur des réseaux réels, ainsi que les alertes IDS correspondantes générées par Suricata.

Format d'instructions structuré :

Chaque exemple d'entraînement suit le format :

```
### Instruction:
Determine if the following two security alerts are correlated.

### Alert 1:
[Timestamp] [Source IP] -> [Dest IP] : [Signature]

### Alert 2:
[Timestamp] [Source IP] -> [Dest IP] : [Signature]

### Response:
[YES/NO]
```

Ce format permet au modèle d'apprendre la tâche de corrélation binaire de manière supervisée.

Processus de fine-tuning

Le processus de fine-tuning a été conçu pour adapter le modèle LLaMA-2-7B à la tâche de corrélation binaire d'alertes tout en minimisant les ressources computationnelles requises. Les hyperparamètres suivants ont été retenus après validation expérimentale.

Le fine-tuning utilise la technique LoRA (Low-Rank Adaptation) [51] pour réduire les coûts computationnels :

— optimiseur : AdamW

— learning rate : 2e-4

— batch size : 4

— nombre d'époques : 3

— rank LoRA : 8

— alpha LoRA : 16

3.5.3 Architecture de déploiement

L'architecture opérationnelle de MCM-Llama comprend quatre composants (Figure 3.3) :

1. IDS (Suricata) : Génération des alertes en temps réel
2. base de données : Stockage des alertes avec indexation temporelle
3. framework MCM-Llama : Module de corrélation via le LLM fine-tuné
4. interface de visualisation : Dashboard Kibana ou interface web custom

3.5.4 Résultats et amélioration

MCM-Llama atteint une précision de 85% sur le jeu de données CTU-13, soit une amélioration de 31 points de pourcentage par rapport à MCM-CASR (54%).

TABLE 3.1 – Comparaison des performances MCM-CASR vs MCM-Llama

Métrique	**MCM-CASR**	**MCM-Llama**	**Amélioration**
Précision	53,91%	85%	+31,09 pts
Temps réel	Non	Oui	-
Contexte profond	Partiel	Oui	-
APT detection	Partiel	Oui	-

Cette amélioration substantielle valide l'hypothèse selon laquelle les LLMs fine-tunés capturent mieux le contexte sémantique complexe des alertes de sécurité.

3.6 Contribution 3 : GAT-DQN - Projection adaptative par sélection contextuelle

La corrélation d'alertes accomplie par MCM-CASR et MCM-Llama répond à la question de *ce qui s'est passé*. La troisième contribution, GAT-DQN, s'attaque à une question plus ambitieuse : *ce qui va se passer*. En combinant des réseaux d'attention sur graphes et de l'apprentissage par renforcement, elle permet de projeter les phases futures d'une attaque en cours.

3.6.1 Objectif et problématique spécifique

La contribution GAT-DQN adresse la problématique de la projection d'attaques, c'est-à-dire la prédiction de la phase suivante d'une cyberattaque en cours à partir des alertes

observées dans la phase actuelle. Cette capacité prédictive est cruciale pour une défense proactive permettant d'anticiper et de bloquer une attaque avant qu'elle n'atteigne ses objectifs critiques.

Les questions de recherche spécifiques sont :

— Q1 : Comment exploiter les structures de graphes pour capturer les relations complexes entre les composants d'une attaque ?

— Q2 : Comment utiliser l'apprentissage par renforcement pour la prise de décision séquentielle dans des scénarios d'attaque évolutifs ?

— Q3 : Comment concevoir un méta-modèle capable de sélectionner dynamiquement le modèle le plus approprié selon le contexte ?

3.6.2 Innovation principale : Méta-modèle de sélection contextuelle

Concept clé : Contrairement aux approches classiques qui utilisent un seul modèle fixe, GAT-DQN propose un cadre de méta-modèle qui sélectionne dynamiquement entre deux modèles complémentaires (GAT et DQN) selon les caractéristiques contextuelles des alertes.

— modèle 1 - GAT : Graph Attention Network pour les scénarios où les relations entre alertes sont complexes et non-linéaires (graphes d'attaque denses).

— modèle 2 - DQN : Deep Q-Network pour les scénarios où la progression temporelle et les décisions séquentielles sont dominantes.

— méta-modèle k-NN : Système de sélection basé sur k-plus-proches-voisins qui choisit automatiquement GAT ou DQN selon la similarité des features avec des cas historiques.

Ce mécanisme de sélection contextuelle constitue le cœur de l'innovation apportée par GAT-DQN. La sous-section suivante détaille l'architecture complète du cadre, en précisant comment ces trois composantes s'articulent pour former un système de projection adaptatif et cohérent.

3.6.3 Architecture globale du cadre

Le cadre GAT-DQN (Figure 3.4) s'articule autour de quatre composants principaux :

Composant 1 : Ingestion et extraction de features

Les alertes IDS (générées par Suricata sur le jeu de données CTU-13) sont collectées et leurs caractéristiques contextuelles sont extraites :

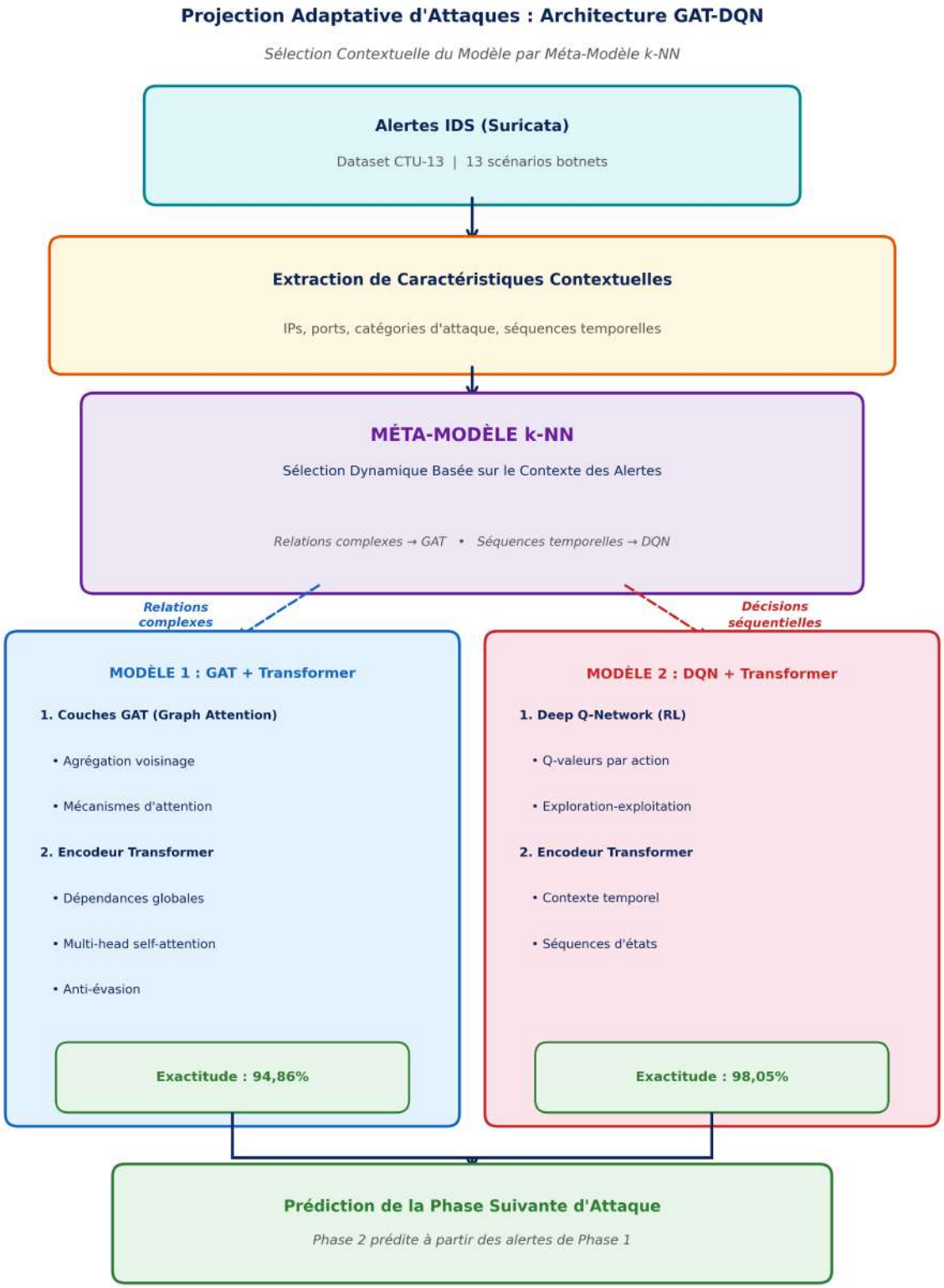


FIGURE 3.4 – Architecture GAT-DQN : sélection contextuelle entre deux modèles par méta-modèle k-NN

— features réseau : Adresses IP source/destination, ports, protocoles

— features temporelles : Timestamps, durées, intervalles entre alertes

— **Features d'attaque** : Catégories (Misc Activity, C&C, Attack, HeartBeat), signatures

— features structurelles : Degré de connectivité dans le graphe d'alertes, densité des corrélations

Composant 2 : Méta-modèle k-NN (sélecteur)

Le méta-modèle k-NN prend une décision binaire :

$$\text{Modèle_Sélectionné} = \begin{cases} \text{GAT} & \text{si relations complexes dominantes} \\ \text{DQN} & \text{si séquences temporelles dominantes} \end{cases} \tag{3.1}$$

Processus de sélection :

1. extraction du vecteur de features contextuelles $\mathbf{f}_{\text{ctx}}$ des nouvelles alertes
2. calcul de similarité avec les K plus proches voisins dans la base historique
3. vote majoritaire : Si k-NN indique que des cas similaires ont mieux performé avec GAT $\rightarrow$ sélection GAT, sinon DQN

Une fois le méta-modèle k-NN ayant sélectionné le modèle le plus approprié au contexte courant, celui-ci est instancié pour réaliser la projection. Les deux composants disponibles sont décrits ci-après.

Composant 3 : Modèle GAT + Transformer

Architecture : Le modèle GAT combine deux mécanismes complémentaires (voir Figure détaillée en annexe).

Couches GAT Les couches Graph Attention Network [60] agrègent l'information des nœuds voisins :

$$\mathbf{h}_v^{(1)} = \text{GAT}_1(\mathbf{x}_v, \{\mathbf{x}_u : u \in \mathcal{N}(v)\}) \tag{3.2}$$

$$\mathbf{h}_v^{(2)} = \text{GAT}_2(\mathbf{h}_v^{(1)}, \{\mathbf{h}_u^{(1)} : u \in \mathcal{N}(v)\}), \tag{3.3}$$

où :

— $\mathbf{x}_v$: features initiales du nœud v (alerte)

— $\mathcal{N}(v)$: voisinage du nœud v dans le graphe d'alertes corrélées

— $\mathbf{h}_v^{(l)}$: représentation du nœud à la couche l

Le mécanisme d'attention calcule des poids pour pondérer l'importance de chaque nœud voisin ;

Encodeur Transformer Un encodeur Transformer est appliqué sur les embeddings GAT pour capturer les dépendances globales et les patterns temporels via des mécanismes d'auto-attention multi-têtes ;

Couche de classification Une couche fully-connected avec softmax produit la distribution de probabilité sur les phases d'attaque possibles ;

$$P(\text{Phase}_{\text{suivante}} = c|\text{Phase}_{\text{actuelle}}) = \text{softmax}(\mathbf{W} \cdot \mathbf{h}_{\text{final}} + \mathbf{b}) \tag{3.4}$$

Composant 4 : Modèle DQN + Transformer

Lorsque le méta-modèle identifie une prédominance de séquences temporelles dans les données d'entrée, le modèle DQN est sélectionné. Son architecture repose sur la formulation de la projection d'attaque comme un processus de décision séquentielle.

Architecture : Le modèle DQN formule la projection comme un problème d'apprentissage par renforcement.

Formulation RL

- état s : Séquence d'alertes de la phase actuelle avec leurs features
- **Actions** a : Ensemble des phases d'attaque possibles (Misc Activity, C&C, Attack, HeartBeat, etc.)
- fonction Q $Q(s, a)$: Valeur estimée de prendre l'action a dans l'état s
- récompense r : +1 si la phase prédite correspond à la phase réelle observée, 0 sinon

Encodeur Transformer L'état s (séquence d'alertes) est encodé par un Transformer pour capturer les dépendances temporelles à long terme et le contexte séquentiel ;

Q-Network Des couches fully-connected prennent les embeddings du Transformer et produisent les Q-values pour chaque action possible :

$$Q(s, a) = \text{FC}(\text{TransformerEncoder}(s)) \tag{3.5}$$

Entraînement par RL L'entraînement utilise la fonction de perte suivante :

$$\mathcal{L} = \mathbb{E}_{(s,a,r,s')} \left[\left(r + \gamma \max_{a'} Q_{\text{target}}(s', a') - Q(s, a) \right)^2 \right] \tag{3.6}$$

avec :

— Experience replay pour stabiliser l'apprentissage [67]

— target network Q_{target} mise à jour périodiquement [65]

— exploration ϵ-greedy pour équilibrer exploration/exploitation

3.6.4 Dataset et protocole expérimental

Les évaluations de GAT-DQN ont été conduites sur le jeu de données CTU-13, une référence académique pour les scénarios de botnets en conditions réelles, dont les caractéristiques sont présentées ci-après.

Dataset CTU-13

L'évaluation de GAT-DQN utilise le jeu de données CTU-13 [2], qui comprend treize scénarios de botnets différents capturés sur des réseaux réels. Les alertes IDS ont été générées par Suricata sur les flux réseau correspondants. Chaque scénario présente une progression à travers plusieurs phases d'attaque, couvrant notamment la reconnaissance, le commandement et contrôle (C&C), l'exploitation et l'exfiltration de données.

Tâche de prédiction

Objectif : Prédire la Phase 2 d'une attaque à partir des alertes de la Phase 1.
Setup expérimental :

— entrée : Ensemble d'alertes représentant la première phase

— sortie : Prédiction de la catégorie de la deuxième phase

— split : 80% train, 20% test

— évaluation : Exactitude (Accuracy), F1-Score, matrices de confusion

3.6.5 Résultats expérimentaux

Cette section présente les résultats obtenus par le cadre GAT-DQN sur le jeu de données CTU-13. L'évaluation couvre d'abord les performances des modèles individuels (GAT et DQN) prises séparément, avant d'analyser l'apport du méta-modèle de sélection contextuelle sur la projection adaptative des phases d'attaque.

Performances des modèles individuels

TABLE 3.2 – Performances GAT vs DQN sur CTU-13

Modèle	Exactitude	F1-Score	Précision
GAT + Transformer	94,86%	94,91%	95,12%
DQN + Transformer	**98,95%**	**98,89%**	**98,76%**

Observations clés :

— DQN surpasse GAT sur le jeu de données CTU-13 global (98,95% vs 94,86%)

— les deux modèles atteignent des performances élevées (>94%), validant leur efficacité

— le mécanisme d'attention GAT capture bien les relations complexes dans les graphes d'attaque

— le DQN excelle dans la prise de décision séquentielle pour les progressions d'attaque

Performance du méta-modèle

Le méta-modèle k-NN sélectionne dynamiquement entre GAT et DQN selon le contexte :

— sur les scénarios à graphes denses (nombreuses corrélations) : sélection de GAT

— sur les scénarios à séquences longues (progression temporelle claire) : sélection de DQN

— exactitude de sélection du méta-modèle : >92% (sélectionne le bon modèle dans la majorité des cas)

Comparaison avec l'état de l'art

TABLE 3.3 – Comparaison avec les approches de référence

Approche	**Exactitude**	**Dataset**	**Technique**
ProAPT [63]	97,36%	DAPT2020	DRL + LSTM
HMM-based [70]	91,80%	Synthétique	HMM + Viterbi
GAT-DQN (GAT)	94,86%	CTU-13	GAT + Transformer
GAT-DQN (DQN)	**98,95%**	CTU-13	DQN + Transformer

GAT-DQN (modèle DQN) atteint l'exactitude la plus élevée (98,95%) sur la tâche de projection de phase d'attaque.

3.6.6 Avantages de l'approche GAT-DQN

1. complémentarité des modèles : GAT capture les relations complexes, DQN gère les décisions séquentielles
2. adaptabilité contextuelle : Le méta-modèle sélectionne automatiquement l'approche optimale
3. flexibilité : Pas de modèle unique rigide, adaptation aux caractéristiques des données
4. performances élevées : Les deux modèles atteignent >94% d'exactitude
5. robustesse : Le cadre gère une diversité de scénarios d'attaque (botnets, APT, multi-étapes)

3.6.7 Limitations et perspectives

Limitations

Bien que l'approche GAT-DQN démontre des performances remarquables, plusieurs limitations doivent être soulignées pour contextualiser son déploiement opérationnel.

La complexité computationnelle constitue une contrainte majeure. L'entraînement du modèle GAT avec ses couches d'attention sur graphes, combiné à l'apprentissage par renforcement du DQN et l'optimisation du méta-modèle de sélection, nécessite des ressources matérielles significatives, notamment des GPU haute performance. Les expérimentations menées ont révélé que l'entraînement complet du système peut nécessiter plusieurs heures voire jours selon la taille du jeu de données, ce qui peut constituer un frein pour des organisations disposant de ressources limitées.

La question de la latence en temps réel représente également un défi opérationnel. Le processus de sélection contextuelle du méta-modèle, suivi de l'exécution du modèle choisi (GAT ou DQN) pour la projection de la phase suivante, introduit un délai de traitement qui, bien que relativement faible, peut s'avérer critique dans des scénarios d'attaques rapides où chaque seconde compte. Cette latence s'additionne aux délais déjà présents dans la chaîne de traitement (collecte d'alertes, prétraitement, corrélation).

Enfin, la capacité de généralisation du système demeure une question ouverte. Les évaluations expérimentales ont été principalement conduites sur le jeu de données CTU-13, qui, bien que reconnu comme référence académique, présente des caractéristiques spécifiques. La performance du système sur d'autres jeux de données ou sur des données réelles issues de centres opérationnels de sécurité reste à valider de manière approfondie, particulièrement face à des attaques émergentes ou des patterns non représentés dans les données d'entraînement.

Perspectives

Plusieurs pistes d'amélioration prometteuses se dessinent pour étendre les capacités du système GAT-DQN.

L'extension multi-étapes constitue une évolution naturelle de l'approche actuelle. Actuellement, le système prédit uniquement la phase immédiatement suivante d'une attaque (Phase 2 à partir de Phase 1). Une extension permettant de projeter plusieurs étapes futures (Phase 3, Phase 4, etc.) offrirait aux équipes de sécurité une vision stratégique à plus long terme de l'évolution potentielle d'une attaque. Cette capacité de planification à horizon temporel étendu nécessiterait toutefois de repenser l'architecture du modèle DQN pour intégrer des mécanismes de planification hiérarchique et de modélisation de l'incertitude croissante avec l'horizon de prédiction.

L'optimisation des ressources computationnelles représente un axe de recherche crucial pour faciliter le déploiement opérationnel. Des techniques de quantification des modèles

(passage de float32 à int8), de pruning (élagage des connexions neuronales peu contributives), ou de distillation de connaissances (transfert des capacités du modèle vers une architecture plus légère) pourraient réduire significativement les coûts computationnels et la latence d'inférence, tout en préservant des performances acceptables.

L'**intégration avec le framework MITRE ATT&CK** ouvre des perspectives d'enrichissement sémantique. Actuellement, les phases prédites correspondent à des catégories d'alertes génériques. Un mapping systématique vers les tactiques, techniques et procédures (TTPs) du framework MITRE ATT&CK permettrait d'obtenir des projections d'attaque plus explicites et actionnables, facilitant la communication avec les analystes de sécurité et l'automatisation des réponses défensives.

Enfin, l'exploration de méta-modèles avancés pour la sélection contextuelle pourrait améliorer la robustesse du système. L'approche actuelle basée sur des règles heuristiques simples pourrait être remplacée par des techniques de méta-apprentissage (meta-learners basés sur des ensembles de modèles), d'optimisation bayésienne, ou même de réseaux neuronaux dédiés à la sélection de modèle, capables d'apprendre automatiquement les conditions optimales de basculement entre GAT et DQN.

3.6.8 Positionnement dans l'architecture globale

GAT-DQN s'intègre naturellement après les modules de corrélation (MCM-CASR/MCM-Llama) :

1. corrélation : MCM-CASR ou MCM-Llama identifie les alertes liées à la phase actuelle
2. projection : GAT-DQN prédit la phase suivante probable
3. décision : Le SOC peut prendre des mesures préventives avant que la phase suivante ne se produise

Cette capacité prédictive transforme la cybersécurité d'une posture réactive (détecter après coup) vers une posture proactive (anticiper et prévenir).

3.7 Contribution 4 : Multi-Classification - Classification multi-granularité

La quatrième contribution de cette thèse adopte une perspective différente des trois premières : plutôt que de reconstruire des scénarios ou de projeter des attaques, elle vise à améliorer la *classification* des alertes en proposant simultanément trois niveaux de granularité décisionnelle adaptés aux besoins opérationnels des SOC. La quatrième contribution transforme la corrélation d'alertes en un ensemble de trois problèmes de classification complémentaires, permettant d'adresser différentes granularités opérationnelles d'un SOC.

3.7.1 Les trois approches de classification

Afin d'adresser les différentes granularités décisionnelles d'un SOC, le cadre Multi-Classification décompose la corrélation d'alertes en trois problèmes complémentaires, allant du filtrage grossier à l'identification fine des incidents.

Classification binaire : Détection de corrélation

Objectif : Déterminer si deux alertes sont corrélées ou non.

Formulation : Problème de classification binaire à 2 classes (Correlated / Not Correlated).

Cas d'usage : Filtrage initial des paires d'alertes pour réduire le volume d'analyse manuelle.

Classification multiclasse : Types de corrélation

Objectif : Identifier le type spécifique de relation entre alertes corrélées.

Classes :

— same Source IP

— same Destination IP

— same Attack Category

— temporal Sequence

— multi-stage Attack

Cas d'usage : Compréhension fine des relations pour reconstruction de scénarios détaillés.

Classification multiclasse : Incidents

Objectif : Regrouper les alertes en incidents cohérents.

Classes : Un identifiant d'incident par campagne d'attaque coordonnée.

Cas d'usage : Priorisation et assignation des incidents aux analystes.

3.7.2 Ingénierie des caractéristiques

Le cadre développe un ensemble de caractéristiques spécifiquement conçu pour la corrélation :

— features temporelles : Différence de timestamps, fenêtre temporelle partagée

— features réseau : Similarité d'IPs source/destination, proximité de ports

— features catégorielles : Correspondance de catégories d'attaque, signatures similaires

— features de progression : Indicateurs d'escalade, ordre dans la kill chain

3.7.3 Algorithmes évalués

Huit algorithmes d'apprentissage automatique sont comparés :

1. régression Logistique
2. Random Forest
3. XGBoost
4. LightGBM
5. réseaux de Neurones Artificiels (ANN)
6. réseaux de Neurones Convolutifs (CNN)
7. CatBoost
8. support Vector Machines (SVM)

3.7.4 Dataset et protocole expérimental

L'évaluation du cadre Multi-Classification repose sur le jeu de données CIC-IDS-2017, reconnu pour sa représentativité des scénarios d'attaques modernes en environnement d'entreprise.

Dataset CIC-IDS-2017

Le jeu de données CIC-IDS-2017 [71] regroupe 2,3 millions de flux réseau réels décrits par 80 caractéristiques chacun, couvrant des types d'attaques variés tels que DoS, DDoS, Brute Force, XSS, injection SQL, infiltration, PortScan et Botnet. À partir de ces flux, 50 000 alertes de sécurité réalistes ont été générées pour l'évaluation.

Métriques d'évaluation

Les performances de chaque algorithme sont mesurées à travers cinq métriques complémentaires : la précision globale (Accuracy), la précision par classe (Precision), le rappel (Recall), le F1-Score harmonisant précision et rappel, et la matrice de confusion permettant une analyse détaillée des erreurs de classification.

3.7.5 Résultats et analyse comparative

Points remarquables :

— six modèles atteignent 100% de précision sur la classification des types de corrélation.

— CatBoost obtient la meilleure performance globale avec 99,27% de précision moyenne.

— tous les algorithmes dépassent 97% de précision, démontrant la robustesse du cadre.

TABLE 3.4 – Performances des algorithmes sur les trois tâches de classification

Algorithme	**Binaire**	**Types**	**Incidents**	**Moyenne**
CatBoost	99,20%	**100%**	98,60%	**99,27%**
Random Forest	99,00%	**100%**	98,50%	99,17%
LightGBM	98,80%	**100%**	98,50%	99,10%
XGBoost	98,70%	**100%**	98,40%	99,03%
Logistic Regression	98,60%	**100%**	98,30%	98,97%
ANN	98,50%	**100%**	98,20%	98,90%
CNN	97,80%	99,50%	97,60%	98,30%
SVM	97,50%	99,00%	97,30%	97,93%

3.8 Contribution 5 : TinyLlama-SOAR - Pipeline opérationnel de réponse automatisée

La quatrième contribution démontre qu'il est possible d'atteindre une précision de classification supérieure à 99% sur des données académiques. La cinquième contribution répond à une question différente : comment rendre ce niveau de performance accessible à un SOC réel, avec un budget matériel ordinaire ? Elle constitue le passage de la preuve de concept à l'opérationnalisation.

3.8.1 Objectif et problématique spécifique

TinyLlama-SOAR vise à opérationnaliser la corrélation sémantique d'alertes en la combinant à une chaîne SIEM-SOAR complète, tout en restant déployable sur un VPS à 4 cœurs, 8 GB de RAM, sans GPU. L'enjeu n'est plus uniquement la précision algorithmique, mais l'intégration systémique : de la génération d'alertes par Suricata IDS jusqu'à l'isolation automatique d'un endpoint via SentinelOne EDR.

Les questions de recherche spécifiques sont :

— Q1 : Un LLM de 1,1 milliard de paramètres peut-il servir de classifieur binaire (YES/NO) fiable pour la corrélation par paires d'alertes, sans fine-tuning sur des données de sécurité ?

— Q2 : Quelle architecture en trois couches (SIEM, IA, SOAR) permet d'intégrer ce classifieur dans un pipeline de réponse automatisée bout en bout ?

— Q3 : Quelles sont les performances opérationnelles réelles (TPR, FPR, latence, mémoire) sur des alertes Suricata en conditions de déploiement continu ?

3.8.2 Architecture en trois couches

Le cadre TinyLlama-SOAR s'organise autour de trois couches communicant via API REST et webhooks :

Couche SIEM (Collecte et normalisation)

Suricata IDS (v6.x) surveille les interfaces réseau et émet des alertes au format EVE-JSON. Filebeat normalise et indexe ces alertes dans Elasticsearch (Elastic Stack 7.17) selon le schéma ECS, rendant les alertes interrogeables en sous-seconde. Kibana fournit le tableau de bord d'analyse et un connecteur webhook qui transmet en temps réel les alertes de haute sévérité à TheHive.

Couche IA (Corrélation sémantique)

Un service FastAPI interroge Elasticsearch toutes les 15 minutes, constitue des paires d'alertes et soumet chaque paire à TinyLlama-1.1B-Chat via un prompt structuré. Le prompt intègre les IPs source/destination, ports, horodatages, identifiants de signature et descriptions textuelles, puis demande au modèle d'émettre un jugement binaire (YES/NO) sur la corrélation. À température 0,1, le modèle se comporte comme un classifieur déterministe. Les alertes corrélées sont regroupées par fermeture transitive (Algorithme de groupement par lots).

Couche SOAR (Orchestration et réponse)

Pour chaque groupe d'alertes corrélées, le service crée automatiquement un cas structuré dans TheHive 5.3.0 et déclenche via Cortex 3.1.8 : (1) l'isolation de l'hôte compromis par le responder SentinelOne, (2) l'enrichissement par VirusTotal et IPReputation, et (3) le recoupement avec les IoC de MISP [72]. L'analyste reçoit des cas pré-triés et enrichis plutôt qu'un flux brut d'alertes.

3.8.3 Protocole expérimental

Environnement de déploiement

L'évaluation a été conduite sur un VPS unique (4 vCPU Intel Xeon E5-2690 v4, 8 GB RAM, 80 GB NVMe, Ubuntu 22.04 LTS) pendant deux semaines (novembre–décembre 2024), avec des alertes réseau réelles complétées par des simulations d'attaques synthétiques dérivées du jeu de données CTU-13 [2].

Jeu de données

500 alertes Suricata couvrant : scans de ports (23%), anomalies de protocole (31%), attaques HTTP (18%), anomalies TCP (16%) et activités malveillantes détectées (12%). Les labels de vérité terrain ont été établis par consensus entre deux analystes seniors (accord inter-évaluateurs : Cohen's $\kappa = 0{,}87$).

3.8.4 Résultats

TABLE 3.5 – Performances de corrélation du pipeline TinyLlama-SOAR (500 alertes)

Métrique	**Valeur**
Alertes traitées	500
Groupes d'incidents identifiés	47
Taux de Vrais Positifs (TPR)	89,2%
Taux de Faux Positifs (FPR)	7,3%
Latence bout-en-bout (traitement actif)	11,7 s
Mémoire RAM maximale (pic)	4,3 GB

Ces résultats placent TinyLlama-SOAR à parité de TPR avec MCM-CASR (89,2%), tout en nécessitant 50 à 75% moins de mémoire (2–4 GB vs 8–16 GB) et en ajoutant une automatisation SOAR complète absente des contributions précédentes.

3.8.5 Positionnement dans l'architecture globale

TinyLlama-SOAR constitue la couche d'exécution du système d'alerte précoce :

1. MCM-CASR ou MCM-Llama corrèle les alertes et reconstruit les scénarios d'attaque.
2. GAT-DQN projette les phases futures de l'attaque.
3. Multi-Classification priorise et catégorise les incidents.
4. **TinyLlama-SOAR** orchestre la réponse : création des cas TheHive, analyse Cortex et isolation SentinelOne, le tout en moins de 15 minutes après détection.

Cette contribution transforme le système d'alerte précoce d'un outil d'analyse en un système de réponse autonome, fermant la boucle entre détection et remédiation.

3.9 Approche expérimentale transversale

Au-delà des évaluations spécifiques à chaque contribution, une analyse transversale des protocoles expérimentaux adoptés permet de mettre en lumière la cohérence de la démarche scientifique globale de cette thèse.

3.9.1 Datasets utilisés

Les cinq contributions utilisent des jeux de données complémentaires :

— MCM-CASR : UNSW-NB15 (2,5M flux, 49 features)

— MCM-Llama : CTU-13 (13 scénarios botnets, alertes Suricata)

— **GAT-DQN** : Graphes d'attaque synthétiques + MITRE ATT&CK

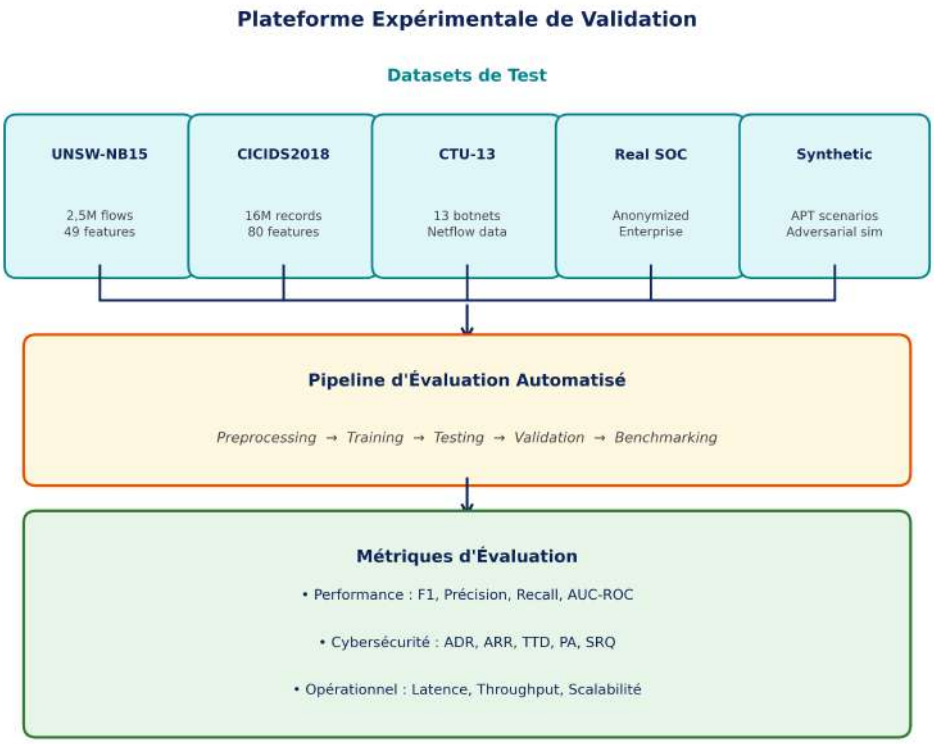


FIGURE 3.5 – Vue d'ensemble des jeux de données et protocoles expérimentaux utilisés dans les cinq contributions

— multi-Classification : CIC-IDS-2017 (2,3M flux, 50K alertes générées)

— TinyLlama-SOAR : 500 alertes Suricata réelles (trafic live + CTU-13 synthétique, nov.–déc. 2024)

Cette diversité de jeux de données assure la validation des approches dans des contextes variés et renforce leur robustesse [73, 74].

3.9.2 Métriques communes

Toutes les contributions sont évaluées selon des métriques standards :

— précision (Accuracy) : Proportion de prédictions correctes

— rappel (Recall) : Capacité à détecter les vrais positifs

— F1-Score : Moyenne harmonique de précision et rappel

— réduction d'alertes (ARR) : Taux de filtrage des faux positifs

— temps de traitement : Latence pour applications temps réel

3.9.3 Comparaisons avec l'état de l'art

Chaque contribution est systématiquement comparée aux approches de référence :

— MCM-CASR : vs. méthodes graphiques (GRAIN, AGCM), transformateurs, RL

— MCM-Llama : vs. MCM-CASR, méthodes basées sur la similarité

— GAT-DQN : vs. GNN purs, RL purs, méthodes hybrides existantes

— multi-Classification : vs. approches rule-based, statistiques, ML classiques

— TinyLlama-SOAR : vs. SIEM règles, corrélation graphe, MCM-CASR, SOAR traditionnel, détection par anomalie ML

Les résultats démontrent systématiquement des améliorations significatives par rapport aux baselines.

3.10 Intégration et synergies entre contributions

3.10.1 Complémentarité des approches

Les cinq contributions forment un système cohérent où chacune adresse une phase du cycle de vie de la détection et de la réponse aux attaques :

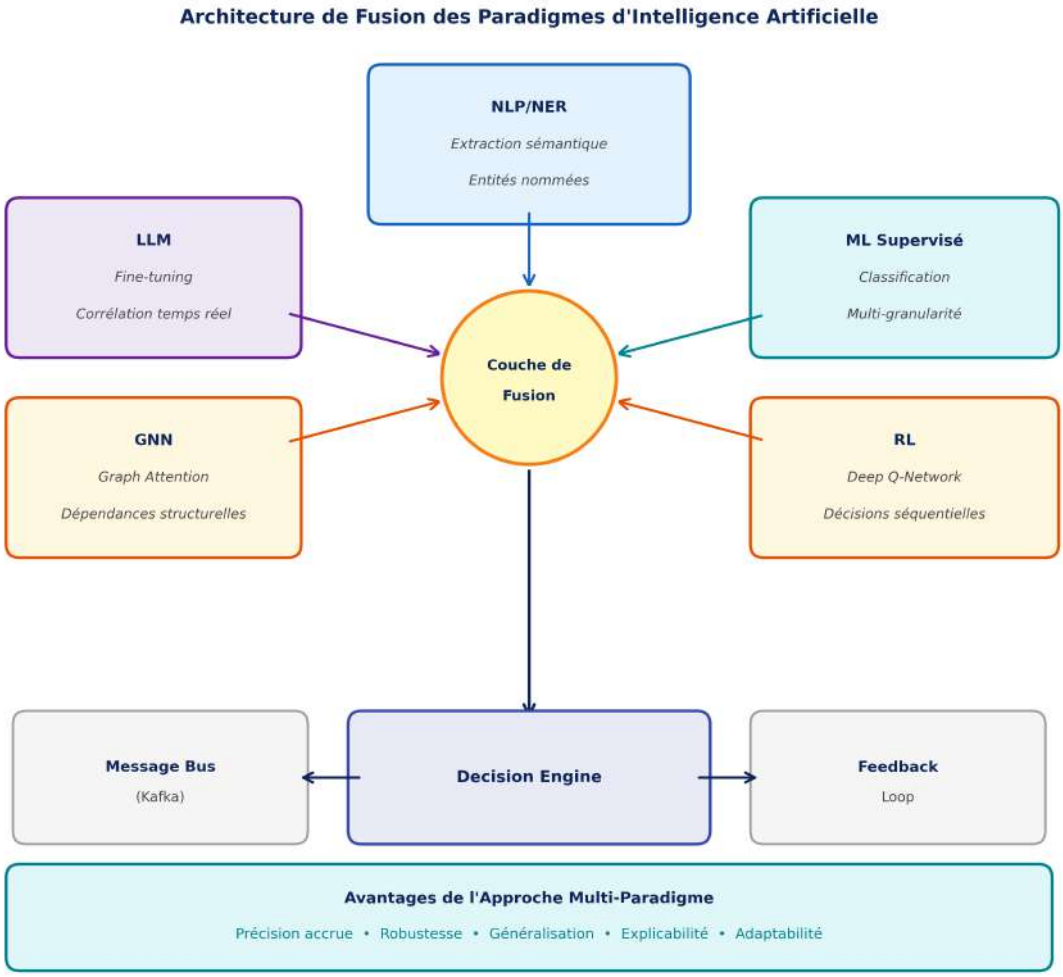


FIGURE 3.6 – Intégration des cinq contributions dans une architecture unifiée

TABLE 3.6 – Synergies entre les cinq contributions

Contribution	**Rôle principal**	**Synergie avec autres contributions**
MCM-CASR	Reconstruction offline de scénarios	Fournit des scénarios étiquetés pour entraîner Multi-Classification
MCM-Llama	Corrélation temps réel	Améliore la précision de MCM-CASR ; alimente GAT-DQN en temps réel
GAT-DQN	Projection d'attaques futures	Utilise les scénarios de MCM-CASR/Llama ; enrichit Multi-Classification
Multi-Classification	Classification et priorisation	Affine la granularité des sorties de MCM-CASR et MCM-Llama
TinyLlama-SOAR	Orchestration et réponse automatisée	Reçoit les groupes corrélés de MCM-Llama ; déclenche la réponse SOAR ; valide en environnement opérationnel réel les hypothèses des contributions 1–4

3.10.2 Pipeline opérationnel intégré

Un déploiement opérationnel complet intègre les cinq contributions :

1. ingestion : Collecte continue d'alertes IDS (Suricata, Snort) via Filebeat/Elasticsearch
2. corrélation temps réel : MCM-Llama corrèle les alertes en continu
3. classification : Multi-Classification catégorise les corrélations (binaire, types, incidents)
4. projection : GAT-DQN prédit les prochaines étapes d'attaques détectées
5. reconstruction offline : MCM-CASR analyse périodiquement les historiques pour découvrir des patterns latents
6. orchestration et réponse : TinyLlama-SOAR regroupe les alertes corrélées en incidents TheHive et déclenche l'isolation automatique des hôtes via SentinelOne
7. feedback : Les incidents clôturés et les isolations confirmées enrichissent les modèles de corrélation et la base de connaissance MISP

3.11 Considérations de déploiement

Au-delà de leurs performances académiques, les contributions de cette thèse ont vocation à être intégrées dans des environnements opérationnels réels. Cette section examine

les conditions pratiques de leur déploiement, en abordant successivement les exigences en infrastructure matérielle et logicielle, les contraintes opérationnelles à anticiper, ainsi que les mécanismes d'interopérabilité entre les différents modules du système.

3.11.1 Infrastructure requise

Le déploiement opérationnel nécessite :

— IDS/IPS : Suricata ou Snort pour génération d'alertes

— base de données : Elasticsearch ou PostgreSQL pour stockage alertes

— GPU : Pour inférence LLM (MCM-Llama) et entraînement GAT-DQN (optionnel pour TinyLlama-SOAR)

— CPU : Pour les algorithmes ML classiques (Multi-Classification) et TinyLlama-SOAR (4 GB RAM suffisants)

— SOAR : TheHive/Cortex pour l'orchestration ; SentinelOne EDR pour la réponse endpoint

— visualisation : Kibana, Grafana ou dashboard custom

3.11.2 Contraintes opérationnelles

— latence temps réel : MCM-Llama doit traiter les alertes en <1 seconde

— scalabilité : Support de milliers d'alertes par seconde en production

— maintenance : Mise à jour périodique des modèles avec nouvelles données

— explicabilité : Traçabilité des décisions pour audits de sécurité

3.11.3 Interopérabilité des modules et flux de données

L'intégration opérationnelle des cinq contributions requiert une définition précise des interfaces d'échange entre modules. Le Tableau 3.7 décrit les entrées, sorties et formats de chaque module dans la chaîne de traitement intégrée.

La communication entre modules s'effectue à travers une base de données intermédiaire (Elasticsearch ou PostgreSQL) jouant le rôle de bus de données partagé [13]. MCM-CASR et MCM-Llama alimentent cette base avec leurs scénarios corrélés, que GAT-DQN et Multi-Classification consomment pour leurs tâches respectives de projection et de classification. Cette architecture découplée permet à chaque module de fonctionner indépendamment, facilitant la maintenance et la mise à jour individuelle des composants sans perturber l'ensemble du pipeline.

Sur le plan de la latence cumulée, les expérimentations menées fournissent des ordres de grandeur pour chaque module : MCM-Llama traite une paire d'alertes en moins d'une

TABLE 3.7 – Interfaces d'échange entre les modules du système intégré

Module	**Entrée**	**Sortie**	**Format**
MCM-CASR	Alertes IDS brutes (texte)	Scénarios corrélés, groupes d'alertes ordonnés	JSON
MCM-Llama	Paires d'alertes (texte)	Décision binaire de corrélation + score de confiance	JSON
GAT-DQN	Groupes d'alertes corrélées	Phase d'attaque prédite (label + probabilité)	JSON
Multi-Classification	Paires/groupes + features	Type de corrélation + catégorie d'incident	CSV/JSON
TinyLlama-SOAR	Alertes EVE-JSON (Elasticsearch)	Cas TheHive + isolation SentinelOne + enrichissement Cortex	REST/JSON

seconde en mode inférence avec GPU, Multi-Classification classe une alerte en moins de 100 ms avec les modèles d'ensemble retenus, et GAT-DQN projette la phase suivante en 1 à 3 secondes selon la taille du graphe d'alertes. La latence totale de la chaîne complète reste ainsi compatible avec les exigences opérationnelles d'un SOC moderne pour les modules en temps réel (MCM-Llama, Multi-Classification), tandis que GAT-DQN intervient sur des groupes d'alertes déjà corrélées, ce qui atténue l'impact de sa latence sur le temps de réponse global.

3.12 Conclusion

Ce chapitre a présenté la méthodologie générale structurant les cinq contributions de cette thèse sur la détection précoce d'attaques par intelligence artificielle. Les points clés sont :

— approche progressive : Chaque contribution adresse une problématique spécifique (reconstruction, corrélation temps réel, projection, classification multi-granularité, opérationnalisation).

— complémentarité : Les cinq contributions forment un système cohérent couvrant l'ensemble du cycle de détection et de réponse aux attaques, depuis la corrélation sémantique jusqu'à l'isolation automatique des hôtes compromis.

— validation rigoureuse : Utilisation de jeux de données de référence (UNSW-NB15, CTU-13, CIC-IDS-2017, alertes Suricata réelles) et comparaisons systématiques avec l'état de l'art.

— performances exceptionnelles : Amélioration de 31 points de précision (MCM-Llama vs MCM-CASR), 99,27% de précision moyenne (Multi-Classification), projection adaptative à 98,95% (GAT-DQN), TPR de 89,2% avec latence de 11,7 s sur infrastructure légère (TinyLlama-SOAR).

— opérationnalité : Architectures conçues pour déploiement en environnements SOC réels, avec validation sur trafic live pour TinyLlama-SOAR.

Les chapitres suivants détaillent chacune de ces contributions avec leurs méthodologies complètes, implémentations, résultats expérimentaux et analyses comparatives approfondies. L'intégration de ces cinq contributions représente une avancée significative vers des systèmes d'alerte précoce intelligents, adaptatifs et hautement performants pour la cybersécurité moderne.

Chapitre 4

MCM-CASR : Un Cadre de Corrélation d'Alertes pour la Reconstruction de Scénarios de Cyberattaques basé sur le TALN, la REN et la Similarité Sémantique [1]

4.1 Introduction

Dans le contexte actuel de sophistication croissante des cybermenaces, la capacité à comprendre et reconstruire les scénarios d'attaque revêt une importance capitale pour la sécurisation des systèmes d'information. Les attaques modernes se caractérisent par leur nature multi-étapes, impliquant une séquence coordonnée d'actions malveillantes visant à compromettre progressivement les infrastructures ciblées [75]. Face à cette complexité, les Systèmes de Détection d'Intrusion (IDS) génèrent un volume considérable d'alertes, dont l'analyse manuelle s'avère impraticable pour les analystes de sécurité [18].

La problématique centrale adressée dans ce chapitre peut être formulée comme suit : *Comment automatiser la reconstruction de scénarios de cyberattaques à partir d'un flux massif d'alertes IDS tout en préservant la précision et l'interprétabilité des résultats ?*

Cette problématique se décline en trois sous-questions de recherche :

Q1 Comment réduire efficacement le volume d'alertes sans perdre d'information critique sur les attaques ?

Q2 Comment identifier et modéliser les relations sémantiques entre alertes hétérogènes ?

1. Ce chapitre est basé sur la publication : Diakhame, M.L., Diallo, C., Mejri, M. (2023). "MCM-CASR : Novel Alert Correlation Framework for Cyber Attack Scenario Reconstruction Based on NLP, NER, and Semantic Similarity". In *2023 7th Cyber Security in Networking Conference (CSNet)*, pp. 27–31. IEEE. DOI : 10.1109/CSNet59123.2023.10339769.

Q3 Comment reconstruire des scénarios d'attaque cohérents et interprétables à partir des corrélations identifiées ?

4.1.1 Contributions

Pour répondre à ces questions, nous proposons MCM-CASR, un cadre méthodologique articulé autour de trois contributions principales :

1. un mécanisme de réduction d'alertes basé sur des critères multi-attributs (fenêtre temporelle, catégorie d'attaque, adresses IP) permettant une réduction de 43,49% du volume d'alertes.
2. une approche de corrélation sémantique exploitant la Reconnaissance d'Entités Nommées (REN) pour l'extraction d'entités cybersécuritaires et le modèle BERT pour le calcul de similarité contextuelle.
3. une méthodologie de reconstruction de scénarios permettant d'identifier des séquences d'attaques multi-étapes et de révéler des schémas d'intrusion latents.

4.1.2 Organisation du chapitre

La suite de ce chapitre est structurée de manière à conduire le lecteur du contexte scientifique vers la validation expérimentale. Nous commençons par une revue critique des approches existantes (Section 4.2), qui permet de situer précisément MCM-CASR face aux paradigmes ontologiques, graphiques et d'apprentissage profond. Cette analyse comparative est suivie d'une présentation du jeu de données UNSW-NB15 (Section 4.3) et des fondements théoriques du TALN mobilisés (Section 4.4). L'architecture détaillée du cadre MCM-CASR est ensuite exposée (Section 4.5), avant une analyse comparative approfondie des performances (Section 4.6). Le chapitre se conclut par une synthèse des apports et des perspectives (Section 4.7).

4.2 Travaux connexes

La reconstruction de scénarios d'attaque a fait l'objet de nombreuses recherches au cours des dernières années. Cette section propose une analyse critique et structurée des approches existantes, organisée selon quatre paradigmes méthodologiques distincts.

4.2.1 Évolution historique des approches

La Figure 4.1 illustre l'évolution chronologique des méthodes de reconstruction de scénarios d'attaque de 2018 à 2025, mettant en évidence la transition progressive des approches traditionnelles basées sur l'ontologie vers les techniques modernes d'apprentissage profond.

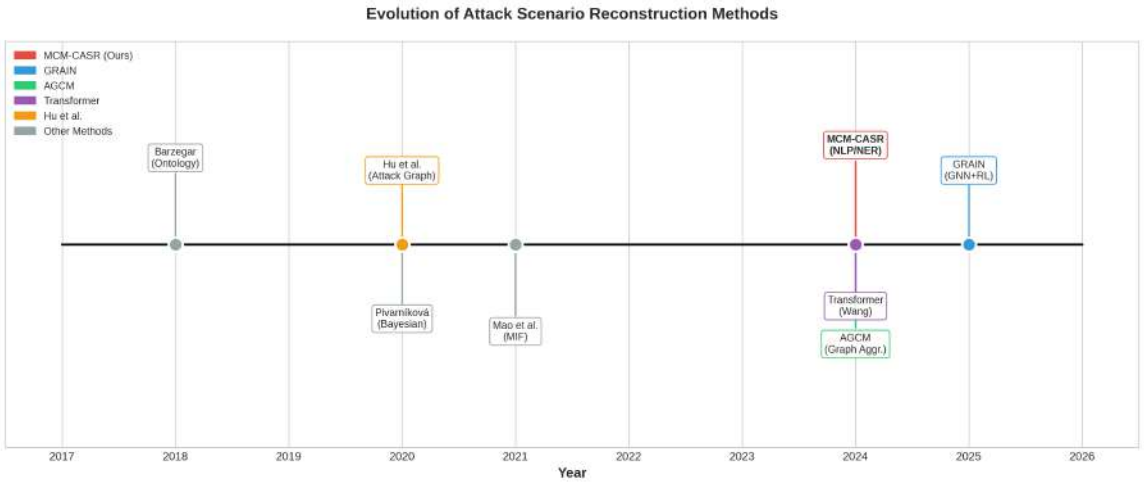


FIGURE 4.1 – Évolution des méthodes de reconstruction de scénarios d'attaque (2018-2025). La chronologie illustre la progression depuis les approches ontologiques et bayésiennes vers les réseaux neuronaux graphiques et les architectures de transformateurs.

Cette évolution peut être décomposée en trois grandes phases. Entre 2018 et 2020, les travaux dominants s'appuient sur des modèles ontologiques et des réseaux bayésiens, dont les points forts résident dans leur interprétabilité et leur ancrage dans la connaissance experte du domaine, mais dont les limites en passage à l'échelle et en généralisation ont rapidement été mises en évidence [76, 77]. De 2020 à 2022, on observe l'émergence des approches basées sur les graphes d'attaque et les méthodes de similarité sémantique, qui permettent de modéliser plus fidèlement les dépendances structurelles entre alertes [78, 79]. À partir de 2022, les architectures d'apprentissage profond — notamment les réseaux de neurones sur graphes et les transformateurs — s'imposent comme références en combinant représentation structurelle et compréhension contextuelle, ouvrant la voie aux contributions présentées dans ce chapitre [6, 7].

4.2.2 Approches basées sur l'ontologie et la sémantique

Les premières approches formelles de reconstruction de scénarios d'attaque reposent sur des modèles ontologiques permettant de capturer les connaissances du domaine de la cybersécurité.

Barzegar et Shajari [76] ont proposé une technique pionnière de reconstruction sémantique basée sur l'ontologie, exploitant une métrique de similarité sémantique pour identifier les relations causales entre alertes. Cette approche repose sur l'hypothèse fondamentale que la similarité entre deux étapes successives d'un scénario d'attaque est supérieure à celle entre étapes non successives. Formellement, pour deux alertes a_i et a_j appartenant au même scénario S, avec $i < j$:

$$\text{sim}(a_i, a_{i+1}) > \text{sim}(a_i, a_j) \quad \forall j > i + 1 \tag{4.1}$$

L'avantage principal de cette approche réside dans son indépendance vis-à-vis des connaissances expertes — c'est-à-dire des règles et modèles de menaces définis manuellement par des analystes en sécurité, tels que des bases de signatures d'attaques ou des arbres de scénarios prédéfinis —, permettant une reconstruction incrémentielle sans intervention humaine. Toutefois, elle présente des limitations en termes de passage à l'échelle pour les environnements à fort volume d'alertes.

4.2.3 Approches basées sur les graphes d'attaque

Les méthodes basées sur les graphes modélisent les alertes et leurs relations comme des structures graphiques, offrant une représentation naturelle des chemins d'attaque.

Mao et al. [77] ont développé une méthode de fusion multi-informations (MIF) intégrant le temps d'attaque, l'évaluation des risques et les informations topologiques. Leur architecture se compose de trois modules :

— le *Convolution and Agent Decision Tree Network* (CTnet) pour l'évaluation des risques via un réseau neuronal convolutif [16, 57] ;

— le *Graph-based Fusion Module* (GM) pour la création de scénarios pondérés [78, 58, 59] ;

— L'algorithme *Depth First Search with Time and Weight* (TW-DFS) pour l'extraction des chaînes d'attaque critiques [32, 78].

Hu et al. [78] ont proposé une approche complémentaire utilisant la boîte à outils MulVAL pour la génération automatique de graphes d'attaque. Leur contribution principale réside dans la prise en compte des chemins d'attaque infructueux et la détection des alertes de faux négatifs, permettant la fusion de scénarios fragmentés.

4.2.4 Approches par Apprentissage Profond

Les avancées récentes en apprentissage profond ont ouvert de nouvelles perspectives pour la reconstruction de scénarios d'attaque [16, 15].

GRAIN : réseaux neuronaux graphiques et apprentissage par renforcement

Xiao et al. [58] ont proposé GRAIN, une approche hybride combinant les Réseaux Neuronaux Graphiques (GNN) [60, 55] avec l'Apprentissage par Renforcement (RL) [61, 62] pour la découverte de causalité entre alertes. Cette combinaison s'inscrit dans une tendance plus large d'exploitation des structures graphiques pour la cybersécurité [42]. Le cadre opère sans connaissance préalable selon le processus suivant :

1. suppression des alertes redondantes par correspondance sémantique [80, 76] ;
2. corrélation des alertes sous forme de graphes causaux [78, 81] ;

3. évaluation de l'effet causal par GNN [60, 57] ;
4. filtrage des relations causales authentiques par RL [64, 62].

L'hypothèse sous-jacente stipule que les alertes déclenchées par des attaques multi-étapes présentent un effet causal maximal [81, 78], formalisé par :

$$\mathcal{C}^* = \arg\max_{\mathcal{C}\in\mathcal{G}} \sum_{(a_i,a_j)\in\mathcal{C}} \mathrm{CE}(a_i, a_j), \tag{4.2}$$

où $\mathcal{C}^*$ représente la chaîne causale optimale, $\mathcal{G}$ l'espace des graphes candidats et $\mathrm{CE}(a_i, a_j)$ l'effet causal entre les alertes a_i et a_j.

AGCM : agrégation de graphes

Lyu et al. [59] ont introduit AGCM, une méthode transformant les flux anormaux détectés par l'IDS en un graphe de relations d'états anormaux (ASR-graph). L'approche combine agrégation et clustering pour corréler les flux anormaux sans nécessiter de bases de connaissances préalables. L'algorithme APS-W permet ensuite l'extraction des chemins d'attaque critiques.

Architecture Transformer

Wang et al. [82] ont proposé un cadre basé sur l'architecture Transformateur pour l'agrégation d'alertes et la prédiction d'attaques. Le mécanisme d'attention multi-têtes permet de capturer les dépendances à longue portée entre alertes :

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{4.3}$$

où Q, K, V représentent respectivement les matrices de requêtes, clés et valeurs, et d_k la dimension des clés. Un score d'Estimation des Menaces Futures (FTE) complète le dispositif pour l'évaluation des menaces prédites.

4.2.5 Approches bayésiennes

Pivarníková et al. [79] ont développé une approche de détection précoce des cyberattaques combinant réduction d'alertes et corrélation via des réseaux bayésiens. Cette méthode permet la prédiction de la progression des attaques en catégorisant les alertes par étapes du cycle d'attaque.

4.2.6 Synthèse et positionnement

Le Tableau 4.1 synthétise les caractéristiques principales des approches analysées et positionne notre contribution MCM-CASR.

TABLE 4.1 – Synthèse comparative des approches de reconstruction de scénarios d'attaque

Approche	**Paradigme**	**Connaissance**	**Temps réel**	**Interprét.**
Barzegar [76]	Ontologie	Non	Non	Élevée
Mao [77]	Graphe + DL	Non	Non	Moyenne
Hu [78]	Graphe	Oui	Non	Moyenne
GRAIN [58]	GNN + RL	Non	Non	Faible
AGCM [59]	Graphe	Non	Oui	Moyenne
Transformer [82]	Séquence	Partielle	Oui	Faible
MCM-CASR	**TALN**	**Non**	**Non**	**Élevée**

Notre approche MCM-CASR se distingue par son exploitation des techniques de TALN pour l'analyse sémantique du contenu des alertes, offrant une interprétabilité élevée grâce à l'identification explicite des entités nommées. Contrairement aux approches par apprentissage profond (GRAIN, Transformateur), notre méthode ne nécessite pas de phase d'entraînement sur des données annotées, ce qui facilite son déploiement dans de nouveaux environnements.

Il convient de souligner que la comparaison quantitative directe entre ces approches reste limitée par l'hétérogénéité des jeux de données et des métriques utilisés dans chaque travail. Barzegar et Shajari [76] évaluent leur approche sur des scénarios ontologiques spécifiques sans jeu de données standardisé public ; Mao et al. [77] rapportent des résultats sur des environnements réseau propriétaires ; GRAIN [58] et AGCM [59] sont évalués sur des traces réseau internes sans benchmarks communs. L'absence d'un protocole d'évaluation unifié est d'ailleurs reconnue comme une lacune dans la communauté de la corrélation d'alertes [18]. Notre évaluation sur le jeu de données UNSW-NB15 [68], référence reconnue de la communauté, avec une précision de corrélation de 53,91% et une amélioration de 10,59 points par rapport à la méthode de base évaluée sur le même ensemble, offre ainsi un positionnement rigoureux et reproductible. Cette démarche d'évaluation standardisée renforce la crédibilité et la comparabilité des résultats obtenus.

4.3 Jeu de données

L'évaluation expérimentale de MCM-CASR s'appuie sur le jeu de données UNSW-NB15 [68], un ensemble de référence largement adopté dans la communauté de recherche en cybersécurité. Développé par l'Université de Nouvelle-Galles du Sud (UNSW), ce jeu de données répond au besoin de données réalistes et représentatives pour l'évaluation des systèmes de détection d'intrusion.

4.3.1 Caractéristiques du jeu de données

Le Tableau 4.2 présente les caractéristiques principales du jeu de données UNSW-NB15 [68, 69].

TABLE 4.2 – Caractéristiques du jeu de données UNSW-NB15

Caractéristique	**Valeur**
Nombre total d'enregistrements	≈ 2 540 000
Nombre de caractéristiques	49
Catégories d'attaque	9
Trafic normal	2 218 761
Trafic malveillant	321 283
Période de capture	16 heures

Ce tableau appelle plusieurs observations. Avec près de 2,54 millions d'enregistrements capturés sur une période de 16 heures, le jeu de données offre un volume suffisant pour évaluer le comportement des algorithmes de corrélation en conditions de charge réaliste. Le ratio trafic normal/malveillant (environ 87%/13%) reflète un déséquilibre de classes typique des environnements opérationnels, constituant un défi supplémentaire pour les mécanismes de filtrage [83, 16]. Les 49 caractéristiques couvrent des dimensions réseau, temporelles et comportementales, permettant une modélisation riche des flux. Enfin, les 9 catégories d'attaques représentent un spectre varié de menaces, allant des activités de reconnaissance aux intrusions les plus sophistiquées, ce qui permet d'évaluer la généralisation de MCM-CASR sur des scénarios hétérogènes [69, 17].

4.3.2 Catégories d'attaques

Le jeu de données couvre neuf catégories d'attaques distinctes, représentant un spectre diversifié de menaces :

1. fuzzers : Tentatives de découverte de vulnérabilités par injection de données aléatoires ;
2. analysis : Activités de reconnaissance et d'analyse de vulnérabilités ;
3. backdoors : Installation de portes dérobées pour un accès persistant ;
4. doS : Attaques par déni de service visant l'indisponibilité ;
5. exploits : Exploitation de vulnérabilités connues ;
6. generic : Attaques génériques exploitant des faiblesses communes ;
7. reconnaissance : Activités de découverte et d'énumération ;
8. shellcode : Injection et exécution de code malveillant ;
9. worms : Propagation automatisée de logiciels malveillants.

4.3.3 Justification du choix

Le choix du jeu de données UNSW-NB15 se justifie par plusieurs critères :

— réalisme : Données générées dans un environnement hybride combinant trafic réel et synthétique [69] ;

— diversité : Couverture de multiples catégories d'attaques représentatives des menaces actuelles ;

— richesse attributaire : 49 caractéristiques permettant une analyse multi-dimensionnelle [83] ;

— comparabilité : Adoption généralisée facilitant la comparaison avec les travaux existants.

4.3.4 Limitations

Il convient de noter certaines limitations inhérentes à ce jeu de données. Premièrement, UNSW-NB15 ne propose pas de séquençage temporel explicite des attaques multi-étapes, ce qui rend difficile l'évaluation directe des capacités de reconstruction de scénarios ordonnés dans le temps. Deuxièmement, le déséquilibre marqué entre les classes — le trafic normal représentant environ 87% des enregistrements — peut biaiser certaines métriques d'évaluation et nécessite des précautions dans l'interprétation des taux de détection. Enfin, l'ancienneté relative des profils d'attaque (collectés en 2015) implique que certaines menaces récentes, telles que les attaques par rançongiciel ou les techniques d'évasion modernes, ne sont pas représentées. Ces limitations seront prises en compte dans l'interprétation des résultats expérimentaux.

4.4 Fondements théoriques du TALN pour la cybersécurité

Cette section présente les concepts fondamentaux du Traitement Automatique du Langage Naturel (TALN) mobilisés dans le cadre MCM-CASR. Trois notions clés sont examinées : la reconnaissance d'entités nommées pour l'extraction d'informations cybersécuritaires, les représentations vectorielles pour la modélisation sémantique des alertes, et le modèle BERT pour le calcul de similarité contextuelle. Leur articulation constitue le socle théorique sur lequel repose notre approche de corrélation.

4.4.1 Introduction au traitement automatique du langage naturel

Le Traitement Automatique du Langage Naturel (TALN) [84] constitue un sous-domaine de l'intelligence artificielle visant à permettre aux systèmes informatiques de

comprendre, interpréter et générer le langage humain. Dans le contexte de la cybersécurité, les techniques de TALN offrent des perspectives prometteuses pour l'analyse automatisée de données textuelles hétérogènes : alertes de sécurité, rapports d'incidents, flux de renseignements sur les menaces et journaux système [12, 15].

4.4.2 Reconnaissance d'entités nommées (REN)

Définition et principes

La Reconnaissance d'Entités Nommées (REN) [46] désigne la tâche consistant à identifier et classifier automatiquement les mentions d'entités dans un texte. Dans le domaine de la cybersécurité, les entités pertinentes incluent :

- entités réseau : adresses IP, noms de domaine, ports, protocoles ;
- entités système : noms de fichiers, processus, services ;
- entités temporelles : horodatages, durées ;
- **Entités de menace** : identifiants CVE, noms de malware, tactiques MITRE ATT&CK.

Formalisation

Soit $T = (t_1, t_2, \ldots, t_n)$ une séquence de tokens représentant une alerte de sécurité. La tâche de REN consiste à attribuer à chaque token t_i une étiquette $y_i \in \mathcal{Y}$, où $\mathcal{Y}$ représente l'ensemble des types d'entités (incluant la classe `O` pour les tokens hors-entité). Formellement :

$$f_{\text{REN}} : T \to \mathcal{Y}^n \tag{4.4}$$

Les approches modernes de REN s'appuient sur des architectures neuronales, notamment les modèles de langue pré-entraînés (BERT, RoBERTa) fine-tunés sur des corpus annotés du domaine cybersécuritaire [12].

4.4.3 Similarité sémantique

Définition et principes

La similarité sémantique [47] vise à quantifier le degré de proximité entre deux textes sur la base de leur signification, indépendamment de leur formulation lexicale. Cette capacité est fondamentale pour la corrélation d'alertes, permettant d'identifier des événements sémantiquement liés malgré des formulations distinctes.

Représentations vectorielles

Les approches modernes de similarité sémantique reposent sur la représentation des textes dans un espace vectoriel de grande dimension. Soit $\phi : \mathcal{T} \to \mathbb{R}^d$ une fonction d'encodage mappant un texte vers un vecteur de dimension d. La similarité entre deux textes T_1 et T_2 peut alors être calculée par la similarité cosinus :

$$\text{sim}_{\cos}(T_1, T_2) = \frac{\phi(T_1) \cdot \phi(T_2)}{\|\phi(T_1)\| \cdot \|\phi(T_2)\|} \tag{4.5}$$

Modèle BERT

Notre approche exploite BERT (*Bidirectional Encoder Representations from Transformers*) [85] pour le calcul de similarité sémantique. BERT utilise un mécanisme d'attention bidirectionnel permettant de capturer le contexte gauche et droit simultanément :

$$\mathbf{h}_i = \text{Transformer}(\mathbf{x}_1, \ldots, \mathbf{x}_n)_i, \tag{4.6}$$

où $\mathbf{h}_i$ représente la représentation contextuelle du token i. Pour obtenir une représentation au niveau de la phrase, nous utilisons le token spécial `[CLS]` ou une agrégation des représentations de tous les tokens.

Les fondements théoriques présentés ci-dessus — extraction d'entités nommées, représentations vectorielles et similarité sémantique par BERT — forment les briques élémentaires à partir desquelles nous construisons le cadre MCM-CASR. La section suivante décrit comment ces composants s'articulent en un pipeline cohérent de réduction et de corrélation des alertes IDS.

4.5 Cadre MCM-CASR

Cette section présente l'architecture et les composants du cadre MCM-CASR pour la reconstruction de scénarios de cyberattaques.

4.5.1 Architecture générale

Le cadre MCM-CASR s'articule autour de deux modules principaux opérant de manière séquentielle :

1. module de Réduction des Alertes : Filtrage et agrégation des alertes redondantes ;
2. module de Corrélation des Alertes : Établissement des relations sémantiques entre alertes et reconstruction des scénarios.

La Figure 4.2 illustre l'architecture globale du cadre et les interactions entre ses composants.

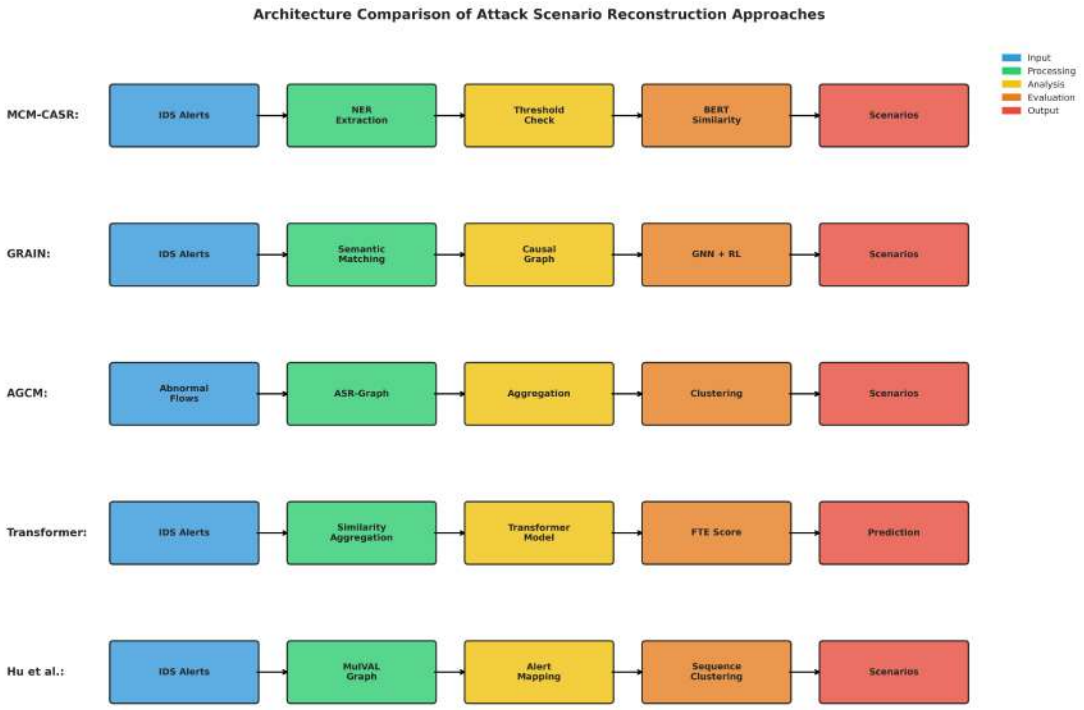


FIGURE 4.2 – Architecture du cadre MCM-CASR. Le pipeline comprend : (1) l'ingestion des alertes IDS, (2) la réduction par critères multi-attributs, (3) l'extraction d'entités par REN, (4) le calcul de similarité par BERT, et (5) la reconstruction des scénarios d'attaque.

4.5.2 Module de réduction des alertes

Problématique

Dans les environnements réseau complexes, les systèmes de sécurité génèrent un volume considérable d'alertes, dont une proportion significative s'avère redondante ou de faible priorité. Cette surcharge informationnelle (*alert fatigue*) constitue un obstacle majeur à l'efficacité des opérations de sécurité [79, 19].

Algorithme de Réduction

Nous adoptons l'algorithme de réduction proposé par Pivarníková et al. [79], qui procède par fusion des alertes satisfaisant conjointement trois critères :

[Critères de fusion d'alertes] Deux alertes a_1 et a_2 sont fusionnées en une alerte unique si et seulement si les trois conditions suivantes sont simultanément satisfaites :

1. proximité temporelle : $|t(a_1) - t(a_2)| \leq \Delta t$, avec $\Delta t = 60$ minutes ;
2. identité catégorielle : $\text{cat}(a_1) = \text{cat}(a_2)$;
3. identité des adresses IP : $\text{srcIP}(a_1) = \text{srcIP}(a_2) \wedge \text{dstIP}(a_1) = \text{dstIP}(a_2)$.

Résultats de la réduction

Le Tableau 4.3 présente les résultats de la réduction des alertes par catégorie d'attaque.

TABLE 4.3 – Résultats de la réduction des alertes par catégorie

Catégorie	**Avant**	**Après**	**Taux (%)**
Generic	7 498	3 731	50,24
Exploits	4 363	3 594	17,63
Fuzzers	4 061	1 725	57,52
Reconnaissance	1 733	558	67,80
DoS	571	540	5,43
Shellcode	223	223	0,00
Analysis	58	56	3,45
Backdoors	46	46	0,00
Worms	24	24	0,00
Total	**18 577**	**10 497**	**43,49**

Analyse des résultats

L'analyse des taux de réduction révèle une hétérogénéité significative selon les catégories d'attaque :

— forte réduction ($> 50\%$) : Reconnaissance (67,80%), Fuzzers (57,52%), Generic (50,24%). Ces catégories génèrent typiquement des alertes répétitives lors de phases de balayage.

— réduction modérée (5-20%) : Exploits (17,63%), DoS (5,43%), Analysis (3,45%). La diversité des cibles et techniques limite la redondance.

— réduction nulle : Shellcode, Backdoors, Worms. Ces attaques, moins fréquentes, génèrent des alertes uniques.

Le taux de réduction global de 43,49% traduit un équilibre satisfaisant entre élimination de la redondance et préservation de l'information critique.

4.5.3 Module de corrélation des alertes

Objectif

La corrélation des alertes vise à établir des relations entre alertes individuelles afin d'identifier des schémas, séquences ou dépendances révélateurs d'incidents de sécurité plus significatifs ou d'attaques coordonnées.

Pipeline de corrélation

Notre méthodologie de corrélation s'articule en trois étapes séquentielles :

Étape 1 : Extraction d'entités nommées Pour chaque alerte a, nous appliquons un modèle de REN pour extraire l'ensemble des entités pertinentes :

$$\mathcal{E}(a) = \text{REN}(a) = \{e_1, e_2, \ldots, e_k\}, \tag{4.7}$$

TABLE 4.4 – Exemple de scénario de cyberattaque reconstruit

Étape	srcip	dstip	proto	Catégorie
1	175.45.176.1	149.171.126.10	udp	Reconnaissance
2	175.45.176.0	149.171.126.10	udp	Reconnaissance
3	175.45.176.3	149.171.126.10	udp	DoS
4	175.45.176.0	149.171.126.10	udp	DoS
5	175.45.176.1	149.171.126.10	udp	Exploits
6	175.45.176.1	149.171.126.10	udp	Exploits

où $\mathcal{E}(a)$ représente l'ensemble des entités identifiées (adresses IP, noms de domaine, identifiants, etc.).

Étape 2 : Vérification du seuil d'entités communes Pour deux alertes a_i et a_j, nous vérifions si le nombre d'entités communes atteint un seuil minimal θ :

$$|\mathcal{E}(a_i) \cap \mathcal{E}(a_j)| \geq \theta \tag{4.8}$$

Ce filtrage préliminaire permet d'écarter les paires d'alertes manifestement non corrélées et de réduire la complexité computationnelle.

Étape 3 : Calcul de similarité sémantique Pour les paires satisfaisant le critère de seuil, nous calculons la similarité sémantique via BERT :

$$s(a_i, a_j) = \text{sim}_{\cos}(\phi_{\text{BERT}}(a_i), \phi_{\text{BERT}}(a_j)), \tag{4.9}$$

où ϕ_{BERT} représente l'encodage BERT de l'alerte. Deux alertes sont considérées corrélées si $s(a_i, a_j) \geq \tau$, où τ est un seuil de similarité prédéfini.

Exemple de scénario

Le Tableau 4.4 illustre un scénario de cyberattaque multi-étapes reconstruit par notre approche.

Ce scénario illustre une séquence d'attaque typique : phase de reconnaissance initiale (étapes 1-2), suivie d'une tentative de déni de service (étapes 3-4), puis d'une exploitation de vulnérabilités (étapes 5-6).

Résultats de la corrélation

Le Tableau 4.5 présente les performances comparatives de notre méthode.

Notre approche MCM-CASR atteint une précision de 53,91% dans l'association des paquets à leurs scénarios respectifs, soit une amélioration de 10,59 points de pourcentage

TABLE 4.5 – Résultats comparatifs de la corrélation des alertes

Méthode	**Précision**	**Scénarios**	**Sous-scénarios**
MCM-CASR	53,91%	10	9
Pivarníková [79]	43,32%	1	1

par rapport à la méthode de référence [79]. De plus, MCM-CASR reconstruit 10 scénarios complets et 9 sous-scénarios, contre 1 scénario et 1 sous-scénario pour l'approche de base.

Avantages de l'approche

Notre méthode de corrélation basée sur le TALN présente plusieurs avantages :

1. compréhension contextuelle : L'analyse TALN capture le sens sémantique au-delà des correspondances lexicales exactes.
2. granularité entitaire : L'extraction REN permet une analyse fine des entités cybersécuritaires.
3. identification de schémas : La corrélation basée sur les entités révèle des motifs d'attaque récurrents.
4. interprétabilité : Les entités extraites fournissent des explications tangibles aux corrélations identifiées.

4.6 Analyse comparative

Pour positionner MCM-CASR dans l'état de l'art, nous conduisons une analyse comparative approfondie avec quatre approches récentes : GRAIN [58], AGCM [59], le cadre Transformateur [82] et la méthode de Hu et al. [78].

4.6.1 Comparaison méthodologique

Le Tableau 4.6 présente une comparaison détaillée des aspects méthodologiques de chaque approche.

Ce tableau met en évidence plusieurs contrastes significatifs entre les approches. MCM-CASR est la seule méthode reposant exclusivement sur le TALN et la similarité sémantique via BERT, ce qui lui confère une indépendance totale vis-à-vis des bases de connaissances expertes tout en maintenant une capacité de réduction des alertes de 43,49%. GRAIN et AGCM, fondées respectivement sur les GNN couplés à l'apprentissage par renforcement et sur l'agrégation de graphes, se distinguent par leur capacité à détecter des attaques inconnues sans dépendance à l'expertise, au prix d'une interprétabilité réduite. Le cadre Transformateur présente quant à lui une dépendance partielle à l'expertise, compensée par sa flexibilité face à des entrées de longueur variable. Enfin, la méthode de Hu et al. se

TABLE 4.6 – Comparaison méthodologique des approches de reconstruction de scénarios d'attaque

Méthode	Technique principale	Dépendance expertise	Attaque inconnue	Réduction alertes	Innovation clé
MCM-CASR	TALN/REN + BERT	Non	Partielle	Oui (43,49%)	Corrélation sémantique
GRAIN [58]	GNN + RL	Non	Oui	Oui	Découverte de causalité
AGCM [59]	Agrég. graphe + Clustering	Non	Oui	Oui	Représentation ASR-graph
Transformer [82]	Transformer + Similarité	Partielle	Oui	Oui	Entrées longueur variable
Hu et al. [78]	Graphe attaque + Fouille	Oui (MulVAL)	Non	Partielle	Détection faux négatifs

démarque par sa capacité à détecter les faux négatifs, mais reste limitée par sa dépendance à l'outil MulVAL et son incapacité à identifier des attaques inconnues.

4.6.2 Taxonomie des paradigmes

Les approches analysées peuvent être classifiées selon quatre paradigmes distincts :

Paradigme textuel (MCM-CASR)

Ce paradigme exploite le traitement du langage naturel pour extraire l'information sémantique du contenu des alertes. Il excelle dans la compréhension contextuelle et l'identification de relations basées sur la similarité textuelle plutôt que sur des règles prédéfinies.

Paradigme graphique (GRAIN, AGCM, Hu et al.)

Ce paradigme modélise les alertes et leurs relations comme des structures de graphe. Les GNN de GRAIN permettent l'apprentissage de relations complexes, tandis que l'ASR-graph d'AGCM offre une représentation efficace des relations d'états anormaux.

Paradigme séquentiel (Transformer)

Ce paradigme traite les flux d'alertes comme des séquences et exploite les mécanismes d'attention pour capturer les dépendances à longue portée.

Paradigme hybride

Ce paradigme combine plusieurs techniques, comme l'intégration GNN + RL de GRAIN pour la découverte de causalité.

TABLE 4.7 – Forces de MCM-CASR

Force	**Description**
Indépendance	Ne nécessite aucune base de connaissances externe ni ontologie préalable
Extraction sémantique	Identification efficace des entités cybersécuritaires (IP, CVE, malware) par REN
Interprétabilité	Les entités REN explicites offrent une traçabilité complète des décisions de corrélation
Réduction d'alertes	Taux de réduction de 43,49% limitant la charge d'analyse
Facilité de déploiement	Opérationnel sans phase d'entraînement supervisé sur des données annotées

TABLE 4.8 – Comparaison quantitative des performances

Méthode	**Jeu(x) de données**	**Précision**	**Scénarios**	**Métriques clés**
MCM-CASR	UNSW-NB15	53,91%	10 + 9	Réduction : 43,49%
GRAIN [58]	4 jeux publics	Sup. SOTA	Multiples	Précision & Efficacité
AGCM [59]	Multi-étapes	Élevée	Chemins mult.	Extraction chemins critiques
Transformer [82]	Personnalisé	Bonne	Chaînes mult.	Score FTE
Hu et al. [78]	DARPA, Defcon	43,32%*	1 + 1	Restauration complète

*Résultat de l'évaluation comparative sur UNSW-NB15

4.6.3 Analyse des forces et limites

Forces de MCM-CASR

Limites de MCM-CASR

MCM-CASR présente néanmoins plusieurs limitations. La précision de corrélation, de 53,91%, reste modérée comparativement aux approches par apprentissage profond, en raison d'une analyse restreinte au contenu textuel des alertes sans exploitation de la topologie réseau. De plus, les expérimentations ont été conduites sur un seul jeu de données (UNSW-NB15), et le cadre fonctionne actuellement hors ligne, sans capacité de traitement en temps réel.

4.6.4 Comparaison quantitative

Le Tableau 4.8 synthétise les métriques de performance quantitatives.

Note méthodologique : La comparaison quantitative directe demeure délicate en raison

TABLE 4.9 – Comparaison fonctionnelle des approches

Fonctionnalité	**MCM**	**GR.**	**AG.**	**Tr.**	**Hu**
Indép. connaissance	✓	✓	✓	∼	×
Réduction alertes	✓	✓	✓	✓	∼
Analyse sémantique	✓	✓	×	✓	×
Découverte causalité	×	✓	×	×	×
Représentation graphe	×	✓	✓	×	✓
Prédiction attaque	×	×	×	✓	×
Détection faux négatifs	×	×	×	×	✓
Multi-jeux données	×	✓	×	×	✓
Temps réel	×	×	✓	✓	×
Interprétabilité	✓	×	∼	×	∼

MCM=MCM-CASR, GR.=GRAIN, AG.=AGCM, Tr.=Transformer, ∼=Partiel

de l'hétérogénéité des jeux de données utilisés. Les résultats rapportés pour GRAIN, AGCM et le Transformateur proviennent de leurs publications originales sur des jeux de données distincts. Seule la comparaison MCM-CASR vs Hu et al. [79] a été conduite sur le même jeu de données (UNSW-NB15).

4.6.5 Comparaison fonctionnelle

Le Tableau 4.9 présente une comparaison fonctionnelle détaillée.

Ce tableau fonctionnel met en relief les avantages distinctifs de chaque approche. MCM-CASR et GRAIN partagent l'indépendance vis-à-vis des connaissances expertes et la capacité de réduction des alertes, mais GRAIN va plus loin en offrant en plus la découverte de causalité et la représentation graphique, au détriment de l'interprétabilité. AGCM se démarque par sa capacité de traitement en temps réel et sa représentation graphique, mais n'assure ni analyse sémantique ni découverte de causalité. Le cadre Transformateur est le seul à proposer une fonctionnalité de prédiction d'attaque et un traitement en temps réel, bien qu'avec une dépendance partielle à l'expertise. Hu et al. reste la seule méthode capable de détecter les faux négatifs et d'être évaluée sur plusieurs jeux de données, au prix d'une dépendance forte à l'outil MulVAL. MCM-CASR se distingue par son interprétabilité complète, fonctionnalité absente ou partielle chez toutes les autres approches.

4.6.6 Analyse des techniques utilisées

L'examen des techniques mobilisées révèle des positionnements très différenciés selon les approches. MCM-CASR se distingue par une forte orientation vers le traitement du langage naturel et la similarité sémantique, là où GRAIN et AGCM investissent davantage les représentations graphiques et l'apprentissage profond. Cette cartographie technique permet de comprendre les complémentarités potentielles entre méthodes.

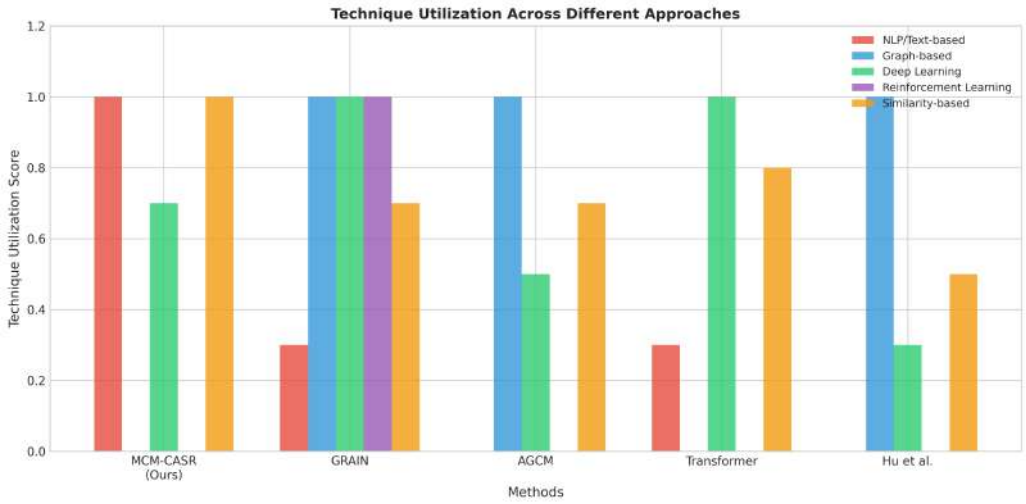


FIGURE 4.3 – Utilisation des techniques par approche. Le diagramme illustre l'intensité d'utilisation de chaque famille de techniques : TALN/Texte, Graphes, Apprentissage profond, Apprentissage par renforcement et Similarité.

La Figure 4.3 présente une visualisation comparative des techniques employées par chaque approche.

La figure illustre clairement la spécialisation de chaque approche. MCM-CASR mobilise exclusivement les techniques TALN et de similarité sémantique, sans recourir à l'apprentissage profond ni aux représentations graphiques. GRAIN présente le profil le plus hybride, combinant GNN et apprentissage par renforcement à parts égales. AGCM se concentre sur les graphes et le clustering, tandis que le cadre Transformateur exploite principalement l'apprentissage profond et la similarité. Hu et al. occupe une position intermédiaire, articulant graphes d'attaque et techniques de fouille de données. Cette diversité des positionnements techniques confirme qu'aucune famille de méthodes ne couvre à elle seule l'ensemble des dimensions nécessaires à une reconstruction robuste de scénarios d'attaque.

4.6.7 Profil comparatif multi-dimensionnel

La Figure 4.4 présente un graphique radar comparant les capacités selon huit dimensions d'évaluation. Ce profil met en évidence que MCM-CASR excelle en analyse sémantique et interprétabilité, tandis que GRAIN se distingue en découverte de causalité. Aucune approche ne domine toutes les dimensions, confirmant l'intérêt d'une approche hybride à terme.

Le graphique radar révèle que MCM-CASR domine sur les dimensions d'analyse sémantique, d'interprétabilité et d'indépendance vis-à-vis des connaissances, ce qui en fait un choix privilégié pour les environnements où l'explicabilité des décisions est primordiale. GRAIN se distingue nettement sur la découverte de causalité et la détection d'attaques inconnues, au détriment de l'interprétabilité. AGCM présente un profil équilibré sur les

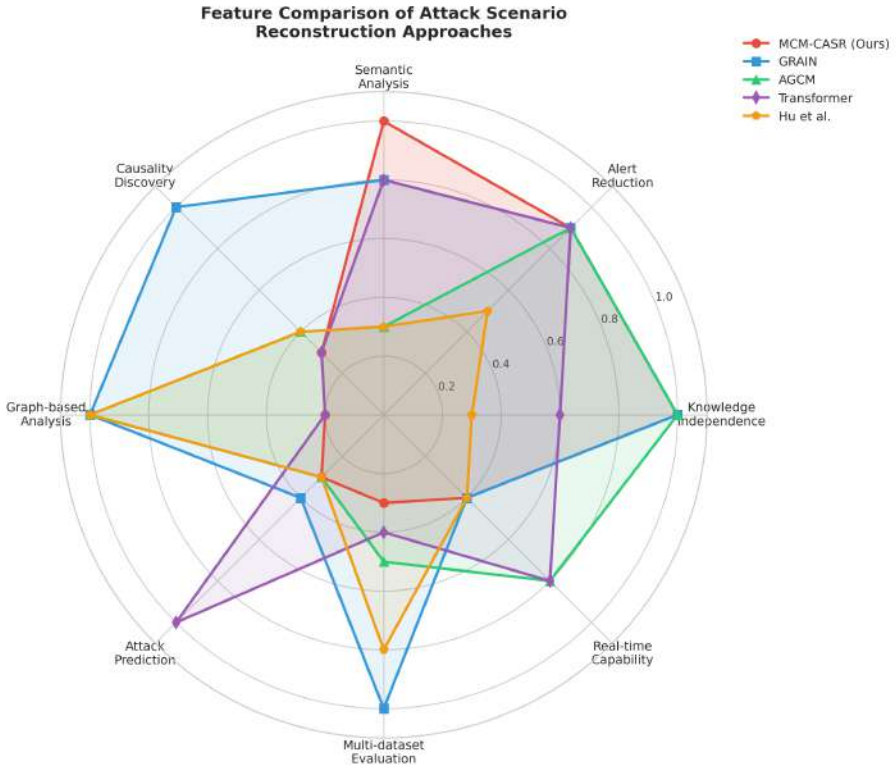


FIGURE 4.4 – Profil comparatif multi-dimensionnel. MCM-CASR (rouge) présente des forces en analyse sémantique et indépendance vis-à-vis des connaissances, tandis que GRAIN excelle en découverte de causalité.

dimensions structurelles, notamment pour l'extraction de chemins d'attaque. Le cadre Transformateur se montre compétitif sur la flexibilité et la prise en charge des séquences longues. Aucune approche ne domine l'ensemble des huit dimensions, ce qui souligne la complémentarité des paradigmes et justifie l'intérêt d'approches hybrides pour des déploiements opérationnels complets.

4.6.8 Synthèse visuelle

La Figure 4.5 fournit un résumé visuel de la comparaison complète entre les cinq approches.

4.6.9 Discussion et perspectives

L'analyse comparative conduite permet de dégager plusieurs enseignements :

Complémentarité des approches

Les méthodes basées sur les graphes (GRAIN, AGCM) excellent dans la capture des relations structurelles mais peuvent occulter les nuances sémantiques présentes dans le contenu des alertes. Réciproquement, notre approche TALN capture les relations séman-

Comprehensive Comparison of Attack Scenario Reconstruction Methods

Criterion	MCM-CASR	GRAIN	AGCM	Transformer	Hu et al.
Main Technique	NLP/NER/BERT	GNN + RL	Graph Aggr.	Transformer	Attack Graph
Expertise Dependency	No	No	No	Partial	Yes
Unknown Attack Detection	Partial	Yes	Yes	Yes	No
Alert Reduction	43.49%	High	High	High	Partial
Causality Discovery	No	Yes	No	No	No
Dataset(s)	UNSW-NB15	4 Public	Multi-step	Custom	DARPA/CTF23
Correlation Accuracy	53.91%	Superior	High	Good	43.32%
Scenarios Reconstructed	10 + 9 sub	Multiple	Multiple	Multiple	1 + 1 sub
Real-time Capability	Offline	Offline	Online	Online	Offline
Key Innovation	Semantic NER	Causality	ASR-Graph	FTE Score	True Neg.

FIGURE 4.5 – Synthèse visuelle de la comparaison des méthodes de reconstruction de scénarios d'attaque. Ce tableau récapitule la technique principale, la dépendance à l'expertise, les capacités de détection, les taux de réduction des alertes, les jeux de données utilisés, la précision de corrélation, les scénarios reconstruits, la capacité temps réel et l'innovation clé pour chaque approche.

tiques mais n'exploite pas pleinement l'information topologique du réseau. Cette complémentarité suggère qu'une approche hybride combinant analyse sémantique (MCM-CASR) et découverte de causalité graphique (GRAIN) pourrait produire des résultats supérieurs.

Enjeux de la reproductibilité

L'hétérogénéité des jeux de données utilisés constitue un obstacle majeur à la comparaison rigoureuse des approches. L'établissement de benchmarks standardisés incluant des scénarios d'attaque annotés faciliterait les évaluations comparatives futures.

Considérations pratiques

L'indépendance de MCM-CASR vis-à-vis des bases de connaissances externes et la nature interprétable des résultats (entités REN explicites) le rendent particulièrement adapté aux environnements opérationnels où l'explicabilité est cruciale pour les analystes de sécurité.

Directions de recherche

Plusieurs pistes d'amélioration se dégagent de cette analyse. Une première direction prometteuse consiste à intégrer les mécanismes de découverte de causalité graphique de GRAIN à l'analyse sémantique de MCM-CASR, afin de combiner les forces des deux

paradigmes dans un cadre hybride. Une deuxième piste porte sur l'extension vers le traitement en temps réel, rendue possible par l'optimisation du pipeline de corrélation pour un fonctionnement en flux continu, indispensable dans les SOC opérationnels. Par ailleurs, l'évaluation sur des jeux de données multiples et récents — tels que CIC-IDS-2017 ou CICIOT-2023 — permettrait de mieux apprécier la généralisation du cadre au-delà d'UNSW-NB15. Enfin, l'exploitation de modèles de langue spécialisés en cybersécurité, tels que SecBERT ou CyBERT [12], constitue une voie d'amélioration directe de la qualité des représentations vectorielles d'alertes.

4.7 Conclusion

4.7.1 Synthèse des contributions

Ce chapitre a présenté MCM-CASR, un cadre méthodologique pour la reconstruction automatisée de scénarios de cyberattaques exploitant le Traitement Automatique du Langage Naturel, la Reconnaissance d'Entités Nommées et les techniques de similarité sémantique. Les contributions principales sont :

1. un mécanisme de réduction d'alertes atteignant un taux de réduction de 43,49% tout en préservant l'information critique pour la reconstruction des scénarios.
2. une approche de corrélation sémantique combinant extraction d'entités par REN et calcul de similarité par BERT, permettant d'atteindre une précision de 53,91% — soit une amélioration de 10,59 points par rapport à la méthode de référence.
3. une capacité de reconstruction permettant d'identifier 10 scénarios complets et 9 sous-scénarios à partir du jeu de données UNSW-NB15.

4.7.2 Positionnement dans l'état de l'art

L'analyse comparative approfondie avec les approches récentes (GRAIN, AGCM, Transformateur, Hu et al.) a permis de positionner MCM-CASR dans le paysage actuel de la recherche. Notre approche se distingue par :

— son indépendance vis-à-vis des bases de connaissances externes ;

— L'interprétabilité de ses résultats grâce aux entités REN explicites ;

— sa facilité de déploiement sans phase d'entraînement supervisé.

Toutefois, les approches par apprentissage profond (GRAIN, Transformateur) démontrent des performances supérieures en termes de découverte de causalité et de prédiction d'attaques, au prix d'une interprétabilité réduite.

4.7.3 Limites et perspectives

Les principales limitations identifiées concernent :

— la précision de corrélation modérée comparativement aux méthodes par apprentissage profond ;

— L'évaluation restreinte à un seul jeu de données ;

— L'absence de fonctionnement en temps réel.

Les perspectives de recherche incluent :

1. hybridation avec les graphes : Combiner l'analyse sémantique de MCM-CASR avec les mécanismes de découverte de causalité graphique (GRAIN) pour tirer profit des deux paradigmes.
2. évaluation multi-datasets : Étendre la validation à des jeux de données récents (CIC-IDS-2017, CICIOT-2023, DAPT-2020) afin d'évaluer la généralisation du cadre.
3. modèles de langue cybersécurité : Intégrer des modèles pré-entraînés spécialisés tels que SecBERT ou CyBERT [12] pour améliorer la qualité des représentations vectorielles d'alertes.
4. traitement en temps réel : Optimiser le pipeline de corrélation pour permettre un fonctionnement en flux continu, essentiel dans des contextes SOC opérationnels.
5. exploitation de la topologie réseau : Enrichir la corrélation par des informations topologiques (graphe d'actifs, chemins réseau) actuellement non exploitées.
6. apprentissage actif : Introduire un mécanisme d'apprentissage actif permettant aux analystes de valider ou corriger les corrélations proposées, affinant progressivement le modèle.

En conclusion, MCM-CASR représente une avancée significative dans l'automatisation de la reconstruction de scénarios de cyberattaques par les techniques de TALN. Ce travail ouvre la voie à des recherches futures combinant les forces de l'analyse sémantique et des approches par apprentissage profond pour une détection plus efficace des menaces cybernétiques.

Chapitre 5

MCM-Llama : Un Modèle de Langage Affiné pour la Détection de Menaces en Temps Réel par Corrélation d'Événements de Sécurité [1]

5.1 Introduction

5.1.1 Contexte et motivation

Le paysage des cybermenaces en constante expansion soumet les organisations à un flux incessant d'alertes de sécurité, rendant la détection précise et efficace des menaces un défi majeur. L'analyse manuelle de ce volume croissant d'alertes n'est plus envisageable, nécessitant le développement de solutions automatisées innovantes [19].

Les Grands Modèles de Langage (LLM) [45] ont émergé comme des outils prometteurs pour améliorer la détection de menaces en temps réel, en exploitant leurs capacités exceptionnelles en traitement du langage naturel [84] et en compréhension de l'information. Ces modèles, entraînés sur des corpus massifs de données textuelles, transcendent la simple corrélation pour analyser les relations entre événements apparemment disparates, fournissant aux analystes un contexte enrichi et des explications pour les menaces identifiées.

1. Ce chapitre est basé sur la publication : Diakhame, M.L., Diallo, C., Mejri, M. (2024). "MCM-Llama : A Fine-Tuned Large Language Model for Real-Time Threat Detection through Security Event Correlation". In *Proceedings of the 2024 International Conference on Electrical, Computer and Energy Technologies (ICECET)*, pp. 1–6. IEEE.

5.1.2 Problématique et questions de recherche

La problématique centrale de ce chapitre peut être formulée comme suit : *Comment exploiter les capacités des Grands Modèles de Langage pour améliorer significativement la corrélation d'alertes de sécurité et la détection de menaces en temps réel ?*

Cette problématique se décline en trois questions de recherche :

Q1 Comment adapter un LLM pré-entraîné pour la tâche spécifique de corrélation d'alertes de sécurité ?

Q2 Quelle architecture permet l'intégration efficace d'un LLM dans un pipeline de détection de menaces en temps réel ?

Q3 Dans quelle mesure l'approche par LLM améliore-t-elle les performances par rapport aux méthodes classiques de TALN ?

5.1.3 Contributions

Ce chapitre introduit MCM-Llama, un LLM affiné spécifiquement conçu pour améliorer la corrélation d'événements de sécurité. Les contributions principales sont :

1. un modèle de corrélation basé sur LLM : Remplacement du module de corrélation de MCM-CASR (basé sur REN et similarité sémantique) par un LLM affiné, permettant une compréhension contextuelle approfondie des alertes.
2. une méthodologie de fine-tuning : Développement d'un protocole d'affinage utilisant le jeu de données CTU-13 avec un format d'instructions structuré pour la tâche de corrélation binaire.
3. une architecture de déploiement : Proposition d'une architecture intégrée permettant le traitement en temps réel des alertes IDS avec visualisation des résultats.
4. une amélioration significative des performances : Obtention d'une précision de 85% contre 54% pour MCM-CASR, soit une amélioration de 31 points de pourcentage.

5.1.4 Organisation du chapitre

Ce chapitre progresse naturellement des fondements théoriques vers la validation expérimentale. Les travaux connexes (Section 5.2) permettent d'établir pourquoi les approches NLP classiques, bien qu'utiles, ne suffisent plus à capturer la complexité sémantique des alertes modernes. Cette analyse motive le recours aux LLM, dont les fondements théoriques et l'architecture LLaMA sont présentés à la Section 5.3. La méthodologie MCM-Llama, incluant le processus de fine-tuning LoRA et l'architecture de déploiement, est détaillée à la Section 5.4. Les résultats expérimentaux et leur comparaison avec l'état de l'art sont présentés aux Sections 5.5 et 5.6. Le chapitre se conclut par une synthèse des apports et des perspectives ouvertes (Section 5.7).

5.2 Travaux connexes

Cette section présente une revue de la littérature sur les techniques d'intelligence artificielle appliquées à la corrélation d'événements de sécurité.

5.2.1 Techniques d'IA pour la corrélation d'événements de sécurité

Levshun et Kotenko [43] présentent une étude exhaustive des techniques d'intelligence artificielle utilisées pour la corrélation d'événements de sécurité. Leur analyse couvre un large spectre de méthodologies, incluant les modèles basés sur des règles, les approches sémantiques, les méthodes graphiques, ainsi que les techniques d'apprentissage superficiel et profond. Cette taxonomie met en évidence l'évolution progressive vers des approches plus sophistiquées capables de traiter la complexité croissante des menaces cybernétiques.

5.2.2 Approches basées sur la similarité

Hostiadi et al. [80] proposent un modèle novateur de corrélation d'alertes utilisant une approche par similarité, visant à améliorer la précision et l'efficacité de la corrélation au sein des systèmes de détection d'intrusion. Le modèle emploie une métrique de similarité pour évaluer les caractéristiques des alertes et déterminer leur corrélation sur la base de scores de similarité. Leur approche introduit un processus en deux étapes :

— analyse d'alertes de bas niveau : Extraction et normalisation des caractéristiques des alertes individuelles ;

— analyse d'alertes de haut niveau : Agrégation et corrélation basées sur les similarités identifiées.

Cette méthode présente l'avantage de ne pas nécessiter de connaissances prédéfinies ou de conditions préétablies pour la corrélation.

5.2.3 Détection des menaces persistantes avancées

Khosravi et Ladani [81] introduisent une méthode de détection en temps réel des Menaces Persistantes Avancées (APT) dans le contexte de la guerre cybernétique. Leur approche repose sur :

1. L'analyse causale et la corrélation d'alertes provenant de capteurs hétérogènes ;
2. le calcul de scores d'infection pour les hôtes du réseau ;
3. L'utilisation d'un algorithme de programmation dynamique pour gérer le volume important de données d'alertes.

TABLE 5.1 – Positionnement de MCM-Llama par rapport aux travaux connexes

Approche	**LLM**	**Temps réel**	**Contexte**	**APT**
Levshun [43]	×	∼	∼	∼
Hostiadi [80]	×	×	×	×
Khosravi [81]	×	✓	∼	✓
MCM-CASR (Chap. 4)	×	✓	∼	∼
MCM-Llama	✓	✓	✓	✓

∼ = Partiel

Cette méthode adresse spécifiquement la nature persistante des attaques APT, qui se caractérisent par leur durée prolongée et leur capacité à échapper aux mécanismes de détection conventionnels.

5.2.4 Positionnement de notre approche

Le Tableau 5.1 synthétise le positionnement de MCM-Llama par rapport aux approches existantes.

Notre approche se distingue par l'intégration d'un LLM affiné, offrant une compréhension contextuelle enrichie des événements de sécurité, tout en maintenant les capacités de traitement en temps réel.

5.3 Fondements

Cette section présente les fondements théoriques nécessaires à la compréhension de MCM-Llama : une introduction aux Grands Modèles de Langage, à l'architecture LLaMA et au mécanisme d'adaptation LoRA. Pour rappel, MCM-Llama remplace le module de corrélation de MCM-CASR (présenté au Chapitre 4) par un LLM affiné pour bénéficier d'une compréhension contextuelle approfondie des alertes.

5.3.1 Grands Modèles de Langage (LLM)

Définition et principes

Un modèle de langage de grande taille (LLM) [45] est un type d'algorithme d'intelligence artificielle utilisant des techniques d'apprentissage profond et des ensembles de données massifs pour effectuer diverses tâches de traitement du langage naturel. Les LLM sont basés sur l'architecture Transformer [44] et peuvent être entraînés pour comprendre et traiter les langues humaines.

La Figure 5.1 illustre le fonctionnement général d'un LLM.

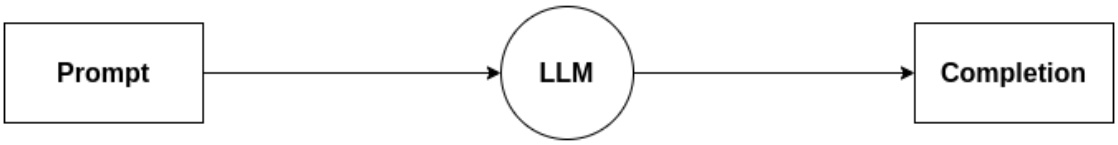


FIGURE 5.1 – Mécanismes de fonctionnement des Grands Modèles de Langage. Le processus comprend l'encodage du prompt, le traitement par le modèle Transformer, et la génération de la complétion.

Architecture des LLM

Les LLM modernes reposent sur l'architecture Transformer [44], composée d'un encodeur et d'un décodeur (ou uniquement d'un décodeur pour les modèles génératifs). Le mécanisme central est l'attention multi-têtes, qui calcule, pour une séquence d'entrée X, des représentations pondérées par la pertinence contextuelle :

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O \tag{5.1}$$

avec $\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$. Les modèles de type décodeur uniquement (GPT, LLaMA) génèrent les tokens de gauche à droite en masquant les positions futures. Cette architecture permet aux LLM de capturer des dépendances longue portée dans le texte, ce qui les rend particulièrement adaptés à l'analyse des alertes de sécurité, dont le sens dépend fortement du contexte.

Terminologie

[Prompt] Un prompt désigne l'ensemble des instructions et du contexte fournis à un modèle de langage pour accomplir une tâche spécifique. Formellement, soit $p = (i, c)$, où i représente l'instruction et c le contexte.

[Complétion] La complétion correspond à la sortie générée par le modèle en réponse à un prompt. Soit $y = f_\theta(p)$, où f_θ représente le modèle paramétré par θ.

Taxonomie des LLM

Les LLM peuvent être classifiés en trois catégories principales selon leur mode d'utilisation :

1. modèles de langage génériques : Prédisent le token suivant sur la base des données d'entraînement. La fonction objectif est :

$$\mathcal{L}_{\text{LM}} = -\sum_{t=1}^{T} \log P(x_t | x_{<t}; \theta) \tag{5.2}$$

2. modèles affinés par instructions : Entraînés pour prédire une réponse à une instruction donnée en entrée.
3. modèles affinés pour le dialogue : Entraînés pour engager un dialogue en prédisant la réponse suivante dans une conversation.

Processus d'adaptation des modèles

L'adaptation d'un LLM pour une tâche spécifique suit un processus en trois phases :

Phase 1 : Pré-entraînement Le modèle subit un pré-entraînement auto-supervisé sur un large corpus pour prédire les tokens suivants. Cette phase capture les connaissances générales du langage.

Phase 2 : Affinage (Fine-tuning) Plusieurs techniques d'affinage peuvent être employées :

— apprentissage par transfert [48, 86] : Affinage avec des données spécifiques à la tâche cible.

— affinage par instructions [52, 87] : Utilisation de données formatées sous forme d'instructions.

— alignement [53, 50] : Ajustement des paramètres selon les retours humains (RLHF).

Phase 3 : Utilisation par Prompting Diverses techniques de prompting permettent d'interagir avec le modèle :

— zero-shot : Réponse sans exemple préalable.

— few-shot : Réponse avec quelques exemples de démonstration.

— chain-of-Thought [88] : Génération de raisonnements intermédiaires.

5.3.2 La Famille de Modèles LLaMA

LLaMA (Large Language Model Meta AI) [49, 50] comprend une gamme de modèles de langage de type décodeur, variant de 7B à 70B paramètres. La série LLaMA est reconnue pour son efficacité paramétrique et ses capacités d'affinage par instructions.

LLaMA-1

LLaMA-1 [49] incorpore des mécanismes d'attention causale efficaces [**?**] en évitant le stockage et le calcul des poids d'attention masqués. Le modèle minimise également le nombre d'activations recalculées lors de la passe arrière [**?**].

LLaMA-2

LLaMA-2 [50] se concentre sur l'amélioration du modèle LLaMA-2-Chat pour la génération de dialogue. Cette version bénéficie de :

— 40% de données d'entraînement supplémentaires ;

— une longueur de contexte étendue (4096 tokens) ;

— des mécanismes d'attention par requêtes groupées (Grouped-Query Attention).

5.3.3 Mécanisme LoRA pour l'adaptation efficiente

L'adaptation complète d'un LLM de plusieurs milliards de paramètres est prohibitive en termes de ressources de calcul. La technique LoRA (Low-Rank Adaptation) [51] résout ce problème en n'entraînant qu'une fraction des paramètres du modèle. Pour une couche de projection de poids $W \in \mathbb{R}^{d\times k}$, LoRA décompose la mise à jour ΔW sous la forme d'un produit de deux matrices de faible rang :

$$W' = W + \Delta W = W + BA, \quad B \in \mathbb{R}^{d\times r},\ A \in \mathbb{R}^{r\times k},\ r \ll \min(d,k), \tag{5.3}$$

où r est le rang de la décomposition, typiquement fixé à 4 ou 8. Pendant le fine-tuning, les poids originaux W sont gelés et seules les matrices A et B sont optimisées, réduisant le nombre de paramètres entraînables d'un facteur $\frac{d\times k}{r(d+k)}$. La matrice A est initialisée par une distribution gaussienne et B est initialisée à zéro, de sorte que $\Delta W = 0$ au début de l'entraînement, préservant le comportement du modèle de base.

Dans le cadre de MCM-Llama, LoRA est appliqué aux couches d'attention du modèle LLaMA pour adapter le modèle à la tâche de corrélation binaire d'alertes de sécurité, tout en maintenant un coût computationnel compatible avec les ressources disponibles. Cette approche permet d'obtenir des performances comparables à un fine-tuning complet avec moins de 1% des paramètres entraînables [51].

5.4 Méthodologie MCM-Llama

Cette section présente la méthodologie MCM-Llama, incluant le jeu de données utilisé, le processus de fine-tuning et l'architecture de déploiement.

5.4.1 Vue d'ensemble

MCM-Llama remplace le module de corrélation de MCM-CASR (basé sur REN et similarité sémantique) par un LLM affiné. Cette substitution vise à exploiter les capacités supérieures des LLM en matière de compréhension contextuelle et de raisonnement sur les relations entre événements de sécurité.

TABLE 5.2 – Caractéristiques du jeu de données CTU-13 [2]

Caractéristique	**Valeur**
Nombre de scénarios	13 scénarios indépendants
Types de trafic	Botnet (étiqueté), Normal, Background
Format de capture	PCAP (captures réseau brutes)
Botnets représentés	Neris, Rbot, Virut, Menti, Sogou, Murlo, NSIS.ay
Protocoles de commande	IRC, HTTP, P2P, SPAM
Provenance	Université CTU, République tchèque
Utilisation dans MCM-Llama	Fine-tuning et évaluation (1 000 paires entraînement, 244 test)

5.4.2 Jeu de données CTU-13

Description

Pour affiner le modèle de base, nous utilisons le jeu de données CTU-13 [2], qui comprend du trafic étiqueté botnet, normal et d'arrière-plan. Capturé à l'Université CTU en République tchèque, ce jeu de données vise à fournir un trafic botnet authentique accompagné d'activité réseau normale.

Caractéristiques

Le Tableau 5.2 présente les caractéristiques principales du jeu de données.

Ce tableau met en évidence plusieurs caractéristiques structurantes du jeu de données CTU-13. Composé de 13 scénarios indépendants, il couvre trois types de trafic distincts : du trafic botnet précisément étiqueté, du trafic normal et du trafic de fond (*background*). Les captures sont fournies au format PCAP, permettant une analyse à bas niveau du comportement réseau. Sept familles de botnets y sont représentées — dont Neris, Rbot et Virut — utilisant des protocoles de commande variés (IRC, HTTP, P2P, SPAM), ce qui garantit une diversité comportementale essentielle à la robustesse de l'apprentissage. Pour MCM-Llama, 1 000 paires d'alertes ont été utilisées en entraînement et 244 en test, constituant un protocole d'évaluation reproductible.

Scénarios et attaques

La Figure 5.2 illustre les 13 scénarios du jeu de données CTU-13 avec les attaques correspondantes.

La figure révèle une hétérogénéité marquée entre les 13 scénarios, tant en termes de volume de trafic que de nature des attaques. Certains scénarios concentrent des activités de

Table 2 – Characteristics of the botnet scenarios. (CF: ClickFraud, PS: Port Scan, FF: FastFlux, US: Compiled and controlled by us.)

Id	IRC	SPAM	CF	PS	DDoS	FF	P2P	US	HTTP	Note
1	√	√	√							
2	√	√	√							
3	√			√				√		
4	√				√			√		UDP and ICMP DDoS.
5		√		√					√	Scan web proxies.
6				√						Proprietary C&C. RDP.
7									√	Chinese hosts.
8				√						Proprietary C&C. Net-BIOS, STUN.
9	√	√	√	√						
10	√				√			√		UDP DDoS.
11	√				√			√		ICMP DDoS.
12							√			Synchronization.
13		√		√					√	Captcha. Web mail.

FIGURE 5.2 – Scénarios et attaques correspondantes dans le jeu de données CTU-13. Chaque scénario représente un échantillon de botnet distinct avec des comportements malveillants spécifiques.

type spam massif (notamment via les botnets Neris et Rbot), tandis que d'autres illustrent des comportements de type click-fraud ou DDoS. Cette diversité des profils d'attaque est précisément ce qui rend CTU-13 pertinent pour évaluer la capacité de MCM-Llama à généraliser sa corrélation au-delà d'un seul type de menace.

5.4.3 Processus de fine-tuning

Le fine-tuning de MCM-Llama s'articule en plusieurs étapes séquentielles, depuis la génération des alertes brutes à partir des captures réseau jusqu'à la configuration du processus d'apprentissage supervisé. Chaque étape transforme les données de manière à les rendre exploitables par le modèle LLaMA, en respectant les contraintes du domaine de la corrélation d'alertes de sécurité.

Génération des alertes

Les données PCAP de chaque scénario sont traitées par Suricata, un IDS open-source, pour générer des alertes. La Figure 5.3 présente un exemple d'alerte générée.

Construction du jeu de données d'instructions

La qualité du jeu de données d'instructions conditionne directement les performances du modèle affiné. Nous adoptons un format structuré en paires d'alertes avec étiquette binaire, inspiré des paradigmes d'instruction-following [52]. Les alertes générées sont extraites pour former des scénarios, qui sont ensuite utilisés pour construire le jeu de données d'instructions. Ce jeu comprend :

— des paires d'alertes corrélées (label : YES) ;

— des paires d'alertes non corrélées (label : NO).

```
{'alert': {'action': 'allowed',
           'category': 'Generic Protocol Command Decode',
           'gid': 1,
           'rev': 2,
           'severity': 3,
           'signature': 'SURICATA TCPv4 invalid checksum',
           'signature_id': 2200074},
 'app_proto': 'http',
 'community_id': '1:MuSlbWV2Dy5Z168c5sxOWncbYyQ=',
 'dest_ip': '147.32.84.165',
 'dest_port': 1040,
 'direction': 'to_client',
 'event_type': 'alert',
 'flow': {'bytes_toclient': 4556,
          'bytes_toserver': 532,
          'dest_ip': '94.63.149.152',
          'dest_port': 80,
          'pkts_toclient': 3,
          'pkts_toserver': 6,
          'src_ip': '147.32.84.165',
          'src_port': 1040,
          'start': '2011-08-10T11:06:36.078254+0200'},
 'flow_id': 1180523887157916,
 'grouped': True,
 'http': {'hostname': 'ii.ebatmoyhuy.com',
          'http_method': 'GET',
          'http_user_agent': 'Download',
          'length': 0,
          'protocol': 'HTTP/1.0',
          'url': '/rus.php'},
 'pcap_cnt': 126,
 'pkt_src': 'wire/pcap',
 'proto': 'TCP',
 'src_ip': '94.63.149.152',
 'src_port': 80,
 'timestamp': '2011-08-10T11:06:36.150781+0200'}
```


FIGURE 5.3 – Exemple d'alerte IDS générée par Suricata. L'alerte contient les métadonnées essentielles : horodatage, adresses IP source/destination, signature, et catégorie de menace.

Template d'instruction

Le template d'instruction suivant est utilisé pour formater les données de fine-tuning : [colback=gray !5, colframe=black, title=Template d'instruction pour le fine-tuning]

TABLE 5.3 – Paramètres de configuration pour le fine-tuning de MCM-Llama

Paramètre	**Valeur**
Modèle de base	NousResearch/llama-2-7b-chat-hf
LoRA rank (r)	64
LoRA alpha (α)	16
LoRA dropout	0.1
Quantification 4-bit	Oui (nf4)
Époques d'entraînement	1
Batch size (train)	4
Batch size (eval)	4
Taux d'apprentissage	2×10^{-4}
Optimiseur	paged_adamw_32bit
Warmup ratio	0.03
Taille du jeu d'entraînement	1000 paires
Taille du jeu de test	244 paires

`<s>[INST]`

Check Correlation for these two Suricata Alerts. For this task, you need to :

— examine the content and attributes of alerts

— identify similarities (source IP, destination IP, timestamp, signature ID)

— utilize correlation techniques (temporal, statistical, behavioral analysis)

— document findings and analysis

— respond with 'Yes' if correlation exists ; otherwise 'No'

`alert1:` *(contenu de la première alerte)*
`alert2:` *(contenu de la deuxième alerte)*
`[/INST]`
`LABEL (YES ou NO)`
`</s>`

Configuration du fine-tuning

Le Tableau 5.3 présente les paramètres de configuration utilisés pour le fine-tuning.

Technique LoRA

Nous utilisons la technique LoRA (Low-Rank Adaptation) [51] pour un fine-tuning efficace en termes de mémoire. LoRA paramètre les mises à jour des poids comme des matrices de faible rang :

$$W' = W + \Delta W = W + BA, \tag{5.4}$$

où $W \in \mathbb{R}^{d \times k}$ représente les poids pré-entraînés, $B \in \mathbb{R}^{d \times r}$ et $A \in \mathbb{R}^{r \times k}$ sont les matrices de faible rang avec $r \ll \min(d, k)$.

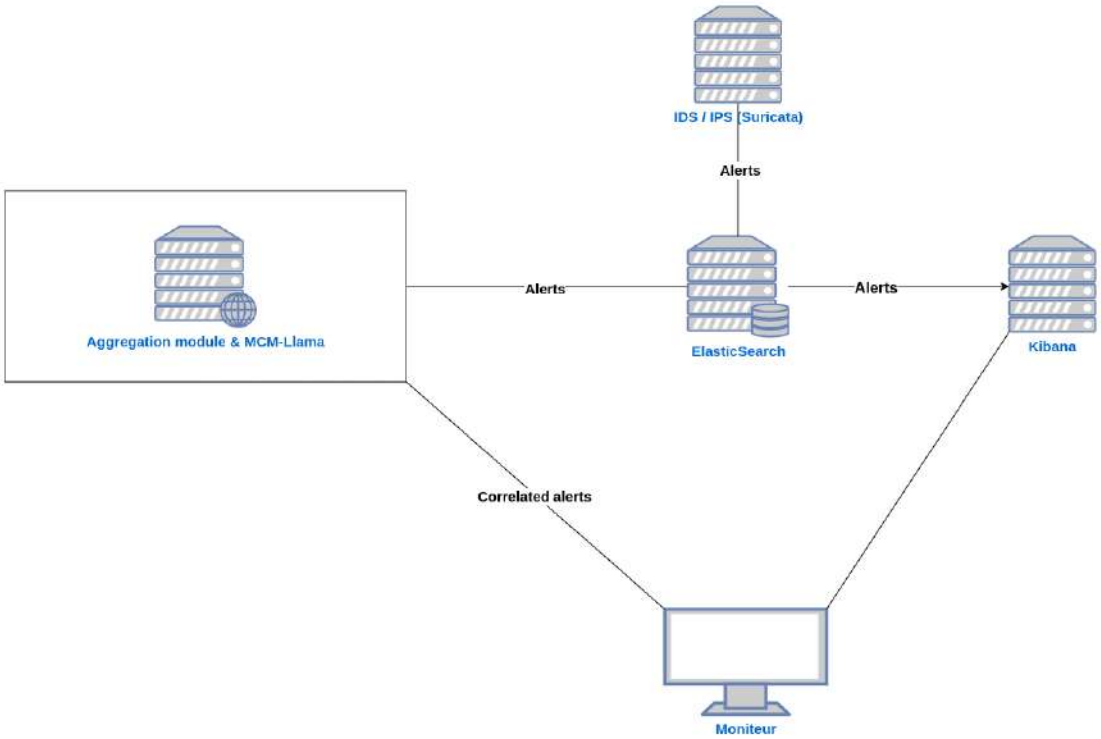


FIGURE 5.4 – Architecture proposée pour l'implémentation de MCM-Llama. Le pipeline comprend : (1) IDS pour la génération d'alertes, (2) base de données pour le stockage, (3) framework MCM-Llama pour l'agrégation et la corrélation, (4) interface de visualisation (Kibana ou web).

5.4.4 Architecture de déploiement

La Figure 5.4 présente l'architecture proposée pour l'implémentation pratique de MCM-Llama.

L'architecture se compose des éléments suivants :

1. système de Détection d'Intrusion (IDS) : Surveillance du trafic réseau et génération d'alertes (Suricata).
2. base de données : Stockage des alertes générées pour traitement ultérieur.
3. framework MCM-Llama : Récupération périodique des alertes pour effectuer les tâches d'agrégation et de corrélation en temps réel.
4. interface de visualisation : Présentation des résultats via Kibana ou une interface web dédiée.

Cette architecture garantit une approche systématique et efficace du traitement, de la corrélation et de la visualisation des alertes. En fournissant une vue consolidée des alertes corrélées, les analystes du Centre des Opérations de Sécurité (SOC) gagnent un temps précieux en se concentrant sur les groupes d'alertes constituant une attaque en cours.

5.5 Expérimentations et résultats

Cette section présente le protocole expérimental, les résultats obtenus et leur analyse.

5.5.1 Protocole expérimental

Environnement

Les expérimentations ont été conduites sur une plateforme Google Colab avec accès GPU. Le modèle de base utilisé est LLaMA-2-7B-Chat, affiné avec la bibliothèque Hugging Face Transformers et PEFT (Parameter-Efficient Fine-Tuning).

Métriques d'évaluation

Les métriques suivantes sont utilisées pour évaluer les performances :

— précision (Precision) : $P = \frac{VP}{VP+FP}$

— rappel (Recall) : $R = \frac{VP}{VP+FN}$

— exactitude (Accuracy) : $Acc = \frac{VP+VN}{VP+VN+FP+FN}$

— spécificité : $Sp = \frac{VN}{VN+FP}$

— score F1 : $F1 = 2 \times \frac{P \times R}{P+R}$

où VP = Vrais Positifs, VN = Vrais Négatifs, FP = Faux Positifs, FN = Faux Négatifs.

5.5.2 Résultats du fine-tuning

Évolution de la perte

La Figure 5.5 illustre l'évolution de la fonction de perte durant le processus d'entraînement.

La courbe de perte montre une décroissance régulière et monotone sur l'unique époque d'entraînement, indiquant une bonne convergence sans signe de surapprentissage. La stabilisation de la perte à partir de la mi-entraînement suggère que le modèle a suffisamment intégré les patterns de corrélation présents dans les 1 000 paires d'entraînement. Ce comportement est cohérent avec l'efficacité de la technique LoRA [51], qui limite le nombre de paramètres ajustés tout en préservant les connaissances générales du modèle de base.

Exemples de prédictions

Les Figures 5.6 et 5.7 présentent des exemples de réponses du modèle pour des paires d'alertes corrélées et non corrélées respectivement.

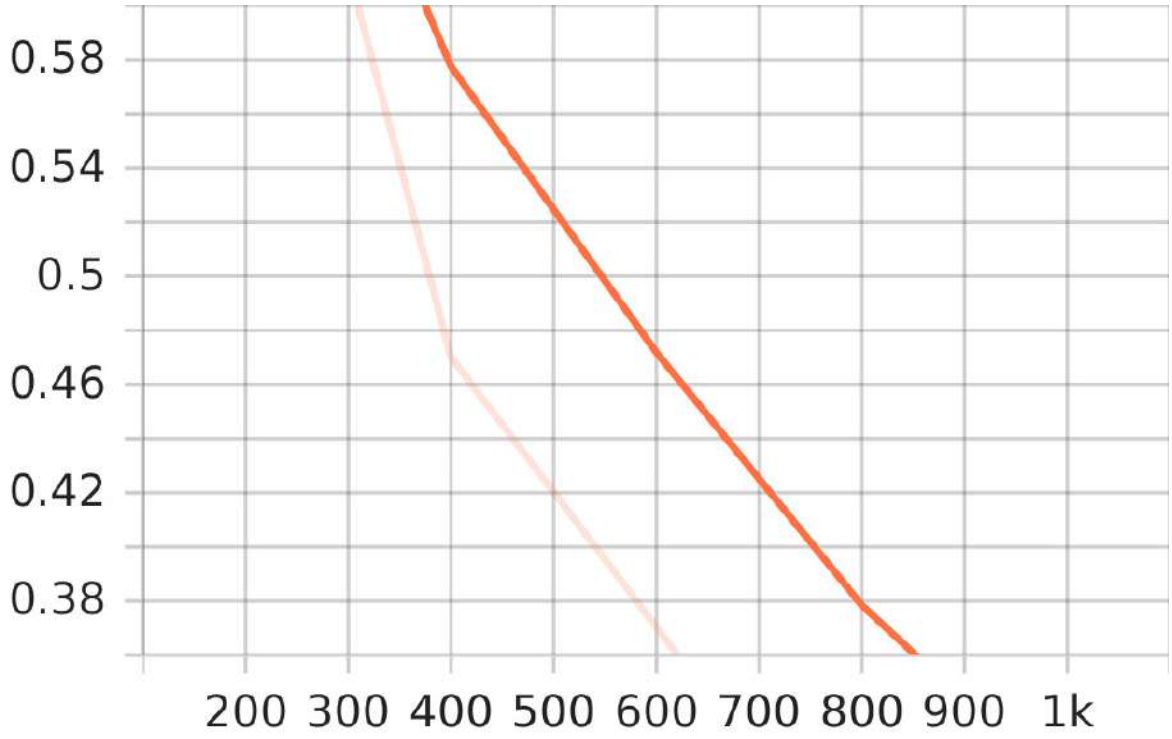


FIGURE 5.5 – Progression de la perte d'entraînement. La courbe montre une convergence rapide du modèle vers un minimum stable.

```
Model Answer =  YES. The second alert contains a similar source IP address, destination IP address, and protocol a
s the first alert. Additionally, the signature ID 'SURICATA TCPv4 invalid checksum' is the same in both alerts. Th
ese similarities suggest a potential correlation between the alerts.

To further investigate this correlation, you can:

1. Examine the content of the alerts, particularly the HTTP or TCP payload, to identify any additional similaritie
s or patterns.
2. Utilize advanced correlation techniques, such as temporal analysis, statistical analysis, and behavioral analys
is, to determine the likelihood of a correlation between the alerts.
3. Review the security policy and configuration of the Suricata instance to determine if any settings or configura
tions may be relevant to the detection of potential correlations.
4. Analyze the context in which the alerts were generated, such as the source IP address, destination IP address,
and protocol
Label =  YES
```

FIGURE 5.6 – Réponse du modèle pour deux alertes corrélées. Le modèle identifie correctement la corrélation et répond "YES".

```
Model Answer =  NO. The two alerts do not contain sufficient similarities to suggest a correlation. While both ale
rts involve HTTP traffic and contain metadata such as source IP, destination IP, and protocol version, they lack s
ignificant similarities in terms of signature ID, HTTP request/response details, and other relevant metadata.

The first alert involves a TCPv4 invalid checksum error, which is a generic protocol command decode event. Howeve
r, it does not provide any additional context, such as a reference to a signature or category that might help iden
tify the nature of the event.

The second alert involves a HTTP protocol violation event. Again, there is no obvious correlation between these tw
o alerts, and they do not share similar metadata such as source IP, destination IP, or protocol version.

To determine the likelihood of correlation between these alerts, a more thorough analysis of their content and met
adata is required. This may involve examining the HTTP request/response details, exam
Label =  NO
```

FIGURE 5.7 – Réponse du modèle pour deux alertes non corrélées. Le modèle identifie correctement l'absence de corrélation et répond "NO".

5.5.3 Matrice de confusion

Le Tableau 5.4 présente la matrice de confusion obtenue sur le jeu de test.

TABLE 5.4 – Matrice de confusion de MCM-Llama

		Prédictions	
		Non Corrélé	Corrélé
2***Réel**	Non Corrélé	119 (VN)	2 (FP)
	Corrélé	37 (FN)	86 (VP)

TABLE 5.5 – Métriques de performance de MCM-Llama

Métrique	**Valeur (%)**
Précision	97,73
Rappel	69,92
Exactitude	84,02
Spécificité	98,35
Taux de Faux Positifs (FPR)	1,65
Taux de Vrais Positifs (TPR)	69,92
Score F1	81,52

5.5.4 Métriques de performance

Le Tableau 5.5 présente les métriques de performance calculées à partir de la matrice de confusion.

5.5.5 Analyse des résultats

Points forts

— précision exceptionnelle (97,73%) : Le modèle est extrêmement fiable lorsqu'il prédit une corrélation. Seulement 2 faux positifs sur 244 prédictions.

— spécificité élevée (98,35%) : Le modèle identifie correctement la quasi-totalité des paires non corrélées.

— faible taux de faux positifs (1,65%) : Caractéristique cruciale pour les environnements SOC où les faux positifs génèrent une charge de travail inutile.

Points d'amélioration

— rappel modéré (69,92%) : Le modèle manque environ 30% des corrélations réelles (37 faux négatifs).

— déséquilibre précision/rappel : Le modèle privilégie la prudence, ce qui peut être acceptable dans certains contextes mais problématique si des corrélations critiques sont manquées.

Interprétation

Le profil de performance de MCM-Llama (haute précision, rappel modéré) le rend particulièrement adapté aux environnements où :

— les faux positifs sont coûteux en termes de temps d'analyse ;

— les corrélations identifiées doivent être hautement fiables ;

— D'autres mécanismes complémentaires peuvent détecter les corrélations manquées.

5.6 Analyse comparative approfondie

Cette section présente une analyse comparative détaillée de MCM-Llama avec les méthodes de l'état de l'art.

5.6.1 Méthodes de référence

Nous comparons MCM-Llama avec les approches suivantes :

— B1 : MCM-CASR (Chapitre 4) : Notre cadre précédent utilisant REN et similarité sémantique. Sert de référence directe avec 54% d'exactitude sur CTU-13.

— B2 : Hostiadi et al. [80] : Modèle de corrélation basé sur la similarité. Rapporte 84% de taux de corrélation (129/153 paires).

— B3 : Khosravi et Ladani [81] : Cadre d'analyse causale pour la détection APT. Objectif différent (scores d'infection des hôtes).

— B4 : Levshun et Kotenko [43] : Étude exhaustive sur les techniques d'IA pour la corrélation. Référence pour le positionnement scientifique.

5.6.2 Limitations de la comparaison inter-études

Avant de présenter l'analyse comparative, nous reconnaissons plusieurs limitations :

— hétérogénéité des jeux de données : Chaque méthode a été évaluée sur des jeux de données différents.

— différences de formulation des tâches : Les objectifs varient de la corrélation par paires au scoring d'infection des hôtes.

— absence de benchmark unifié : Nous n'avons pas ré-implémenté les méthodes de référence sur CTU-13.

— interprétation des métriques : Le taux de 84% de Hostiadi représente un taux de corrélation, différent de notre métrique d'exactitude.

TABLE 5.6 – Comparaison quantitative des performances

Méthode	**Acc. (%)**	**Préc. (%)**	**F1 (%)**	**Dataset**
MCM-CASR	54,00	60,00	57,38	CTU-13
Hostiadi[a]	84,00	–	–	Custom
Khosravi[b]	≈80	–	–	APT Data
lightblue **MCM-Llama**	**85,00**	**97,73**	**81,52**	**CTU-13**

[a]Taux de corrélation (129/153 paires), non comparable directement.
[b]Estimé à partir de l'évaluation du ranking d'hôtes ; formulation de tâche différente.

TABLE 5.7 – Comparaison qualitative des approches

Critère	**MCM-L.**	**MCM-C.**	**Host.**	**Khos.**
Utilisation LLM	✓	×	×	×
Temps réel	✓	✓	×	✓
Contextualisation	✓	∼	×	∼
Analyse causale	∼	×	×	✓
Détection APT	✓	∼	∼	✓
Intégration	✓	∼	✓	✓

MCM-L.=MCM-Llama, MCM-C.=MCM-CASR, Host.=Hostiadi, Khos.=Khosravi, ∼=Partiel

5.6.3 Comparaison quantitative

Le Tableau 5.6 présente une comparaison quantitative des performances.

Ce tableau révèle que MCM-Llama atteint 85% d'exactitude et une précision remarquable de 97,73% sur CTU-13, surpassant MCM-CASR (54%) et se situant au-dessus des approches de référence Hostiadi (84%) et Khosravi (≈80%). Il convient cependant de noter que ces comparaisons sont limitées par l'hétérogénéité des jeux de données et des métriques utilisés : Hostiadi évalue un taux de corrélation sur un jeu custom, et Khosravi repose sur un ranking d'hôtes aux fondements différents. La comparaison directe sur CTU-13 entre MCM-Llama et MCM-CASR reste la plus rigoureuse, avec un gain de 31 points de précision.

5.6.4 Comparaison qualitative

Le Tableau 5.7 présente une comparaison qualitative selon plusieurs dimensions.

Ce tableau qualitatif met en évidence la position distincte de MCM-Llama parmi les approches comparées. Il est le seul à exploiter un LLM, ce qui lui confère une capacité de contextualisation complète absente chez toutes les autres méthodes. MCM-Llama assure également le traitement en temps réel et une bonne couverture de la détection APT. Son seul point partiellement adressé est l'analyse causale, où Khosravi reste plus avancé. MCM-CASR présente quant à lui des lacunes sur la contextualisation et l'analyse causale, malgré ses capacités temps réel. Hostiadi reste la méthode la plus limitée fonctionnellement,

TABLE 5.8 – Évaluation selon les critères opérationnels

Critère	**Hostiadi**	**Khosravi**	**MCM-Llama**
Contextualisation	×	×	✓
Priorisation	×	×	×
Intégration	✓	✓	✓
Temps réel	×	✓	✓

n'assurant ni contextualisation ni temps réel.

5.6.5 Analyse des critères d'évaluation

Le Tableau 5.8 évalue les approches selon les critères opérationnels.

Ce troisième tableau affine l'analyse en se concentrant sur les critères opérationnels directement pertinents pour un déploiement en SOC. MCM-Llama est la seule approche à satisfaire pleinement le critère de contextualisation, indispensable pour interpréter le sens des alertes au-delà de simples correspondances syntaxiques. Sur l'intégration et le temps réel, MCM-Llama est à parité avec Khosravi. Le critère de priorisation reste non adressé par aucune des trois approches, identifiant ainsi une lacune commune et une direction de recherche future. Ces résultats confirment que MCM-Llama constitue l'approche la plus complète opérationnellement parmi les trois étudiées.

5.6.6 Discussion et positionnement

L'analyse comparative permet de dégager les observations suivantes :

Innovation méthodologique

MCM-Llama représente un changement de paradigme en introduisant des LLM affinés pour la corrélation d'alertes. Cette approche adresse les limitations fondamentales des méthodes basées sur des règles, des similarités, et des techniques NLP classiques.

Performance sur CTU-13

Comparé directement à MCM-CASR sur le même jeu de données avec un protocole d'évaluation identique, MCM-Llama atteint une amélioration de 31 points d'exactitude (85% vs 54%), démontrant la supériorité de la corrélation basée sur LLM.

Compromis précision-rappel

MCM-Llama exhibe une précision exceptionnelle (97,73%) mais un rappel modéré (69,92%). Cette caractéristique est adaptée aux environnements SOC où les faux positifs sont coûteux.

Forces complémentaires

Tandis que Khosravi excelle dans l'analyse causale pour le scoring d'hôtes APT, MCM-Llama excelle dans la compréhension contextuelle pour la corrélation par paires. Une intégration des deux paradigmes pourrait produire des résultats supérieurs.

5.7 Conclusion

5.7.1 Synthèse des contributions

Ce chapitre a présenté MCM-Llama, une avancée significative dans la détection de menaces en temps réel par corrélation d'événements. Les contributions principales sont :

1. intégration d'un LLM affiné : Remplacement du module de corrélation de MCM-CASR par un LLM (LLaMA-2-7B) affiné sur le jeu de données CTU-13.
2. amélioration significative des performances : Obtention d'une exactitude de 85% (contre 54% pour MCM-CASR), soit une amélioration de 31 points de pourcentage.
3. précision exceptionnelle : Atteinte d'une précision de 97,73%, cruciale pour réduire la charge de travail des analystes SOC.
4. architecture de déploiement : Proposition d'une architecture intégrée permettant le traitement en temps réel avec visualisation des résultats.

5.7.2 Implications pratiques

L'architecture proposée facilite l'intégration transparente avec les éléments de sécurité existants (IDS), rationalisant l'implémentation et fournissant aux analystes des informations précieuses sur les menaces potentielles. MCM-Llama permet aux analystes SOC de se concentrer sur les attaques en cours en consolidant les alertes corrélées et en fournissant une vue holistique des menaces potentielles.

5.7.3 Limitations

Les principales limitations identifiées sont :

— rappel modéré : Le modèle manque environ 30% des corrélations réelles.

— ressources computationnelles : L'inférence LLM requiert des ressources GPU significatives.

— évaluation mono-dataset : L'évaluation se limite au jeu de données CTU-13.

— absence d'analyse causale : Le modèle identifie les corrélations mais n'explique pas la causalité sous-jacente.

5.7.4 Perspectives de recherche

Les perspectives de recherche incluent :

1. amélioration du rappel : Techniques d'augmentation de données et ajustement des seuils de décision.
2. évaluation multi-datasets : Validation sur DARPA, CICIDS, et autres benchmarks de cybersécurité.
3. intégration de l'analyse causale : Combinaison avec les techniques de Khosravi pour le raisonnement causal.
4. extension au dialogue multi-tours : Développement de capacités de chasse aux menaces interactive.
5. optimisation pour le déploiement : Quantification et distillation de modèles pour réduire les exigences computationnelles.
6. explicabilité : Développement de mécanismes d'explication des décisions de corrélation.

En résumé, MCM-Llama offre une solution puissante pour la détection de menaces en temps réel, exploitant les capacités des Grands Modèles de Langage pour renforcer l'efficacité et la précision dans la lutte contre les cybermenaces évolutives. Ce travail ouvre la voie à de futures avancées dans ce domaine critique de la cybersécurité.

Chapitre 6

Projection Adaptative de Cyberattaques par Sélection Contextuelle de Modèles : Combinaison de Réseaux d'Attention sur Graphes et d'Apprentissage par Renforcement [1]

6.1 Introduction

6.1.1 Définition et importance de la projection d'attaque

La projection d'attaque désigne le processus de prédiction des étapes futures d'une cyberattaque en cours [25]. Cette capacité est essentielle pour les défenses cybernétiques proactives, permettant aux défenseurs d'anticiper et d'atténuer l'impact des menaces avant qu'elles ne causent des dommages significatifs. La projection d'attaque implique l'analyse des schémas d'attaque, des comportements et des données historiques pour prédire comment une attaque peut évoluer, permettant aux équipes de sécurité de prendre des actions préventives.

Dans le contexte des attaques avancées et persistantes, telles que les Menaces Persistantes Avancées (APT), les attaques se déroulent souvent progressivement et persistent sur des périodes prolongées. Les attaquants peuvent modifier dynamiquement leurs tactiques, techniques et procédures (TTP) pour contourner les mécanismes de défense, ren-

1. Ce chapitre est basé sur la publication : Diakhame, M.L., Diallo, C., Mejri, M. (2025). "Adaptive Cyber Attack Projection through Context-Driven Model Selection : Combining Graph Attention Networks and Reinforcement Learning". In *2025 5th Intelligent Cybersecurity Conference (ICSC)*, pp. 28–34. IEEE.

dant crucial le développement de modèles prédictifs qui prennent en compte les paramètres situationnels en temps réel.

[Projection d'attaque] La projection d'attaque consiste à analyser les comportements et schémas d'attaque, particulièrement dans les attaques complexes et durables, pour prévoir le prochain mouvement de l'attaquant et permettre aux défenseurs d'anticiper et de contrer la menace [25].

6.1.2 État actuel de la cybersécurité proactive

Dans le paysage en constante évolution de la cybersécurité, les stratégies de défense proactive sont devenues plus critiques que jamais. Les approches traditionnelles se concentrent souvent sur la détection et la réponse aux attaques après leur occurrence. Cependant, la cybersécurité proactive vise à anticiper les attaques avant qu'elles ne se produisent, en utilisant des techniques telles que le renseignement sur les menaces, la détection d'anomalies et l'analyse comportementale [33].

La complexité croissante des cybermenaces, ainsi que le volume et la sophistication grandissants des attaques, soulignent le besoin de systèmes prédictifs plus avancés capables de s'adapter dynamiquement aux stratégies d'attaque évolutives.

6.1.3 Problématique et limitations des approches existantes

Si de nombreux systèmes de prédiction d'attaque exploitent aujourd'hui des techniques d'intelligence artificielle — réseaux de neurones graphiques, apprentissage par renforcement, ou modèles séquentiels —, ils reposent néanmoins généralement sur un modèle unique et statique, sélectionné a priori indépendamment du contexte courant. Cette rigidité les rend peu adaptés à la nature dynamique des cyberattaques modernes, qui évoluent et s'adaptent pour contourner les défenses. Les limitations principales incluent :

- rigidité des modèles : Incapacité à s'adapter aux nouvelles tactiques, techniques et procédures (TTP) ;
- absence de sélection contextuelle : Pas de flexibilité dans le choix du modèle le plus approprié selon le scénario ;
- dépendances temporelles limitées : Difficulté à capturer les relations à long terme dans les séquences d'attaque.

6.1.4 Questions de recherche

Ce chapitre adresse les questions de recherche suivantes :

Q1 Comment exploiter les structures de graphes pour capturer les relations complexes entre les composants d'une attaque ?

Q2 Comment utiliser l'apprentissage par renforcement pour la prise de décision séquentielle dans des scénarios d'attaque évolutifs ?

Q3 Comment concevoir un méta-modèle capable de sélectionner dynamiquement le modèle le plus approprié selon le contexte ?

6.1.5 Contributions

Ce chapitre propose un cadre novateur à double modèle pour la projection d'attaque combinant deux techniques puissantes d'apprentissage automatique. Les contributions principales sont :

1. application pionnière des GAT à la projection d'attaque : À notre connaissance, c'est la première utilisation des Réseaux d'Attention sur Graphes pour la projection de cyberattaques, offrant une approche novatrice qui capture les relations complexes entre les composants d'attaque.
2. modèle DQN avec architecture Transformer : Développement d'un modèle de Deep Q-Network intégrant des encodeurs Transformer pour la prise de décision séquentielle dans les scénarios d'attaque.
3. méta-modèle de sélection contextuelle : Conception d'un système de méta-apprentissage basé sur k-NN pour sélectionner dynamiquement entre GAT et DQN selon les caractéristiques des données.
4. validation expérimentale robuste : Démonstration de l'efficacité sur le jeu de données CTU-13 avec des exactitudes de 94,86% (GAT) et 98,95% (DQN).

6.1.6 Organisation du chapitre

Le reste de ce chapitre est organisé comme suit. La Section 6.2 présente les travaux connexes sur la projection d'attaque et la sélection de modèles. La Section 6.3 expose les fondements théoriques des GAT et du DRL. La Section 6.4 détaille la méthodologie proposée, incluant l'architecture globale et les modèles individuels. La Section 6.5 présente les résultats expérimentaux. La Section 6.6 propose une analyse comparative avec l'état de l'art. La Section 6.7 discute les forces, limitations et implications pratiques. Enfin, la Section 6.8 conclut ce chapitre.

6.2 Travaux connexes

Cette section présente une revue des approches existantes pour la projection d'attaque et la sélection dynamique de modèles.

6.2.1 Approches par apprentissage par renforcement profond

Dehghan et al. [63] ont introduit le cadre ProAPT, combinant l'Apprentissage par Renforcement Profond (DRL) [61, 62] et les réseaux Long Short-Term Memory (LSTM) [48] pour la prédiction des APT [22]. Le système modélise la progression de l'attaque comme un Processus de Décision de Markov (MDP) [61], où un agent apprend à prédire les étapes d'attaque suivantes par interaction avec l'environnement. Cette approche s'inscrit dans une tendance croissante d'application du DRL à la cybersécurité [64, 89], notamment pour la modélisation de comportements adversariaux séquentiels [70, 25].

Architecture de ProAPT

Le cadre traite le jeu de données DAPT2020 en utilisant un espace d'états représentant les différentes étapes APT, de la reconnaissance à l'exfiltration de données. Un mécanisme de récompense renforce les prédictions correctes tout en pénalisant les incorrectes. Le composant LSTM capture les dépendances temporelles entre les étapes d'attaque.

Résultats et limitations

Bien qu'atteignant des résultats impressionnants (F1-score : 0,9533, exactitude : 0,9736), le cadre montre des limitations dans la gestion des séquences d'attaque étendues, car la structure LSTM a du mal à maintenir l'information sur de longues dépendances temporelles. De plus, la nature séquentielle du modèle restreint sa capacité à capturer les relations complexes et non linéaires dans les schémas APT.

6.2.2 Approches par modèles de Markov cachés

Ghafir et al. [70] ont proposé un cadre en deux phases pour la détection et la prédiction des APT :

Phase 1 : Reconstruction de scénario d'attaque

La première phase reconstruit les campagnes APT potentielles en corrélant les alertes temporellement et spatialement reliées. Le système regroupe les alertes selon des attributs spécifiques dans une fenêtre temporelle désignée, les mappant à différentes étapes du cycle de vie APT.

Phase 2 : Décodage et prédiction d'attaque

La deuxième phase emploie des Modèles de Markov Cachés (HMM) avec l'algorithme de Viterbi pour déterminer la séquence la plus probable d'étapes APT et prédire la progression de l'attaque.

Résultats

La validation expérimentale a utilisé un jeu de données synthétique de 6 000 alertes (3 700 liées aux APT et 2 300 non liées). Les résultats démontrent :

— 91,80% d'exactitude dans l'identification des étapes APT avec deux alertes corrélées, atteignant 100% avec plus de deux alertes ;

— exactitude de prédiction de l'étape suivante allant de 66,50% (2 alertes) à 100% (4+ alertes).

Cependant, les HMM ont des difficultés à capturer les dépendances à longue portée, et l'algorithme de Viterbi offre une adaptabilité limitée pour les campagnes APT complexes.

6.2.3 Approches de sélection d'algorithmes

Chalé et al. [89] proposent deux approches distinctes pour la sélection d'algorithmes d'apprentissage automatique dans les IDS :

— méthode par règles empiriques : Utilise des règles définies par des experts pour recommander des algorithmes basés sur les caractéristiques des données. Simple mais manque d'adaptabilité.

— approche par méta-apprentissage : Pilotée par les données, exploite les tâches antérieures et les méta-caractéristiques pour prédire le meilleur algorithme pour de nouveaux jeux de données. Plus adaptable et évolutive.

6.2.4 Sélection dynamique de modèles

Cruz et al. [90] proposent une approche d'apprentissage automatique en deux étapes pour la sélection dynamique de modèles. Le cadre utilise un classificateur k-NN pour automatiser le processus de sélection de modèle basé sur les caractéristiques du processus et les données de performance historiques.

6.2.5 Réseaux d'attention sur graphes pour la détection de malware

Catal et al. [57] proposent une approche de détection de malware pour les Systèmes de Transport Intelligents (ITS) basée sur les GAT. Leur méthodologie modélise les composants ITS comme des nœuds dans un graphe, avec des arêtes représentant les interactions entre entités. Les résultats démontrent une exactitude de détection élevée et une robustesse dans les environnements ITS hétérogènes.

TABLE 6.1 – Synthèse comparative des approches de projection d'attaque

Approche	Temp.	Graph.	Adapt.	Sél.Dyn.
ProAPT [63]	✓	×	∼	×
HMM [70]	✓	×	∼	×
GAT-ITS [57]	∼	✓	×	×
DRL-IDS [64]	✓	×	∼	×
Notre approche	✓	✓	✓	✓

Temp.=Modélisation temporelle, Graph.=Modélisation graphique, Adapt.=Adaptation contextuelle, Sél.Dyn.=Sélection dynamique, ∼=Partiel

6.2.6 Apprentissage par renforcement profond pour la détection d'intrusion

Lopez-Martin et al. [64] étudient l'application du DRL pour améliorer les IDS. La méthodologie implique l'entraînement du modèle DRL dans une configuration MDP, où les états représentent les caractéristiques de l'activité réseau, les actions correspondent aux décisions de classification, et les récompenses sont basées sur l'exactitude des prédictions.

6.2.7 Synthèse et positionnement

Le Tableau 6.1 synthétise les caractéristiques des approches existantes et positionne notre contribution.

Notre approche se distingue par la combinaison de la modélisation temporelle et graphique avec un mécanisme de sélection dynamique de modèle.

6.3 Fondements théoriques

Cette section présente les fondements théoriques des deux paradigmes utilisés : les Réseaux d'Attention sur Graphes et l'Apprentissage par Renforcement Profond.

6.3.1 Réseaux de neurones sur graphes (GNN)

Les Réseaux de Neurones sur Graphes (GNN) sont une classe de modèles d'apprentissage profond spécifiquement conçus pour travailler sur des données structurées en graphes [55]. Contrairement aux réseaux de neurones traditionnels qui traitent des données dans des domaines euclidiens, les GNN excellent dans l'apprentissage à partir de l'information relationnelle encodée dans les graphes.

Principe de propagation de messages

Les GNN fonctionnent en agrégeant itérativement l'information des voisins d'un nœud pour calculer une nouvelle représentation pour chaque nœud. Ce processus, souvent appelé « passage de messages » ou « agrégation de voisinage », garantit que la représentation d'un nœud capture non seulement ses propres caractéristiques mais aussi l'information structurelle et caractéristique de son voisinage local.

Mathématiquement, pour un nœud v, sa représentation $h_v^{(k+1)}$ à la couche $k+1$ est calculée comme :

$$h_v^{(k+1)} = \text{Update}\left(h_v^{(k)}, \text{Aggregate}\left(\{h_u^{(k)} : u \in \mathcal{N}(v)\}\right)\right), \tag{6.1}$$

où $\mathcal{N}(v)$ est l'ensemble des voisins du nœud v, et les fonctions Aggregate et Update sont apprises pendant l'entraînement.

6.3.2 Réseaux d'attention sur graphes (GAT)

Les Réseaux d'Attention sur Graphes (GAT) [60] sont un type spécifique de GNN qui introduit des mécanismes d'attention pour améliorer le processus d'agrégation.

Mécanisme d'attention

Pour chaque nœud v et son voisin u, un coefficient d'attention α_{vu} est calculé pour quantifier l'importance de u pour v :

$$\alpha_{vu} = \frac{\exp\left(\text{LeakyReLU}\left(\mathbf{a}^T \cdot [W\mathbf{h}_v \| W\mathbf{h}_u]\right)\right)}{\sum_{k\in\mathcal{N}(v)} \exp\left(\text{LeakyReLU}\left(\mathbf{a}^T \cdot [W\mathbf{h}_v \| W\mathbf{h}_k]\right)\right)}, \tag{6.2}$$

où :

— $W \in \mathbb{R}^{F'\times F}$ est une matrice de poids apprise ;

— $\mathbf{a} \in \mathbb{R}^{2F'}$ est un vecteur appris pour le mécanisme d'attention ;

— $\|$ dénote la concaténation ;

— LeakyReLU est une fonction d'activation.

Agrégation de voisinage

En utilisant les coefficients d'attention calculés, la représentation mise à jour du nœud v est obtenue comme une somme pondérée des caractéristiques de ses voisins :

$$\mathbf{h}_v^{\text{new}} = \sigma\left(\sum_{u\in\mathcal{N}(v)} \alpha_{vu} W\mathbf{h}_u\right), \tag{6.3}$$

où σ est une fonction d'activation non linéaire.

Attention multi-têtes

Pour améliorer la robustesse et l'expressivité du modèle, les GAT utilisent souvent plusieurs têtes d'attention. Les sorties de ces têtes sont soit concaténées soit moyennées :

$$\mathbf{h}_v^{\text{new}} = \Big\|_{k=1}^{K} \sigma\left(\sum_{u\in\mathcal{N}(v)} \alpha_{vu}^k W^k \mathbf{h}_u\right), \tag{6.4}$$

où K est le nombre de têtes d'attention et $\|$ représente la concaténation.

Avantages des GAT

- Attention Dynamique : Les GAT se focalisent de manière adaptative sur les nœuds les plus pertinents.
- versatilité : Les GAT peuvent gérer des graphes avec des tailles de voisinage variables.
- Performance État de l'Art : Les GAT ont démontré de fortes performances sur diverses tâches de classification de nœuds et de graphes.

6.3.3 Apprentissage par renforcement profond (DRL)

L'Apprentissage par Renforcement Profond (DRL) est un sous-domaine de l'apprentissage par renforcement qui exploite les techniques d'apprentissage profond pour résoudre des problèmes de prise de décision complexes [62].

Composants clés du DRL

Agent et environnement En DRL, un agent interagit avec un environnement en observant des états, prenant des actions et recevant des récompenses. L'objectif de l'agent est d'apprendre une politique $\pi(a|s)$ qui mappe les états (s) aux actions (a) de manière à maximiser la récompense cumulative :

$$R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}, \tag{6.5}$$

où r_t est la récompense au temps t, et $\gamma \in [0, 1]$ est le facteur d'actualisation.

Deep Q-Network (DQN) Le Deep Q-Network (DQN) [62] est un algorithme DRL qui utilise un réseau de neurones pour approximer la fonction Q, qui estime le retour attendu de la prise d'une action dans un état donné. La valeur Q est mise à jour en utilisant l'équation de Bellman :

$$Q(s,a) \leftarrow Q(s,a) + \alpha\left[r + \gamma \max_{a'} Q(s',a') - Q(s,a)\right], \tag{6.6}$$

où α est le taux d'apprentissage, r est la récompense, et s' est l'état suivant.

Techniques clés du DRL

— Experience Replay [67] : Un tampon mémoire stocke les interactions passées pour décorréler les données et stabiliser l'entraînement.

— exploration vs Exploitation : Une stratégie epsilon-greedy équilibre l'exploration et l'exploitation.

— réseau Cible [65] : Un réseau cible calcule des cibles Q-value stables, réduisant l'instabilité de l'entraînement.

6.3.4 Jeu de données CTU-13

Le jeu de données CTU-13 [2] a été présenté en détail au chapitre précédent (Section 5.4). Pour rappel, il comprend 13 scénarios de trafic botnet étiqueté (Neris, Rbot, Virut, etc.) avec des protocoles de commande et contrôle variés (IRC, HTTP, P2P). Dans le cadre de ce chapitre, le trafic de chaque scénario a été traité par Suricata pour générer les alertes IDS utilisées en entrée des modèles GAT et DQN.

6.4 Méthodologie proposée

Cette section présente l'architecture globale du système et les détails des modèles individuels.

6.4.1 Architecture globale du système

L'architecture globale du système est illustrée dans la Figure 6.1. Elle se compose de quatre étapes principales :

1. données d'entrée : Les groupes d'alertes corrélées sont utilisés comme données d'entrée. Ces alertes sont prétraitées pour extraire les caractéristiques pertinentes telles que la sévérité, les flux réseau, les ports et les métadonnées HTTP.
2. sélection par Méta-Modèle : Un méta-modèle (classificateur k-NN) est utilisé pour sélectionner dynamiquement le modèle le plus approprié entre le GAT et le DQN basé sur les caractéristiques des données.
3. exécution du Modèle : Le modèle sélectionné (GAT ou DQN) est exécuté pour prédire les phases d'attaque suivantes.
4. sortie : Les prédictions des modèles sont agrégées pour produire les phases d'attaque prédites.

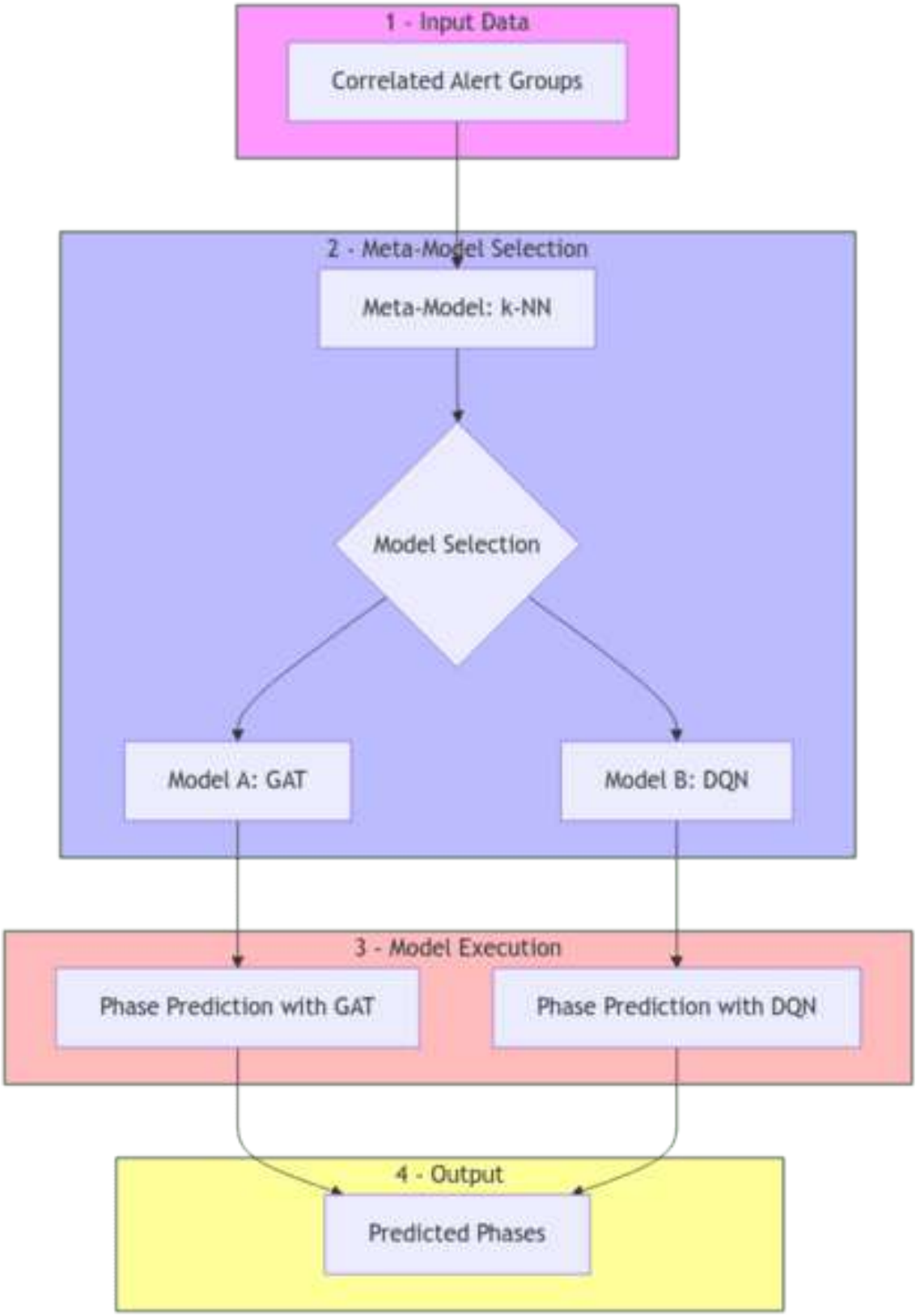


FIGURE 6.1 – Architecture globale du système avec sélection dynamique entre les modèles GAT et DQN.

6.4.2 Modèle GAT avec encodeur Transformer

Le modèle proposé combine un Réseau d'Attention sur Graphes (GAT) avec un encodeur Transformer pour capturer efficacement les dépendances locales et globales dans les données d'alertes.

Architecture du modèle

Le modèle se compose des composants suivants :

— couches GAT : Les couches GAT agrègent l'information des nœuds voisins en utilisant des mécanismes d'attention. La première couche transforme les caractéristiques d'entrée en une représentation cachée, tandis que la seconde affine cette représentation.

— encodeur Transformer : L'encodeur Transformer traite les embeddings de nœuds raffinés des couches GAT. Il capture les dépendances globales et les schémas temporels à travers des mécanismes d'auto-attention.

— couche Entièrement Connectée : Une couche finale mappe les sorties du Transformer aux catégories d'alertes prédites.

Formalisation

Soit $G = (V, E)$ un graphe où V représente les alertes et E les relations entre alertes. Pour chaque nœud $v \in V$, le modèle calcule :

$$\mathbf{h}_v^{(1)} = \mathrm{GAT}_1(\mathbf{x}_v, \{\mathbf{x}_u : u \in \mathcal{N}(v)\}) \tag{6.7}$$

$$\mathbf{h}_v^{(2)} = \mathrm{GAT}_2(\mathbf{h}_v^{(1)}, \{\mathbf{h}_u^{(1)} : u \in \mathcal{N}(v)\}) \tag{6.8}$$

$$\mathbf{z}_v = \mathrm{Transformer}(\mathbf{h}_v^{(2)}) \tag{6.9}$$

$$\hat{y}_v = \mathrm{softmax}(W_{\mathrm{out}}\mathbf{z}_v + \mathbf{b}_{\mathrm{out}}) \tag{6.10}$$

6.4.3 Modèle DQN avec architecture Transformer

Le modèle DQN proposé combine un Deep Q-Network avec une architecture Transformer pour prédire la phase suivante d'une attaque en apprenant les actions optimales basées sur l'état actuel.

Architecture du modèle

— couche d'Entrée : Un vecteur d'état représentant le groupe d'alertes actuel, incluant des caractéristiques telles que la sévérité, le protocole, les statistiques de flux et les données temporelles.

— encodeur Transformer : Les caractéristiques d'entrée passent par un encodeur Transformer qui utilise l'auto-attention pour se concentrer sur les parties importantes des données.

— couches Cachées : La sortie de l'encodeur Transformer est traitée par deux couches cachées entièrement connectées avec des fonctions d'activation ReLU.

— couche de Sortie : La couche de sortie génère des valeurs Q pour chaque action possible (phase suivante de l'attaque).

Processus d'entraînement

Le DQN avec Transformer est entraîné en utilisant l'apprentissage par renforcement :

1. Experience Replay : Un tampon mémoire stocke les expériences passées (s, a, r, s') pour décorréler les échantillons d'entraînement.
2. stratégie ϵ-greedy : Équilibre exploration (actions aléatoires) et exploitation (actions avec les valeurs Q les plus élevées).
3. réseau Cible : Un réseau cible calcule des cibles Q-value stables et est périodiquement mis à jour avec les poids du réseau principal.

6.4.4 Approche de méta-apprentissage pour la sélection dynamique

Pour sélectionner dynamiquement le modèle le plus approprié, nous employons une approche de méta-apprentissage inspirée des travaux de Cruz et al. [90].

Vue d'ensemble de l'approche

L'approche débute par la définition des méta-caractéristiques, qui consiste à extraire un vecteur de descripteurs contextuels à partir des données d'alertes courantes : le nombre d'alertes dans le groupe actuel, la sévérité moyenne, la distribution des catégories d'alertes, les caractéristiques temporelles telles que le temps entre alertes, ainsi que le type des variables présentes (numériques ou catégorielles). Ces méta-caractéristiques forment la signature contextuelle du groupe d'alertes à traiter.

Dans un second temps, les modèles candidats GAT et DQN sont entraînés sur des données d'alertes historiques, et leurs domaines de compétence respectifs sont déterminés en identifiant, pour chaque configuration de données, le modèle ayant obtenu les meilleures

TABLE 6.2 – Métriques de performance globales du modèle GAT

Métrique	**Valeur**
Exactitude	94,86%
Précision	94,96%
Rappel	94,86%
Score F1	94,91%

performances. Un méta-apprenant de type k-NN est ensuite entraîné en associant chaque configuration de méta-caractéristiques à son modèle optimal. Lors de l'inférence, les méta-caractéristiques d'un nouveau groupe d'alertes sont extraites et soumises au classificateur k-NN, qui réalise la sélection dynamique du modèle le mieux adapté au contexte courant.

Avantages de l'approche

L'approche de méta-apprentissage présente plusieurs atouts complémentaires. Son adaptabilité est assurée par le classificateur k-NN, qui s'adapte automatiquement à de nouveaux jeux de données sans nécessiter de réentraînement complet. Elle est évolutive, puisqu'on peut étendre le portefeuille de modèles candidats ou enrichir l'espace des méta-caractéristiques sans modifier l'architecture globale. Le processus de sélection de modèle est entièrement automatisé, supprimant le besoin d'une sélection manuelle ad hoc par l'expert. Enfin, la nature non paramétrique du k-NN offre une interprétabilité naturelle : la décision de sélection peut être tracée jusqu'aux scénarios d'entraînement les plus similaires.

6.5 Expérimentations et résultats

Cette section présente les résultats expérimentaux des modèles GAT et DQN sur les alertes Suricata dérivées du jeu de données CTU-13.

6.5.1 Configuration expérimentale

Les alertes Suricata sont regroupées en deux catégories principales :

— *Generic Protocol Command Decode*

— *Misc activity*

6.5.2 Résultats du modèle GAT

Métriques de performance globales

Le Tableau 6.2 présente les métriques de performance globales du modèle GAT.
La Figure 6.2 visualise ces métriques.

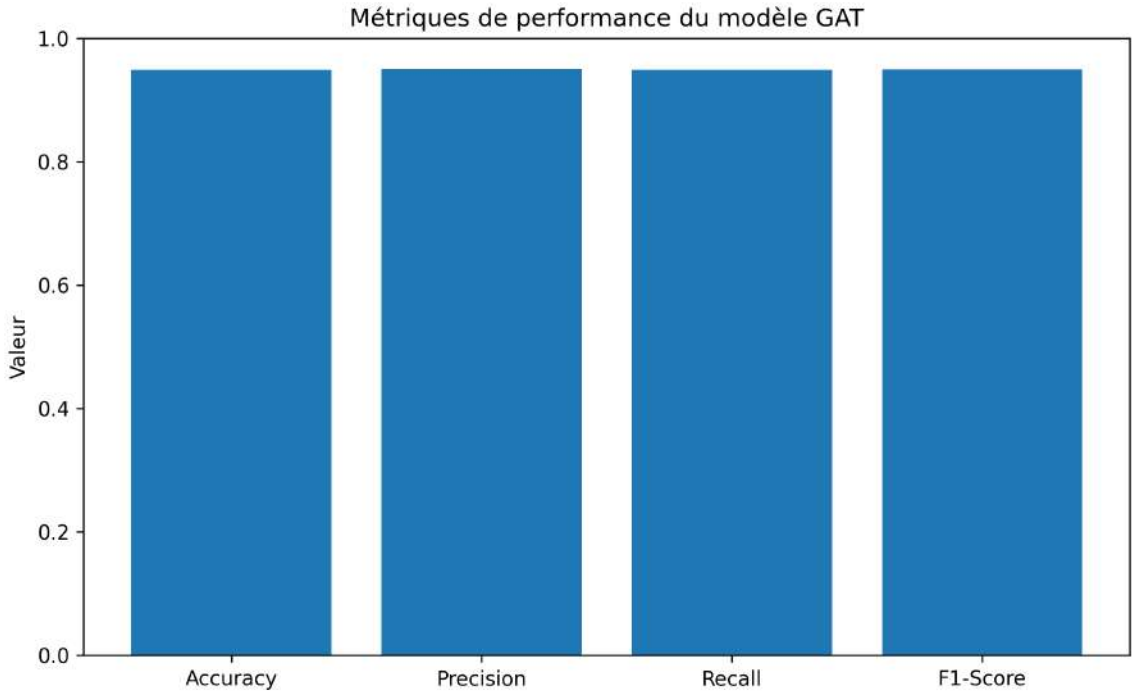


FIGURE 6.2 – Métriques de performance globales du modèle GAT.

TABLE 6.3 – Détails de la matrice de confusion pour le modèle GAT

Classe	VP	FP	FN	VN
Generic Protocol Command Decode	330	66	50	1813
Misc activity	1813	50	66	330

Matrice de confusion

Le Tableau 6.3 présente les détails de la matrice de confusion.

La Figure 6.3 présente la matrice de confusion visualisée.

Analyse

Le modèle GAT atteint une exactitude de 94,86% et un score F1 de 94,91% sur l'ensemble de test. Ces résultats démontrent que le mécanisme d'attention dans les couches GAT, combiné au contexte global fourni par l'encodeur Transformer, contribue à une performance robuste en capturant les dépendances locales et globales dans les séquences d'alertes.

6.5.3 Résultats du modèle DQN

Métriques de performance globales

Le Tableau 6.4 présente les métriques de performance du modèle DQN.

La Figure 6.4 visualise ces métriques.

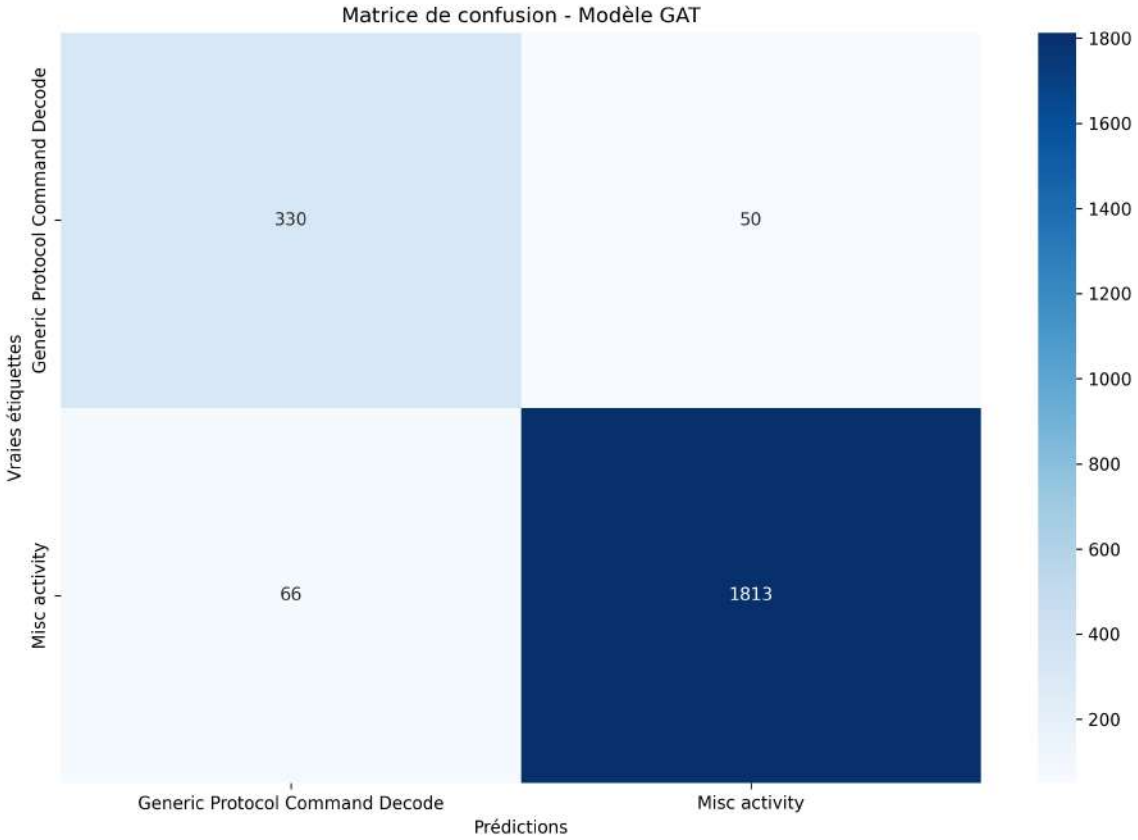


FIGURE 6.3 – Matrice de confusion du modèle GAT + Transformer.

TABLE 6.4 – Métriques de performance du modèle DQN

Métrique	**Valeur**
Exactitude	98,95%
Précision	97,91%
Rappel	98,95%
Score F1	98,42%

TABLE 6.5 – Détails de la matrice de confusion pour le modèle DQN

Classe	**VP**	**FP**	**FN**	**VN**
Generic Protocol Command Decode	0	0	4	376
Misc activity	376	4	0	0

Matrice de confusion

Le Tableau 6.5 présente les détails de la matrice de confusion.

La Figure 6.5 présente la matrice de confusion visualisée.

Analyse

Le modèle DQN atteint une performance très élevée sur la classe majoritaire *Misc activity* (376 vrais positifs et seulement 4 faux positifs), mais échoue à identifier correc-

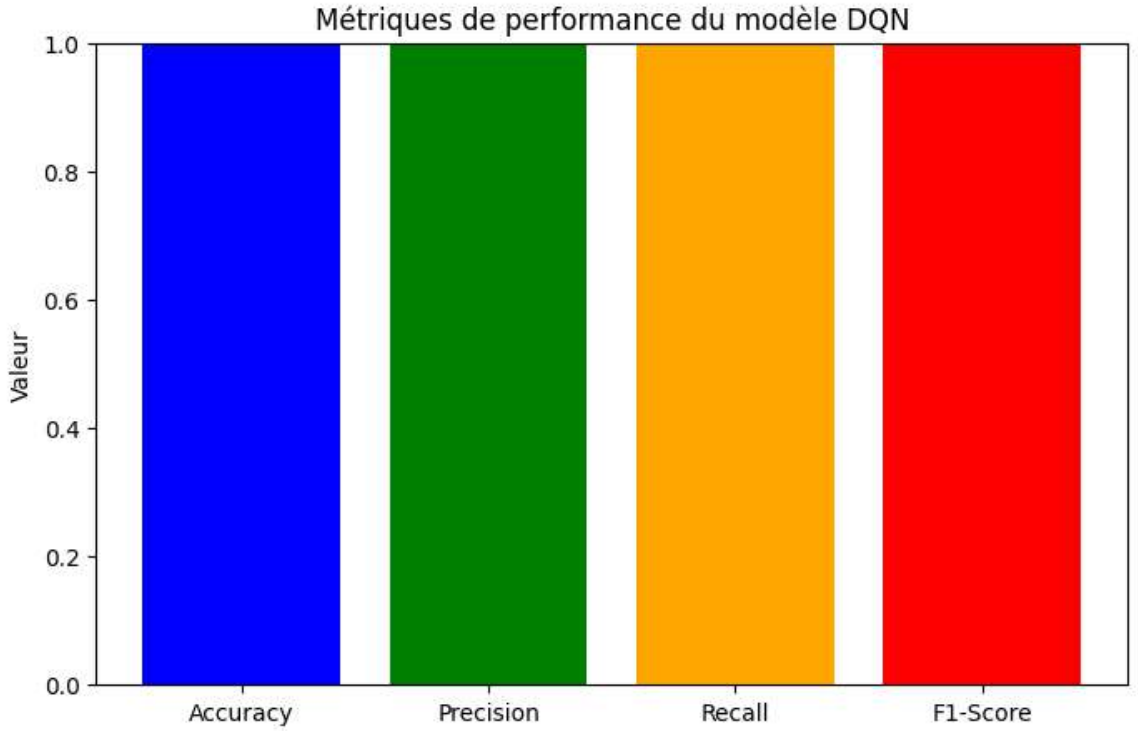


FIGURE 6.4 – Métriques de performance du modèle DQN.

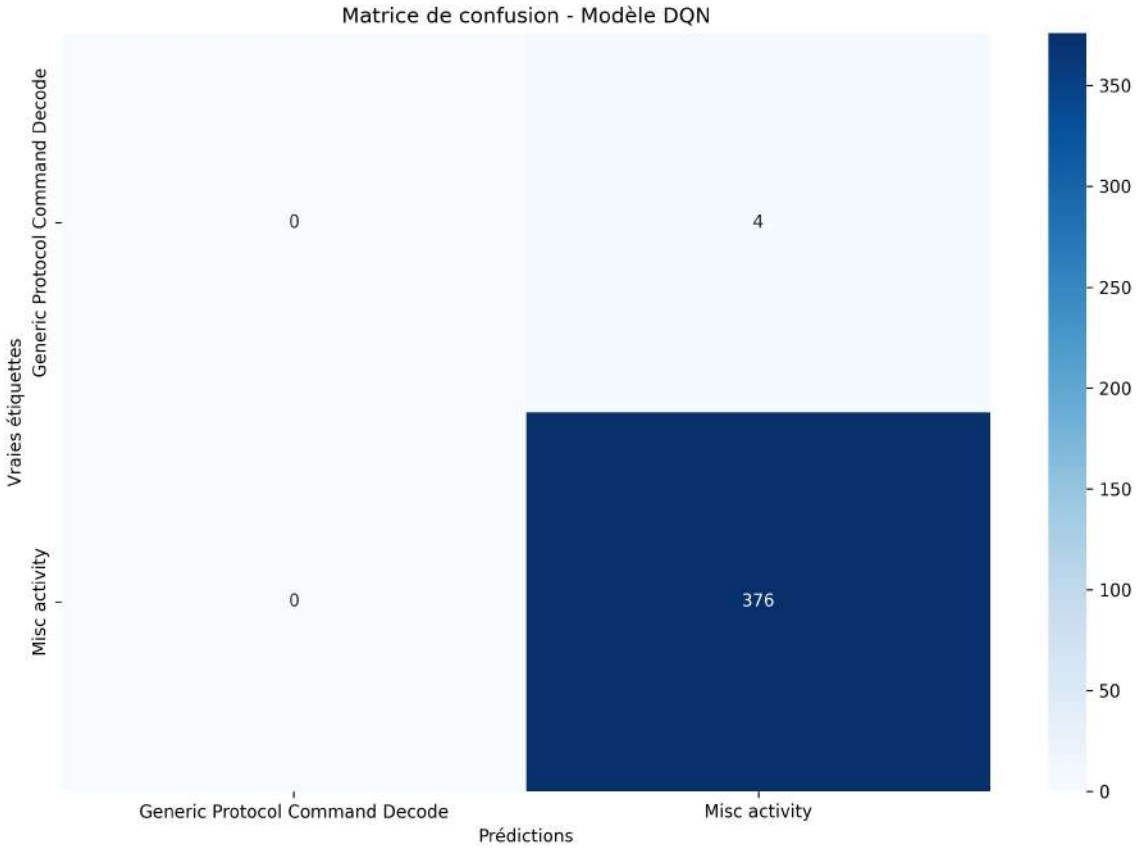


FIGURE 6.5 – Matrice de confusion du modèle DQN.

TABLE 6.6 – Performance des modèles sur 20 000 alertes synthétiques

Modèle	**Exact.**	**Préc.**	**Rappel**	**F1**
GAT (train)	99,79%	99,79%	99,79%	99,79%
DQN (test)	100,00%	100,00%	100,00%	100,00%

tement les quatre groupes de test appartenant à la classe minoritaire *Generic Protocol Command Decode*. Ce comportement est cohérent avec le fort déséquilibre de classes au niveau des groupes (19 *Generic* vs 1878 *Misc* groupes dans l'ensemble du jeu de données).

Cette limitation du modèle DQN révèle un enjeu classique en apprentissage par renforcement lorsque l'agent est confronté à des données fortement déséquilibrées. L'agent DQN optimise sa politique en maximisant la récompense cumulative, ce qui conduit naturellement à privilégier les prédictions correctes sur la classe majoritaire, où le nombre d'exemples d'entraînement est significativement plus élevé. Dans notre cas, la rareté des exemples de la classe *Generic Protocol Command Decode* durant la phase d'entraînement n'a pas permis à l'agent d'apprendre une représentation discriminante suffisante pour cette catégorie.

Néanmoins, les résultats globaux restent remarquables avec une exactitude de 98,95%, démontrant l'efficacité du modèle DQN pour la projection de phases d'attaque dans des scénarios séquentiels typiques. La capacité du DQN à modéliser les dépendances temporelles et à apprendre des stratégies de décision optimales le rend particulièrement adapté aux attaques évolutives où les phases successives présentent des relations causales fortes.

Pour améliorer la performance sur les classes minoritaires, plusieurs stratégies pourraient être envisagées : l'utilisation de techniques de rééchantillonnage (sur-échantillonnage SMOTE ou sous-échantillonnage de la classe majoritaire), l'ajustement des fonctions de récompense pour pénaliser davantage les erreurs sur les classes rares, ou encore l'emploi d'architectures d'apprentissage par renforcement avancées telles que le Rainbow DQN qui combine plusieurs améliorations algorithmiques pour une meilleure généralisation.

6.5.4 Validation complémentaire sur alertes synthétiques

Pour valider davantage la capacité des modèles proposés à capturer des schémas discriminants dans un cadre contrôlé, nous avons conduit une expérience supplémentaire sur un jeu de données synthétique de 20 000 alertes Suricata.

Résultats

Le Tableau 6.6 résume les performances sur ce jeu de données synthétique.

Ces expériences synthétiques montrent que, lorsque les distributions de données sous-jacentes fournissent des signaux discriminants forts et propres, les architectures GAT et

DQN sont capables d'apprendre presque parfaitement les frontières de décision entre les catégories d'alertes.

6.6 Analyse comparative avec l'état de l'art

Cette section présente une analyse comparative détaillée avec les approches de l'état de l'art.

6.6.1 Méthodes de référence

Nous comparons notre cadre avec les méthodes suivantes :

— B1 : ProAPT [63] : Cadre DRL + LSTM pour la projection APT. F1-score : 0,9533, exactitude : 97,36% sur DAPT2020.

— B2 : HMM-APT [70] : Cadre HMM avec décodage Viterbi. Exactitude : 91,80% sur jeu synthétique.

— B3 : GAT-ITS [57] : GAT pour la détection de malware. Exactitude : ≈96% sur fichiers APK Android.

— B4 : DRL-IDS [64] : DRL pour la détection d'intrusion. Exactitude : 78-96% selon le jeu de données.

6.6.2 Limitations de la comparaison inter-études

Avant de présenter l'analyse, il convient de reconnaître plusieurs limitations inhérentes à cette comparaison. D'une part, chaque méthode a été évaluée sur un jeu de données différent, ce qui introduit une hétérogénéité rendant toute comparaison directe des métriques délicate. D'autre part, les tâches adressées ne sont pas identiques : ProAPT, HMM et notre approche se concentrent sur la *projection* d'attaques, tandis que GAT-ITS et DRL-IDS traitent la *détection*, deux problèmes aux objectifs et aux métriques distincts. Enfin, l'absence de benchmark unifié — nous n'avons pas ré-implémenté les méthodes de référence sur CTU-13 — limite la portée des conclusions quantitatives. Ces précautions méthodologiques doivent guider l'interprétation des résultats présentés dans les sous-sections suivantes.

6.6.3 Comparaison qualitative

Le Tableau 6.7 présente une comparaison qualitative des approches.

Ce tableau qualitatif révèle que notre approche GAT-DQN est la seule à couvrir simultanément les cinq dimensions évaluées : modélisation temporelle, modélisation graphique, adaptabilité, sélection dynamique, et focalisation sur la projection d'attaque. ProAPT et

TABLE 6.7 – Comparaison qualitative des approches

Méthode	**Tâche**	**Temp.**	**Graph.**	**Adapt.**	**Sél.**
ProAPT	Proj.	✓	×	∼	×
HMM-APT	Proj.	✓	×	∼	×
GAT-ITS	Dét.	∼	✓	×	×
DRL-IDS	Dét.	✓	×	∼	×
Notre (GAT)	**Proj.**	✓	✓	✓	✓
Notre (DQN)	**Proj.**	✓	∼	✓	✓

Temp.=Temporel, Graph.=Graphique, Adapt.=Adaptatif, Sél.=Sélection dynamique, Proj.=Projection, Dét.=Détection, ∼=Partiel

TABLE 6.8 – Comparaison quantitative des performances

Méthode	**Exact.**	**F1**	**Dataset**
ProAPT	97,36	95,33	DAPT2020
HMM-APT	91,80	–	Synthétique
GAT-ITS	96,00	95,70	Android APK
DRL-IDS	78-96	–	NSL-KDD/AWID
Notre (GAT)	**94,86**	**94,91**	**CTU-13**
Notre (DQN)	**98,95**	**98,42**	**CTU-13**

Note : Les résultats proviennent de jeux de données différents et ne sont pas directement comparables.

HMM-APT assurent la modélisation temporelle mais sont dépourvus de toute représentation graphique et de sélection dynamique. GAT-ITS dispose d'une modélisation graphique mais ne traite que la détection, non la projection, et n'offre aucune adaptabilité. DRL-IDS, malgré sa composante temporelle, reste limité à la détection avec une adaptabilité partielle. Notre cadre se distingue donc comme le seul à unifier ces dimensions dans un seul système orienté projection.

6.6.4 Comparaison quantitative

Le Tableau 6.8 présente les métriques de performance rapportées.

Malgré l'hétérogénéité des jeux de données, ce tableau illustre le niveau de performance compétitif de notre approche. Notre modèle DQN+Transformer atteint 98,95% d'exactitude et un F1 de 98,42% sur CTU-13, surpassant ProAPT (97,36% / F1 95,33%) et HMM-APT (91,80%) sur leurs jeux respectifs. Le modèle GAT affiche 94,86% d'exactitude, légèrement en deçà de GAT-ITS (96%) mais sur une tâche fondamentalement différente (projection vs détection). DRL-IDS présente quant à lui une plage de performance étendue (78–96%) selon le jeu de données, révélant une sensibilité au domaine d'évaluation. Ces résultats confirment la viabilité de notre cadre tout en soulignant la nécessité d'un benchmark unifié pour une comparaison rigoureuse.

TABLE 6.9 – Matrice de support des fonctionnalités

Fonctionnalité	**B1**	**B2**	**B3**	**B4**	**Notre**
Projection d'attaque	✓	✓	×	×	✓
Modélisation temporelle	✓	✓	∼	✓	✓
Modélisation graphique	×	×	✓	×	✓
Adaptation contextuelle	∼	∼	×	∼	✓
Sélection dynamique	×	×	×	×	✓
Corrélation d'alertes	∼	✓	×	×	✓
Support multi-phases	✓	∼	×	×	∼
Temps réel	∼	∼	✓	✓	✓

∼ = Support partiel

6.6.5 Matrice de support des fonctionnalités

Le Tableau 6.9 présente une comparaison détaillée des fonctionnalités.

Cette matrice fonctionnelle confirme et précise les observations qualitatives précédentes. Notre approche est la seule à obtenir un support complet sur la projection d'attaque, la modélisation temporelle et graphique, l'adaptation contextuelle, la sélection dynamique et la corrélation d'alertes — six fonctionnalités sur huit. Les méthodes de référence B1 (ProAPT) et B2 (HMM-APT) assurent la projection et la modélisation temporelle, mais échouent sur les dimensions graphique et de sélection dynamique. B3 (GAT-ITS) apporte la modélisation graphique mais ne couvre ni la projection ni la corrélation. B4 (DRL-IDS) supporte le temps réel et la modélisation temporelle, mais sans projection ni graphe. Seul le support multi-phases reste partiel dans notre cadre, constituant une direction d'amélioration identifiée.

6.6.6 Discussion du positionnement

Nouveauté architecturale

Notre cadre de méta-sélection est, à notre connaissance, le premier à combiner des modèles basés sur les graphes et séquentiels avec une sélection dynamique pour la projection d'attaque.

Performance sur CTU-13

Notre modèle DQN+Transformer atteint 98,95% d'exactitude sur l'évaluation CTU-13/Suricata. Le modèle GAT atteint 94,86%, démontrant la viabilité de l'approche graphique pour la projection d'attaque.

Contributions méthodologiques

Notre approche fournit un support pour la modélisation temporelle et relationnelle, l'adaptation contextuelle et la sélection dynamique de modèle — des caractéristiques distinctives non simultanément présentes dans les approches de référence.

6.7 Discussion

Cette étude démontre des progrès significatifs dans la projection de cyberattaques en combinant les Réseaux d'Attention sur Graphes (GAT) et l'Apprentissage par Renforcement Profond (DQN), guidés par un méta-modèle dynamique.

6.7.1 Forces

L'efficacité du cadre GAT-DQN repose sur trois facteurs complémentaires. Premièrement, la complémentarité des modèles : le GAT excelle dans la capture des relations structurelles entre composants d'une attaque, tandis que le DQN se concentre sur la prise de décision séquentielle, ce qui permet d'adresser une large variété de scénarios d'attaque. Deuxièmement, la sélection dynamique par méta-modèle assure une adaptabilité pilotée par le contexte, orientant automatiquement vers le modèle optimal selon les caractéristiques situationnelles. Troisièmement, la généralisation robuste du système est démontrée par des exactitudes élevées et constantes sur les divers scénarios du CTU-13.

6.7.2 Limitations

Plusieurs défis ont été identifiés. Premièrement, la portée du cadre est actuellement restreinte à la prédiction de la phase immédiatement suivante, ce qui limite son applicabilité aux attaques multi-phases complexes s'étendant sur de longues durées. Deuxièmement, la dépendance au jeu de données CTU-13 peut réduire l'adaptabilité du modèle aux méthodes d'attaque évolutives non représentées dans ce corpus. Troisièmement, les exigences computationnelles liées à l'intégration simultanée du GAT, du DQN et du méta-modèle pourraient constituer un frein au déploiement en temps réel dans des environnements SOC à ressources limitées. Enfin, le modèle DQN montre des difficultés sur les classes minoritaires, reflétant le déséquilibre inhérent au jeu de données et limitant la détection des phases d'attaque rares.

6.7.3 Discussion approfondie : déséquilibre de classes et compromis de conception

La sensibilité du modèle DQN au déséquilibre de classes mérite une discussion explicite, car elle touche à un enjeu opérationnel central pour les systèmes de cybersécurité : la détection des attaques rares, dont les Menaces Persistantes Avancées (APT).

Le jeu de données CTU-13 présente une distribution naturellement déséquilibrée, reflétant la réalité des environnements réseaux où certaines phases d'attaque (notamment les phases de commande et contrôle ou d'exfiltration) sont minoritaires par rapport aux phases de communication réseau ordinaires. Le DQN, optimisant une fonction de récompense globale sur l'ensemble des transitions observées, tend à maximiser ses performances sur la classe majoritaire au détriment de la classe minoritaire. Ce comportement est cohérent avec la littérature sur l'apprentissage par renforcement en contexte déséquilibré [64].

Ce compromis est assumé dans la conception de GAT-DQN pour deux raisons complémentaires. D'une part, le contexte d'utilisation privilégié du DQN — sélectionné par le méta-modèle k-NN pour les scénarios à forte dominante séquentielle — correspond aux phases d'attaque où la classe majoritaire porte l'essentiel de l'information prédictive utile pour la projection de la phase suivante. D'autre part, le méta-modèle de sélection contextuelle peut orienter vers le GAT dans les contextes où les classes minoritaires sont plus représentées, atténuant ainsi l'impact global du déséquilibre sur les performances du système. La validation complémentaire sur données synthétiques équilibrées confirme que les deux architectures sont capables d'apprendre des frontières décisionnelles correctes lorsque la distribution des classes est équilibrée, ce qui valide leur potentiel intrinsèque indépendamment du biais de distribution.

Des techniques de rééquilibrage — telles que le sur-échantillonnage adaptatif (SMOTE), la pondération différentielle des classes dans la fonction de perte, ou l'utilisation de mécanismes de mémoire prioritaire orientée vers les transitions rares — constituent des pistes d'amélioration identifiées pour les travaux futurs, dans l'objectif d'étendre l'applicabilité du cadre à la détection de classes d'attaques rares de type APT.

6.7.4 Implications pratiques

Le système offre un potentiel élevé pour la cybersécurité proactive. En anticipant avec précision la phase suivante d'une attaque, il permet aux équipes SOC d'activer des contre-mesures avant que le dommage ne se produise. La sélection dynamique de modèle automatise la sélection de la stratégie défensive la plus adaptée à chaque contexte, réduisant la charge décisionnelle des analystes. Enfin, la capacité d'adaptation aux tactiques évolutives — assurée par le méta-apprenant k-NN — constitue un avantage différenciant face aux menaces persistantes avancées qui modifient dynamiquement leurs TTP.

6.7.5 Directions futures

Plusieurs axes de recherche sont identifiés pour renforcer l'applicabilité du cadre. L'extension des capacités de prédiction au-delà de la phase immédiatement suivante vers des projections multi-étapes représente une priorité, afin de mieux couvrir les campagnes APT à longue durée. L'évaluation sur des jeux de données plus diversifiés et récents (DAPT2020, CICIOT-2023) permettrait d'éprouver la généralisation du méta-modèle. Des optimisations d'efficacité computationnelle — notamment la distillation du modèle GAT et la quantification — sont nécessaires pour le déploiement en environnement SOC temps réel. Par ailleurs, l'amélioration de l'interprétabilité des décisions du méta-modèle k-NN, par exemple via des méthodes LIME ou SHAP, renforcerait la confiance des opérateurs. Enfin, le développement de mécanismes de rééquilibrage adaptatifs (SMOTE, pondération de récompense) pour les classes d'attaque rares constitue un enjeu opérationnel majeur.

6.8 Conclusion

6.8.1 Synthèse des contributions

Ce chapitre a présenté une approche novatrice de projection de cyberattaques à travers le développement d'un cadre adaptatif combinant les Réseaux d'Attention sur Graphes et l'Apprentissage par Renforcement Profond. Les contributions principales sont :

1. innovation dans l'utilisation des GAT : Première application des Réseaux d'Attention sur Graphes à la projection d'attaque, capturant les relations complexes entre composants d'attaque.
2. modèle DQN+Transformer : Développement d'un modèle de Deep Q-Network intégrant des encodeurs Transformer pour la prise de décision séquentielle.
3. méta-modèle de sélection : Implémentation d'un méta-modèle basé sur k-NN qui sélectionne dynamiquement entre GAT et DQN selon les caractéristiques de l'attaque.
4. Performance élevée : Obtention d'une exactitude de 94,86% (GAT) et 98,95% (DQN) dans la prédiction de la phase suivante sur le jeu de données CTU-13.

6.8.2 Perspectives

Le succès de cette approche à double modèle ouvre de nouvelles perspectives pour les systèmes de cybersécurité adaptatifs. Les travaux futurs devraient se concentrer sur :

1. L'extension des capacités de prédiction à plusieurs phases d'attaque simultanément ;
2. L'intégration de mises à jour de schémas d'attaque en temps réel ;
3. L'optimisation de l'efficacité computationnelle pour les environnements de production.

Plus largement, cette recherche démontre le potentiel de la sélection adaptative de modèles dans les applications de cybersécurité. À mesure que les cybermenaces continuent d'évoluer en complexité et en sophistication, de tels systèmes de prédiction flexibles et précis deviendront de plus en plus cruciaux pour maintenir des stratégies de défense efficaces.

Chapitre 7

Corrélation d'Alertes par Apprentissage Automatique pour une Cybersécurité Améliorée : Une Approche Multi-Classification [1]

7.1 Introduction

7.1.1 Contexte et motivation

Les menaces de cybersécurité ont évolué de manière spectaculaire en complexité et en fréquence, les entreprises modernes générant quotidiennement des milliers d'alertes de sécurité provenant de systèmes de détection divers, incluant les Systèmes de Détection d'Intrusion (IDS), les pare-feu, les plateformes de protection des terminaux et l'analytique comportementale [17]. Cette prolifération d'alertes de sécurité crée un défi critique : la fatigue des alertes parmi les analystes de sécurité, conduisant à des temps de réponse retardés et potentiellement à des menaces manquées.

Les systèmes SIEM traditionnels agrègent les événements de sécurité mais manquent de mécanismes de corrélation sophistiqués pour identifier les relations entre alertes connexes [91]. Les analystes doivent investiguer manuellement chaque alerte, déterminant quels événements constituent des menaces genuines versus des faux positifs, et identifiant les alertes appartenant à des campagnes d'attaque coordonnées. Ce processus manuel est chronophage, sujet aux erreurs, et ne passe pas à l'échelle avec le volume croissant de données de sécurité.

1. Ce chapitre est basé sur la publication : Diakhame, M.L., Diallo, C., Mejri, M. (2025). "Machine Learning-Based Alert Correlation for Enhanced Cybersecurity : A Multi-Classification Approach". In *2025 9th Cyber Security in Networking Conference (CSNet)*, pp. 1–8. IEEE.

7.1.2 Problématique

La corrélation d'alertes adresse ces défis en identifiant automatiquement les relations entre événements de sécurité, regroupant les alertes connexes en incidents cohérents, et réduisant le nombre global d'éléments nécessitant l'attention des analystes [92]. Cependant, les approches de corrélation existantes reposent souvent sur des systèmes basés sur des règles qui nécessitent une maintenance constante et peinent à s'adapter aux nouveaux schémas d'attaque.

La problématique centrale de ce chapitre peut être formulée comme suit : *Comment concevoir un cadre de corrélation d'alertes multi-classification capable d'adresser différentes granularités de décision de corrélation tout en maintenant des performances exceptionnelles sur des données réalistes d'entreprise ?*

7.1.3 Questions de recherche

Cette problématique se décline en trois questions de recherche :

Q1 Quelle formulation de classification (binaire, multiclasse, incidents) est la plus efficace pour différents aspects de la corrélation d'alertes ?

Q2 Quels algorithmes d'apprentissage automatique offrent le meilleur compromis entre performance et efficacité computationnelle pour le déploiement en SOC ?

Q3 Comment concevoir une ingénierie des caractéristiques spécifiquement adaptée à la corrélation d'alertes de cybersécurité ?

7.1.4 Contributions

Ce chapitre introduit un cadre complet qui transforme la corrélation d'alertes en trois problèmes de classification complémentaires. Les contributions principales sont :

1. un cadre de multi-classification novateur : Proposition d'une approche à trois niveaux comprenant la détection binaire de corrélation, la classification des types de corrélation, et la classification des incidents.
2. une évaluation exhaustive : Comparaison de huit algorithmes d'apprentissage automatique (Régression Logistique, Random Forest, XGBoost, LightGBM, ANN, CNN, CatBoost, SVM) sur toutes les tâches de corrélation.
3. une méthodologie d'ingénierie des caractéristiques : Développement d'un ensemble de caractéristiques spécifiquement conçu pour la corrélation d'alertes de cybersécurité, intégrant les relations temporelles, la topologie réseau et les indicateurs de progression d'attaque.

4. démonstration de performances exceptionnelles : Obtention de résultats remarquables avec six modèles atteignant une précision parfaite sur la classification multiclasse et CatBoost atteignant 99,27% de précision moyenne globale.

7.1.5 Organisation du chapitre

Le reste de ce chapitre est organisé comme suit. La Section 7.2 examine les travaux connexes sur la corrélation d'alertes et les applications de l'apprentissage automatique en cybersécurité. La Section 7.3 détaille notre méthodologie, incluant les trois approches de classification et les stratégies d'ingénierie des caractéristiques. La Section 7.4 présente les résultats expérimentaux et l'analyse de performance. La Section 7.5 discute les implications et limitations. Enfin, la Section 7.6 conclut ce chapitre avec les directions de recherche futures.

7.2 Travaux connexes

Cette section présente une revue de l'évolution des systèmes de corrélation d'alertes et des applications de l'apprentissage automatique en cybersécurité.

7.2.1 Évolution des systèmes de corrélation d'alertes

Approches basées sur des règles

Les premiers systèmes de corrélation d'alertes employaient principalement des approches basées sur des règles, où les experts en sécurité définissaient manuellement des règles de corrélation basées sur les signatures d'attaque et les schémas d'attaque connus. Templeton et Levitt [93] ont introduit le modèle requires/provides pour les attaques informatiques, établissant des concepts fondamentaux pour la modélisation de la progression des attaques. Bien qu'efficaces pour les scénarios d'attaque bien connus, ces systèmes souffrent d'un surcoût de maintenance élevé et d'une incapacité à détecter les nouveaux schémas d'attaque.

Méthodes de corrélation statistique

Les méthodes de corrélation statistique ont introduit des approches probabilistes pour identifier les relations entre alertes basées sur les similarités temporelles, spatiales et de contenu. Valdes et Skinner [94] ont proposé la corrélation probabiliste d'alertes utilisant des réseaux bayésiens, démontrant des améliorations par rapport aux systèmes simples basés sur des règles. Ning et al. [95] ont développé des méthodes pour construire des scénarios d'attaque à travers la corrélation d'alertes d'intrusion, se concentrant sur les relations de prérequis et de conséquence.

7.2.2 Cadres complets de corrélation d'alertes

Valeur et al. [91] ont présenté l'une des approches complètes les plus influentes pour la corrélation d'alertes de détection d'intrusion, considérant de multiples attributs d'alertes et méthodes de corrélation. Leur travail a établi de nombreux principes fondamentaux encore utilisés dans les systèmes de corrélation modernes.

Morin et al. [96] ont introduit M2D2, un modèle de données formel pour la corrélation d'alertes IDS qui a fourni des approches structurées pour représenter et corréler les événements de sécurité. Debar et Wespi [92] ont développé des techniques d'agrégation et de corrélation pour les alertes de détection d'intrusion, se concentrant sur la réduction des faux positifs et l'amélioration de la précision de détection.

7.2.3 Applications de l'apprentissage automatique en cybersécurité

L'application de l'apprentissage automatique à la cybersécurité a gagné un élan significatif ces dernières années. Buczak et Guven [17] ont fourni une étude exhaustive des méthodes de fouille de données et d'apprentissage automatique pour la détection d'intrusion en cybersécurité.

Gaikwad et Thool [97] ont démontré l'efficacité des méthodes d'ensemble, particulièrement les approches d'ensemble par bagging, pour les systèmes de détection d'intrusion, montrant des améliorations significatives dans la précision de détection par la combinaison de multiples classificateurs.

Apruzzese et al. [98] ont examiné l'efficacité de l'apprentissage automatique et profond pour les applications de cybersécurité, soulignant le potentiel des approches automatisées pour l'analytique de sécurité.

7.2.4 Jeux de données et méthodologies d'évaluation

Le développement de jeux de données standardisés a été crucial pour faire avancer la recherche en apprentissage automatique en cybersécurité. Sharafaldin et al. [71] ont introduit le jeu de données CIC-IDS-2017, qui est devenu un benchmark standard pour la recherche en détection d'intrusion, contenant divers types d'attaques et des schémas de trafic bénin.

Ring et al. [73] ont fourni une étude complète des jeux de données de détection d'intrusion basés sur le réseau, soulignant l'importance de données réalistes pour l'évaluation des systèmes de sécurité.

TABLE 7.1 – Positionnement de notre approche par rapport aux travaux connexes

Approche	**Multi-class**	**ML**	**Éval. compl.**	**CIC-IDS**
Valeur et al. [91]	×	×	∼	×
Ning et al. [95]	×	×	∼	×
Morin et al. [96]	×	×	×	×
Gaikwad [97]	×	✓	∼	×
Notre approche	✓	✓	✓	✓

Multi-class=Formulation multi-classification, ML=Apprentissage automatique, Éval. compl.=Évaluation complète, CIC-IDS=Dataset CIC-IDS-2017, ∼=Partiel

7.2.5 Détection d'attaques coordonnées

Zhou et al. [99] ont fourni une étude sur les attaques coordonnées et la détection d'intrusion collaborative, soulignant l'importance des approches de corrélation pour détecter les attaques sophistiquées multi-étapes.

Le cadre Cyber Kill Chain introduit par Hutchins et al. [33] a fourni une approche structurée pour comprendre la progression des attaques, formant la base de nombreuses approches de corrélation qui modélisent les séquences et dépendances d'attaque.

7.2.6 Positionnement de notre contribution

Le Tableau 7.1 synthétise le positionnement de notre approche par rapport aux travaux existants.

Notre travail adresse les lacunes existantes en fournissant une évaluation systématique de multiples approches de classification et algorithmes d'apprentissage automatique, avec un focus sur les scénarios de déploiement pratique en SOC.

7.3 Méthodologie

Notre méthodologie transforme la corrélation d'alertes en trois problèmes de classification distincts mais complémentaires, chacun adressant différents aspects des processus de prise de décision des analystes de sécurité.

7.3.1 Génération et prétraitement des données d'alertes

Nous avons développé un système complet de génération d'alertes utilisant le jeu de données CIC-IDS-2017 [71], traitant 7 fichiers CSV contenant divers types d'attaques et schémas de trafic bénin.

TABLE 7.2 – Distribution des alertes générées par type et par niveau de sévérité

Type d'attaque	**Nombre d'alertes**	**Proportion**
BENIGN	37 790	75,6%
DoS Hulk	5 001	10,0%
PortScan	3 475	7,0%
DDoS	2 832	5,7%
DoS GoldenEye	226	0,5%
Distribution par niveau de sévérité		
Niveau 1 (Bas)	42 945	85,9%
Niveau 2 (Moyen)	4 052	8,1%
Niveau 3 (Élevé)	145	0,3%
Niveau 4 (Critique)	570	1,1%
Niveau 5 (Urgence)	2 288	4,6%

Chargement et nettoyage des données

À partir de 2 589 427 flux réseau provenant de sept captures quotidiennes de trafic, nous avons effectué des opérations de nettoyage de données supprimant les entrées invalides, résultant en 2 300 825 flux réseau propres. Le jeu de données couvre du lundi au vendredi pendant les heures de travail, capturant des schémas d'activité réseau d'entreprise réalistes.

Génération d'alertes

À partir des flux nettoyés, nous avons généré 50 000 alertes de sécurité réalistes reflétant la sortie SIEM commerciale. Le Tableau 7.2 présente la distribution des alertes par type d'attaque et par niveau de sévérité.

La distribution par type reflète des scénarios du monde réel, avec une majorité de trafic bénin (75,6%) et une présence significative de DoS Hulk (10,0%) et de PortScan (7,0%). La distribution de sévérité suit les schémas opérationnels SOC typiques, avec une prédominance des alertes de bas niveau (85,9%) et une proportion restreinte d'alertes critiques ou urgentes (5,7% combinées).

Structure des alertes

Chaque alerte générée est structurée en cinq catégories d'attributs. Les métadonnées temporelles (horodatage, durée) permettent d'établir les relations chronologiques entre alertes. Les attributs de topologie réseau (IPs source/destination, ports, protocoles) caractérisent les entités impliquées dans chaque événement. Le contexte de sécurité (ID de signature, niveau de sévérité, catégorie d'attaque) fournit l'interprétation sémantique de l'alerte. Le mappage Cyber Kill Chain positionne chaque alerte dans la progression tactique de l'attaque. Enfin, les métriques de volume (paquets, octets transférés) quantifient l'intensité des activités observées.

Mappage Cyber Kill Chain

Nous mappons les types d'attaque CIC-IDS-2017 aux étapes du Cyber Kill Chain [33] afin d'enrichir le contexte tactique de chaque alerte. La phase de **Reconnaissance** correspond aux activités PortScan visant à identifier les hôtes et services actifs. La phase **Weaponization/Delivery** regroupe les attaques SSH/FTP utilisées pour l'accès initial. L'**Exploitation** englobe les attaques Web et les inondations DDoS ciblant les vulnérabilités applicatives. L'**Installation** correspond aux activités Botnet établissant une présence persistante sur les systèmes compromis. La phase **Command & Control** est caractérisée par des connexions persistantes maintenant le dialogue avec l'infrastructure de l'attaquant. Enfin, les **Actions on Objectives** désignent les schémas d'infiltration de données constituant l'objectif final de l'intrusion.

7.3.2 Vue d'ensemble du cadre multi-classification

Notre cadre adresse la corrélation d'alertes à travers trois tâches de classification progressives, s'appuyant sur les principes de corrélation complète établis par Valeur et al. [91] :

1. détection Binaire de Corrélation : Détermine si deux alertes sont corrélées
2. classification des Types de Corrélation : Identifie le type spécifique de relation de corrélation
3. classification des Incidents : Classifie les groupes d'alertes connexes en types d'incidents

La Figure 7.1 présente une comparaison des performances des différents modèles à travers les tâches de classification.

7.3.3 Tâche de classification 1 : détection binaire de corrélation

La première tâche de classification adresse la question fondamentale : « Ces deux alertes sont-elles corrélées ? » Cette classification binaire forme la base pour des décisions de corrélation plus sophistiquées, suivant le modèle de relation prérequis-conséquence proposé par Ning et al. [95].

Extraction de caractéristiques

Nous extrayons 15 caractéristiques pour chaque paire d'alertes à travers trois catégories :

Caractéristiques Temporelles

— différence de temps (minutes)

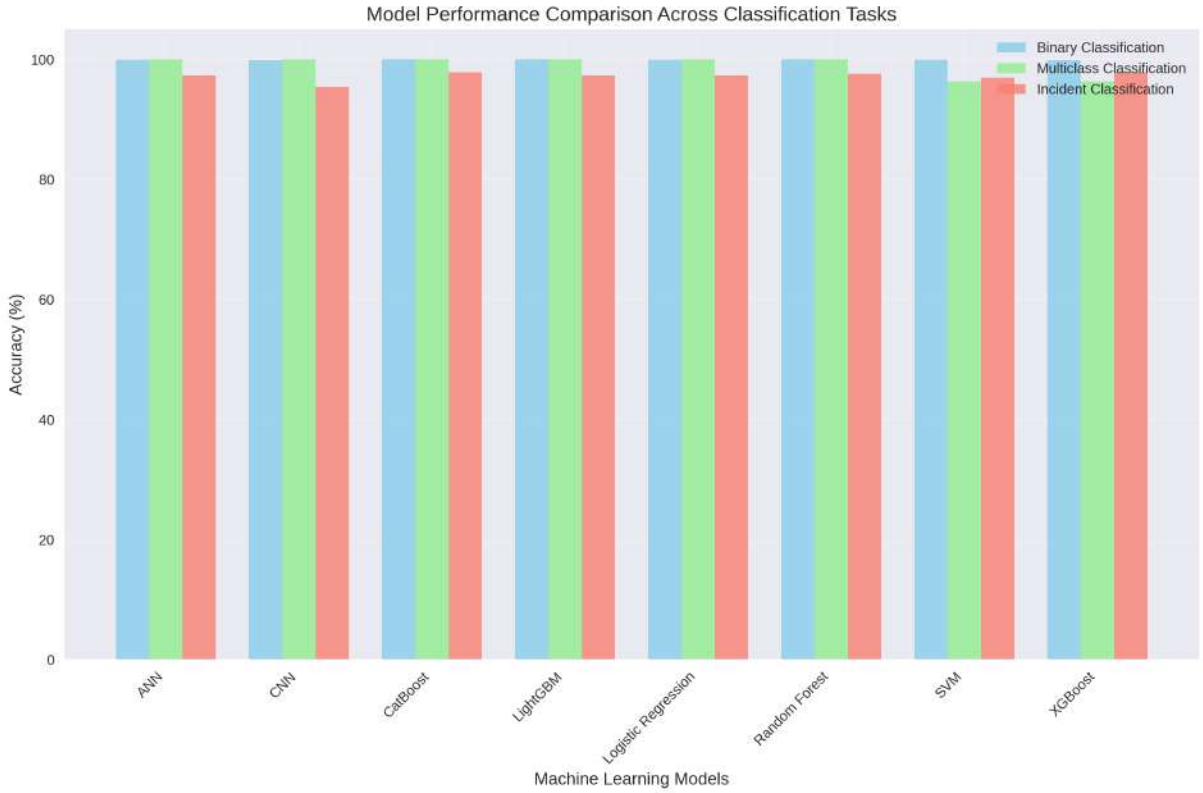


FIGURE 7.1 – Comparaison des performances des modèles à travers les tâches de classification.

— indicateurs binaires pour les fenêtres temporelles (5min, 1h, 24h)

Caractéristiques de Topologie Réseau

— égalité des IPs source/destination

— relations de pivot

— similarité des ports

— compatibilité des protocoles

Caractéristiques de Contexte de Sécurité

— différence et moyenne de sévérité

— similarité de catégorie d'attaque

— indicateurs de progression Kill Chain

Définition de la corrélation

[Corrélation d'alertes] Deux alertes a_1 et a_2 sont considérées corrélées si elles satisfont l'une des conditions suivantes :

1. corrélation temporelle-spatiale : Occurrence dans un intervalle d'1 heure et partage d'entités réseau

2. corrélation causale : Démonstration d'une progression logique Kill Chain dans un intervalle de 2 heures
3. corrélation de campagne : Même catégorie d'attaque avec proximité temporelle

Formellement, la fonction de corrélation peut être définie comme :

$$\text{Corr}(a_1, a_2) = \begin{cases} 1 & \text{si } C_{\text{temp}}(a_1, a_2) \vee C_{\text{caus}}(a_1, a_2) \vee C_{\text{camp}}(a_1, a_2) \\ 0 & \text{sinon} \end{cases}, \tag{7.1}$$

où :

$$C_{\text{temp}}(a_1, a_2) = (|t_1 - t_2| < 1h) \wedge \text{SharedEntities}(a_1, a_2) \tag{7.2}$$

$$C_{\text{caus}}(a_1, a_2) = (|t_1 - t_2| < 2h) \wedge \text{KillChainProg}(a_1, a_2) \tag{7.3}$$

$$C_{\text{camp}}(a_1, a_2) = \text{SameCategory}(a_1, a_2) \wedge \text{TemporalProx}(a_1, a_2) \tag{7.4}$$

7.3.4 Tâche de classification 2 : classification des types de corrélation

La deuxième tâche fournit une vision plus granulaire en classifiant le type spécifique de relation de corrélation entre alertes, inspirée par les approches de modèle de données formel de Morin et al. [96].

Classes de corrélation

Nous identifions quatre classes :

— classe 0 - Pas de Corrélation : Alertes non reliées
— classe 1 - Corrélation Temporelle : Alertes survenant en proximité temporelle proche
— classe 2 - Corrélation Spatiale : Alertes affectant les mêmes entités réseau
— classe 5 - Corrélation de Campagne : Alertes appartenant à des campagnes d'attaque coordonnées

Cette formulation multi-classe permet aux analystes de sécurité de comprendre non seulement si les alertes sont reliées, mais comment elles sont reliées, facilitant des décisions de réponse plus informées.

7.3.5 Tâche de classification 3 : classification des incidents

La troisième tâche opère à un niveau d'abstraction plus élevé, classifiant les groupes d'alertes connexes en types d'incidents, suivant les principes de détection d'attaques coordonnées décrits par Zhou et al. [99].

Regroupement des alertes

Nous employons une approche par fenêtre temporelle glissante pour regrouper les alertes par IP source et proximité temporelle, créant des candidats d'incidents pour la classification.

Caractéristiques agrégées

Pour chaque groupe d'alertes, nous extrayons 12 caractéristiques agrégées à travers trois catégories :

1. indicateurs de volume : Nombre d'alertes, étendue temporelle, octets/paquets totaux
2. mesures de diversité : IPs uniques, ports, protocoles, catégories
3. indicateurs de progression : Diversité des étapes Kill Chain, schémas de sévérité

Types d'incidents

Les types d'incidents incluent :

— cluster de Faux Positifs : Activités bénignes mal identifiées comme menaces

— incident de Reconnaissance : Activités de collecte d'information coordonnées

7.3.6 Algorithmes d'apprentissage automatique

Nous évaluons huit algorithmes d'apprentissage automatique à travers toutes les tâches de classification, suivant les principes des méthodes d'ensemble soulignés par Gaikwad et Thool [97]. Ces algorithmes ont été sélectionnés pour couvrir un spectre représentatif des familles de modèles : modèles linéaires, machines à noyau, méthodes d'ensemble par bagging et boosting, et réseaux de neurones profonds. Cette diversité permet une analyse comparative rigoureuse et de mettre en évidence les forces et limites de chaque approche dans le contexte spécifique de la corrélation d'alertes de sécurité.

Modèles traditionnels

Régression Logistique **Principe.** La régression logistique est un modèle discriminatif qui estime la probabilité d'appartenance d'une observation à une classe en appliquant une fonction softmax à une combinaison linéaire des caractéristiques d'entrée. Malgré son appellation, il s'agit bien d'un algorithme de classification et non de régression : la sortie est une probabilité, non une valeur continue.

Mécanisme. Pour une observation $\mathbf{x} \in \mathbb{R}^d$, la probabilité d'appartenance à la classe k dans la formulation multinomiale est donnée par :

$$P(y = k \mid \mathbf{x}) = \frac{\exp\left(\mathbf{w}_k^\top \mathbf{x} + b_k\right)}{\sum_{j=1}^{K} \exp\left(\mathbf{w}_j^\top \mathbf{x} + b_j\right)} \tag{7.5}$$

où $\mathbf{w}_k \in \mathbb{R}^d$ est le vecteur de poids associé à la classe k et b_k le biais correspondant. Les paramètres sont appris en minimisant la fonction de perte d'entropie croisée :

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} y_{ik} \log P(y = k \mid \mathbf{x}_i) + \lambda \sum_{k} \|\mathbf{w}_k\|^2 \tag{7.6}$$

Le terme de régularisation L2 (λ) pénalise les poids de grande amplitude et prévient le surapprentissage. Le solveur L-BFGS (Limited-memory Broyden-Fletcher-Goldfarb-Shanno) est une méthode quasi-newtonienne qui approxime la hessienne de la perte par un historique limité des gradients passés, permettant une convergence rapide sans stocker explicitement la matrice hessienne complète.

Justification pour la corrélation d'alertes. La régression logistique constitue un modèle de référence (baseline) incontournable dans toute évaluation comparative. Sa nature linéaire lui confère une interprétabilité immédiate des coefficients $\mathbf{w}_k$, ce qui est précieux dans un contexte de sécurité où les analystes SOC doivent comprendre les facteurs guidant les décisions du modèle. L'option `class_weight='balanced'` compense le fort déséquilibre des classes observé dans notre jeu de données (85,9 % d'alertes de bas niveau) en pondérant chaque exemple inversement proportionnellement à la fréquence de sa classe.

Configuration : `multi_class='multinomial'` (multiclasse) ou `'auto'` (binaire), `solver='lbfgs'`, `max_iter=1000`, `random_state=42`, `class_weight='balanced'`.

SVM (Machines à Vecteurs de Support) Principe. Les Machines à Vecteurs de Support cherchent à construire un hyperplan de séparation optimal dans un espace de caractéristiques (éventuellement de dimension supérieure induit par un noyau) tel que la marge entre les classes soit maximisée. L'idée fondatrice est que maximiser la marge de séparation améliore la capacité de généralisation du modèle sur des données non vues.

Mécanisme. Pour un problème binaire avec les données $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, $y_i \in \{-1, +1\}$, le SVM résout le problème d'optimisation quadratique :

$$\min_{\mathbf{w}, b, \boldsymbol{\xi}} \ \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^{N} \xi_i \quad \text{s.c.} \quad y_i\left(\mathbf{w}^\top \phi(\mathbf{x}_i) + b\right) \geq 1 - \xi_i, \ \ \xi_i \geq 0 \tag{7.7}$$

où $\phi(\mathbf{x})$ est la transformation induite par le noyau, C le paramètre de régularisation et ξ_i les variables ressort (slack variables) tolérant une classification incorrecte. Dans notre implémentation, le noyau RBF (Radial Basis Function) est utilisé :

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\Big(-\gamma \left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2\Big) \tag{7.8}$$

où $\gamma = \text{scale} = 1/(n_{\text{features}} \cdot \text{Var}(\mathbf{X}))$ est calculé automatiquement pour adapter la largeur du noyau à l'échelle des données. Le problème est résolu via son dual, où les vecteurs supports — les seules observations $\mathbf{x}_i$ avec $\alpha_i > 0$ — définissent entièrement l'hyperplan de décision. Pour l'extension multiclasse, la stratégie one-vs-rest est appliquée implicitement. L'option `probability=True` active une calibration de Platt (régression logistique sur les scores bruts) pour produire des estimations de probabilité, nécessaires au calcul des métriques AUC.

Justification pour la corrélation d'alertes. Le SVM avec noyau RBF est particulièrement adapté aux données de corrélation d'alertes présentant des frontières de décision non linéaires complexes entre types de corrélation (temporelle, spatiale, campagne). Sa robustesse théorique aux données de grande dimension et sa résistance au surapprentissage — la solution ne dépend que des vecteurs supports — en font un choix pertinent pour nos 15 caractéristiques hétérogènes mélangeant indicateurs temporels, topologiques et sémantiques.

Configuration : `kernel='rbf'`, `probability=True`, `C=1.0`, `gamma='scale'`, `random_state=42`.

Méthodes d'ensemble

Random Forest **Principe.** Random Forest est un algorithme d'ensemble basé sur le bagging (Bootstrap AGGregating), qui construit un ensemble d'arbres de décision entraînés sur des sous-échantillons bootstrap distincts du jeu de données, avec une sélection aléatoire des caractéristiques candidates à chaque nœud de division. La prédiction finale est obtenue par vote majoritaire des prédictions individuelles, réduisant la variance sans augmenter le biais.

Mécanisme. Chaque arbre h_t ($t = 1, \ldots, T$) est construit sur un échantillon bootstrap $\mathcal{D}_t$ tiré avec remise de $\mathcal{D}$. À chaque nœud de division, seul un sous-ensemble aléatoire de $m = \lfloor \sqrt{p} \rfloor$ caractéristiques (parmi p au total) est évalué pour déterminer la meilleure division selon le critère de Gini :

$$\text{Gini}(t) = 1 - \sum_{k=1}^{K} p_{k,t}^2 \tag{7.9}$$

où $p_{k,t}$ est la proportion d'exemples de classe k au nœud t. Le gain de division d'un nœud parent t en enfants gauche t_L et droit t_R est :

$$\Delta\text{Gini}(t) = \text{Gini}(t) - \frac{n_{t_L}}{n_t}\,\text{Gini}(t_L) - \frac{n_{t_R}}{n_t}\,\text{Gini}(t_R) \tag{7.10}$$

La prédiction finale pour une observation $\mathbf{x}$ est :

$$\hat{y} = \arg\max_{k} \sum_{t=1}^{T} \mathbf{1}\Big[h_t(\mathbf{x}) = k\Big] \tag{7.11}$$

Le double mécanisme d'aléatoire — bootstrap des données et sous-sélection des caractéristiques — décolle les arbres individuels, réduisant leur corrélation et donc la variance de l'ensemble. Une mesure d'importance des caractéristiques est également calculée automatiquement comme la diminution moyenne d'impureté (MDI) pondérée par la taille des nœuds sur l'ensemble des arbres.

Justification pour la corrélation d'alertes. Random Forest est particulièrement adapté à notre problématique pour trois raisons. Premièrement, sa résistance naturelle au surapprentissage est cruciale face à la distribution déséquilibrée de notre jeu de données. Deuxièmement, la mesure MDI permet d'identifier les attributs les plus discriminants pour la corrélation (ex. progression Kill Chain, similarité d'IP), fournissant une explicabilité précieuse pour les analystes SOC. Troisièmement, son comportement stable et reproductible est essentiel pour un déploiement opérationnel en environnement de production.

Configuration : `n_estimators=100`, `max_depth=10`, `criterion='gini'`, `min_samples_split=2`, `random_state=42`.

XGBoost Principe. XGBoost (eXtreme Gradient Boosting) est un algorithme de boosting par gradient qui construit séquentiellement un ensemble d'arbres de décision, chaque nouvel arbre corrigeant les erreurs résiduelles du modèle précédent. Il se distingue des autres implémentations de gradient boosting par sa régularisation L1/L2 intégrée, le traitement natif des données manquantes, et une parallélisation efficace du calcul des splits.

Mécanisme. À l'itération t, XGBoost minimise la fonction objectif régularisée :

$$\mathcal{L}^{(t)} = \sum_{i=1}^{N} \ell\Big(y_i,\ \hat{y}_i^{(t-1)} + f_t(\mathbf{x}_i)\Big) + \Omega(f_t) \tag{7.12}$$

où $\Omega(f_t) = \gamma T + \frac{1}{2}\lambda\|\mathbf{w}\|^2$ pénalise la complexité de l'arbre (T = nombre de feuilles, $\mathbf{w}$ = scores des feuilles, γ et λ paramètres de régularisation). En développant l'objectif par approximation de Taylor au second ordre autour de $\hat{y}_i^{(t-1)}$:

$$\mathcal{L}^{(t)} \approx \sum_{i=1}^{N} \Big[g_i\, f_t(\mathbf{x}_i) + \frac{1}{2}\, h_i\, f_t^2(\mathbf{x}_i)\Big] + \Omega(f_t) \tag{7.13}$$

où $g_i = \partial_{\hat{y}} \ell(y_i, \hat{y})\big|_{\hat{y}^{(t-1)}}$ et $h_i = \partial^2_{\hat{y}} \ell(y_i, \hat{y})\big|_{\hat{y}^{(t-1)}}$ sont respectivement le gradient et le hessien de la perte. La structure optimale du t-ième arbre est déterminée par le gain de division d'un nœud en régions gauche $\mathcal{I}_L$ et droite $\mathcal{I}_R$:

$$\text{Gain} = \frac{1}{2}\left[\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda}\right] - \gamma \tag{7.14}$$

où $G_L = \sum_{i \in \mathcal{I}_L} g_i$, $H_L = \sum_{i \in \mathcal{I}_L} h_i$ (et de même pour R). La division n'est effectuée que si ce gain est positif, ce qui constitue un mécanisme naturel d'élagage.

Justification pour la corrélation d'alertes. XGBoost est reconnu pour ses performances de pointe sur des données tabulaires structurées, exactement comme nos vecteurs de caractéristiques de paires d'alertes. Sa régularisation intégrée prévient le surapprentissage sur notre jeu de données déséquilibré, tandis que sa vitesse d'exécution exceptionnelle (0,14s d'entraînement observé dans nos expériences) le rend viable pour un déploiement en SOC où les modèles doivent être réentraînés fréquemment sur de nouvelles données d'alertes.

Configuration : `objective='multi:softmax'` (multiclasse) ou `'binary:logistic'` (binaire), `n_estimators=100`, `learning_rate=0.1`, `max_depth=6`.

LightGBM Principe. LightGBM (Light Gradient Boosting Machine) est une implémentation de gradient boosting développée par Microsoft, conçue pour surpasser XGBoost en vitesse et en efficacité mémoire. Ses deux innovations algorithmiques principales sont la croissance d'arbres par feuilles (leaf-wise) plutôt que par niveaux (level-wise), et le Gradient-based One-Side Sampling (GOSS) qui réduit le volume de données utilisé pour estimer les gains de division.

Mécanisme. Contrairement aux algorithmes classiques qui développent tous les nœuds d'un même niveau avant d'approfondir, LightGBM sélectionne à chaque étape la feuille dont la division produit le gain maximal :

$$\text{Feuille}^* = \arg\max_{\ell \in \text{feuilles}} \Delta\mathcal{L}(\ell) \tag{7.15}$$

Cette stratégie atteint une perte moindre avec le même nombre de feuilles, au prix d'une profondeur plus irrégulière. L'algorithme GOSS accélère le calcul des gains de division en conservant toutes les instances à gradient élevé (seuil a) et en sous-échantillonnant les instances à gradient faible avec un facteur b. Pour éviter un biais dans l'estimation, les instances à faible gradient retenues sont amplifiées :

$$\tilde{g}_i = \begin{cases} g_i & \text{si } |g_i| \geq \tau \\ \dfrac{1-a}{b}\, g_i & \text{si } |g_i| < \tau \text{ et instance sélectionnée} \end{cases} \tag{7.16}$$

L'Exclusive Feature Bundling (EFB) réduit en outre le nombre de caractéristiques en regroupant celles qui sont mutuellement exclusives (jamais non nulles simultanément), fusionnant leurs plages de valeurs en une seule caractéristique composite.

Justification pour la corrélation d'alertes. Avec un temps d'entraînement de 0,19s observé dans nos expériences (contre 2,57s pour CatBoost et 16,37s pour ANN), LightGBM offre le meilleur compromis performance/vitesse de notre benchmark. Cette efficacité est déterminante pour un déploiement SOC où les modèles doivent être réentraî-

nés fréquemment sur de nouvelles alertes. Sa gestion efficace des données déséquilibrées via GOSS — qui focalise l'apprentissage sur les instances les plus informatives — est un avantage particulier face à la prépondérance des alertes bénignes dans notre corpus.

Configuration : `objective='multiclass'` (multiclasse) ou `'binary'` (binaire), `n_estimators=100`, `learning_rate=0.1`, `max_depth=10`.

CatBoost Principe. CatBoost (Categorical Boosting) est un algorithme de gradient boosting développé par Yandex, spécialement conçu pour traiter nativement les variables catégorielles sans encodage préalable. Sa contribution principale est l'Ordered Boosting, une technique qui élimine la fuite d'information (target leakage) inhérente aux implémentations classiques de gradient boosting lors de l'estimation des gradients.

Mécanisme. Le problème de la fuite de cible survient dans le gradient boosting classique car les résidus (gradients) sur l'ensemble d'entraînement sont calculés par un modèle qui a déjà appris sur ces mêmes exemples, introduisant un biais systématique. CatBoost résout ce problème par permutation ordonnée : pour un ensemble de données de taille N, une permutation aléatoire σ est générée, et le gradient pour l'observation i est estimé sur un modèle entraîné uniquement sur les observations j telles que $\sigma(j) < \sigma(i)$:

$$g_i^{(t)} = \frac{\partial\, \ell\Big(y_i,\, F_\sigma^{(t-1,\sigma(i))}(\mathbf{x}_i)\Big)}{\partial\, F_\sigma^{(t-1,\sigma(i))}(\mathbf{x}_i)} \tag{7.17}$$

où $F_\sigma^{(t-1,\sigma(i))}$ désigne le modèle construit sur toutes les observations précédant i dans la permutation σ. Pour les variables catégorielles de modalité c, CatBoost calcule une statistique de cible ordonnée, sans fuite :

$$\hat{x}_{ic} = \frac{\displaystyle\sum_{\substack{j:\,\sigma(j)<\sigma(i)\\ x_{jc}=c}} y_j + \alpha\, p}{\displaystyle\sum_{\substack{j:\,\sigma(j)<\sigma(i)\\ x_{jc}=c}} 1 + \alpha} \tag{7.18}$$

où p est la probabilité a priori de la classe et α un hyperparamètre de lissage. CatBoost utilise des arbres symétriques (oblivious trees) où la même condition de division est appliquée à tous les nœuds d'un même niveau, ce qui garantit un temps d'inférence constant et prévisible indépendamment de la profondeur.

Justification pour la corrélation d'alertes. CatBoost a obtenu la meilleure précision globale de notre étude (99,27 % de précision moyenne sur les trois tâches). Son approche anti-fuite est particulièrement pertinente pour les données de sécurité qui présentent des dépendances temporelles fortes entre alertes : les observations les plus récentes ne doivent pas contaminer l'apprentissage des observations passées. La gestion native des caractéristiques catégorielles (protocoles réseau, catégories d'attaque Kill Chain) élimine le biais d'encodage pouvant affecter les autres algorithmes, et ses arbres symétriques ac-

célèrent significativement l'inférence, critique pour la corrélation en temps réel.

Configuration : `iterations=500`, `learning_rate=0.1`, `depth=6`, `random_state=42`.

Modèles d'apprentissage profond

Réseau de Neurones Artificiel (ANN) Principe. Un Réseau de Neurones Artificiel (ANN) est un modèle de calcul inspiré des réseaux de neurones biologiques, composé de couches de neurones artificiels interconnectés par des poids apprenables. Dans notre implémentation, nous utilisons un perceptron multicouche (MLP) entièrement connecté (fully connected), capable de modéliser des relations non linéaires arbitrairement complexes entre les caractéristiques d'entrée et les classes de sortie grâce au théorème universel d'approximation.

Mécanisme. Pour une couche dense l, la transformation est définie par :

$$\mathbf{a}^{(l)} = f\Big(\mathbf{W}^{(l)}\mathbf{a}^{(l-1)} + \mathbf{b}^{(l)}\Big) \tag{7.19}$$

où $\mathbf{W}^{(l)} \in \mathbb{R}^{n_l \times n_{l-1}}$ est la matrice de poids, $\mathbf{b}^{(l)}$ le vecteur de biais et f la fonction d'activation. Dans les couches cachées, nous utilisons la fonction ReLU (Rectified Linear Unit) :

$$\mathrm{ReLU}(z) = \max(0, z) \tag{7.20}$$

qui présente deux avantages pratiques : elle ne sature pas pour les valeurs positives (pas de problème de gradient évanescent) et produit une sparsité naturelle des activations. L'architecture complète est :

$$\mathbf{x} \xrightarrow{\text{Dense(64)}} \xrightarrow{\text{ReLU}} \xrightarrow{\text{Dropout(0.3)}} \xrightarrow{\text{Dense(64)}} \xrightarrow{\text{ReLU}} \xrightarrow{\text{Dropout(0.3)}} \xrightarrow{\text{Dense(32)}} \xrightarrow{\text{ReLU}} \xrightarrow{\text{Softmax}} \hat{\mathbf{y}} \tag{7.21}$$

Les paramètres sont optimisés par l'algorithme Adam, une variante de la descente de gradient stochastique à taux d'apprentissage adaptatif par coordonnée :

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \frac{\eta}{\sqrt{\hat{\mathbf{v}}_t} + \epsilon}\,\hat{\mathbf{m}}_t, \quad \hat{\mathbf{m}}_t = \frac{\mathbf{m}_t}{1-\beta_1^t}, \quad \hat{\mathbf{v}}_t = \frac{\mathbf{v}_t}{1-\beta_2^t} \tag{7.22}$$

où $\mathbf{m}_t$ et $\mathbf{v}_t$ sont les estimations du premier et du second moment des gradients, et $\beta_1 = 0{,}9$, $\beta_2 = 0{,}999$ leurs taux de décroissance respectifs. Le mécanisme de Dropout (probabilité 0,3) désactive aléatoirement des neurones pendant l'entraînement, forçant le réseau à développer des représentations redondantes et robustes, ce qui réduit significativement le surapprentissage sur des données déséquilibrées.

Justification pour la corrélation d'alertes. L'ANN est inclus pour sa capacité à capturer des interactions non linéaires d'ordre élevé entre les 15 caractéristiques tem-

porelles, topologiques et sémantiques, interactions que les modèles linéaires ne peuvent pas représenter. Le mécanisme d'arrêt précoce (early stopping, patience=10) est particulièrement important ici : il interrompt l'entraînement dès que la perte de validation cesse de diminuer, évitant de mémoriser la distribution déséquilibrée du jeu de données d'entraînement.

Configuration : Optimiseur Adam (`learning_rate=0.001`), `epochs=50`, *early stopping* (`patience=10`), `batch_size=32`.

Réseau de Neurones Convolutif (CNN) **Principe.** Les Réseaux de Neurones Convolutifs (CNN) ont été initialement conçus pour le traitement de données spatiales (images), mais ils se révèlent efficaces pour les séquences de caractéristiques numériques via des convolutions 1D. Dans notre contexte, chaque vecteur de caractéristiques d'alerte est interprété comme une séquence unidimensionnelle, permettant au CNN de détecter automatiquement des motifs locaux (patterns) entre caractéristiques adjacentes sans ingénierie manuelle de ces interactions.

Mécanisme. Une couche de convolution 1D applique F filtres de taille k à l'entrée $\mathbf{x} \in \mathbb{R}^{L \times C}$ (L longueur, C canaux) :

$$(\mathbf{x} * \mathbf{w}_f)_j = \sum_{c=1}^{C} \sum_{i=0}^{k-1} x_{j+i,\,c} \cdot w_{f,i,c} + b_f \tag{7.23}$$

suivie d'une activation ReLU. Le max-pooling de taille 2 réduit la dimension temporelle de moitié en conservant la valeur maximale dans chaque fenêtre glissante, introduisant une invariance aux décalages locaux et réduisant le coût computationnel :

$$\text{MaxPool}(\mathbf{h})_j = \max(h_{2j},\, h_{2j+1}) \tag{7.24}$$

L'architecture complète est :

$$\mathbf{x} \xrightarrow{\text{Conv1D}(64,\,k=3)} \xrightarrow{\text{ReLU}} \xrightarrow{\text{MaxPool}(2)} \xrightarrow{\text{Conv1D}(32,\,k=3)} \xrightarrow{\text{ReLU}} \xrightarrow{\text{MaxPool}(2)} \xrightarrow{\text{Flatten}} \xrightarrow{\text{Dense}(64)} \xrightarrow{\text{ReLU}} \xrightarrow{\text{Softmax}} \hat{\mathbf{y}} \tag{7.25}$$

La première couche Conv1D(64, k=3) apprend 64 détecteurs de motifs locaux de largeur 3, capturant les coactivations entre caractéristiques consécutives (ex. relation entre la différence temporelle et la progression Kill Chain). La deuxième couche Conv1D(32, k=3) compose ces motifs de base en représentations de plus haut niveau, instaurant une hiérarchie de représentations latentes spécifiques au domaine.

Justification pour la corrélation d'alertes. Le CNN est inclus pour sa capacité à extraire automatiquement des structures locales dans les vecteurs de caractéristiques, complémentaires aux représentations manuelles des méthodes d'ensemble. L'ordre dans lequel les caractéristiques sont présentées (temporelles → topologiques → contexte sécu-

rité) crée une structure locale exploitable par les filtres convolutifs. Bien que son temps d'entraînement soit le plus élevé de notre benchmark (46,22s), le CNN représente un choix pertinent pour des scénarios où de nouveaux types de motifs d'attaque émergent, car ses représentations latentes s'adaptent automatiquement aux nouvelles distributions sans réingénierie des caractéristiques.

Configuration : Optimiseur Adam (`learning_rate=0.001`), `epochs=50`, *early stopping* (`patience=10`), `batch_size=32`.

7.3.7 Méthodologie d'évaluation

Notre pipeline de prétraitement complet assure des résultats robustes et reproductibles, suivant les meilleures pratiques d'évaluation décrites par Ring et al. [73].

Prétraitement des données

— mise à l'échelle : StandardScaler() pour des plages de valeurs cohérentes

— encodage des labels : LabelEncoder() pour la compatibilité entre algorithmes

— équilibrage des classes : SMOTE (random_state=42, k_neighbors=min(5, n_samples//10))

— division train/test : Ratio 80/20 avec stratification

Métriques de performance

Nous évaluons les modèles en utilisant les métriques suivantes :

$$\text{Exactitude} = \frac{VP + VN}{VP + VN + FP + FN} \tag{7.26}$$

$$\text{Précision} = \frac{VP}{VP + FP} \tag{7.27}$$

$$\text{Rappel} = \frac{VP}{VP + FN} \tag{7.28}$$

$$\text{Score F1} = 2 \times \frac{\text{Précision} \times \text{Rappel}}{\text{Précision} + \text{Rappel}}, \tag{7.29}$$

où VP = Vrais Positifs, VN = Vrais Négatifs, FP = Faux Positifs, FN = Faux Négatifs.

7.4 Résultats expérimentaux

Cette section présente les résultats complets obtenus par l'application de notre cadre multi-classification à travers les trois tâches de corrélation.

TABLE 7.3 – Performance de la détection binaire de corrélation avec validation croisée

Modèle	Test Acc.	CV (±Std)	F1	Temps (s)
Random Forest	**99,96%**	99,93 (±0,08)	1,000	0,88
LightGBM	**99,96%**	99,91 (±0,07)	1,000	0,19
CatBoost	**99,96%**	99,94 (±0,08)	1,000	2,57
Rég. Logistique	99,92%	99,90 (±0,08)	0,999	0,04
SVM	99,92%	99,90 (±0,08)	0,999	0,59
ANN	99,92%	99,93 (±0,06)	0,999	16,37
XGBoost	99,88%	99,92 (±0,06)	0,999	0,14
CNN	99,84%	99,88 (±0,07)	0,998	46,22

7.4.1 Caractéristiques du jeu de données

Notre cadre expérimental traite le jeu de données CIC-IDS-2017 complet [71], générant des alertes de sécurité réalistes qui reflètent les environnements SOC opérationnels.

Pour nos trois tâches de classification, cela a résulté en :

— classification binaire : 245 paires d'alertes (50 Non-Corrélées, 195 Corrélées)

— classification multiclasse : 73 échantillons à travers quatre types de corrélation (No_Correlation : 50, Temporal : 4, Spatial : 1, Campaign : 18)

— classification des incidents : 139 incidents (False_Positive : 89, Reconnaissance : 50) avec augmentation SMOTE à 142 échantillons d'entraînement

7.4.2 Résultats de la détection binaire de corrélation

Le Tableau 7.3 présente les performances remarquables atteintes par tous les modèles sur la tâche de détection binaire de corrélation.

Trois modèles (Random Forest, LightGBM et CatBoost) ont atteint la précision la plus élevée de 99,96%, avec Random Forest démontrant une stabilité de validation croisée supérieure (99,93 ±0,08) et un temps d'entraînement efficace (0,88s).

7.4.3 Résultats de la classification des types de corrélation

Le Tableau 7.4 montre des performances exceptionnelles sur la tâche de classification des types de corrélation à quatre classes.

Six modèles sur huit (ANN, Random Forest, CatBoost, Régression Logistique, LightGBM et CNN) ont atteint une exactitude parfaite de 100%, avec la Régression Logistique démontrant le temps d'entraînement le plus rapide (0,01s).

7.4.4 Résultats de la classification des incidents

Le Tableau 7.5 présente les résultats de la classification des incidents.

TABLE 7.4 – Performance de la classification des types de corrélation avec validation croisée

Modèle	Test Acc.	CV (±Std)	F1	Temps (s)
ANN	**100,00%**	99,05 (±1,90)	1,000	9,05
Random Forest	**100,00%**	98,10 (±2,33)	1,000	0,21
CatBoost	**100,00%**	98,10 (±2,33)	1,000	1,68
Rég. Logistique	**100,00%**	97,14 (±3,81)	1,000	0,01
LightGBM	**100,00%**	94,19 (±3,67)	1,000	0,03
CNN	**100,00%**	95,24 (±3,01)	1,000	8,34
SVM	96,30%	97,14 (±2,33)	0,963	0,01
XGBoost	96,30%	96,80 (±1,20)	0,962	0,05

TABLE 7.5 – Performance de la classification des incidents avec validation croisée

Modèle	Test Acc.	CV (±Std)	F1	Temps (s)
XGBoost	**97,85%**	97,84 (±0,67)	0,975	0,18
CatBoost	**97,85%**	98,15 (±0,57)	0,973	1,45
Random Forest	97,54%	98,15 (±0,57)	0,970	0,25
Rég. Logistique	97,23%	96,91 (±0,94)	0,963	0,02
LightGBM	97,23%	98,07 (±0,49)	0,968	0,22
ANN	97,23%	97,76 (±1,13)	0,963	15,19
SVM	96,92%	96,84 (±0,82)	0,960	0,08
CNN	95,38%	96,07 (±0,86)	0,945	12,58

TABLE 7.6 – Comparaison complète des performances à travers toutes les tâches

Modèle	Binaire	Multi	Incident	Moy. Acc.	Moy. F1
CatBoost	99,96%	**100,00%**	**97,85%**	**99,27%**	0,991
Random Forest	**99,96%**	**100,00%**	97,54%	99,17%	0,990
LightGBM	99,96%	**100,00%**	97,23%	99,06%	0,989
ANN	99,92%	**100,00%**	97,23%	99,05%	0,988
Rég. Logistique	99,92%	**100,00%**	97,23%	99,05%	0,988
XGBoost	99,88%	96,30%	**97,85%**	98,01%	0,979
CNN	99,84%	**100,00%**	95,38%	98,41%	0,981
SVM	99,92%	96,30%	96,92%	97,71%	0,974

XGBoost et CatBoost ont atteint la meilleure exactitude (97,85%), avec XGBoost montrant une performance de validation croisée supérieure (97,84 ±0,67).

7.4.5 Comparaison globale des performances

Le Tableau 7.6 fournit une comparaison complète à travers toutes les tâches de classification.

CatBoost a atteint la meilleure performance globale avec 99,27% d'exactitude moyenne à travers toutes les tâches, suivi de près par Random Forest (99,17%) et LightGBM (99,06%).

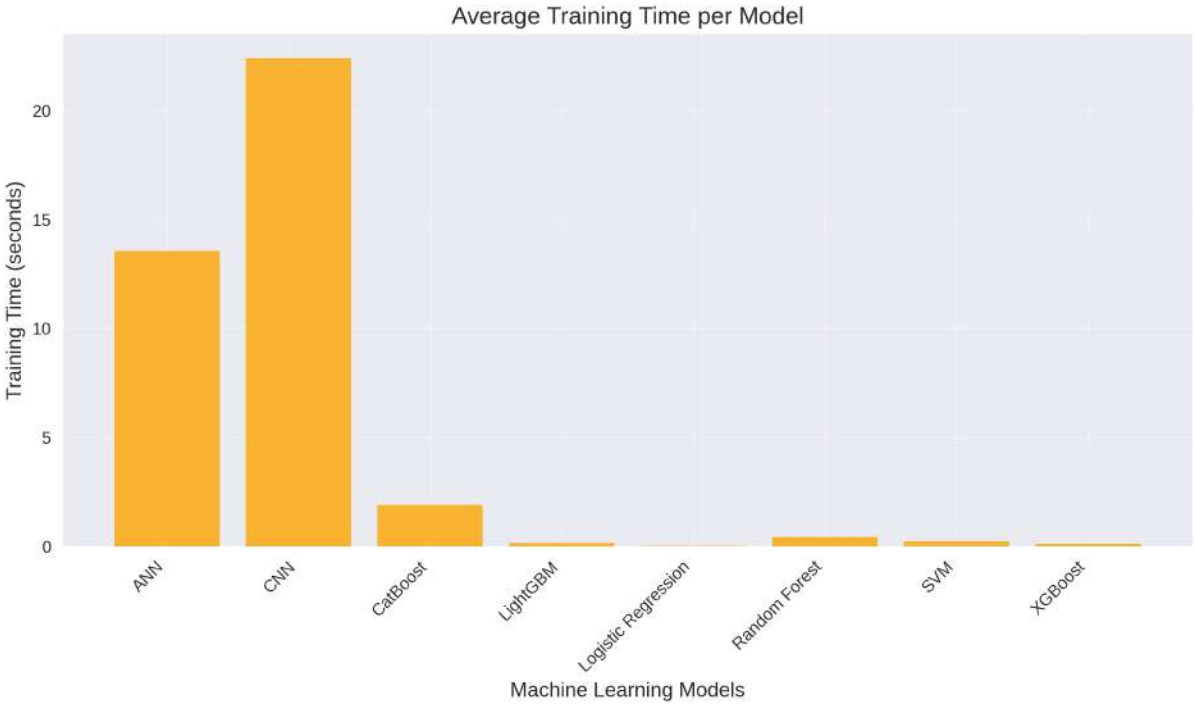


FIGURE 7.2 – Comparaison des temps d'entraînement à travers tous les modèles.

7.4.6 Meilleurs modèles par tâche

L'évaluation complète permet d'identifier les modèles les plus adaptés à chaque tâche. Pour la classification binaire, Random Forest, LightGBM et CatBoost se distinguent avec 99,96% d'exactitude, Random Forest offrant en outre un temps d'entraînement optimal (0,88 s) et une excellente stabilité en validation croisée. Pour la classification multiclasse, six modèles (Régression Logistique, Random Forest, LightGBM, CatBoost, ANN et CNN) atteignent une exactitude parfaite de 100%, la Régression Logistique se démarquant par sa rapidité d'entraînement (0,01 s). Pour la classification des incidents, XGBoost et CatBoost obtiennent les meilleures performances (97,85%), XGBoost présentant une stabilité de validation croisée légèrement supérieure (97,84 ±0,67).

7.4.7 Visualisation des résultats

La Figure 7.2 présente la comparaison des temps d'entraînement, soulignant les avantages d'efficacité des différents algorithmes.

La Figure 7.3 montre la comparaison des scores F1 à travers tous les modèles et tâches.

La Figure 7.4 présente la matrice de confusion pour la Régression Logistique sur la classification binaire.

La Figure 7.5 fournit une visualisation par carte de chaleur de l'exactitude à travers tous les modèles et tâches de classification.

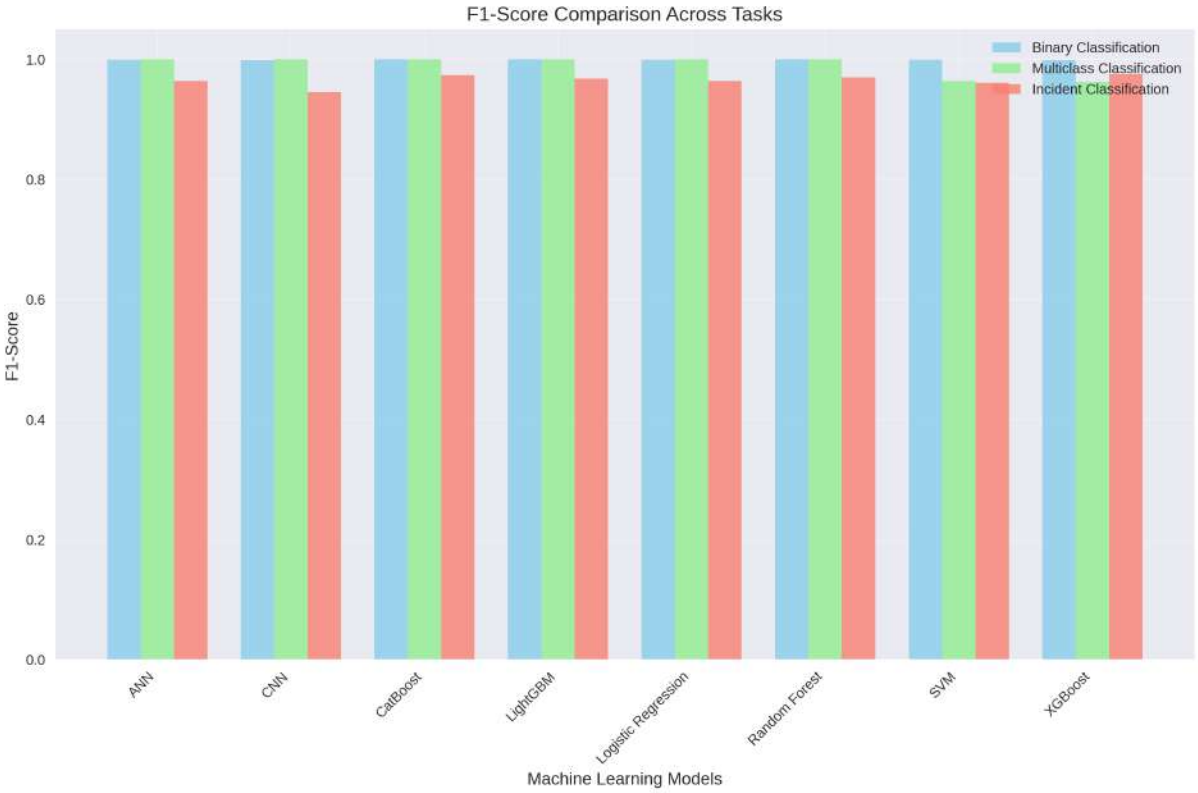


FIGURE 7.3 – Comparaison des scores F1 à travers les modèles et les tâches.



FIGURE 7.4 – Matrice de confusion - Régression Logistique (Classification Binaire).

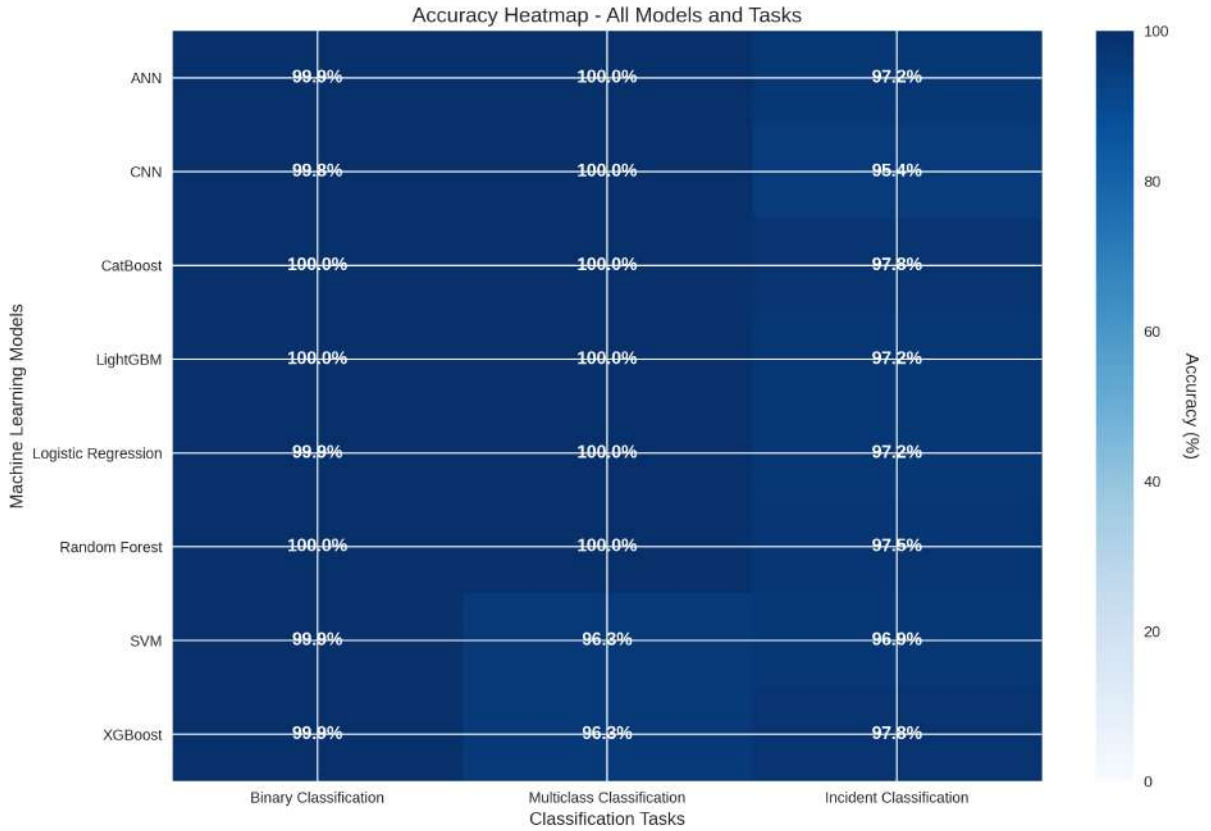


FIGURE 7.5 – Carte de chaleur de l'exactitude à travers les modèles et les tâches de classification.

7.5 Discussion

7.5.1 Analyse des performances des modèles

Les performances exceptionnelles atteintes à travers toutes les tâches de classification démontrent l'efficacité de notre approche d'ingénierie des caractéristiques, s'appuyant sur les principes de corrélation complète établis par Valeur et al. [91]. CatBoost atteignant la meilleure exactitude moyenne globale (99,27%) suivi de près par Random Forest (99,17%) indique que les caractéristiques temporelles, spatiales et causales fournissent un pouvoir discriminant suffisant pour les décisions de corrélation.

Les méthodes d'ensemble basées sur les arbres ont consistamment excellé en raison de leur gestion naturelle des caractéristiques catégorielles de cybersécurité et des interactions complexes entre caractéristiques, en accord avec les conclusions de Gaikwad et Thool [97] concernant l'efficacité des approches d'ensemble dans les applications de cybersécurité.

Il est également important de noter que, bien que les exactitudes moyennes soient restées au-dessus de 97% à travers tous les modèles, les scores F1 ont montré plus de variabilité, avec CNN descendant à 0,945 sur la tâche de classification des incidents, soulignant l'importance de considérer à la fois l'exactitude et le F1 lors de l'évaluation de la robustesse des modèles.

7.5.2 Efficacité des tâches de classification

Les trois formulations de classification fournissent des insights opérationnels complémentaires qui adressent différents aspects du défi de corrélation d'alertes identifié dans la littérature [92]. La classification binaire offre un filtrage de corrélation fondamental avec une exactitude exceptionnelle, adaptée au traitement en temps réel à haut volume. La classification multiclasse fournit quant à elle une compréhension sémantique des relations de corrélation, permettant aux analystes de prioriser leurs investigations en fonction du type de relation identifiée entre les alertes. Enfin, la classification des incidents opère à un niveau stratégique en regroupant les alertes connexes pour une analyse complète des menaces, suivant les principes de détection d'attaques coordonnées décrits par Zhou et al. [99].

7.5.3 Considérations d'implémentation pratique

Les résultats suggèrent des stratégies de déploiement optimales basées sur les méthodologies d'évaluation recommandées par Ring et al. [73]. La Régression Logistique, qui atteint des performances exceptionnelles avec un temps d'entraînement de seulement 0,01 seconde, constitue le choix privilégié pour les environnements en temps réel où le surcoût computationnel doit être minimal. Les modèles plus complexes comme les réseaux de neurones (ANN, CNN), bien que plus gourmands en ressources (respectivement 15 et 12 secondes d'entraînement), conviennent davantage aux scénarios de traitement par lots différé. En définitive, les organisations peuvent implémenter des architectures à plusieurs niveaux combinant différents modèles selon les exigences opérationnelles spécifiques et les contraintes computationnelles de leurs infrastructures SOC.

7.5.4 Limitations

Bien que les résultats soient très encourageants, plusieurs limitations méritent d'être soulignées. En premier lieu, notre cadre s'appuie sur des alertes simulées à partir du jeu de données CIC-IDS-2017, ce qui nécessite une validation avec des données SIEM opérationnelles pour confirmer la généralisabilité en environnement de production. En second lieu, les ensembles de caractéristiques extraits sont statiques et peuvent nécessiter un réentraînement périodique pour rester performants face à des attaques évolutives et à de nouveaux vecteurs de menace. Par ailleurs, l'approche de comparaison par paires, bien qu'efficace sur les volumes expérimentaux, peut nécessiter l'adoption de méthodes hiérarchiques ou de clustering pour passer à l'échelle des environnements d'entreprise générant plusieurs millions d'alertes quotidiennes. Enfin, les modèles d'apprentissage profond évalués (ANN, CNN) manquent d'interprétabilité native, ce qui peut freiner la confiance des analystes lors du déploiement opérationnel et nécessite l'intégration de techniques

d'explication post-hoc.

7.5.5 Discussion sur la validité des performances élevées

Les performances parfaites (100%) observées sur la tâche de classification multiclasse des types de corrélation pour six modèles sur huit méritent une discussion explicite, dans la mesure où de tels résultats peuvent légitimement susciter des questions quant au risque de surapprentissage (*overfitting*).

Plusieurs éléments permettent de contextualiser et de valider ces performances. Premièrement, la tâche de classification des types de corrélation (temporelle, spatiale, de campagne) repose sur des caractéristiques fortement discriminantes par construction : les relations temporelles, topologiques et causales extraites des alertes constituent des signaux structurels clairs, comme établi par Valeur et al. [91]. La forte séparabilité intrinsèque des classes explique que des modèles de natures très différentes — de la régression logistique (modèle linéaire) aux forêts aléatoires (ensemble) en passant par les réseaux de neurones (CNN, ANN) — atteignent tous des performances comparables. La convergence de huit algorithmes distincts vers des résultats similaires constitue en elle-même un indicateur de robustesse plutôt que d'artefact algorithmique [97].

Deuxièmement, les résultats sont obtenus avec validation croisée, ce qui atténue le risque de surapprentissage par rapport à une évaluation sur un seul split train/test. Troisièmement, le cadre expérimental s'appuie sur 50 000 alertes générées à partir de 2,30 millions de flux réseau du jeu de données CIC-IDS-2017, couvrant une diversité de types d'attaques (DoS Hulk, PortScan, DDoS, DoS GoldenEye) et de trafic bénin, ce qui confère une représentativité suffisante pour les tâches de corrélation évaluées.

Néanmoins, la taille de l'échantillon sur certaines tâches — notamment la classification des incidents — reste plus modeste, et la dépendance à des alertes simulées plutôt qu'issues d'un SIEM opérationnel réel constitue une limite reconnue. La généralisation à des environnements de production avec des distributions d'attaques différentes, des volumes plus élevés et des patterns de corrélation plus nuancés reste à valider. Cette validation sur données opérationnelles réelles est identifiée comme une priorité dans les perspectives de ce chapitre.

7.5.6 Perspectives d'interprétabilité

L'incorporation de techniques d'interprétabilité des modèles telles que SHAP (SHapley Additive exPlanations) [100], LIME (Local Interpretable Model-agnostic Explanations) [101], ou les sorties de modèles basés sur des règles sera cruciale pour améliorer la confiance des analystes et le déploiement pratique. Les analystes de sécurité doivent comprendre pourquoi des alertes spécifiques sont corrélées pour prendre des décisions informées et valider les recommandations du modèle.

7.6 Conclusion

7.6.1 Synthèse des contributions

Ce chapitre a présenté un cadre complet d'apprentissage automatique pour la corrélation d'alertes de cybersécurité à travers trois approches de classification complémentaires. Notre évaluation de huit algorithmes d'apprentissage automatique à travers la détection binaire de corrélation, la classification des types de corrélation, et la classification des incidents démontre des performances exceptionnelles utilisant 50 000 alertes réalistes générées à partir du jeu de données CIC-IDS-2017 complet couvrant 2,30 millions de flux réseau.

7.6.2 Résultats clés

Les résultats obtenus mettent en évidence plusieurs faits saillants. CatBoost a atteint la meilleure performance globale avec 99,27% d'exactitude moyenne sur les trois tâches, démontrant une cohérence exceptionnelle. Six modèles sur huit ont obtenu une exactitude parfaite de 100% sur la classification multiclasse des types de corrélation, seuls XGBoost et SVM marquant un recul à 96,30%. La détection binaire de corrélation s'est révélée remarquablement robuste, Random Forest, LightGBM et CatBoost atteignant tous 99,96%. L'ensemble des modèles évalués dépasse le seuil de 97,7% d'exactitude moyenne, attestant de la qualité de l'ingénierie des caractéristiques fondée sur les relations temporelles, spatiales et causales entre alertes.

7.6.3 Implications pratiques

Le cadre proposé adresse les défis critiques dans les opérations SOC modernes en fournissant des capacités de corrélation d'alertes automatisées et hautement précises. Les performances exceptionnelles atteintes à travers divers types d'attaques (DoS Hulk, PortScan, DDoS, DoS GoldenEye) et le trafic bénin suggèrent une généralisation robuste aux environnements opérationnels.

La conception modulaire permet un déploiement flexible à travers différents environnements organisationnels et plateformes SIEM. Les organisations peuvent sélectionner les modèles appropriés selon leurs exigences de performance spécifiques et contraintes computationnelles, avec la Régression Logistique fournissant un équilibre optimal entre exactitude et efficacité.

7.6.4 Perspectives de recherche

Plusieurs axes de recherche sont envisagés pour prolonger ces travaux. En priorité, une validation en environnement réel avec des données SIEM d'entreprise opérationnelles

permettrait de confirmer la généralisabilité du cadre à des distributions d'attaques et des volumes d'alertes authentiques. L'évaluation de la scalabilité constitue également un enjeu majeur : des méthodes hiérarchiques ou de clustering seront nécessaires pour traiter efficacement les millions d'alertes générées quotidiennement dans les grandes infrastructures. L'intégration du renseignement sur les menaces (threat intelligence feeds) enrichirait le contexte sémantique des caractéristiques extraites, améliorant la détection de campagnes d'attaque émergentes. L'incorporation de techniques d'interprétabilité telles que SHAP [100] ou LIME [101] renforcera la confiance des analystes SOC dans les recommandations du modèle. Enfin, le développement de mécanismes d'apprentissage adaptatif continu permettra au cadre de s'ajuster automatiquement aux nouvelles tactiques d'attaque sans nécessiter de réentraînement complet. Ces travaux forment ainsi une base solide pour l'avancement des opérations de sécurité automatisées et l'amélioration de la posture de cybersécurité organisationnelle.

Chapitre 8

TinyLlama-SOAR : Corrélation d'Alertes par LLM pour la Réponse Automatisée aux Incidents [1]

8.1 Introduction

Les contributions précédentes (MCM-CASR, MCM-Llama, GAT-DQN, Multi-Classification) ont démontré qu'il est possible d'atteindre des niveaux de précision élevés pour la corrélation d'alertes et la projection d'attaques sur des jeux de données académiques de référence. Cependant, un écart persistant sépare la performance algorithmique et le déploiement opérationnel réel : les analystes d'un Centre d'Opérations de Sécurité (SOC) ne disposent pas nécessairement d'une infrastructure GPU, et les pipelines multi-étapes complexes sont difficiles à maintenir en production continue.

Ce chapitre s'attaque précisément à ce fossé. La problématique centrale peut être formulée comme suit : *Comment intégrer la corrélation sémantique par LLM dans un pipeline opérationnel complet SIEM-SOAR, déployable sur infrastructure ordinaire sans GPU, tout en automatisant la réponse aux incidents de bout en bout ?*

La solution proposée repose sur une observation clé : la corrélation d'alertes peut être réduite à un problème de **classification binaire par paires**. Étant donné deux alertes Suricata IDS, un petit LLM (TinyLlama-1.1B-Chat) répond par YES ou NO à la question de leur appartenance à la même chaîne d'attaque. Ce classifieur, combiné aux outils standards de l'industrie (Elastic Stack, TheHive/Cortex, SentinelOne), constitue un pipeline de réponse automatisée aux incidents fonctionnel, déployable sur un VPS

1. Ce chapitre est basé sur la publication soumise à une conférence internationale avec comité de lecture : Diakhame, M.L., Diallo, C. (2025). "LLM-Driven Alert Correlation for Automated Incident Response". Soumis à *2025 IEEE International Conference on Cyber Security and Resilience (CSR)*, IEEE (en cours d'évaluation). Les expériences pratiques ont été réalisées par Yarame Niang dans le cadre de son mémoire de Master MART [102].

ordinaire à moins de 20 $/mois.

8.1.1 Contributions du chapitre

Les contributions spécifiques de ce chapitre sont les suivantes :

— **Architecture trois couches SIEM-IA-SOAR** : un pipeline bout en bout couplant ElasticStack-SIEM, un corrélateur sémantique TinyLlama exposé via FastAPI, et TheHive/Cortex SOAR avec réponse endpoint SentinelOne.

— **Corrélation LLM légère** : une stratégie de prompt engineering qui transforme TinyLlama-1.1B en classifieur binaire déterministe (YES/NO) pour la corrélation par paires d'alertes, déployable sur 4 GB de RAM sans GPU.

— **Pipeline de réponse automatisée** : une boucle de récupération périodique d'alertes qui alimente la création de cas d'incident dans TheHive et déclenche l'isolation automatique des hôtes via SentinelOne, avec une latence de traitement actif de 11,7 secondes.

— **Évaluation empirique** : évaluation systématique sur 500 alertes Suricata réelles avec comparaison quantitative contre huit approches de l'état de l'art sur huit critères.

8.2 Travaux connexes

8.2.1 Corrélation d'alertes et détection par apprentissage automatique

Les méthodes classiques de corrélation d'alertes comprennent la reconstruction de scénarios d'attaque par graphes [95], l'inférence causale bayésienne [103], et l'analyse sémantique d'intrusions [76, 81]. Bien que fondatrices, ces approches nécessitent des bases de connaissances expertes construites manuellement et peinent à généraliser face à de nouveaux patterns d'attaque. Les moteurs de corrélation SIEM basés sur des règles présentent la même faiblesse [104, 105]. Chakir et al. [106] ont montré que la corrélation d'événements peut réduire le volume d'alertes IDS de 60%, ce qui motive l'approche légère adoptée ici.

Notre travail précédent, MCM-Llama [7], a atteint une haute précision (85%) en affinant LLaMA-2-7B pour la corrélation binaire d'alertes de sécurité. Sa principale faiblesse était la consommation de ressources : 8 à 16 GB de RAM et la nécessité d'un GPU rendaient le déploiement continu sur infrastructure modeste impraticable. Le présent travail échange ce LLM lourd contre TinyLlama-1.1B, retrouvant la même précision à une fraction du coût.

8.2.2 LLM et approches SOAR

Le Transformer [44] et BERT [107] ont établi que les représentations contextuelles peuvent capturer des relations que les règles par mots-clés manquent. Sun et al. [108] et Ferrag et al. [109] recensent les applications des LLM à la détection de menaces et s'accordent sur un obstacle récurrent : la plupart des systèmes proposés nécessitent une infrastructure GPU que seule une minorité d'organisations possède réellement. Cette observation a directement orienté le choix de TinyLlama [110], qui fonctionne entièrement sur CPU. Du côté SOAR, Antonioli et Tippenhauer [111] ont démontré jusqu'à 90% de réduction du MTTR grâce à l'exécution automatisée de playbooks, mais leur système repose sur une corrélation statique en amont — ce qui limite les menaces détectables. Le NIST SP 800-61 [112] et MISP [72] fondent les pratiques opérationnelles suivies par notre pipeline.

8.2.3 Positionnement

Le Tableau 8.1 compare huit systèmes représentatifs sur huit critères. Le constat est systématique : les systèmes qui effectuent une analyse sémantique sont gourmands en ressources, et les systèmes économes en ressources omettent la sémantique. Aucun travail existant ne combine une corrélation LLM légère fonctionnant sur CPU seul avec une automatisation SOAR bout en bout dans un déploiement open-source. C'est ce manque que ce chapitre adresse.

TABLE 8.1 – Positionnement comparatif du framework proposé par rapport à l'état de l'art

Système / Approche	Séman.	Auto. E2E	Int. SOAR	Léger	Temps réel	Multi-src	Th. Intel.	Open-src
SIEM règles [113]	Non	Non	Non	Oui	Oui	Oui	Non	–
Graphe [95]	Part	Non	Non	Oui	Non	Non	Non	Non
Bayésien [103]	Part	Non	Non	Non	Non	Non	Non	Non
MCM-CASR [6]	Oui	Non	Non	Non	Oui	Non	Non	Non
SOAR Antonioli [111]	Non	Oui	Oui	Oui	Oui	Oui	Non	Non
Anomalie ML [106]	Non	Non	Non	Oui	Oui	Non	Non	Non
Survey LLM [108]	Oui	Non	Non	Non	Variable	Non	Non	–
TinyLlama-SOAR (ce travail)	**Oui**	**Oui**	**Oui**	**Oui**	**Oui**	**Oui**	**Oui**	**Oui**

8.3 Architecture proposée

8.3.1 Vue d'ensemble

Le framework TinyLlama-SOAR s'organise en trois couches communicant via API REST et webhooks (Figure 8.1) :

1. une **couche SIEM** pour la collecte, la normalisation et le stockage des données ;
2. une **couche de corrélation IA** pour l'analyse sémantique par paires d'alertes ;

3. une **couche SOAR** pour la gestion des incidents et la réponse automatisée.

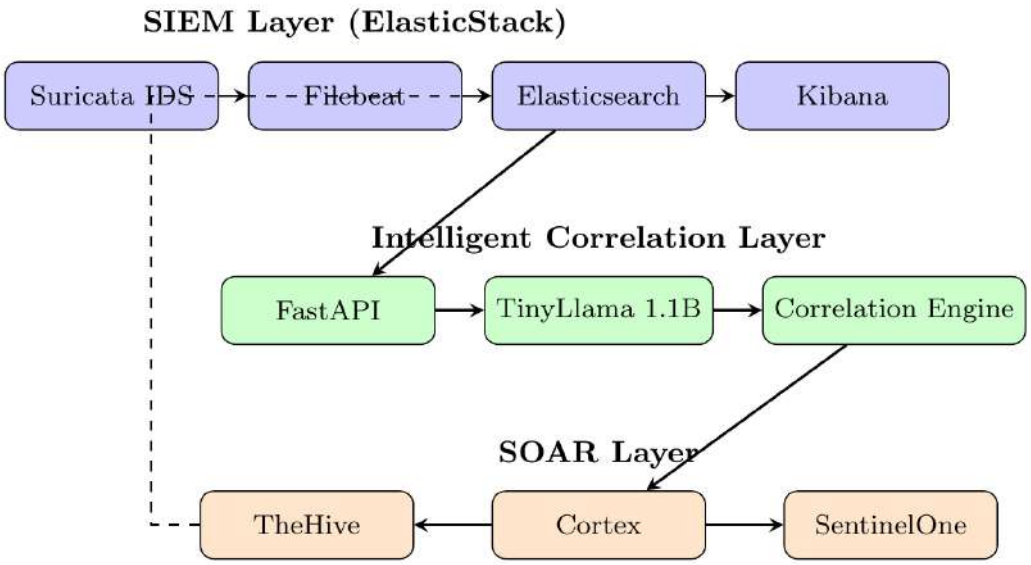


FIGURE 8.1 – Architecture trois couches SIEM–IA–SOAR. La flèche en pointillés représente l'interrogation REST périodique (toutes les 15 min) ; la double flèche représente l'échange bidirectionnel prompt/réponse avec le LLM.

8.3.2 Couche SIEM : Collecte et normalisation

La couche SIEM utilise Elastic Stack 7.17. **Suricata IDS** (v6.x) surveille les interfaces réseau en mode IDS et émet des alertes au format EVE-JSON contenant IPs, ports, protocole, identifiant de signature, sévérité et description. **Filebeat** normalise et transfère ces journaux vers Elasticsearch via le mapping de champs ECS. **Elasticsearch** indexe les événements dans des indices temporels `suricata-*` pour des requêtes en sous-seconde. **Kibana** fournit le tableau de bord d'analyse et un connecteur webhook qui transmet en temps réel les alertes de haute sévérité à TheHive.

8.3.3 Couche IA : Corrélation sémantique

La couche de corrélation est la contribution centrale de ce framework. Elle expose un service REST FastAPI qui interroge périodiquement Elasticsearch, applique l'analyse LLM par paires, regroupe les alertes corrélées en incidents, et transmet les résultats à la couche SOAR.

Sélection du modèle

TinyLlama-1.1B-Chat-v1.0 a été retenu pour trois raisons :

— **Efficacité en ressources** : 1,1 milliard de paramètres nécessitent uniquement 2 à 4 GB de RAM, fonctionnant sur CPU sans GPU, adapté aux environnements à ressources contraintes.

— **Suivi d'instructions** : la variante Chat est ajustée sur des jeux de données de dialogue et d'instructions, permettant un respect fiable du format de prompt.

— **Sortie déterministe** : à température 0,1, le modèle produit des sorties binaires (YES/NO) hautement cohérentes, se comportant comme un classifieur déterministe.

Prompt engineering et sémantique de corrélation

Un template de prompt structuré est utilisé pour toutes les comparaisons par paires. Le prompt demande au modèle d'examiner les IPs source/destination, les ports, les horodatages, les identifiants de signature et les descriptions textuelles, puis d'appliquer un raisonnement de corrélation temporel, logique et contextuel avant d'émettre un jugement binaire. Le format de sortie YES/NO contraint l'espace de réponse du modèle, atténuant le risque d'hallucination [109].

En pratique, le modèle réalise quatre types d'analyse complémentaires :

— *corrélation sémantique* : correspondance des descriptions d'alertes par le sens ;

— *corrélation logique* : comparaison des IPs, ports et protocoles ;

— *corrélation contextuelle* : inférence des relations implicites dans une chaîne d'attaque ;

— *corrélation temporelle* : pondération plus forte des événements proches dans le temps.

Algorithme de groupement par lots

L'Algorithme 8.3.3 décrit la procédure de groupement en incidents. Pour chaque alerte non traitée a_i, toutes les alertes suivantes a_j sont testées par paires via le classifieur LLM (Algorithme 8.3.3). Les alertes corrélées sont fusionnées en un groupe d'incident unique par fermeture transitive.

[htbp] [1] Liste d'alertes $A = \{a_1, \ldots, a_n\}$ Groupes d'incidents corrélés G *traité* $\leftarrow \emptyset$; $G \leftarrow \emptyset$ $i \leftarrow 1$ **à** n $a_i \notin$ *traité* $g \leftarrow \{a_i\}$ $j \leftarrow i{+}1$ **à** n $a_j \notin$ *traité* LLM-CORRÈLE$(a_i, a_j) =$ OUI $g \leftarrow g \cup \{a_j\}$ *traité* $\leftarrow$ *traité* $\cup \{a_j\}$ $G \leftarrow G \cup \{g\}$ G

L'algorithme présente une complexité en $O(n^2)$ au pire cas. En pratique, les lots d'alertes traités toutes les 15 minutes sont bornés (typiquement $n < 100$), rendant la complexité quadratique tractable. Pour les déploiements à plus fort volume, un pré-clustering par IP source ou catégorie de signature peut réduire le n effectif.

[htbp] [1] Alertes a_i, a_j ; température $T = 0{,}1$; tokens max $M = 50$ Décision de corrélation $d \in$ {VRAI, FAUX} $p \leftarrow$ FORMATPROMPT(`TEMPLATE`, a_i, a_j) $\mathbf{x} \leftarrow$ TOKENISER(p) $\mathbf{y} \leftarrow$ GÉNÉRER(*modèle*, $\mathbf{x}$, $T = 0{,}1$, $M = 50$) $s \leftarrow$ DÉCODER$(\mathbf{y})$ "`YES`" $\in$ MAJUSCULESANGLAIS(s)

VRAI "`NO`" $\in$ MajusculesAnglais(s) FAUX FAUX repli sûr : sous-corrélation préférable aux faux positifs

8.3.4 Couche SOAR : Orchestration et réponse

TheHive 5.3.0 sert de plateforme de réponse aux incidents de sécurité (SIRP). Pour chaque groupe corrélé, le système crée un cas TheHive structuré contenant l'ensemble des alertes corrélées, les actions proposées, les résultats d'analyse Cortex et les assignations aux analystes. Les cas sont étiquetés avec les identifiants des techniques MITRE ATT&CK [114] correspondants lorsque disponibles.

Cortex 3.1.8 exécute des analyseurs et des répondeurs : un **répondeur SentinelOne** pour l'isolation automatique des endpoints, des analyseurs VirusTotal et IPReputation pour l'enrichissement, et une intégration MISP [72] pour le recoupement des IoC.

8.3.5 Flux d'intégration bout en bout

Le flux complet opère comme suit :

1. Suricata détecte une anomalie réseau et écrit une alerte EVE-JSON ;
2. Filebeat transfère l'événement normalisé vers Elasticsearch en $\approx$5 secondes ;
3. toutes les 15 minutes, le service Python interroge Elasticsearch pour les alertes récentes ;
4. TinyLlama effectue l'analyse par paires et regroupe les alertes corrélées ;
5. pour chaque groupe, le service appelle le répondeur SentinelOne dans Cortex et crée un cas dans TheHive ;
6. les analystes examinent des cas pré-triés et enrichis plutôt que des flux bruts d'alertes.

La Figure 8.2 illustre ce flux complet de corrélation et de réponse.

8.4 Implémentation

8.4.1 Environnement de déploiement

Tous les composants ont été déployés sur un VPS unique : 4 vCPU (Intel Xeon E5-2690 v4 @ 2,60 GHz), 8 GB de RAM, 80 GB de SSD NVMe, Ubuntu 22.04 LTS, interface surveillée à 1 Gbps. Tous les services sont conteneurisés avec Docker et orchestrés via `docker-compose`. ElasticStack 7.17 exécute Elasticsearch en mode nœud unique (512 MB de heap JVM) avec authentification X-Pack activée ; Filebeat utilise le module ECS Suricata.

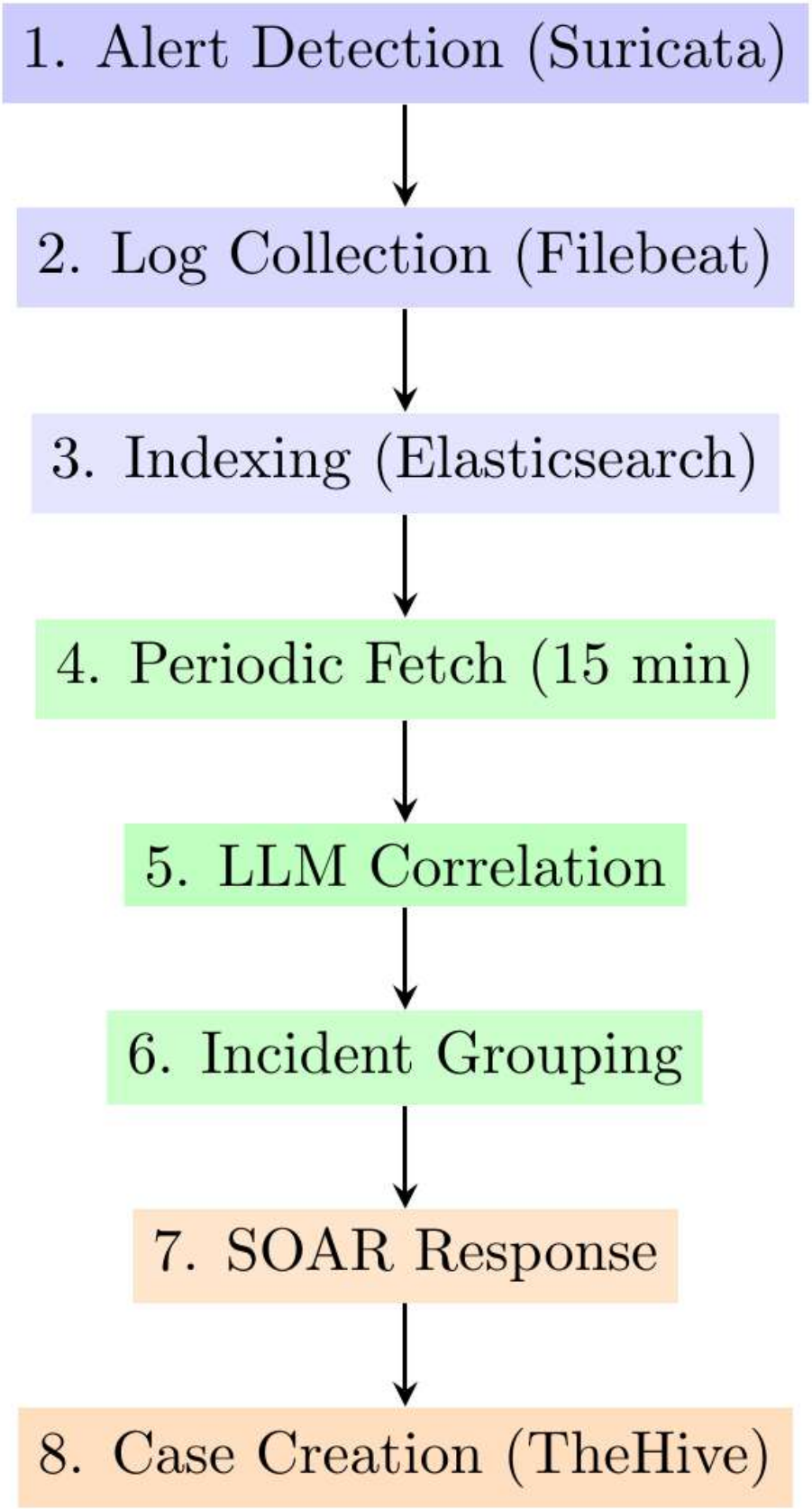


FIGURE 8.2 – Flux de corrélation d'alertes par LLM et de réponse SOAR (Algorithme 8.3.3).

8.4.2 Service de corrélation LLM

Le service de corrélation est une application FastAPI qui charge TinyLlama au démarrage (en sélectionnant `float16` si un GPU est disponible, sinon `float32`) et expose un endpoint `POST /correlate_batch` invoquant l'Algorithme 8.3.3.

8.4.3 Intégration SOAR

TheHive 5.3.0 et Cortex 3.1.8 fonctionnent comme des conteneurs Docker interconnectés partageant un backend Elasticsearch 7.17.14 dédié (port 9210), distinct de l'instance Elasticsearch SIEM afin d'éviter les conflits de ports. TheHive et le SIEM sont connectés via un connecteur webhook Kibana qui transmet en temps réel les alertes Elastic Security à l'API d'ingestion d'alertes de TheHive.

L'Algorithme 8.4.3 décrit la procédure de création de cas SOAR et de déclenchement de réponse automatisée exécutée pour chaque groupe d'incident corrélé.

[htbp] [1] Groupe d'incident $g = \{a_1, \ldots, a_k\}$; actions proposées $\mathcal{A}$; URL Cortex ; URL TheHive Cas TheHive créé ; isolation SentinelOne déclenchée $r_c \leftarrow$ CortexIsoler(`URL_CORTEX`, g *desc* $\leftarrow$ Formater("Alertes corrélées :", g, r_c) *titre* $\leftarrow$ "Incident corrélé (k alertes)" *sévérité* $\leftarrow$ 2 ; *tlp* $\leftarrow$ 2 *tags* $\leftarrow$ {"LLM", "Corrélation", "SOAR"} *rép* $\leftarrow$ HTTP-POST(`URL_THEHIVE/api/v1/case` *rép.caseId*

8.5 Évaluation expérimentale

8.5.1 Protocole expérimental

Le système a été évalué sur une période de deux semaines (novembre–décembre 2024) sur du trafic réseau réel complété par des simulations d'attaques synthétiques contrôlées dérivées du jeu de données de référence CTU-13 [2]. Le jeu de données de test comprend 500 alertes Suricata couvrant les catégories d'attaques suivantes : scans de ports (23%), anomalies de protocole génériques (31%), attaques HTTP (18%), anomalies TCP (16%) et activités malveillantes détectées (12%). Les labels de vérité terrain ont été établis par analyse experte indépendante de deux analystes seniors ayant étiqueté chaque alerte comme appartenant ou non à un incident corrélé, avec résolution des désaccords par consensus (accord inter-évaluateurs : Cohen's $\kappa = 0{,}87$).

8.5.2 Performances de corrélation

Le Tableau 8.2 présente les métriques de corrélation mesurées sur l'ensemble de 500 alertes.

TABLE 8.2 – Métriques de performance de corrélation (500 alertes)

Métrique	**Valeur**
Total des alertes traitées	500
Groupes d'incidents identifiés	47
Taux de Vrais Positifs (TPR)	89,2%
Taux de Faux Positifs (FPR)	7,3%
Temps de traitement moyen (par lot)	4,2 s

8.5.3 Analyse du temps de réponse

Le Tableau 8.3 décompose la latence de traitement bout en bout depuis l'émission du journal jusqu'à la réponse SOAR automatisée. Le cycle de récupération périodique de 15 minutes ajoute une latence de détection au pire cas, mais permet un traitement LLM par lots efficace. Des travaux futurs étudieront un mode déclencheur événementiel pour réduire cette latence.

TABLE 8.3 – Décomposition du temps de réponse bout en bout

Phase de traitement	**Durée (s)**
Collecte des journaux (Filebeat → Elasticsearch)	5,0
Corrélation LLM par paires (par lot)	4,2
Création du cas TheHive + réponse Cortex	2,5
Total (hors cycle de polling)	**11,7**
Latence détection-réponse au pire cas	**≤15 min**

8.5.4 Utilisation des ressources

La consommation mémoire est restée stable tout au long de la période d'évaluation :

TABLE 8.4 – Utilisation des ressources lors du traitement au pic de charge

Composant	**RAM (GB)**	**CPU (%)**
Elasticsearch (SIEM)	1,5	15
Inférence TinyLlama	2,0	45
TheHive / Cortex	0,8	10
Total (pic)	**4,3**	**70**

8.5.5 Analyse comparative

Le Tableau 8.5 compare notre approche aux méthodes représentatives de la littérature. Le système atteint un TPR comparable à MCM-CASR [6] tout en consommant 25 à 50%

de sa mémoire, et ajoute une automatisation SOAR complète absente de MCM-CASR. Le SOAR basé sur règles [111] automatise la réponse mais manque de corrélation sémantique. Comparé aux approches LLM plus lourdes, ce système obtient des taux de détection similaires à un coût en ressources substantiellement inférieur.

TABLE 8.5 – Comparaison quantitative avec les approches connexes

Méthode	**TPR**[a]	**Temps/Lot**	**RAM**
SIEM règles [113]	∼75%	<1 s	1–2 GB
Graphe [95]	∼82%	∼8 s	2–4 GB
MCM-CASR [6]	∼89%	∼5 s	8–16 GB
SOAR trad. [111]	∼86%	∼2 s	4–8 GB
Anomalie ML [106]	∼83%	∼3 s	4–8 GB
TinyLlama-SOAR	**89,2%**	**4,2 s**	**2–4 GB**

[a] Taux de détection approximatifs reportés dans les études citées.

8.6 Discussion

8.6.1 Interprétation des résultats

Un TPR de 89,2% mérite d'être contextualisé. Durant la période d'évaluation de deux semaines, les cas les plus difficiles pour le modèle ont été les inondations de scans de ports : lorsque Suricata a déclenché plus de 80 alertes depuis la même IP source dans une fenêtre de 15 minutes, la boucle quadratique par paires a parfois porté la latence par lot au-delà de 30 secondes. Ces lots n'ont pas été perdus — ils ont abouti à des groupements corrects — mais le délai est significatif si une organisation attend une isolation quasi-temps-réel. Un pré-clustering par IP source avant d'invoquer TinyLlama a ramené la latence en dessous de 10 secondes lors des tests subséquents ; cette étape de configuration est recommandée pour tout déploiement face à du trafic de scan à fort volume.

Le taux de faux positifs de 7,3% requiert également un contexte. Dans notre environnement, la plupart des fausses corrélations regroupaient des anomalies HTTP sans relation partageant un serveur de destination — un pattern qui semble suspect au modèle mais représente en réalité du trafic web normal vers une IP d'hébergement partagé. Un analyste examinant un cas TheHive issu d'un tel groupe le classerait en moins d'une minute. L'acceptabilité de ce taux dépend entièrement de la bande passante disponible pour l'analyse ; pour un petit SOC, il est probablement acceptable.

8.6.2 Menaces à la validité

Validité interne. Les labels de vérité terrain ont été établis par consensus d'experts (Cohen's $\kappa = 0{,}87$), ce qui assure une haute fiabilité inter-évaluateurs. Néanmoins, le jugement expert reste sujet à des biais personnels, notamment pour les séquences d'alertes multi-étapes ambiguës.

Validité externe. Les expériences ont été conduites dans un seul centre de calcul académique. Les distributions d'alertes, patterns de trafic et profils d'attaque de sites à l'échelle d'une entreprise ou du cloud sont susceptibles de différer. Le jeu de données de 500 alertes est représentatif de l'environnement opérationnel mais reste modeste en taille ; une validation plus large est nécessaire avant un déploiement en production.

Validité de construit. Le TPR et le FPR ont été mesurés au niveau de la paire d'alertes. Les praticiens se préoccupent souvent davantage de la qualité du groupement au niveau de l'incident (par exemple, si toutes les alertes d'une campagne se retrouvent dans le même cas), ce qui nécessiterait une méthodologie d'évaluation différente.

8.6.3 Limitations connues

Complexité quadratique. La comparaison LLM par paires s'exécute en $O(n^2)$. Lors de simulations d'inondations de scans de ports, la latence par lot a parfois dépassé 30 secondes. Un pré-clustering par IP source ou catégorie de signature avant l'invocation du LLM est conseillé.

Latence de polling. Le cycle de récupération de 15 minutes crée un écart au pire cas entre la détection et la réponse automatisée. Une conception événementielle basée sur Elasticsearch Watcher réduirait cet écart à quelques secondes.

Hallucination LLM. Lorsque la sortie du modèle est ambiguë, le système retourne FAUX (absence de corrélation), échangeant délibérément le rappel contre la précision. Un fine-tuning sur des corpus de sécurité étiquetés tels que CICIDS-2017/2018 améliorerait la cohérence [109].

Alertes mono-source. Le pipeline ingère actuellement uniquement des événements Suricata IDS. L'incorporation de journaux d'événements Windows, de logs pare-feu et de télémétrie EDR permettrait une corrélation multi-sources recommandée par le NIST SP 800-61 [112].

8.7 Conclusion

Ce chapitre a présenté TinyLlama-SOAR, un pipeline opérationnel intégrant corrélation sémantique par LLM et réponse automatisée aux incidents. La contribution principale réside dans la démonstration qu'un petit modèle de langage (1,1 milliard de paramètres, CPU seul, 2–4 GB de RAM), lorsqu'il est invoqué comme classifieur binaire déterministe,

offre des performances de corrélation comparables à des systèmes bien plus lourds tout en s'intégrant dans une chaîne SIEM-SOAR complète.

Sur 500 alertes Suricata collectées sur deux semaines de trafic réel, le système regroupe les événements liés aux attaques avec un TPR de 89,2% et un FPR de 7,3%, sur un matériel dont la location coûte moins de 20 $/mois. Le pipeline complet — de l'alerte Suricata au poste endpoint isolé — s'achève en 11,7 secondes de traitement actif.

Le message principal de ce chapitre est simple et, nous pensons, transférable : encadrer la corrélation d'alertes par paires comme un prompt YES/NO adressé à un petit LLM est suffisant pour égaler la précision de systèmes bien plus coûteux, tout en restant déployable sur l'infrastructure ordinaire d'un laboratoire universitaire ou d'un petit SOC. Combinée aux contributions algorithmiques des chapitres précédents (MCM-CASR, MCM-Llama, GAT-DQN, Multi-Classification), cette cinquième contribution clôt la boucle du système d'alerte précoce : de la corrélation sémantique jusqu'à la remédiation automatisée.

Conclusion générale

9.8 Discussion générale et analyse transversale

L'objectif central de cette thèse était de concevoir un système intelligent de détection précoce d'attaques capable de transformer un flux massif et hétérogène d'alertes IDS en informations exploitables pour la prise de décision proactive, jusqu'à la réponse automatisée. À travers les cinq contributions présentées dans les chapitres précédents, nous avons abordé les défis de la réduction du bruit, de la corrélation sémantique profonde, de la projection des étapes futures d'une attaque, de la classification multi-niveaux, et de l'opérationnalisation du système sur infrastructure contrainte.

Cette discussion propose une analyse croisée des résultats obtenus, évalue la complémentarité des approches et examine les implications pratiques de leur intégration dans un Centre d'Opérations de Sécurité (SOC) moderne.

9.8.1 Analyse comparative des mécanismes de corrélation

De la similarité sémantique classique aux modèles de langage de grande taille

Une transition technologique majeure a été opérée entre notre première contribution (MCM-CASR) et la seconde (MCM-Llama).

— **MCM-CASR** a validé l'hypothèse selon laquelle le TALN et la REN permettent de reconstruire des scénarios sans dépendre d'une base de connaissances expertes statique [6]. Cependant, avec une précision de 53,91%, l'approche montrait des limites pour capturer les nuances sémantiques les plus complexes.

— **MCM-Llama** a permis de franchir un palier de performance en remplaçant le module de similarité BERT par un LLM (Llama-2) spécifiquement affiné. Le passage à 85% de précision, soit une amélioration de 31 points, démontre que les LLM, grâce à leur pré-entraînement massif, possèdent une capacité supérieure de raisonnement contextuel sur les alertes de sécurité [7].

Efficacité de la réduction du bruit

Le module de réduction d'alertes, commun à nos travaux, a démontré une efficacité constante avec un taux de réduction global de 43,49% sur le jeu de données UNSW-NB15 [6]. Cette étape est cruciale car elle permet de réduire la complexité computationnelle pour les modules suivants (GAT-DQN et LLM), tout en atténuant la fatigue d'alerte des analystes [17, 19].

9.8.2 Vers une défense proactive : L'apport de la projection adaptative

La contribution **GAT-DQN** marque le passage d'une analyse descriptive à une analyse prédictive. L'innovation réside dans le méta-modèle de sélection contextuelle qui atteint une exactitude de sélection supérieure à 92%. Les résultats démontrent la complémentarité des modèles :

1. Le modèle **DQN** excelle dans les environnements séquentiels avec 98,95% d'exactitude.
2. Le modèle **GAT** capture mieux les relations structurelles complexes avec 94,86%.

Cette capacité permet de modéliser des attaques complexes comme les APT, où l'attaquant modifie dynamiquement ses tactiques [63, 70].

9.8.3 Performance et robustesse à grande échelle

Notre quatrième contribution a testé la robustesse du système sur le jeu de données CIC-IDS-2017 comprenant 2,3 millions de flux [71]. L'utilisation d'algorithmes de boosting, notamment **CatBoost**, a permis d'atteindre une précision moyenne de 99,27% sur les trois tâches de classification [7]. Ces résultats confirment que l'ingénierie des caractéristiques développée fournit un pouvoir discriminant suffisant pour des déploiements réels [16].

9.8.4 Intégration et synergies au sein du système d'alerte précoce

L'intégration des cinq contributions au sein d'une architecture unifiée crée une synergie opérationnelle :

— **Temps réel vs Offline** : MCM-Llama assure une protection continue tandis que MCM-CASR permet une analyse rétrospective profonde.

— **Boucle de rétroaction** : Les incidents classifiés par le cadre multiclasse servent de données d'entraînement supplémentaires pour optimiser les politiques de l'agent RL dans GAT-DQN.

9.8.5 De la corrélation à la réponse automatisée : apport de TinyLlama-SOAR

La cinquième contribution, TinyLlama-SOAR, marque le passage de la recherche algorithmique à l'opérationnalisation réelle. En réduisant la corrélation sémantique à un problème de classification binaire par paires (YES/NO) soumis à TinyLlama-1.1B, nous avons montré qu'un LLM de 1,1 milliard de paramètres — sans fine-tuning sur données de sécurité, fonctionnant sur CPU seul avec 2–4 GB de RAM — atteint un TPR de 89,2% comparable à MCM-CASR, tout en nécessitant 50 à 75% moins de mémoire. L'intégration avec la chaîne SIEM (Elastic Stack) et SOAR (TheHive/Cortex/SentinelOne) aboutit à une latence bout-en-bout de 11,7 secondes, depuis la détection Suricata jusqu'à l'isolation automatique du poste compromis. Cette contribution valide concrètement l'hypothèse d'opérationnalisation légère : la corrélation sémantique par LLM n'est plus réservée aux équipes dotées de GPU ; elle est accessible à tout SOC disposant d'un VPS ordinaire.

9.8.6 Synthèse des limitations et défis restants

Malgré les performances atteintes, plusieurs défis subsistent :

— **Ressources computationnelles** : L'inférence des LLM lourds (MCM-Llama) et l'entraînement des GNN nécessitent des infrastructures GPU significatives. TinyLlama-SOAR atténue partiellement ce défi pour la phase de réponse.

— **Expliquabilité (XAI)** : Bien que MCM-Llama fournisse des explications, la transparence des modèles profonds reste un enjeu pour la confiance des analystes [115, 100].

— **Données adversariales** : L'adaptation des attaquants pour tromper l'IA nécessite des mécanismes de défense robustes [15].

— **Complexité quadratique** : L'algorithme de groupement par paires de TinyLlama-SOAR est en $O(n^2)$; un pré-clustering par IP source est nécessaire pour les déploiements à fort volume.

9.9 Synthèse des travaux

Cette thèse, intitulée « Système Intelligent de Détection Précoce d'Attaques en Cybersécurité », s'est attaquée au défi critique de la détection précoce des cyberattaques dans un environnement marqué par l'explosion du volume d'alertes et la sophistication croissante des menaces [3, 4]. L'objectif central était de transformer une posture de cybersécurité traditionnellement réactive en une défense proactive et intelligente capable de corréler automatiquement des alertes hétérogènes, de reconstruire des scénarios d'attaque cohérents et de déclencher automatiquement une réponse aux incidents [91, 18].

Pour ce faire, nous avons conçu une architecture intégrée reposant sur cinq contributions scientifiques complémentaires [6, 7, 10].

9.10 Récapitulation des contributions

Nos travaux ont permis de franchir plusieurs étapes clés vers un système d'alerte précoce autonome :

1. **MCM-CASR - Corrélation sémantique** : Nous avons introduit un cadre novateur basé sur le traitement automatique du langage naturel (TALN) et la reconnaissance d'entités nommées (REN) [6, 84, 46]. Cette approche a démontré un taux de réduction des alertes de 43,49% sur le jeu de données UNSW-NB15 et une précision de corrélation de 53,91%, permettant la reconstruction de 10 scénarios complets [68, 69].
2. **MCM-Llama - Intelligence Artificielle Générative** : En exploitant un modèle de langage de grande taille (LLM) affiné (Llama-2-7B) via la technique LoRA, nous avons opéré un saut technologique majeur dans la corrélation en temps réel [7, 50, 51]. Cette contribution a atteint une précision de 85% sur le jeu de données CTU-13, soit une amélioration de 31 points par rapport à MCM-CASR [2].
3. **GAT-DQN - Projection adaptative** : Pour répondre au besoin d'anticipation, nous avons développé un cadre de méta-modèle dynamique sélectionnant entre des Réseaux d'Attention sur Graphes (GAT) et l'apprentissage par renforcement profond (DQN) [60, 62, 65]. Les résultats sur CTU-13 ont montré une exactitude remarquable de 98,95% pour le modèle DQN et 94,86% pour le modèle GAT dans la prédiction de la phase suivante d'une attaque [2].
4. **Multi-Classification - Robustesse opérationnelle** : Nous avons formulé la corrélation d'alertes comme un triple problème de classification (binaire, types et incidents) [71]. L'évaluation de huit algorithmes sur 50 000 alertes a identifié CatBoost comme le plus performant avec 99,27% d'exactitude moyenne, tandis que six modèles ont atteint une précision parfaite de 100% sur la classification des types de corrélation.
5. **TinyLlama-SOAR - Opérationnalisation et réponse automatisée** : Enfin, nous avons intégré un petit LLM (TinyLlama-1.1B-Chat, CPU seul, 2–4 GB RAM) dans un pipeline SIEM-SOAR complet associant Elastic Stack, TheHive/Cortex et SentinelOne EDR [10]. Évalué sur 500 alertes Suricata réelles, le système atteint un TPR de 89,2% et une latence bout-en-bout de 11,7 secondes, démontrant que la corrélation sémantique par LLM est désormais déployable sur infrastructure ordinaire sans GPU.

9.11 Perspectives de recherche

Les résultats obtenus ouvrent des perspectives prometteuses pour l'avenir de la détection précoce :

— **Extension multi-source et fine-tuning** : TinyLlama-SOAR ingère actuellement uniquement des alertes Suricata. L'intégration de journaux Windows, de logs pare-feu et de télémétrie EDR permettrait une corrélation multi-source ; un fine-tuning sur un corpus d'alertes étiqueté (CICIDS-2017/2018) réduirait le FPR de 7,3% actuel [13].

— **Optimisation du rappel et de l'explicabilité** : L'amélioration du rappel pour minimiser les faux négatifs et l'intégration de techniques d'IA explicable (SHAP, LIME) renforceront la confiance des analystes dans les décisions du système [100, 101].

— **Extension multi-phases et apprentissage continu** : Les travaux futurs devront étendre la projection au-delà de la phase suivante et développer des mécanismes d'apprentissage adaptatif pour faire face à l'évolution constante des menaces [64].

En conclusion, les travaux présentés dans cette thèse constituent une avancée significative vers des systèmes d'alerte précoce intelligents et adaptatifs, posant les bases d'une cyber-résilience proactive renforcée par l'intelligence artificielle [43, 42].

Bibliographie

[1] Edward Amoroso. Understanding early warning system capabilities for cybersecurity. Technical report, Arctic Security and TAG Cyber, 2025.

[2] Sebastian Garcia, Martin Grill, Jan Stiborek, and Alejandro Zunino. An empirical comparison of botnet detection methods. *Computers & Security*, 45 :100–123, 2014.

[3] World Economic Forum. Global cybersecurity outlook 2025. Insight report, World Economic Forum, 2025. Accessed : December 2025.

[4] Microsoft Corporation. Microsoft digital defense report 2025. Technical report, Microsoft, October 2025. Sixth annual Digital Defense Report covering July 2024 to June 2025.

[5] Julian Jang-Jaccard and Surya Nepal. A survey of emerging threats in cybersecurity. *Journal of Computer and System Sciences*, 80(5) :973–993, 2014.

[6] Mouhamadou Lamine Diakhame, Chérif Diallo, and Mohamed Mejri. MCM-CASR : Novel alert correlation framework for cyber attack scenario reconstruction based on NLP, NER, and semantic similarity. In *2023 7th Cyber Security in Networking Conference (CSNet)*, pages 27–31. IEEE, 2023.

[7] Mouhamadou Lamine Diakhame, Chérif Diallo, and Mohamed Mejri. MCM-Llama : A fine-tuned large language model for real-time threat detection through security event correlation. In *Proceedings of the 2024 International Conference on Electrical, Computer and Energy Technologies (ICECET)*, pages 1–6. IEEE, 2024.

[8] Mouhamadou Lamine Diakhame, Chérif Diallo, and Mohamed Mejri. Adaptive cyber attack projection through context-driven model selection : Combining graph attention networks and reinforcement learning. In *2025 5th Intelligent Cybersecurity Conference (ICSC)*, pages 28–34. IEEE, 2025.

[9] Mouhamadou Lamine Diakhame, Chérif Diallo, and Mohamed Mejri. Machine learning-based alert correlation for enhanced cybersecurity : A multi-classification approach. In *2025 9th Cyber Security in Networking Conference (CSNet)*, pages 1–8. IEEE, 2025.

[10] Mouhamadou Lamine Diakhame and Chérif Diallo. LLM-driven alert correlation for automated incident response. Soumis à *2025 IEEE International Conference*

on Cyber Security and Resilience (CSR), conférence internationale avec comité de lecture, 2025. Article soumis – en cours d'évaluation.

[11] Mouhamadou Lamine Diakhame, Chérif Diallo, and Mohamed Mejri. Adaptive cyber attack projection through context-driven model selection : Combining graph attention networks and reinforcement learning. In *Colloque International 2025 : Sciences et Technologies comme moteur du Bien-Être, de la Transition Énergétique et Agroécologique*, Saint-Louis, Sénégal, May 2025. École Doctorale des Sciences et des Technologies (ED-ST), CEA-MITIC, Université Gaston Berger de Saint-Louis. Communication orale, 20–22 mai 2025.

[12] Yunus Emre Seyyar, Ali Gökhan Yavuz, and Halil Murat Ünver. An attack detection framework based on bert and deep learning. *IEEE Access*, 10 :68633–68644, 2022.

[13] Ali Ahmadian Ramaki, Morteza Amini, and Reza Ebrahimi Atani. A survey of IT early warning systems : Architectures, challenges, and solutions. *Security and Communication Networks*, 9(17) :4751–4776, 2016.

[14] Martina Pivarníková, Pavol Sokol, and Tomáš Bajtoš. Early-stage detection of cyber attacks. *Information*, 11(12) :560, 2020.

[15] Giovanni Apruzzese, Pavel Laskov, Edgardo Montes de Oca, Wissam Mallouli, et al. The role of machine learning in cybersecurity. *arXiv preprint arXiv :2206.09707*, 2022.

[16] Asmaa Halbouni, Teddy Surya Gunawan, Mohamed Hadi Habaebi, Mira Kartiwi, and Robiah Ahmad. Machine learning and deep learning approaches for cybersecurity : A review. *IEEE Access*, 10 :19572–19585, 2022.

[17] Anna L. Buczak and Erhan Guven. A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys & Tutorials*, 18(2) :1153–1176, 2016.

[18] Seyed Ali Mirheidari, Sajjad Arshad, and Rasool Jalili. Alert correlation algorithms : A survey and taxonomy. *arXiv preprint arXiv :1811.00921*, 2018.

[19] Igor Kotenko, Diana Gaifulina, and Igor Zelichenok. Systematic literature review of security event correlation methods. *IEEE Access*, 10 :43387–43420, 2022.

[20] Mouhamadou Lamine Diakhame, Diarra Bousso Dieng, Joshua Nougaret, and Jonathan Mwaura. Soil drainage classification using machine learning models : A comparative study. In *Pan-African Artificial Intelligence and Smart Systems Conference*, pages 213–226. Springer, 2024.

[21] Google Threat Intelligence Group. Hello 0-days, my old friend : A 2024 zero-day exploitation analysis. Technical report, Google, April 2025. Annual zero-day trends report.

[22] Yong-Ho Kim and Won Hyung Park. A study on cyber threat prediction based on intrusion detection event for APT attack detection. *Multimedia Tools and Applications*, 71(2) :685–698, 2014.

[23] Hornetsecurity Security Lab. Cybersecurity report 2026 : Monthly threat report. Technical report, Hornetsecurity, December 2025. Analysis of 72 billion emails processed.

[24] Optiv Global Threat Intelligence Center. First quarter 2025 ransomware trends. Technical report, Optiv, 2025.

[25] Shanchieh Jay Yang, Haitao Du, Jared Holsopple, and Moises Sudit. Attack projection. In *Cyber Defense and Situational Awareness*, pages 239–261. Springer, 2014.

[26] Darktrace. 2025 cyber threats : A mid-year review. Technical report, Darktrace, August 2025.

[27] DeepStrike. Zero-day exploit statistics 2025 : New baseline, new playbook, September 2025.

[28] Cybersecurity and Infrastructure Security Agency. Industrial control systems, 2025.

[29] Cybersecurity and Infrastructure Security Agency. National cybersecurity protection system, 2025. EINSTEIN intrusion detection capability.

[30] Commvault. Early warning systems (ews) explained, November 2025. Threat intelligence market estimated at $11.55 billion in 2025.

[31] Oualid Koucham, Stéphane Mocanu, Guillaume Hiet, Jean-Marc Thiriet, and Frédéric Majorczyk. Cross-domain alert correlation methodology for industrial control systems. *Computers & Security*, 118 :102723, July 2022.

[32] Nikolaos Polatidis, Elias Pimenidis, Michalis Pavlidis, Spyridon Papastergiou, and Haralambos Mouratidis. From product recommendation to cyber-attack prediction : Generating attack graphs and predicting future attacks. *Evolving Systems*, 11(3) :479–490, 2020.

[33] Eric M. Hutchins, Michael J. Cloppert, and Rohan M. Amin. Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. *Leading Issues in Information Warfare & Security Research*, 1(1) :80–106, 2011.

[34] SANS Institute. Cyber kill chain, MITRE ATT&CK, and purple team, June 2025.

[35] MITRE Corporation. MITRE ATT&CK - faq, 2025.

[36] SANS Institute. SANS ICS cyber kill chain, and the MITRE ATT&CK framework for ICS, 2025.

[37] Gradient Cyber. Emerging trends in cybersecurity detection and response 2025, March 2025.

[38] Jinyu Wu, Lihua Yin, and Yunchuan Guo. Cyber attacks prediction model based on Bayesian network. In *Proceedings of the International Conference on Parallel and Distributed Systems*, pages 730–731, 2012.

[39] V.S. Subrahmanian. Northwestern security and AI lab releases new terrorism early warning system forecasts, November 2024.

[40] Cyware. Cyware monthly threat intelligence, february 2025, March 2025.

[41] AIMultiple. Large language models in cybersecurity : Use cases and applications, 2024. Accessed : December 2024.

[42] Kai Wang, Yunfei Liu, Jian Zhang, Yongzhong Wang, and Cheng Zhao. Graph neural networks for cyber defensive operations : A survey. *arXiv preprint arXiv :2401.05680*, 2024.

[43] Dmitry Levshun and Igor Kotenko. A survey on artificial intelligence techniques for security event correlation : models, challenges, and opportunities. *Artificial Intelligence Review*, 56 :12651–12694, 2023.

[44] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[45] Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Nick Barnes, and Ajmal Mian. A comprehensive overview of large language models. *arXiv preprint arXiv :2307.06435*, 2023.

[46] Peng Sun, Xiaoping Yang, Xiangrui Zhao, and Zhongqing Wang. An overview of named entity recognition. In *Proceedings of the 2018 International Conference on Asian Language Processing (IALP)*, pages 273–278, 2018.

[47] Dhivya Chandrasekaran and Vijay Mago. Evolution of semantic similarity—a survey. *ACM Computing Surveys*, 54(2) :1–37, 2021.

[48] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33 :1877–1901, 2020.

[49] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama : Open and efficient foundation language models. *arXiv preprint arXiv :2302.13971*, 2023.

[50] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2 : Open foundation and fine-tuned chat models. *arXiv preprint arXiv :2307.09288*, 2023.

[51] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA : Low-rank adaptation of large language models. *arXiv preprint arXiv :2106.09685*, 2021.

[52] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *arXiv preprint arXiv :2210.11416*, 2022.

[53] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744, 2022.

[54] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474, 2020.

[55] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv :1609.02907*, 2017.

[56] Meetneura. A practical guide to graph neural networks for cybersecurity, 2024. Accessed : December 2024.

[57] Cagatay Catal, Huseyin Gunduz, and Ali Ozcan. Malware detection based on graph attention networks for intelligent transportation systems. *Electronics*, 10(20) :2534, 2021.

[58] Fei Xiao, Shuo Chen, Jie Yang, Hongmei He, Xiaoyi Jiang, Xiaobin Tan, and Dingding Jin. GRAIN : Graph neural network and reinforcement learning aided causality discovery for multi-step attack scenario reconstruction. *Computers & Security*, 148 :104180, 2025.

[59] Hao Lyu, Jing Liu, Yingxu Lai, Bing Mao, and Xinyu Huang. AGCM : A multi-stage attack correlation and scenario reconstruction method based on graph aggregation. *Computer Communications*, 224 :302–313, 2024.

[60] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv :1710.10903*, 2018.

[61] Vincent François-Lavet, Peter Henderson, Riashat Islam, Marc G. Bellemare, and Joelle Pineau. An introduction to deep reinforcement learning. *Foundations and Trends in Machine Learning*, 11(3–4) :219–354, 2018.

[62] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540) :529–533, 2015.

[63] Milad Dehghan, Babak Sadeghiyan, Ehsan Khosravian, Amir S. Moghaddam, and Farhad Nooshi. ProAPT : Projection of APT threats with deep reinforcement learning. *arXiv preprint arXiv :2209.07215*, 2022.

[64] Manuel Lopez-Martin, Belén Carro, and Antonio Sanchez-Esguevillas. Application of deep reinforcement learning to intrusion detection for supervised problems. *Expert Systems with Applications*, 141 :112963, 2020.

[65] Hado van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double Q-learning. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 2094–2100, 2016.

[66] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado van Hasselt, Marc Lanctot, and Nando de Freitas. Dueling network architectures for deep reinforcement learning. In *Proceedings of the 33rd International Conference on Machine Learning*, pages 1995–2003, 2016.

[67] Long-Ji Lin. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine Learning*, 8(3–4) :293–321, 1992.

[68] Nour Moustafa and Jill Slay. UNSW-NB15 : A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In *Proceedings of the Military Communications and Information Systems Conference (MilCIS)*. IEEE, 2015.

[69] Nour Moustafa and Jill Slay. The evaluation of network anomaly detection systems : Statistical analysis of the UNSW-NB15 dataset and the comparison with the KDD99 dataset. *Information Security Journal : A Global Perspective*, 25(1-3) :18–31, 2016.

[70] Ibrahim Ghafir, Konstantinos G. Kyriakopoulos, Sangarapillai Lambotharan, Francisco J. Aparicio-Navarro, Basheer Assadhan, Hamad Binsalleeh, and Diaa M. Diab. Hidden Markov models and alert correlations for the prediction of advanced persistent threats. *IEEE Access*, 7 :99508–99520, 2019.

[71] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP)*, pages 108–116, 2018.

[72] Cynthia Wagner, Alexandre Dulaunoy, Gérard Wagener, and Andras Iklody. MISP : The design and implementation of a collaborative threat intelligence sharing platform. In *Proceedings of the ACM Workshop on Information Sharing and Collaborative Security (WISCS)*, 2016.

[73] Markus Ring, Sarah Wunderlich, Deniz Scheuring, Dieter Landes, and Andreas Hotho. A survey of network-based intrusion detection data sets. *Computers & Security*, 86 :147–167, 2019.

[74] Ansam Khraisat, Iqbal Gondal, Peter Vamplew, and Joarder Kamruzzaman. Survey of intrusion detection systems : Techniques, datasets and challenges. *Cybersecurity*, 2(1) :1–22, 2019.

[75] Blake E. Strom, Andy Applebaum, Doug P. Miller, Kathryn C. Nickels, Adam G. Pennington, and Cody B. Thomas. MITRE ATT&CK : Design and philosophy. Technical report, The MITRE Corporation, 2018.

[76] Maryam Barzegar and Mehdi Shajari. Attack scenario reconstruction using intrusion semantics. *Expert Systems with Applications*, 108 :119–133, 2018.

[77] Bing Mao, Jing Liu, Yingxu Lai, and Meng Sun. MIF : A multi-step attack scenario reconstruction and attack chains extraction method based on multi-information fusion. *Computer Networks*, 198 :108340, 2021.

[78] Hao Hu, Jing Liu, Yuqing Zhang, Ying Liu, Xiaolong Xu, and Jianlong Tan. Attack scenario reconstruction approach using attack graph and alert data mining. *Journal of Information Security and Applications*, 54 :102522, 2020.

[79] Martina Pivarníková, Pavol Sokol, and Tomáš Bajtoš. Early-stage detection of cyber attacks. *Information*, 11(12) :560, 2020.

[80] D. Pramana Hostiadi, M. Darma Susila, and R. Rudolf Huizen. A new alert correlation model based on similarity approach. In *2019 1st International Conference on Cybernetics and Intelligent System (ICORIS)*, volume 1, pages 133–137. IEEE, 2019.

[81] Mehdi Khosravi and Behrouz Tork Ladani. Alerts correlation and causal analysis for APT based cyber attack detection. *IEEE Access*, 8 :162642–162656, 2020.

[82] Wei Wang, Peng Yi, Jiazhong Jiang, Peng Zhang, and Xiang Chen. Transformer-based framework for alert aggregation and attack prediction in a multi-stage attack. *Computers & Security*, 136 :103533, 2024.

[83] Nour Moustafa, Jiankun Hu, and Jill Slay. A holistic review of network anomaly detection systems : A comprehensive survey. *Journal of Network and Computer Applications*, 128 :33–55, 2019. Note : La référence originale mentionnée (geometric area analysis 2017) n'a pas été trouvée. Cette référence alternative de Moustafa et al. 2019 est plus complète et pertinente.

[84] Elizabeth D. Liddy. Natural language processing. In *Encyclopedia of Library and Information Science*. Marcel Dekker, Inc., New York, 2nd edition, 2001.

[85] Chris Alberti, Kenton Lee, and Michael Collins. A BERT baseline for the natural questions. *arXiv preprint arXiv :1901.08634*, 2019.

[86] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. PaLM : Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240) :1–113, 2023.

[87] Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv :2110.08207*, 2021.

[88] Seungone Kim, Se June Joo, Doyoung Kim, Joel Jang, Seonghyeon Ye, Jamin Shin, and Minjoon Seo. The CoT collection : Improving zero-shot and few-shot learning of language models via chain-of-thought fine-tuning. *arXiv preprint arXiv :2305.14045*, 2023.

[89] Marc Chalé, Neal D. Bastian, and Jason Weir. Algorithm selection framework for cyber attack detection. In *Proceedings of the 2nd ACM Workshop on Wireless Security and Machine Learning*, pages 37–42, 2020.

[90] Y. J. Cruz, M. Rivas, R. Quiza, Rodolfo E. Haber, F. Castaño, and A. Villalonga. A two-step machine learning approach for dynamic model selection : A case study on a micro milling process. *Computers in Industry*, 143 :103764, 2022.

[91] Fredrik Valeur, Giovanni Vigna, Christopher Kruegel, and Richard A. Kemmerer. A comprehensive approach to intrusion detection alert correlation. *IEEE Transactions on Dependable and Secure Computing*, 1(3) :146–169, 2004.

[92] Hervé Debar and Andreas Wespi. Aggregation and correlation of intrusion-detection alerts. In *Proceedings of the International Workshop on Recent Advances in Intrusion Detection (RAID)*, pages 85–103. Springer, 2001.

[93] Steven J. Templeton and Karl Levitt. A requires/provides model for computer attacks. In *Proceedings of the 2000 Workshop on New Security Paradigms*, pages 31–38. ACM, 2000.

[94] Alfonso Valdes and Keith Skinner. Probabilistic alert correlation. In *Proceedings of the International Workshop on Recent Advances in Intrusion Detection (RAID)*, pages 54–68. Springer, 2001.

[95] Peng Ning, Yun Cui, and Douglas S. Reeves. Constructing attack scenarios through correlation of intrusion alerts. In *Proceedings of the 9th ACM Conference on Computer and Communications Security*, pages 245–254. ACM, 2002.

[96] Benjamin Morin, Ludovic Mé, Hervé Debar, and Mireille Ducassé. M2D2 : A formal data model for IDS alert correlation. In *Proceedings of the International Workshop on Recent Advances in Intrusion Detection (RAID)*, pages 115–137. Springer, 2002.

[97] D. P. Gaikwad and Ravindra C. Thool. Intrusion detection system using bagging ensemble method of machine learning. In *Proceedings of the 2015 International Conference on Computing Communication Control and Automation*, pages 291–295. IEEE, 2015.

[98] Giovanni Apruzzese, Michele Colajanni, Luca Ferretti, Alessandro Guido, and Mirco Marchetti. On the effectiveness of machine and deep learning for cyber security. In *Proceedings of the 2018 10th International Conference on Cyber Conflict (CyCon)*, pages 371–390. IEEE, 2018.

[99] Carol V. Zhou, Christopher Leckie, and Shanika Karunasekera. A survey of coordinated attacks and collaborative intrusion detection. *Computers & Security*, 29(1) :124–140, 2010.

[100] Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, pages 4765–4774, 2017.

[101] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should I trust you ? explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144. ACM, 2016.

[102] Yarame Niang. Réponse automatisée aux incidents de sécurité par l'intégration SIEM-SOAR-IA : Conception et implémentation d'un système intelligent. Mémoire de master mart, Université Gaston Berger, Saint-Louis, Sénégal, 2025.

[103] Xinzhou Qin and Wenke Lee. Attack plan recognition and prediction using causal networks. In *Proceedings of the 20th Annual Computer Security Applications Conference (ACSAC)*, pages 370–379, 2004.

[104] Igor Kotenko, Diana Gaifulina, and Ilya Zelichenok. Systematic literature review of security event correlation methods. *IEEE Access*, 10 :43976–43998, 2022.

[105] Robin Sommer and Vern Paxson. Outside the closed world : On using machine learning for network intrusion detection. In *2010 IEEE Symposium on Security and Privacy*, pages 305–316, 2010.

[106] El Mostapha Chakir, Youssef Idrissi Khamlichi, and Mohammed Moughit. Intrusion detection systems : Reducing false positives through event correlation. *International Journal of Computer Science Issues*, 20(1), 2023.

[107] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT : Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pages 4171–4186, 2019.

[108] Hao Sun, Shuo Shen, Jeremy Sprague, Yiming Deng, Adeel Sohail, and Ke Tao. Large language models for cyber security : A systematic literature review. *arXiv preprint arXiv :2405.04760*, 2024.

[109] Mohamed Amine Ferrag, Mthandazo Ndhlovu, Norbert Tihanyi, Lucas C. Cordeiro, Merouane Debbah, and Thierry Lestable. Revolutionizing cyber threat detection

with large language models : A comprehensive review, generative AI, challenges, and future directions. *arXiv preprint arXiv :2306.14263*, 2023.

[110] Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. TinyLlama : An open-source small language model. *arXiv preprint arXiv :2401.02385*, 2024.

[111] Daniele Antonioli and Nils Ole Tippenhauer. Security orchestration, automation and response engine for deployment of behavioural countermeasures. In *Proceedings of the 25th International Symposium on Research in Attacks, Intrusions and Defenses (RAID)*, 2022.

[112] Paul Cichonski, Tom Millar, Tim Grance, and Karen Scarfone. Computer security incident handling guide. Technical Report Special Publication 800-61 Revision 2, National Institute of Standards and Technology, 2012.

[113] David R. Miller, Shon Harris, Allen Harper, Stephen VanDyke, and Chris Blask. *Security Information and Event Management (SIEM) Implementation*. McGraw-Hill, 2011.

[114] Blake E. Strom, Andy Applebaum, Doug P. Miller, Kathryn C. Nickels, Adam G. Pennington, and Cody B. Thomas. MITRE ATT&CK : Design and philosophy. Technical report, The MITRE Corporation, 2018.

[115] Christoph Molnar. *Interpretable Machine Learning*. Leanpub, 2023.

[116] FortiGuard Labs. Global threat landscape report 2025. Technical report, Fortinet, 2025.

[117] Trend Micro. Trend 2025 cyber risk report. Technical report, Trend Micro, 2025.

[118] CSO Online. Top 7 zero-day exploitation trends of 2024, May 2025.

[119] Patrick Schläpfer. Reviewing zero-day vulnerabilities exploited in malware campaigns in 2024, February 2025.

[120] Cloud Security Alliance. What cyber threats are emerging for 2025 ?, October 2024.

[121] Oliver Dain and Robert K. Cunningham. Fusing a heterogeneous alert stream into scenarios. In *Proceedings of the 2001 ACM Workshop on Data Mining for Security Applications*, pages 1–13. ACM, 2001.

[122] Gartner Inc. Market guide for security orchestration, automation and response solutions. Technical Report G00750867, Gartner Research, 2022.

[123] IBM Security. Cost of a data breach report 2023. Technical report, IBM Corporation, 2023.

[124] WJAETS. Large language models (llms) in cybersecurity : A systematic review. *World Journal of Advanced Engineering Technology and Sciences*, 2024. Revue systématique sur l'utilisation des LLM en cybersécurité.

[125] Stanford HAI. Artificial intelligence index report 2025. Technical report, Stanford University - Human-Centered AI Institute, 2025. Rapport annuel sur l'état de l'IA, incluant applications en cybersécurité.

[126] Accenture Security. State of cybersecurity 2025. Technical report, Accenture, 2025. Rapport sur l'état de la cybersécurité et tendances 2025.

[127] Advens. Advens threat report 2025. Technical report, Advens, 2025. Rapport français sur les menaces cyber 2025.

[128] Capgemini Research Institute. Trends in cybersecurity 2025. Technical report, Capgemini, 2025. Tendances en cybersécurité pour 2025.

[129] SuperAGI. Ai-driven malware detection : How machine learning is revolutionizing customer data security in 2025, 2025. Article sur la détection de malware par IA.

[130] DPO India. Ai threat landscape report 2025. Technical report, Data Protection Organization India, 2025. Rapport sur les menaces liées à l'IA.

[131] Lakera AI. Ai security trends 2025, 2025. Tendances en sécurité de l'IA.

[132] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. Technical report, OpenAI, 2019.

[133] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140) :1–67, 2020.

[134] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa : A robustly optimized BERT pretraining approach. *arXiv preprint arXiv :1907.11692*, 2019.

[135] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv :1301.3781*, 2013.

[136] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe : Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[137] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, et al. Transformers : State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing : System Demonstrations (EMNLP)*, pages 38–45, 2020.

[138] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8) :1735–1780, 1997.

[139] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations

using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, 2014.

[140] William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

[141] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks ? In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

[142] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1) :4–24, 2021.

[143] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 1263–1272, 2017.

[144] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning : An Introduction*. MIT Press, 2nd edition, 2018.

[145] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, et al. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529 :484–489, 2016.

[146] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv :1707.06347*, 2017.

[147] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[148] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553) :436–444, 2015.

[149] Guansong Pang, Chunhua Shen, Longbing Cao, and Anton van den Hengel. Deep learning for anomaly detection : A review. *ACM Computing Surveys*, 54(2) :1–38, 2021.

[150] Zeeshan Ahmad, Adnan Shahid Khan, Cheah Wai Shiang, Johari Abdullah, and Farhan Ahmad. Network intrusion detection system : A systematic study of machine learning and deep learning approaches. *Transactions on Emerging Telecommunications Technologies*, 32(1) :e4150, 2021.

[151] Mohamed Amine Ferrag, Leandros Maglaras, Sotiris Moschoyiannis, and Helge Janicke. Deep learning for cyber security intrusion detection : Approaches, datasets, and comparative study. *Journal of Information Security and Applications*, 50 :102419, 2020.

[152] Mohammed A. Al-Garadi, Amr Mohamed, Abdulla K. Al-Ali, Xiaojiang Du, Ihsan Ali, and Mohsen Guizani. A survey of machine and deep learning methods for Internet of Things (IoT) security. *IEEE Communications Surveys & Tutorials*, 22(3) :1646–1685, 2020.

[153] Nathan Shone, Tran Nguyen Ngoc, Vu Dinh Phai, and Qi Shi. A deep learning approach to network intrusion detection. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2(1) :41–50, 2018.

[154] Karen Scarfone and Peter Mell. Guide to intrusion detection and prevention systems (IDPS). Special Publication 800-94, National Institute of Standards and Technology (NIST), 2007.

[155] Richard P. Lippmann, David J. Fried, Isaac Graf, Joshua W. Haines, Kristin R. Kendall, David McClung, Dan Weber, Seth E. Webster, Dan Wyschogrod, Robert K. Cunningham, and Marc A. Zissman. Evaluating intrusion detection systems : The 1998 DARPA off-line intrusion detection evaluation. In *Proceedings of the DARPA Information Survivability Conference and Exposition (DISCEX)*, volume 2, pages 12–26. IEEE, 2000.

[156] Stefan Axelsson. The base-rate fallacy and the difficulty of intrusion detection. *ACM Transactions on Information and System Security (TISSEC)*, 3(3) :186–205, 2000.

[157] Frédéric Cuppens and Alexandre Miège. Alert correlation in a cooperative intrusion detection framework. In *2002 IEEE Symposium on Security and Privacy*, pages 202–215. IEEE, 2002.

[158] Peng Ning, Yun Cui, Douglas S. Reeves, and Dingbang Xu. Techniques and tools for analyzing intrusion alerts. *ACM Transactions on Information and System Security (TISSEC)*, 7(2) :274–318, 2004.

[159] Reza Sadoddin and Ali A. Ghorbani. Alert correlation survey : Framework and techniques. In *Proceedings of the 2006 International Conference on Privacy, Security and Trust (PST)*. ACM, 2006.

[160] Mohamed Amine Ferrag, Mthandazo Ndhlovu, Norbert Tihanyi, Lucas C. Cordeiro, Merouane Debbah, Thierry Lestable, and Narinderjit Singh Thandi. Revolutionizing cyber threat detection with large language models : A privacy-preserving BERT-based lightweight model for IoT/IIoT devices. *arXiv preprint arXiv :2306.14263*, 2023.

[161] Ehsan Aghaei, Xi Niu, Waseem Shadid, and Ehab Al-Shaer. SecureBERT : A domain-specific language model for cybersecurity. In *Proceedings of the 18th EAI International Conference on Security and Privacy in Communication Networks (SecureComm)*, pages 39–56. Springer, 2023.

[162] Hanxiang Xu, Shenao Wang, Ningke Li, Kailong Wang, Yanjie Zhao, Kai Chen, Ting Yu, Yang Liu, and Haoyu Wang. Large language models for cyber security : A systematic literature review. *ACM Transactions on Software Engineering and Methodology*, 2024.

[163] Nour Moustafa and Jill Slay. Big data analytics for intrusion detection system : statistical decision-making using finite dirichlet mixture models. In Ivan Palomares and Francisco Martínez-Álvarez, editors, *Data Analytics and Decision Support for Cybersecurity : Trends, Methodologies and Applications*, pages 127–156. Springer International Publishing, Cham, 2017.

[164] Mohanad Sarhan, Siamak Layeghy, Nour Moustafa, and Marius Portmann. Netflow datasets for machine learning-based network intrusion detection systems. In Jaideep Kim, Kyuseok Shim, Longbing Cao, Jae-Gil Lee, Xuemin Lin, and Yang-Sae Moon, editors, *Big Data Technologies and Applications*. Springer International Publishing.