RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université de Mohamed El-Bachir El-Ibrahimi - BBA

Faculté des Mathématiques et Informatiques

MÉMOIRE

Présenté en vue de l'obtention du diplôme

Master en Informatique

Spécialité : Ingénierie de l'Informatique Décisionnelle

Thème

Détection Intelligente des Anomalies et Menaces Cybernétiques dans l'IoBT à l'aide du Deep Learning

Présenté par:

- Chaima CHEIKH
- Yousra Kawther GUISSOUS

Soutenu le: 11/06/2025, devant la commission d'examen suivante:

Dr. Salah MAACHE	Enseignant à l'unversité de BBA	Président
Dr. Hassina BENSEFIA	Enseignante à l'unversité de BBA	Examinatrice
Dr. Oussama SENOUCI	Enseignant à l'unversité de BBA	Encadrant

Promotion 2024 / 2025

**“Je n’ai jamais rêvé de succès,
J’ai travaillé pour ça.”**

- Estee Lauder -

Remerciements

Louange à **Dieu**, Le Tout-Puissant, pour la santé, le courage, la patience et la volonté qu'Il nous a accordés tout au long de ce travail.

Nous exprimons nos sincères remerciements à notre encadrant, **Dr. Senouci Oussama**, de l'Université Mohamed El-Bachir El-Ibrahimi de Bordj Bou Arreridj, Algérie, pour son accompagnement attentif, sa disponibilité constante et la richesse de ses conseils, qui ont grandement contribué à la qualité de ce mémoire.

ous adressons nos remerciements les plus sincères et les plus respectueux aux membres du jury, Monsieur Salah Maâch et Madame Hassina Bensefia et, pour l'honneur qu'ils nous font en acceptant d'évaluer ce travail.

Nos remerciements vont aussi à l'ensemble de nos enseignants et aux cadres pédagogiques de l'Université Mohamed El-Bachir El-Ibrahimi, en particulier ceux de la Faculté des Mathématiques et Informatique, pour leurs efforts constants dans le développement de nos compétences.

Un merci chaleureux à nos chers parents, pour leur amour inconditionnel, leurs sacrifices, leur soutien indéfectible et leur confiance en nous. Leur présence a été une source inestimable de force et de motivation.

Nous exprimons également notre gratitude à nos frères, sœurs, ami(e)s, et à toutes les personnes qui, de près ou de loin, ont contribué à la réalisation de ce travail.

Enfin, que toutes les personnes qui nous sont chères trouvent ici l'expression de notre profonde reconnaissance.

Dédicace

À mes parents, piliers de tous mes systèmes instables,

Vous qui avez toujours cru en moi, même quand mes lignes de code refusaient de le faire. Merci pour votre amour inconditionnel, vos encouragements silencieux mais puissants, Et pour avoir été la source constante d'une énergie... bien plus stable que celle de mon laptop.

À ma sœur,

Merci d'avoir été mon refuge discret et mon énergie douce dans les moments les plus instables.

À la Nuit, bien sûr – pas celle romantique des poétesses,

Mais la mienne : un mix de lampe torche, de clavier luminescent

Et de videoprocs qui dansaient comme des lucioles schizophrènes.

« Merci de m'avoir laissée croire que 3h du mat était un fuseau horaire valide. »

Et à mes amies, Yousra, Amel, Merry, Ala et toutes celles qui ont été là sans jamais compter les heures,

Merci pour les éclats de rire qui fendaient le stress comme une fonction bien factorisée, Pour les pauses-café qui ressemblaient à des thérapies express, Et pour vos mots simples mais essentiels : "Tu vas y arriver."

⋆ ***Chaima*** ⋆

Dédicace

Ce travail est dédié à ceux qui ont été mes forces invisibles tout au long de ce parcours.

À ***mes parents****, dont l'amour sans faille et les sacrifices constants ont fait de moi la personne que je suis aujourd'hui. Vous êtes mes modèles et mes guides, et chaque pas que je fais est le reflet de vos valeurs.*

À mes ***frères*** *et* ***sœurs*** *et à toute* ***ma famille****,* ***mes tantes****,* ***Rima*** *et* ***Fariel****, pour leur soutien constant, leur patience et leurs encouragements. Votre présence a été mon moteur dans les moments difficiles.*

À mes amies précieuses, ***Amani****,* ***Amira****,* ***Soundous****,* ***Chaima****,* ***Amel*** *pour vos rires et votre soutien inconditionnel. Cette réussite est aussi la vôtre.*

À toutes celles et ceux qui m'ont permis de croire en moi 07.06, je vous remercie du fond du cœur.

⋆ ***Yousra Kawther*** ⋆

Résumé

Ce mémoire s'inscrit dans le champ stratégique de la cybersécurité militaire, en explorant le potentiel de l'Intelligence Artificielle (IA), de l'apprentissage automatique et de l'apprentissage profond pour renforcer la sécurité et l'autonomie des systèmes connectés dans les réseaux Internet of Battlefield Things (IoBT). Après avoir mis en évidence les limites des protocoles traditionnels face aux menaces sophistiquées, nous avons étudié et optimisé le protocole BioBT, dédié à l'authentification biométrique des entités IoBT et à la garantie de l'intégrité des données au sein du système C3I (Commandement, Contrôle, Communications et Renseignement). L'amélioration proposée repose sur l'intégration d'un modèle LSTM (Long Short-Term Memory), une architecture de réseau neuronal récurrent particulièrement adaptée à l'analyse de séquences temporelles. Ce modèle apporte une seconde couche d'authentification dynamique, venant compléter la vérification initiale assurée par la blockchain. En simulant des comportements biométriques issus de dispositifs IoBT, le système est capable de détecter les anomalies et d'authentifier les objets de manière fiable et réactive. Les résultats expérimentaux confirment la robustesse de l'approche : le modèle LSTM atteint une précision de 98% dans la tâche d'authentification, tout en détectant 77.94% des tentatives d'intrusion simulées. De plus, la latence moyenne d'inférence reste compatible avec les exigences temps réel des environnements tactiques. Ces performances démontrent l'efficacité de la combinaison entre IA, biométrie et blockchain pour sécuriser les infrastructures militaires intelligentes, et ouvrent la voie à des déploiements opérationnels de plus grande envergure.

Mots-clés : Intelligence Artificielle, Apprentissage Automatique, Apprentissage Profond, LSTM, Blockchain, Authentification Biométrique, IoBT, C3I, Cybersécurité.

Abstract

This thesis is positioned within the strategic domain of military cybersecurity, exploring the potential of Artificial Intelligence (AI), Machine Learning (ML), and Deep Learning (DL) to enhance the security and autonomy of connected systems in Internet of Battlefield Things (IoBT) networks. After identifying the limitations of traditional protocols in the face of increasingly sophisticated threats, we studied and optimized the BioBT protocol, which is dedicated to biometric authentication of IoBT entities and ensuring data integrity within the C3I system (Command, Control, Communications, and Intelligence). The proposed enhancement relies on the integration of a Long Short-Term Memory (LSTM) model, a type of recurrent neural network architecture particularly suited to the analysis of temporal sequences. This model introduces a second layer of dynamic authentication, complementing the initial blockchain-based verification. By simulating biometric behaviors from IoBT devices, the system is capable of reliably detecting anomalies and authenticating objects in real time. Experimental results confirm the robustness of the proposed approach: the LSTM model achieved 98% accuracy in biometric authentication and detected 77.94% of simulated intrusion attempts. Furthermore, the average inference latency remains well within the real-time constraints of tactical environments. These performances demonstrate the effectiveness of combining AI, biometrics, and blockchain technologies to secure intelligent military infrastructures and pave the way for broader operational deployment.

Keywords: Artificial Intelligence, Machine Learning, Deep Learning, LSTM, Blockchain, Biometric Authentication, IoBT, C3I, Cybersecurity.

ملخص

تندرج هذه المذكرة ضمن المجال الاستراتيجي للأمن السيبراني العسكري، حيث تستكشف إمكانيات الذكاء الاصطناعي ، التعلم الآلي، والتعلم العميق، في تعزيز أمن واستقلالية الأنظمة المتصلة داخل شبكات إنترنت الأشياء في ساحة المعركة IoBT. بعد تحليل حدود البروتوكولات التقليدية في مواجهة التهديدات المعقدة، قمنا بدراسة وتحسين بروتوكول BioBT المخصص للمصادقة البيومترية لكيانات IoBT وضمان سلامة البيانات داخل نظام القيادة والسيطرة والاتصالات والاستخبارات C3I.

يرتكز التحسين المقترح على دمج نموذج LSTM (الذاكرة طويلة وقصيرة الأمد)، وهو نوع متقدم من الشبكات العصبية المتكررة، يُستخدم بكفاءة لتحليل السلاسل الزمنية. يسمح هذا النموذج بإضافة طبقة ثانية من المصادقة الديناميكية تُكمل التحقق الأولي الذي توفره تقنية البلوكشين. ومن خلال محاكاة سلوكيات بيومترية صادرة عن أجهزة IoBT، يُمكِّن النظام من التمييز بين الكيانات السليمة والمشبوهة بسرعة ودقة.

وقد أكدت النتائج التجريبية فعالية النهج المقترح، حيث بلغ معدل الدقة ٩٨٪ في مهمة المصادقة البيومترية، كما تم الكشف عن ٧٧.٩٤٪ من محاولات التسلل المحاكية. علاوة على ذلك، أظهرت فترة الاستجابة الزمنية (زمن الاستدلال) توافقًا جيدًا مع متطلبات البيئات التكتيكية في الزمن الحقيقي. تعكس هذه النتائج القوة التكاملية بين الذكاء الاصطناعي، والبيومتريات، والبلوكشين في تأمين البنية التحتية للأنظمة العسكرية الذكية، وتمهّد الطريق نحو نشرها على نطاق أوسع في تطبيقات ميدانية واقعية.

الكلمات المفتاحية: الذكاء الاصطناعي، التعلم الآلي، التعلم العميق، LSTM، البلوكشين، المصادقة البيومترية، IoBT، نظام C3I، الأمن السيبراني.

Table of contents

Liste des figures **13**

Liste des tables **14**

Abréviations **15**

Introduction Générale **1**

1 Généralités sur le Contexte **5**

1.1 Introduction 5

1.2 Contexte Militaire et Défense Moderne 5

1.2.1 Évolution des Conflits et des Menaces Militaires 6

1.2.1.1 Nature des Conflits Modernes 6

1.2.1.2 Menaces Actuelles et Futures 6

1.2.2 Technologies Employées dans les Systèmes Militaires 7

1.2.2.1 Capteurs et Systèmes de Surveillance 7

1.2.2.2 Systèmes de Communication et de Commandement . . 9

1.2.2.3 Systèmes de Défense Aérienne et Maritime 9

1.2.3 Défis Actuels en Matière de Défense 10

1.2.4 Transition des Approches de Défense : Des Systèmes Traditionnels aux Systèmes Intelligents 12

1.3 Introduction du Deep Learning dans le Domaine Militaire 14

1.3.1 Définition et Concepts Clés du Deep Learning 14

1.3.1.1 Artificial Neural Network (ANN) 14

1.3.1.2 Deep Neural Network (DNN) 15

1.3.1.3 Convolutional Neural Network (CNN) 15

1.3.1.4 Recurrent Neural Network (RNN) 16
1.3.1.5 Long Short-Term Memory (LSTM) 16
1.3.2 Applications du Deep Learning dans le Secteur Militaire 16
1.3.2.1 Analyse de Données de Surveillance 17
1.3.2.2 Reconnaissance d'Images et Détection de Cibles 17
1.3.2.3 Détection Automatique d'Anomalies 18
1.3.3 Avantages du Deep Learning pour la Défense 18
1.3.3.1 Capacité à Traiter de Grandes Masses de Données . . 18
1.3.3.2 Précision et Réduction des Faux Positifs 19
1.3.3.3 Automatisation et Réactivité en Temps Réel 19
1.4 Conclusion 19

2 État de l'art **21**
2.1 Introduction 21
2.2 Méthodologie et Objectifs de Recherche 21
2.2.1 Objectifs de Recherche 22
2.2.2 Méthodes et Outils de Recherche 22
2.2.2.1 Recherche bibliographique systématique 22
2.2.2.2 Outils d'analyse et de traitement 23
2.2.3 Critères de sélection des articles 24
2.2.4 Mots-clés utilisés 24
2.3 Classification des Protocoles de Détection des Menaces Classiques . . . 25
2.3.1 Protocoles Basés sur la Détection d'Anomalies 26
2.3.1.1 Détection d'intrusion basée sur les anomalies sensibles au protocole (ABID) 26
2.3.1.2 Détection d'anomalies dans les systèmes MIL-STD-1553 27
2.3.1.3 ADMOST : Détection et correction d'anomalies en ligne 28
2.3.2 Protocoles de Surveillance et de Sécurité des Réseaux 28
2.3.2.1 BIoBT : Authentification autonome par blockchain dans l'IoBT 28
2.3.2.2 Système de défense active pour réseaux critiques . . . 28
2.3.2.3 Sécurité des réseaux de capteurs sans fil militaires . . . 29
2.3.2.4 Surveillance intrinsèque des fibres optiques 29

2.3.3 Techniques Traditionnelles de Cyberdéfense 29
2.3.3.1 Pare-feu (Firewall) 29
2.3.3.2 NetFlow 30
2.3.3.3 SNMP (Simple Network Management Protocol) 30
2.4 Limites des Protocoles Classiques 30
2.4.1 Manque de Réactivité et d'Adaptabilité 31
2.4.2 Limites dans les Scénarios Complexes 31
2.4.3 Fiabilité et Faux Positifs 32
2.5 Apport de l'Intelligence Artificielle dans les Systèmes de Détection des Menaces 32
2.5.1 Avancées des Modèles de Machine Learning pour la Sécurité Militaire 32
2.5.1.1 Apprentissage Supervisé et Non Supervisé : 33
2.5.1.2 Deep Learning: 34
2.5.1.3 Avantages par Rapport aux Protocoles Classiques . . . 36
2.5.2 Intelligence Artificielle pour l'Optimisation des Stratégies de Défense 37
2.5.3 Applications Pratiques et Perspectives Futures 39
2.5.3.1 Applications Pratiques 39
2.5.3.2 Perspectives Futures 43
2.6 Conclusion 44

3 Protocole Blockchain-Based Autonomous Authentication and Integrity for IoBT in C3I System (BioBT) 45
3.1 Introduction 45
3.2 Protocole BIoBT 46
3.3 Architecture de BIoBT dans le système C3I 46
3.3.1 Acteurs clés 46
3.3.2 Composants connectés 46
3.3.3 Vue abstraite du système C3I 47
3.3.4 Symboles et leurs définitions 47
3.3.5 Phases Techniques Détaillées de BIoBT 47
3.3.6 Limitations de BIoBT 55
3.3.6.1 Confidentialité sur une Blockchain Publique 56

3.3.6.2 Absence de Validation à Grande Échelle 57
3.3.6.3 Forte Dépendance à Ethereum 57
3.3.6.4 Complexité d'Intégration dans l'Infrastructure Militaire 57
3.3.6.5 Risques de Centralisation de l'Autorité CCC 58
3.4 Conclusion 58

4 Amélioration du Protocole BioBT par un Modèle LSTM Basé sur le Deep Learning pour l'Authentification Biométrique 59
4.1 Introduction 59
4.2 Motivations et Objectifs 60
4.2.1 Renforcement de la Résilience Sécuritaire dans le Contexte IoBT 60
4.2.2 Fusion Biométrie Temporelle et Blockchain 60
4.2.3 Réduction des Taux d'Erreur et Précision d'Authentification . . 60
4.2.4 Optimisation pour Dispositifs Contraints et Environnements Tactiques 61
4.2.5 Objectifs Stratégiques de l'Intégration du Deep Learning 61
4.3 Exploration, Analyse et Prétraitement des Données 62
4.3.1 Description du Jeu de Données 62
4.3.2 Visualisation et Corrélation des Données 63
4.3.3 Prétraitement Spécifique aux Réseaux LSTM 64
4.4 Architecture du Modèle LSTM pour l'Authentification Biométrique . . 66
4.4.1 Structure du Réseau Séquentiel 67
4.4.2 Hyperparamètres du Modèle 68
4.4.3 Régularisation et Stratégies d'Entraînement 68
4.4.4 Fonction de Perte Utilisée 70
4.4.5 Métriques d'Évaluation du Modèle LSTM 70
4.5 Déploiement Simulé du Modèle LSTM dans un Réseau IoBT Sécurisé . 76
4.5.1 Paramètres de la Simulation 77
4.5.2 Déploiement Réseau et Visualisation Initiale 77
4.5.3 Création du système CCC_Sys sur la Blockchain 78
4.5.4 Association des objets IoBT au CCC_Sys 80
4.5.5 Authentification et Transmission des Données 81
4.5.6 Authentification Biométrique par le Modèle LSTM 83

4.5.7 Visualisation des Résultats de l'Authentification LSTM 86
4.5.8 Révocation d'une entité IoBT 89
4.6 Conclusion 90

Conclusion Générale **91**

Annexe **93**

Bibliographie **119**

List of Figures

2.1 Classification des protocoles conventionnels de détection et prévention des menaces. 27

3.1 Composants du système BIoBT [1]. 47
3.2 Vue abstraite du système C3I [1]. 47
3.3 Architecture de création du système `CCC_Sys` 50
3.4 Processus d'association sécurisée entre IoBT et `CCC_Sys` 52
3.5 Authentification et validation d'intégrité des données par la blockchain avant acceptation par le `CCC_Sys`. 54
3.6 Suppression sécurisée d'un IoBT du registre `CCC_Sys` via révocation. . . 56

4.1 Aperçu des cinq premières lignes du jeu de données 64
4.2 Matrice de corrélation. 65
4.3 Évolution de la précision d'entraînement et de validation au cours des époques. 72
4.4 Évolution de la perte d'entraînement et de validation au cours des époques. 73
4.5 Matrice de confusion illustrant les performances de classification entre IoBT authentifiés (1) et non authentifiés (0). 74
4.6 Topologie initiale des nœuds IoBT 79
4.7 Distribution des probabilités d'authenticité prédites par le modèle LSTM. 86
4.8 Décision d'authentification biométrique (LSTM) sur les 68 nœuds validés par la blockchain 87
4.9 Temps de latence d'inférence par nœud IoBT (en millisecondes). 88

List of Tables

1.1 Comparaison des approches de défense 13

3.1 Symboles utilisés dans le système Biobot. 48

4.1 Statistiques descriptives de quelques attributs du dataset. 63
4.2 Extrait des corrélations fortes identifiées. 64
4.3 Architecture du modèle LSTM. 68
4.4 Hyperparamètres utilisés pour l'entraînement du modèle LSTM 68
4.5 Rapport de Classification du Modèle LSTM 76
4.6 Paramètres de simulation du protocole BIoBT amélioré 78
4.7 Exemples d'association des objets IoBT au `CCC_Sys` 81

Abréviations

- **SC** : Smart Contract.
- **CCC**: Command and Control Center.
- **LIDAR** : Light Detection and Ranging.
- **C3I** : Command, Control, Communications and Intelligence.
- **IoT** : Internet of Things.
- **IoBT** : Internet of Battlefield Things.
- **CNN** : Convolutional Neural Network.
- **LSTM** : Long Short-Term Memory.
- **ML** : Machine Learning.
- **DL** : Deep Learning.
- **AI** : Artificial Intelligence.
- **RL** : Reinforcement Learning.
- **BioBT**: Blockchain-Based Autonomous Authentication and Integrity for IoBT in C3I System.

Introduction Générale

Les conflits modernes ont profondément évolué, mettant en lumière la complexité croissante des menaces militaires dans un monde interconnecté et technologiquement avancé. Face à cette réalité, les approches traditionnelles de défense peinent à répondre efficacement aux attaques hybrides, aux cybermenaces et aux enjeux tactiques imprévisibles. Dans ce contexte, l'intégration de technologies émergentes telles que l'intelligence artificielle (IA) et le Deep Learning (DL) dans les systèmes de défense devient une nécessité stratégique.

Ce mémoire s'inscrit dans cette dynamique d'innovation technologique, en explorant l'apport du Deep Learning, notamment des réseaux LSTM, dans la détection des menaces militaires et l'optimisation des stratégies de défense. Nous analysons également l'architecture d'un protocole d'authentification biométrique basé sur la blockchain dans un environnement IoBT (Internet of Battlefield Things), afin d'illustrer l'interopérabilité entre plusieurs technologies de pointe au service de la sécurité militaire.

Problématique & Motivations

Avec l'essor des technologies et l'augmentation des dispositifs connectés dans les environnements de sécurité critique (IoBT), il devient essentiel de garantir la sécurité et l'intégrité des systèmes. L'authentification biométrique, en utilisant des données biologiques uniques, apparaît comme une solution prometteuse pour sécuriser ces systèmes, mais elle pose des défis, notamment en raison du traitement de grandes quantités de données en temps réel.

Ce mémoire explore l'utilisation des modèles LSTM (Long Short-Term Memory) pour l'authentification biométrique dans des systèmes IoBT, en évaluant leur efficacité pour offrir des solutions de sécurité robustes. L'objectif est de démontrer comment ces

modèles peuvent garantir une authentification précise et fiable, tout en fournissant des recommandations pour leur application dans des scénarios réels.

Objectifs & Contributions

Ce mémoire a pour objectif de développer une solution d'authentification biométrique adaptée aux exigences sécuritaires de l'Internet des objets de champ de bataille (IoBT), en s'appuyant sur les réseaux de neurones LSTM. Dans un contexte où les dispositifs militaires connectés sont vulnérables à diverses attaques, l'utilisation de données biométriques, combinée à la puissance des modèles LSTM pour le traitement des séquences temporelles, constitue une approche pertinente pour garantir une identification fiable et sécurisée. Le travail consiste à analyser les approches existantes, concevoir un modèle LSTM, l'entraîner sur des données biométriques réelles, et évaluer ses performances en termes de précision, de robustesse et d'efficacité. Les contributions principales de ce mémoire incluent la proposition d'une architecture d'authentification intelligente, l'évaluation expérimentale de son efficacité, et des recommandations pratiques pour son intégration dans les systèmes IoBT. Cette étude vise ainsi à renforcer la sécurité des environnements militaires en exploitant les avancées de l'intelligence artificielle appliquée à la biométrie.

Organisation du mémoire

Le mémoire est organisé en quatre chapitres distincts, chacun traitant d'un aspect spécifique de notre travail :

- **Chapitre 1: Généralités sur le Contexte**

  Nous exposons le cadre théorique et stratégique dans lequel s'inscrit notre travail. Ce chapitre met en lumière l'évolution des menaces militaires, la complexité croissante des conflits modernes, ainsi que les nouveaux défis que doivent relever les systèmes de défense. Il présente également les technologies émergentes, telles que les capteurs, les drones, les systèmes de communication sécurisés, ainsi que l'intégration progressive de l'intelligence artificielle (IA) et du Deep Learning (DL) dans les environnements militaires. Une attention particulière est accordée

au concept d'Internet des objets de champ de bataille (IoBT) et au système C3I, qui constituent le socle technologique de notre étude.

- **Chapitre 2: État d'art**

  Dans ce chapitre nous proposons une revue critique des approches existantes en matière de détection des menaces et de cybersécurité dans un contexte militaire. Ce chapitre présente une classification des protocoles classiques de surveillance, ainsi que leurs limites en termes de réactivité, de précision et d'adaptabilité. Nous y analysons également l'apport de l'intelligence artificielle, du machine learning et du deep learning dans l'optimisation des systèmes de défense. Enfin, ce chapitre justifie le recours à des modèles intelligents et distribués pour renforcer la sécurité des réseaux IoBT et répondre aux défis actuels de la défense.

- **Chapitre 3: Protocole Blockchain-Based Autonomous Authentication and Integrity for IoBT in C3I System (BioBT**

  Nous détaillons le protocole de sécurité BioBT, qui constitue la base de notre contribution. Ce chapitre décrit l'architecture du protocole dans le système C3I, les différents acteurs et composants impliqués, ainsi que les phases techniques de son fonctionnement. Nous abordons aussi les limitations du protocole, telles que le manque de confidentialité en blockchain publique, l'absence de tests à grande échelle, et la dépendance à la plateforme Ethereum. Ce chapitre prépare ainsi le terrain pour les améliorations proposées dans le chapitre suivant.

- **Chapitre 4: Amélioration du Protocole BioBT par un Modèle LSTM Basé sur le Deep Learning pour l'Authentification Biométrique**

  Ce chapitre décrit les motivations ayant conduit à l'intégration d'un modèle LSTM dans le protocole BioBT, dans le but d'améliorer la précision de l'authentification biométrique et la détection des intrusions. Nous détaillons le jeu de données utilisé, les étapes de prétraitement, la structure du modèle, les hyperparamètres, ainsi que les métriques d'évaluation. Les résultats expérimentaux obtenus sont analysés, démontrant l'efficacité du modèle proposé dans des environnements tactiques contraints, avec une précision de 98% pour l'authentification et un taux de détection d'anomalies de 77,94 %. Ce chapitre se conclut par une discussion sur la robustesse, la scalabilité et les perspectives d'intégration du modèle dans

des scénarios militaires réels.

Chapter 1

Généralités sur le Contexte

1.1 Introduction

Le contexte sécuritaire mondial actuel, marqué par une évolution rapide des menaces et des technologies, impose aux systèmes de défense de s'adapter en permanence. Les conflits modernes ne se limitent plus aux affrontements conventionnels: ils intègrent désormais des dimensions cybernétiques, informationnelles et technologiques, rendant la défense plus complexe et multidimensionnelle. Dans ce contexte, l'intégration des technologies avancées, telles que les capteurs intelligents, les systèmes de surveillance automatisés et les réseaux de communication sécurisés, devient essentielle pour anticiper et contrer des menaces de plus en plus sophistiquées. Ce chapitre vise à présenter les généralités du contexte militaire et de la défense moderne, en mettant en lumière l'évolution des conflits, les nouvelles menaces, ainsi que l'apport des technologies émergentes, en particulier le Deep Learning, dans la transformation des stratégies de défense et de détection des menaces.

1.2 Contexte Militaire et Défense Moderne

Cette section traite de l'évolution des menaces militaires, de la complexité des conflits modernes, de l'impact des nouvelles technologies dans les systèmes de défense, et des défis que cela représente pour l'adaptation des stratégies de défense.

1.2.1 Évolution des Conflits et des Menaces Militaires

Après le début des années 1990 et la fin de la guerre froide, les tensions géopolitiques entre les blocs Est et Ouest se sont grandement atténuées. Cela a laissé à penser au public que le risque d'une guerre mondiale avait fortement diminué, et que l'on entrait dans l'ère de la « mondialisation heureuse », économiquement favorable aux échanges commerciaux. Cela a aussi signifié le licenciement de quantités massives des forces armées de la plupart des pays occidentaux. Cependant, certains états ont décidé que le moment était idéal pour réinvestir dans la défense et commencer à développer de nouvelles possibilités dans les domaines émergents comme le cyberespace, l'espace, et la guerre de l'information [2].

1.2.1.1 Nature des Conflits Modernes

Les conflits modernes se caractérisent généralement par une complexité croissante et un caractère asymétrique. Les guerres traditionnelles entre états sont en déclin, les conflits contemporains étant des guerres de faible intensité impliquant des acteurs non étatiques tels que des groupes terroristes ou criminels. Les ennemis maîtrisent de plus en plus des stratégies irrégulières telles que le parti pris et la cyber-guerre pour compenser leur infériorité démontrée. L'impact environnemental est un facteur clé ici, puisque certaines armes modernes durent pendant de nombreuses années, endommageant des écosystèmes complets [3].

1.2.1.2 Menaces Actuelles et Futures

Comprennent non seulement les affrontements militaires classiques mais aussi des domaines émergents tels que la cybersécurité, la guerre de l'information, la militarisation de l'espace et de nouvelles technologies avancées avec par exemple, des drones et des armes autonomes. Les menaces futures s'étendront vraisemblablement aux nouvelles technologies à mesure que leur utilisation se généralisera, ce qui compliquera considérablement les moyens de défense traditionnels. Par conséquent, les forces innovantes devront être rapidement adoptées pour protéger correctement les infrastructures critiques et la nature [3] .

1.2.2 Technologies Employées dans les Systèmes Militaires

Dans le domaine de la défense nationale, les technologies avancées jouent un rôle crucial dans les systèmes militaires modernes. L'une des technologies clés est le radar avancé, capable de détecter et d'analyser des cibles à longue distance, donnant ainsi à l'armée un avantage significatif dans la préparation aux menaces potentielles. De plus, l'intelligence artificielle est largement utilisée pour améliorer la précision des décisions et la réponse aux situations commentaires. Les drones ont également gagné en importance dans les opérations militaires, car ils permettent d'effectuer des missions de reconnaissance et de combat sans mettre en danger la vie des soldats. En outre, la cybertechnologie est devenue essentielle pour la défense militaire, protégeant les systèmes militaires contre les cyberattaques et assurant la sécurité des informations sensibles. Pour rester prêt et exceller face aux défis de sécurité, investissez dans les technologies modernes ne sont pas un luxe mais une nécessité stratégique [4].

1.2.2.1 Capteurs et Systèmes de Surveillance

Les forces militaires modernes s'appuient sur des technologies de capteurs avancées afin d'assurer une surveillance permanente, une reconnaissance précise et une collecte de données en temps réel. Ces technologies permettent de détecter, d'identifier et de suivre des cibles terrestres, aériennes et maritimes, renforçant ainsi la capacité de prise de décision stratégique et tactique.

1. **Principales Technologies de Capteurs Militaires**

   Les dispositifs ci-dessous représentent les piliers de la surveillance militaire actuelle:

   - **Radars (Détection et Télémétrie par Radio)**

     Le radar est un système électronique capable de détecter, localiser et suivre des objets en mouvement tels que des avions, des drones ou des navires, en mesurant leur distance, leur direction et leur altitude [5]. Leur fonction principale repose sur l'émission et la réception d'ondes radio pour détecter la présence d'objets. Il existe plusieurs types de radars, chacun adapté à des usages spécifiques : les radars à balayage électronique, offrant une haute résolution et une adaptation rapide aux contre-mesures électroniques ; les radars Doppler, spécialisés dans la détection de la vitesse et de la direction

; et les radars à synthèse d'ouverture (SAR), capables de produire des images haute définition, même de nuit ou dans des conditions météorologiques difficiles. Ces technologies trouvent leurs applications dans la surveillance de l'espace aérien, la défense antimissile et la reconnaissance maritime.

- **LiDAR (Light Detection and Ranging)**

  Le LiDAR est une technologie de télédétection utilisant des impulsions laser pour mesurer les distances et générer des représentations tridimensionnelles de l'environnement [6]. Sa fonction principale est la cartographie 3D à l'aide de faisceaux laser, offrant une précision centimétrique et la capacité à détecter des objets dissimulés ou camouflés. Le LiDAR est particulièrement utile dans des domaines tels que la détection de mines, la reconnaissance de terrain et l'analyse topographique en zone ennemie.

- **Drones**

  Les drones militaires, qu'ils soient télépilotés ou autonomes, jouent un rôle fondamental dans les opérations modernes. Ils sont principalement utilisés pour la surveillance de zones hostiles sans exposer des vies humaines, le suivi en temps réel des mouvements ennemis, l'appui aux missions des forces spéciales, ainsi que l'évaluation des dommages après une opération de combat.

- **Satellites Militaires**

  Les satellites dédiés à l'usage militaire sont équipés de capteurs puissants leur permettant d'assurer plusieurs fonctions stratégiques. Ils sont utilisés pour l'observation terrestre à grande échelle grâce à des technologies optiques, infrarouges ou de radar spatial. Ils assurent également la transmission sécurisée des communications entre les bases, les troupes et les centres de commandement, et participent au guidage précis des armements stratégiques.

2. **Rôle des Capteurs dans la Collecte et l'Analyse de Données**

Ces capteurs jouent un rôle déterminant dans l'acquisition et l'exploitation de

données stratégiques :

- **Ciblage de précision** : Identification exacte des cibles pour les frappes chirurgicales.
- **Surveillance proactive** : Détection des menaces émergentes ou dissimulées.
- **Traitement intelligent des données** :
  - Réduction du temps de réaction décisionnel sur le terrain.
  - Intégration de l'Intelligence Artificielle (IA) pour l'analyse rapide et automatique des données recueillies.

1.2.2.2 Systèmes de Communication et de Commandement

Systèmes de communication et de commandement sont des éléments essentiels dans les forces armées pour assurer la coordination et l'efficacité des opérations militaires. La recherche souligne l'importance du modèle C3I, qui a évolué pour devenir le modèle C4I en intégrant la technologie Informatique. Ce modèle étendu permet une gestion plus performante des informations et une meilleure transmission des données entre les différentes unités militaires, qu'elles soient terrestres, maritimes, ou aériennes. Les systèmes de communication, qu'ils soient basés sur la radio, le satellite ou d'autres technologies avancées, sont conçus pour crypter les transmissions et garantir la sécurité des messages. Ces équipements sont capables de fonctionner dans des environnements difficiles, offrant ainsi une fiabilité indispensable dans des situations critiques, notamment dans la gestion des armes nucléaires. En somme, les systèmes de communication et de commandement jouent un rôle central dans la coordination des opérations militaires, en assurant une communication rapide, sécurisée et efficace entre les différents acteurs, et améliorer la réactivité des forces sur le terrain [7].

1.2.2.3 Systèmes de Défense Aérienne et Maritime

Les systèmes de défense aérienne et maritime sont des infrastructures et des technologies mises en place pour protéger un pays contre les menaces provenant de l'air et de la mer. Cela inclut des radars, des avions de chasse, des navires de guerre et des systèmes de surveillance qui détectent, interceptent et neutralisent les menaces potentielles. Ces systèmes sont essentiels pour garantir la souveraineté nationale, la

sécurité des frontières et la protection des intérêts stratégiques en mer et dans l'espace aérien [7].

1.2.3 Défis Actuels en Matière de Défense

À l'ère de la numérisation et de l'interconnexion globale, les stratégies de défense contemporaine sont de plus en plus tributaires des technologies de pointe, des données massives et de leur traitement intelligent. Cette dépendance induit à la fois des avantages opérationnels significatifs (tels que la rapidité de réponse et la précision des décisions) et des défis critiques, notamment en matière de cybersécurité, d'interopérabilité et de gestion de l'information.

Afin d'assurer une défense efficace et résiliente, il est essentiel d'adopter une posture proactive, capable d'anticiper les menaces émergentes tout en maintenant la sécurité et l'intégrité des systèmes d'information stratégiques [8].

Les environnements militaires modernes sont confrontés à une transformation rapide, où la quantité d'information, la vitesse d'exécution, et la complexité des menaces imposent de nouveaux standards technologiques. Les systèmes intelligents, pour être efficaces, doivent relever plusieurs défis fondamentaux que nous illustrons ci-dessous par des modèles et équations simplifiées.

1. **Complexité et Vitesse des Menaces**

   L'émergence rapide de menaces complexes peut être modélisée comme un processus exponentiel de génération d'événements :

   $$N(t) = N_0 \cdot e^{\alpha t} \tag{1.1}$$

   où $N(t)$ est le nombre d'attaques ou événements à l'instant t, N_0 la quantité initiale, et α un facteur d'accélération lié à la complexité croissante des menaces. Cela met en évidence le besoin de systèmes de réaction à faible latence.

2. **Interopérabilité et Intégration des Systèmes**

   Le défi d'interopérabilité peut être représenté par une mesure d'entropie de compatibilité entre systèmes S_i :

$$H = -\sum_{i=1}^{n} p(S_i) \cdot \log p(S_i) \tag{1.2}$$

Une entropie élevée indique une diversité de protocoles et normes difficilement intégrables, suggérant un besoin d'harmonisation des formats et interfaces.

3. **Vulnérabilités Cybersécuritaires**

   La probabilité P_c de compromission d'un système peut être exprimée comme une fonction inverse de sa robustesse R et de sa complexité C :

$$P_c = \frac{1}{R \cdot \log(C+1)} \tag{1.3}$$

   Plus le système est robuste (R élevé), et plus la complexité d'accès est grande (C), moins il est vulnérable aux cyberattaques.

4. **Gestion du Volume et de la Vitesse des Données**

   La contrainte de traitement de données en temps réel est modélisée par l'inégalité :

$$T_{process} \leq T_{acquisition} \tag{1.4}$$

   où $T_{process}$ est le temps de traitement, et $T_{acquisition}$ le temps entre deux captures de données. Si $T_{process} > T_{acquisition}$, un goulot d'étranglement apparaît, ce qui mène à une surcharge et à une perte d'efficacité.

5. **Limites des Approches Traditionnelles**

   Le manque d'adaptabilité des systèmes traditionnels peut se traduire par une fonction de rigidité f_{rigid} :

$$f_{rigid} = \frac{\Delta R}{\Delta C} \tag{1.5}$$

   où ΔR représente le changement de réponse du système et ΔC le changement dans les conditions externes. Une faible valeur de f_{rigid} indique une faible capacité d'adaptation, typique des infrastructures obsolètes.

1.2.4 Transition des Approches de Défense : Des Systèmes Traditionnels aux Systèmes Intelligents

Le secteur militaire connaît une transition technologique profonde, marquée par le passage progressif des approches de défense traditionnelles, fondées sur des processus humains, des analyses manuelles et des systèmes rigides, vers des solutions automatisées, intelligentes, et adaptatives basées sur le Machine Learning (ML) et l'intelligence artificielle (IA) [9].

Ce basculement n'est pas arbitraire, mais résulte d'un ensemble de facteurs déterminants :

- **L'explosion du volume des données opérationnelles** générées par les capteurs, les drones, les satellites et les communications sur le terrain dépasse largement les capacités d'analyse humaine.
- **La montée en complexité et en rapidité des menaces** notamment les cyberattaques asymétriques et multi-domaines, rend les systèmes classiques inadaptés.
- **La nécessité d'une réactivité en temps réel** pour éviter les pertes humaines et matérielles impose des systèmes capables de décision automatique assistée.
- **Les exigences de résilience et d'interopérabilité** dans un environnement composé de dispositifs hétérogènes, interconnectés et géographiquement dispersés.

Ainsi, l'intégration de l'IA représente un changement de paradigme dans les doctrines de défense, en rendant les systèmes plus intelligents, plus rapides, et plus robustes face à des environnements dynamiques et imprévisibles. Table 1.1 présente une comparaison synthétique que nous avons réalisée entre les deux paradigmes, selon des métriques clés de performance. Elle met en lumière les justifications opérationnelles et techniques qui poussent les institutions de défense à adopter des architectures intelligentes.

Face à ce constat, les forces de défense doivent entamer une révision structurelle et technologique de leurs modèles. Cette transition passe par :

- **La refonte des architectures de commandement et de contrôle** pour accueillir des modules IA interopérables et déployables en temps réel.

Table 1.1: Comparaison des approches de défense

Métrique	**Défense Traditionnelle**	**Défense basée sur ML/IA**
Latence de Réaction	Élevée (secondes à minutes, traitement manuel)	Faible (millisecondes, traitement en temps réel)
Capacité d'analyse de données	Limitée par les ressources humaines et outils statiques	Massivement parallèle, traitement automatisé de Big Data
Complexité opérationnelle gérée	Moyenne, adaptée à des schémas connus et linéaires	Élevée, traitement de schémas non linéaires, imprévus, multidimensionnels
Taux d'erreurs humaines	Élevé (fatigue, stress, surcharge cognitive)	Faible (automatisation, supervision minimale)
Précision des décisions	Variable, dépendante du jugement humain	Optimisée par apprentissage à partir de données historiques
Scalabilité	Faible, infrastructure centralisée et rigide	Élevée, architecture modulaire, distribuée
Résilience aux cybermenaces	Réponse réactive, vulnérable aux attaques ciblées	Prédiction proactive, réponse automatisée et adaptative

- **L'implémentation de chaînes décisionnelles hybrides** combinant jugement humain et capacités prédictives automatisées.
- **La mise en place de plateformes de fusion de données** capables d'intégrer, nettoyer, analyser et visualiser l'information de manière cohérente.
- **La formation continue du personnel militaire** à l'usage des technologies d'intelligence artificielle et à l'éthique de leur utilisation.
- **L'adoption de cadres réglementaires** garantissant l'utilisation responsable, transparente et sécurisée de ces technologies dans les opérations militaires.

Cette évolution représente plus qu'une innovation technologique, elle constitue une réponse stratégique et nécessaire à la complexité grandissante du champ de bataille moderne. Le succès des forces armées de demain dépendra en grande partie de leur capacité à exploiter intelligemment les ressources informationnelles à leur disposition.

1.3 Introduction du Deep Learning dans le Domaine Militaire

L'intégration du Deep Learning (DL) dans le secteur militaire revêt une importance stratégique croissante. Ce domaine exige des solutions avancées pour faire face à la complexité et à l'évolution rapide des menaces et des environnements opérationnels. Cette intégration explore les motivations et les opportunités offertes par l'application du DL aux opérations militaires, tout en abordant les implications éthiques et sécuritaires liées à son utilisation. Par ailleurs, les concepts fondamentaux du DL ainsi que ses applications actuelles en contexte militaire seront présentés. Enfin, cette introduction vise à établir une compréhension globale des enjeux, des technologies et des usages du DL dans le cadre militaire.

1.3.1 Définition et Concepts Clés du Deep Learning

Le DL, ou Apprentissage Profond, est une branche avancée de l'IA qui repose sur des architectures de réseaux de neurones composées de multiples couches hiérarchiques. Il permet à une machine d'extraire automatiquement des représentations pertinentes à partir de données brutes, sans intervention humaine directe. Les concepts clés du DL incluent plusieurs types de réseaux spécialisés, chacun adapté à une catégorie de problèmes spécifiques tels que la reconnaissance visuelle, l'analyse séquentielle ou la compréhension du langage. Les sections suivantes présentent ces architectures fondamentales:

1.3.1.1 Artificial Neural Network (ANN)

Les réseaux de neurones artificiels (ANN) représentent l'un des fondements du DL. Inspirés du fonctionnement du cerveau humain, ils sont constitués de couches de neurones interconnectés capables d'apprendre des représentations complexes à partir de données. Chaque neurone artificiel reçoit une entrée pondérée, applique une fonction d'activation, puis transmet une sortie aux couches suivantes. Les ANN sont particulièrement efficaces pour l'analyse et la classification de données volumineuses, grâce à leur capacité à détecter automatiquement des motifs non linéaires. Dans un contexte militaire, ces réseaux permettent de modéliser des comportements dynamiques

et d'optimiser la prise de décision à partir de données multisources, comme les signaux radar ou les flux de communication. Leur flexibilité en fait un outil essentiel pour le traitement intelligent de l'information, notamment dans les systèmes d'alerte avancée ou la surveillance autonome [10].

1.3.1.2 Deep Neural Network (DNN)

Les réseaux de neurones profonds (DNN) sont une extension des ANN, caractérisés par un grand nombre de couches cachées entre les couches d'entrée et de sortie. Cette profondeur structurelle permet aux DNN de modéliser des relations complexes et abstraites dans les données, rendant ces modèles particulièrement adaptés à la détection d'anomalies dans des environnements comme l'Internet des objets (IoT). En contexte militaire, leur capacité à traiter des flux de données hétérogènes issus de multiples capteurs les rend cruciaux pour la cybersécurité et la surveillance en temps réel. L'apprentissage des DNN s'effectue généralement par rétropropagation, avec l'utilisation de fonctions d'activation comme ReLU ou Sigmoid et de techniques de régularisation telles que le dropout pour éviter le surapprentissage. Leur performance a été démontrée à travers des expérimentations où des architectures telles que le Multilayer Perceptron (MLP) ont atteint une précision de détection supérieure à 98 %, tout en conservant une bonne capacité de généralisation sur des données non vues [11].

1.3.1.3 Convolutional Neural Network (CNN)

Les réseaux de neurones convolutifs (CNN) sont un type de DNN spécialement conçu pour le traitement des images. Ils sont capables d'extraire automatiquement les caractéristiques pertinentes à partir de données visuelles, ce qui élimine le besoin d'une étape manuelle d'extraction de caractéristiques. Un CNN est constitué de couches de convolution, de regroupement (pooling) et de couches entièrement connectées, qui permettent d'apprendre progressivement des représentations de plus en plus abstraites de l'image. Grâce à cette architecture, les CNN sont particulièrement efficaces pour des tâches telles que la classification, la reconnaissance et la détection d'objets dans tous les types d'images [12].

1.3.1.4 Recurrent Neural Network (RNN)

Les réseaux de neurones récurrents (RNN) sont des modèles conçus pour traiter des données séquentielles comme le texte ou les séries temporelles. Leur particularité est leur architecture avec boucles qui maintient une mémoire des entrées précédentes, permettant de prendre en compte le contexte temporel. Contrairement aux réseaux feedforward classiques, les RNN peuvent traiter des séquences de longueur variable. Cependant, ils rencontrent des difficultés avec les dépendances à long terme, ce qui a conduit au développement de variantes plus performantes comme les LSTM et GRU. Ces modèles sont particulièrement utiles en traitement du langage naturel pour des tâches comme la traduction ou l'analyse de texte [13].

1.3.1.5 Long Short-Term Memory (LSTM)

Les réseaux à mémoire long terme (LSTM) sont un type de RNN capables de traiter des données séquentielles tout en conservant des informations sur de longues périodes. Contrairement aux RNN classiques, les LSTM évitent les problèmes de perte ou explosion du gradient grâce à une architecture composée de portes qui contrôlent le flux d'information : la porte d'oubli, la porte d'entrée, et la porte de sortie, ainsi qu'un état de cellule qui transporte la mémoire.

Dans l'étude analysée, un modèle LSTM a été utilisé pour l'analyse de sentiments sur des textes. L'architecture comprend une couche d'embedding, des couches LSTM, et une couche de sortie avec activation sigmoïde pour classer les sentiments comme positifs ou négatifs. Les résultats montrent une bonne précision lorsque les modèles sont entraînés sur des jeux de données spécifiques, mais des performances plus faibles en cas de généralisation à d'autres contextes [14].

1.3.2 Applications du Deep Learning dans le Secteur Militaire

Les applications actuelles du DL dans le domaine militaire sont diverses et couvrent un large éventail de sujets de gestion et d'analyse de l'information. En facilitant la communication et le partage d'informations entre les secteurs militaire et civil, l'intégration civilo-militaire (ICM) joue un rôle essentiel dans le renforcement de la sécurité nationale. Grâce à l'optimisation des systèmes de gestion de l'information, les technologies de DL permettent une meilleure analyse des données de surveillance, une

meilleure reconnaissance d'images et une meilleure cyberdéfense. Grâce à des systèmes de gestion de l'information centralisés et décentralisés basés sur l'entropie structurelle, ces technologies améliorent l'efficacité des processus de communication militaire et permettent une adaptation flexible aux exigences en constante évolution de la défense [15].

1.3.2.1 Analyse de Données de Surveillance

Le DL est utilisé par les groupes militaires pour évaluer de vastes volumes de données complexes collectées à partir de diverses sources. Ces sources comprennent des données open source, telles que des photos, des vidéos et des discours provenant de plateformes telles que des véhicules aériens sans pilote (UAV), des évaluations d'experts, l'Internet des objets (IoT) et l'Internet des objets de champ de bataille (IoBT), des données biométriques et des dossiers personnels. Des algorithmes de DL sont souvent nécessaires pour traiter les données, en particulier lorsqu'elles ne sont pas structurées ou qu'elles sont volumineuses. Ces algorithmes basés sur des réseaux neuronaux, qui sont excellents pour traiter des données militaires complexes, comprennent les CNN, les algorithmes antagonistes profonds et les algorithmes de mémoire à long terme (LSTM) [16].

1.3.2.2 Reconnaissance d'Images et Détection de Cibles

Un type de réseau neuronal profond qui excelle dans l'analyse de photos et de vidéos est le CNN. De ce fait, ils sont idéaux pour des utilisations militaires telles que la localisation de cibles à l'aide de véhicules aériens sans pilote (UAV). Le DL est utilisé pour améliorer les compétences de détection d'objets à mesure que la technologie des drones devient plus sophistiquée. Pour que ces véhicules fonctionnent, les techniques de détection d'objets sont essentielles. Bien qu'il existe des utilisations évidentes de la reconnaissance d'images et de l'identification de cibles dans les drones, ces technologies sont probablement également utilisées dans d'autres applications militaires. Par exemple, les caméras de sécurité pourraient suivre des individus d'intérêt ou identifier d'éventuelles menaces à l'aide de la reconnaissance d'images. Ces méthodes pourraient être utilisées pour analyser les images satellite et localiser les mouvements de troupes, les armes ou les infrastructures de l'adversaire [16].

1.3.2.3 Détection Automatique d'Anomalies

Les LSTM, spécialisés dans le traitement des données séquentielles, sont largement utilisés pour la détection d'anomalies dans les systèmes militaires. Grâce à leur capacité à mémoriser les dépendances temporelles à long terme, les LSTM apprennent les comportements normaux dans des séquences de données issues de capteurs (vitesse, altitude, température, etc.). Lorsqu'un événement anormal survient tel qu'une défaillance d'un capteur ou un comportement inattendu d'un dispositif, le modèle détecte un écart significatif entre la séquence réelle et celle prédite. Cette capacité à surveiller en continu des flux de données temporelles rend les LSTM particulièrement adaptés aux contextes opérationnels, comme la surveillance de drones ou la détection précoce de pannes dans les systèmes embarqués.

1.3.3 Avantages du Deep Learning pour la Défense

Le DL offre des avantages stratégiques majeurs dans le domaine militaire, notamment la capacité à traiter de vastes volumes de données hétérogènes, à améliorer la précision des systèmes de détection en réduisant les faux positifs, et à automatiser des tâches complexes avec une réactivité en temps réel. Ces atouts renforcent l'efficacité des opérations de défense et la prise de décision stratégique.

1.3.3.1 Capacité à Traiter de Grandes Masses de Données

Le DL excelle dans l'analyse de vastes ensembles de données complexes, ce qui est crucial pour le secteur de la défense, où les informations proviennent de multiples sources telles que les satellites, les capteurs IoT et les systèmes radar. Par exemple, une étude de l'Université du Missouri a démontré que des modèles de DL pouvaient identifier des sites de missiles sol-air en Chine avec une précision de 90 %, réduisant le temps d'analyse de 60 heures à seulement 42 minutes. Cette efficacité permet aux agences de renseignement de traiter rapidement des volumes massifs de données, améliorant ainsi la surveillance et la prise de décision stratégique.

1.3.3.2 Précision et Réduction des Faux Positifs

L'un des défis majeurs des systèmes de détection traditionnels est le taux élevé de faux positifs, qui peut entraîner une surcharge des analystes et des ressources. Le DL, grâce à sa capacité d'apprentissage profond, permet de réduire significativement ces faux positifs. Par exemple, une recherche a montré que l'utilisation de modèles de DL pour la détection d'anomalies dans les systèmes de sécurité permettait de diminuer les faux positifs de 10 % par rapport aux méthodes traditionnelles. Cette amélioration de la précision est essentielle pour la défense, où une alerte erronée peut avoir des conséquences graves [17].

1.3.3.3 Automatisation et Réactivité en Temps Réel

Le DL permet l'automatisation de tâches complexes et offre une réactivité en temps réel, ce qui est vital pour les opérations militaires. Par exemple, des systèmes de surveillance par drones utilisant des réseaux de neurones profonds peuvent identifier des comportements violents en temps réel, permettant une intervention rapide. De plus, dans le domaine de la cybersécurité, le DL peut détecter et répondre automatiquement aux menaces, réduisant ainsi le temps de réaction et minimisant les dommages potentiels.

1.4 Conclusion

En conclusion, l'analyse du contexte militaire contemporain met en évidence la nécessité d'une adaptation constante face à la diversité et à la complexité croissantes des menaces. L'évolution des technologies, notamment l'essor du Deep Learning et de l'Intelligence Artificielle, offre de nouvelles perspectives pour renforcer l'efficacité des systèmes de défense, tant au niveau de la surveillance que de la prise de décision en temps réel. Les capacités de traitement massif des données, la précision accrue des algorithmes et l'automatisation des processus constituent des atouts majeurs pour anticiper, détecter et neutraliser les menaces dans un environnement militaire en perpétuelle mutation. Ces constats justifient pleinement l'exploration de solutions innovantes, telles que celles présentées dans les chapitres suivants, pour répondre aux défis de la sécurité et de la défense du futur.

Dans le chapitre suivant, nous nous intéresserons aux travaux de recherche déjà réalisés dans ce domaine. Nous passerons en revue les différentes approches utilisées pour la détection des menaces, en évaluant leurs avantages, leurs limites, et en soulignant les apports concrets de l'IA dans ce domaine stratégique.

Chapter 2

État de l'art

2.1 Introduction

L'évolution des menaces dans le domaine militaire et l'émergence de nouveaux défis sécuritaires nécessitent une adaptation constante des stratégies de défense. Dans ce contexte, l'intelligence artificielle et le deep learning jouent un rôle central dans l'amélioration des capacités de détection et de prévention des menaces. Ce chapitre présente une revue approfondie des approches existantes en matière de détection des menaces, en mettant en lumière les protocoles classiques et leurs limites. Nous abordons également les apports de l'intelligence artificielle dans l'optimisation des stratégies de défense, en mettant l'accent sur les techniques modernes permettant d'anticiper et de neutraliser les menaces émergentes. Enfin, nous explorons les perspectives futures qui pourraient révolutionner le domaine de la sécurité militaire grâce aux avancées technologiques.

2.2 Méthodologie et Objectifs de Recherche

Cette partie décrit la démarche scientifique adoptée pour réaliser les objectifs établis, en plus des outils et méthodes employés dans le processus de recherche. La méthodologie repose sur une analyse rigoureuse des données, la sélection des sources pertinentes, et l'application de techniques avancées comme le machine learning pour les applications militaires. L'objectif est d'optimiser la détection des menaces, d'optimiser les tactiques défensives et de prévoir les actions adverses pour garantir une supériorité

stratégique et un renforcement de la sécurité, voici comment organiser chaque partie:

2.2.1 Objectifs de Recherche

L'objectif principal de ce chapitre est d'étudier en détail les recherches précédentes concernant la détection des menaces et l'amélioration des tactiques de défense dans un contexte militaire, en utilisant des méthodes traditionnelles et contemporaines, y compris l'intelligence artificielle et l'apprentissage automatique. Cette revue a pour objectif spécifique de :

- Comprendre les fondements de base des méthodes actuelles de détection des menaces et de défense militaire.
- Déterminer les limites des protocoles traditionnels employés dans les systèmes militaires (réactivité, précision, adaptabilité, et cetera).
- Il est essentiel de justifier l'introduction de solutions basées sur l'intelligence artificielle et le machine learning, dans le but d'apporter par la suite une amélioration significative à notre contribution.

2.2.2 Méthodes et Outils de Recherche

Pour mener à bien cette étude, une recherche bibliographique systématique a été réalisée en utilisant les bases de données suivantes :

2.2.2.1 Recherche bibliographique systématique

La collecte des travaux de recherche a été réalisée à partir de bases scientifiques reconnues telles que IEEE Xplore, SpringerLink, ScienceDirect, Google Scholar et JSTOR.

- **IEEE Xplore**: La bibliothèque numérique IEEE Xplore propose près de quatre millions d'articles dans divers secteurs, offrant une richesse de connaissances aux organisations pour améliorer ou découvrir des percées, avec des outils de recherche puissants pour des recherches pertinentes.

- **ScienceDirect :** Est une plateforme numérique d'Elsevier offrant un accès à des articles et ouvrages scientifiques soumis à l'examen par des experts dans divers secteurs tels que la science, la technologie et la médecine.

- **SpringerLink**: Est un site web offre l'accès à des millions de publications scientifiques, y compris ouvrages, articles de périodiques, chapitres et actes de colloques édités par la maison d'édition Springer. Elle englobe un vaste éventail de domaines, y compris les sciences, la technologie, la médecine et les sciences sociales.

- **Google Scholar**: Plateforme de recherche en ligne dotée d'un algorithme raffiné qui repère des ressources universitaires comme des articles soumis à l'examen par des pairs, des thèses, des livres, des résumés et des documents techniques venant de tous les coins du globe. Il assiste les chercheurs dans la recherche de matériaux appropriés et dans l'identification des sources d'information initiales. En outre, il offre la fonctionnalité de datation de Google pour un suivi aisé.

- **JSTOR** : bibliothèque digitale fournissant accès à des milliers de périodiques universitaires, d'ouvrages et de documents originaux, simplifiant la numérisation et la conservation des collections académiques.

2.2.2.2 Outils d'analyse et de traitement

Dans le cadre de ce travail, plusieurs outils ont été mobilisés pour l'analyse et le traitement des données.

Draw.io : Est un logiciel en ligne gratuit permettant de créer facilement des diagrammes, schémas, organigrammes et modèles conceptuels. Il est particulièrement utilisé pour représenter visuellement des systèmes, des processus ou des architectures, grâce à une interface intuitive et une large bibliothèque de formes.

Zotero: Pour organiser les références et générer la bibliographie, c'est un logiciel open source pour la gestion de références qui facilite l'organisation et la gestion efficiente des sources de recherche. Zotero se distingue en tant qu'instrument non commercial, aligné sur l'idée d'une infrastructure ouverte gérée par des chercheurs, ce qui le rend idéal pour les universitaires et les chercheurs travaillant dans des environnements collaboratifs [18].

2.2.3 Critères de sélection des articles

La sélection des articles scientifiques s'est appuyée sur un ensemble de critères rigoureux visant à garantir la qualité, la pertinence et l'actualité des sources utilisées dans le cadre de cette recherche. Les critères retenus sont les suivants :

- **Actualité :** Seules les publications récentes, datées entre 2018 et 2025, ont été considérées, afin de s'assurer de la prise en compte des avancées les plus récentes dans les domaines concernés.
- **Pertinence thématique :** Les articles sélectionnés devaient aborder des thématiques directement en lien avec les axes de recherche du projet, à savoir l'intelligence artificielle, la technologie blockchain, ainsi que les applications dans le domaine de la défense militaire et des systèmes C3I (Command, Control, Communication and Intelligence).
- **Crédibilité scientifique :** L'analyse s'est limitée aux publications issues de revues à comité de lecture reconnues, telles que celles indexées par IEEE, Springer, Elsevier, Wiley ou ACM. Cela garantit la fiabilité méthodologique et la rigueur scientifique des travaux mobilisés.
- **Impact académique :** Une attention particulière a été portée aux articles cités dans d'autres travaux de recherche ou publiés dans des conférences et journaux de référence, ce qui témoigne de leur influence et de leur reconnaissance dans la communauté scientifique.

2.2.4 Mots-clés utilisés

Dans le cadre de la revue de littérature, une recherche documentaire approfondie a été effectuée à l'aide de mots-clés soigneusement choisis afin de couvrir les thématiques centrales du projet. Ces mots-clés ont été sélectionnés dans les deux langues principales de la littérature scientifique, à savoir l'anglais et le français, et reflètent les concepts clés liés à l'intelligence artificielle, à la cybersécurité, à la blockchain et aux applications militaires.

1. **En anglais :**

- *"Machine learning military applications"*
- *"Threat detection in military systems"*
- *"Blockchain IoT authentication"*
- *"Anomaly detection in defense applications"*
- *"Intrusion detection for military networks"*
- *"Deep learning in cyber defense"*
- *"Supervised and unsupervised learning for military use"*
- *"Cyber defense strategies in military contexts"*

2. **En français :**

- *"Détection des menaces par l'intelligence artificielle"*
- *"Sécurité des systèmes C3I"*
- *"Authentification basée sur la blockchain dans les réseaux IoT militaires"*
- *"Optimisation des stratégies de défense par apprentissage automatique"*
- *"Systèmes de défense autonomes et intelligents"*
- *"Détection d'intrusion dans les réseaux tactiques"*

Ces mots-clés ont permis d'orienter les recherches vers des publications pertinentes et spécialisées, en assurant une couverture exhaustive des domaines technologiques et stratégiques ciblés.

2.3 Classification des Protocoles de Détection des Menaces Classiques

La détection des menaces dans les systèmes militaires repose sur divers protocoles visant à identifier, prévenir et contrer les attaques potentielles. Ces protocoles peuvent être classés en plusieurs catégories en fonction de leur approche et de leur mode de fonctionnement. Parmi eux, on retrouve les méthodes basées sur la détection d'anomalies, les protocoles de surveillance et de sécurité des réseaux, ainsi que les techniques traditionnelles de cyberdéfense. Chaque approche présente des avantages et des

limitations qui influencent leur efficacité face aux menaces émergentes et en constante évolution. Cette section explore ces différentes classifications en mettant en évidence leurs principes, leurs applications et leurs performances dans un contexte militaire.

La Figure 2.1 présente une classification des protocoles conventionnels de détection et prévention des menaces en trois grandes familles, proposée dans le cadre de notre travail :

- **Détection d'anomalies** : ABID, MIL-STD-1553, ADMOST
- **Surveillance et sécurité des réseaux** : BIoBT, systèmes actifs, capteurs sans fil, fibres optiques
- **Cyberdéfense traditionnelle** : Firewalls, NetFlow, SNMP

Cette structuration permet de mieux comprendre les complémentarités entre ces approches et d'identifier les points à renforcer pour bâtir une architecture de défense cohérente et robuste.

2.3.1 Protocoles Basés sur la Détection d'Anomalies

Les protocoles basés sur la détection d'anomalies identifient les comportements inhabituels au sein des systèmes militaires en analysant les écarts par rapport aux modèles normaux de fonctionnement. Ces approches incluent des méthodes comme la détection d'intrusion basée sur les anomalies du brouillage (ABID), l'analyse des communications avioniques avec l'algorithme CUSUM et l'utilisation du protocole ADMOST pour la surveillance des données de vol des drones.

2.3.1.1 Détection d'intrusion basée sur les anomalies sensibles au protocole (ABID)

Les auteurs proposent une méthode ABID pour détecter les brouilleurs intelligents ciblant des paquets critiques, comme les messages "Hello" du protocole OSPF. En s'appuyant sur l'analyse du SNR et du type de paquet, ainsi que sur des tests statistiques (ex. : Hotelling's T^2), la méthode atteint une précision de détection proche de 100 % avec un faible taux de faux positifs (2 %). Elle est adaptée aux environnements militaires où la protection des communications critiques est primordiale [19].

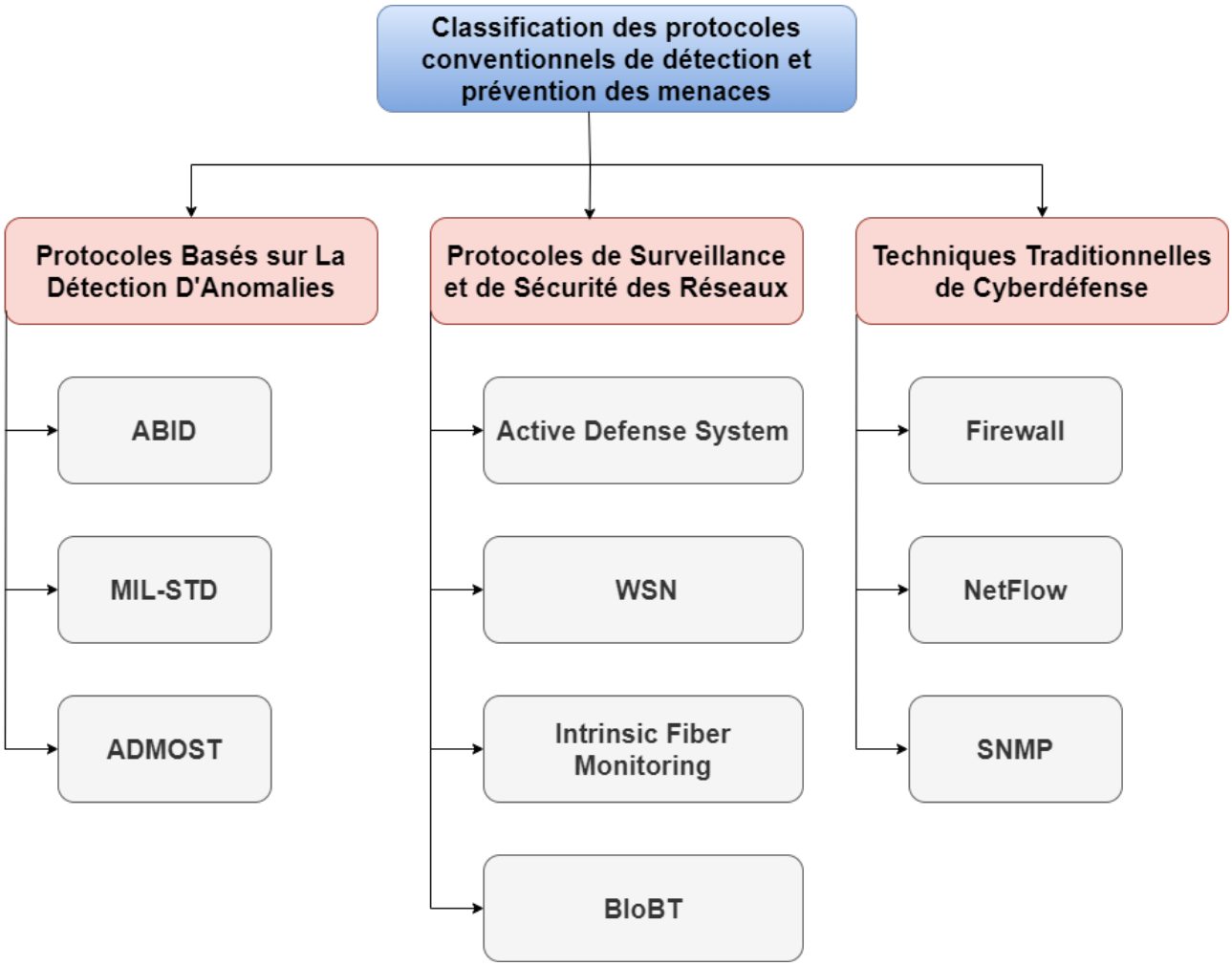

Figure 2.1: Classification des protocoles conventionnels de détection et prévention des menaces.

Cependant, les conditions extrêmes des terrains militaires, la sophistication des attaques, et les exigences de faible latence limitent son efficacité. Une approche hybride combinant IA et détection par signature est recommandée pour pallier ces limites.

2.3.1.2 Détection d'anomalies dans les systèmes MIL-STD-1553

Le protocole MIL-STD-1553 est couramment utilisé dans les systèmes avioniques. L'algorithme CUSUM permet une détection non supervisée d'anomalies en comparant les écarts cumulés à une moyenne de référence. Cela permet de détecter rapidement des comportements suspects sans nécessiter de données étiquetées [20].

Malgré son efficacité, le protocole souffre de limitations telles qu'un débit réduit, une architecture centralisée à point de défaillance unique, et une sensibilité aux attaques sophistiquées. La combinaison de CUSUM avec des techniques d'IA pourrait améliorer

la robustesse de la détection.

2.3.1.3 ADMOST : Détection et correction d'anomalies en ligne

Le protocole ADMOST détecte et corrige en temps réel les anomalies dans les données de vol des drones, en suivant dynamiquement un sous-espace représentatif des données multivariées. En cas d'anomalie, les valeurs aberrantes sont remplacées par des estimations fiables, ce qui améliore la stabilité et la sécurité du système [21].

Ses performances dépendent toutefois de la qualité initiale des données, de la complexité du suivi en ligne, et des contraintes des systèmes embarqués à faible puissance. Son efficacité diminue également face à des données fortement hétérogènes ou fragmentées.

2.3.2 Protocoles de Surveillance et de Sécurité des Réseaux

Les protocoles de sécurité réseau jouent un rôle crucial dans la protection des infrastructures militaires. Qu'il s'agisse de l'authentification des entités connectées, de la surveillance proactive des flux de données ou de la détection physique des intrusions, ces protocoles visent à assurer la résilience, la confidentialité et l'intégrité des communications critiques.

2.3.2.1 BIoBT : Authentification autonome par blockchain dans l'IoBT

Le protocole BIoBT propose une authentification décentralisée pour les objets connectés militaires via blockchain, garantissant l'intégrité et la traçabilité des communications C3I sans tiers de confiance [1].

Malgré ses atouts, il pose des défis de consommation énergétique, de scalabilité, et de dépendance à une infrastructure blockchain robuste, ce qui limite son adoption sur des dispositifs embarqués aux ressources restreintes.

2.3.2.2 Système de défense active pour réseaux critiques

Ce système multi-modulaire, initialement conçu pour les réseaux de surveillance électrique, intègre des fonctions de collecte, d'analyse et de réaction aux menaces en temps réel, avec un accent sur la protection proactive des ressources sensibles [22].

Bien qu'adaptable aux réseaux militaires, il nécessite des ajustements pour les infrastructures existantes, et peut être limité par la disponibilité des ressources de calcul et par sa dépendance à une bibliothèque de stratégies préexistantes.

2.3.2.3 Sécurité des réseaux de capteurs sans fil militaires

Ce protocole en trois phases (établissement des clés ECC, découverte des voisins, communication sécurisée) garantit la confidentialité et l'authenticité des échanges entre nœuds dans les réseaux militaires [23].

Cependant, la dépendance à une station de base pour l'authentification crée un point de défaillance. La cryptographie ECC, bien qu'optimisée, reste coûteuse pour les capteurs à faible consommation, et le protocole reste vulnérable à des attaques avancées.

2.3.2.4 Surveillance intrinsèque des fibres optiques

Cette approche permet de détecter physiquement toute tentative de manipulation des câbles en analysant les signaux lumineux transitant dans la fibre. Elle offre une surveillance passive, précise et à faible taux de faux positifs pour les réseaux sensibles comme SIPRNET ou JWICS [24].

Néanmoins, elle ne couvre pas les menaces logicielles et ne s'applique qu'aux infrastructures en fibre optique, avec un risque résiduel d'alertes liées à des perturbations non malveillantes (ex. : vibrations externes).

2.3.3 Techniques Traditionnelles de Cyberdéfense

Les techniques traditionnelles de cyberdéfense regroupent des outils éprouvés tels que les pare-feu, NetFlow et SNMP. Bien qu'indispensables, ces solutions montrent des limites face aux menaces modernes et doivent être renforcées par des approches intelligentes comme l'IA ou l'apprentissage automatique.

2.3.3.1 Pare-feu (Firewall)

Les pare-feu contrôlent le trafic réseau à l'aide de règles basées sur les adresses, ports ou protocoles. Les versions proxy inspectent les couches applicatives, mais peu-

vent créer des goulots d'étranglement. L'architecture ASP améliore les performances en redirigeant le trafic légitime vers des chemins accélérés [25].

Cependant, les pare-feu traditionnels restent limités contre les attaques sophistiquées (zero-day, évasion) et nécessitent des règles statiques, souvent inefficaces face à des menaces dynamiques. Leur efficacité dépend d'une intégration avec des outils plus intelligents, comme l'IDS ou les modèles IA.

2.3.3.2 NetFlow

NetFlow analyse les métadonnées des flux réseau pour identifier des anomalies de trafic, comme les attaques DDoS. Il est apprécié pour sa capacité à traiter de grands volumes en temps réel, particulièrement utile dans les environnements militaires [26].

Ses principales limites résident dans l'absence d'analyse de contenu, la perte potentielle d'informations due à l'échantillonnage, et l'impossibilité de bloquer les menaces. Il doit donc être couplé à des solutions complémentaires pour offrir une protection active et granulaire.

2.3.3.3 SNMP (Simple Network Management Protocol)

SNMP permet la surveillance des équipements via une base de données MIB et des échanges de requêtes GET/SET entre agents et gestionnaire. Il offre une visibilité sur l'état du réseau et des périphériques, utile dans les réseaux tactiques militaires [27].

Toutefois, il souffre d'une surcharge liée aux chaînes OID, d'une inefficacité en multidiffusion, et de limites dans les environnements à faible bande passante. Il reste mal adapté aux conditions dynamiques et contraignantes des réseaux militaires mobiles.

2.4 Limites des Protocoles Classiques

Les protocoles classiques de détection des menaces ont longtemps constitué le socle de la protection des infrastructures militaires et des réseaux sensibles. Toutefois, face à la complexité croissante des cyberattaques modernes, ces approches présentent plusieurs limitations structurelles. Leur rigidité, leur faible adaptabilité et leur propension aux erreurs de détection soulignent la nécessité d'intégrer des solutions

plus dynamiques, notamment basées sur l'intelligence artificielle et l'apprentissage automatique.

2.4.1 Manque de Réactivité et d'Adaptabilité

Bien que fiables dans des environnements stables, les protocoles classiques peinent à réagir efficacement face à des menaces évolutives et imprévues. Leur faible capacité d'adaptation en temps réel compromet leur efficacité opérationnelle.

- **Détection d'anomalies** : Ces techniques nécessitent des données historiques de qualité et sont sensibles aux conditions extrêmes (ex. : interférences électromagnétiques, changements topologiques rapides). Les attaques avancées peuvent imiter un comportement normal, réduisant leur efficacité.
- **Surveillance des réseaux** : Malgré leur approche proactive, ces systèmes souffrent de limitations en capacité de calcul, ralentissant la réactivité. Leur modularité requiert également des ajustements complexes dans les contextes militaires.
- **Cyberdéfense traditionnelle** : Reposant sur des règles statiques, ces méthodes sont peu efficaces contre les menaces zero-day et ne permettent pas une réponse agile face aux attaques nouvelles.

2.4.2 Limites dans les Scénarios Complexes

Les menaces modernes sont souvent asymétriques, coordonnées et multi-vecteurs, rendant leur détection difficile avec des outils classiques peu adaptés à ces environnements dynamiques.

- **Détection d'anomalies** : Ces systèmes détectent des tendances générales, mais leur efficacité diminue face à des attaques sporadiques ou furtives.
- **Surveillance réseau** : Leur dépendance à la qualité des données peut limiter leur performance contre des attaques discrètes ou manipulées.
- **Cyberdéfense** : L'analyse limitée aux métadonnées (comme dans NetFlow) ne permet pas d'identifier le contenu malveillant, ce qui est problématique dans le cas d'attaques sophistiquées.

2.4.3 Fiabilité et Faux Positifs

Un défi majeur réside dans le taux élevé de faux positifs et de faux négatifs, qui nuit à la réactivité des opérateurs et affaiblit la chaîne décisionnelle dans les contextes critiques.

- **Systèmes basés sur les anomalies** : La variabilité des environnements militaires peut perturber les modèles, générant des détections erronées ou manquées.
- **Protocoles de surveillance** : La centralisation de certaines fonctions (ex. : gestion des clés dans les WSN ou BIoBT) crée des points de défaillance uniques, augmentant la vulnérabilité.
- **Méthodes traditionnelles** : Ces systèmes nécessitent des mises à jour constantes pour rester pertinents et génèrent un volume de données élevé, complexifiant leur exploitation en temps réel.

2.5 Apport de l'Intelligence Artificielle dans les Systèmes de Détection des Menaces

L'intelligence artificielle révolutionne les systèmes de détection des menaces en renforçant leur capacité à analyser des volumes massifs de données en temps réel et à identifier des attaques sophistiquées avec une précision accrue. Grâce aux avancées du machine learning et du deep learning, les modèles prédictifs permettent d'anticiper les comportements malveillants et d'optimiser les stratégies de défense. Ces technologies offrent une meilleure adaptabilité face aux menaces émergentes, surpassant ainsi les limitations des protocoles classiques.

2.5.1 Avancées des Modèles de Machine Learning pour la Sécurité Militaire

L'IA, en particulier les modèles de ML et DL, offre des progrès considérables dans l'identification des menaces pour les systèmes militaires. Ces technologies dépassent les protocoles traditionnels en matière de réactivité, d'adaptabilité et de précision, tout en diminuant les taux de faux positifs. Voici un examen approfondi de ces progrès :

2.5.1.1 Apprentissage Supervisé et Non Supervisé :

Les modèles ML, qu'ils soient Supervised ou non, présentent des aptitudes de détection des menaces largement supérieures aux techniques conventionnelles.

1. **Apprentissage Supervisé :** est une technique de ML qui s'appuie sur des données annotées pour former des modèles capables de prévoir des résultats sur des données inédites. Chaque donnée d'entrée est liée à une réponse attendue, ce qui permet au modèle de comprendre le lien entre les deux. Ce processus comporte la collecte de données, le choix des caractéristiques, l'apprentissage, l'évaluation et la prédiction [28].

   On l'emploie dans différents secteurs tels que la vision par ordinateur, la finance et la cybersécurité. Par exemple, on est en mesure d'entraîner un modèle supervised afin de repérer des attaques par déni de service (DDoS) ou des intrusions sur le réseau, en se référant à des indicateurs comme les adresses IP, les ports et les volumes de trafic. Ces modèles démontrent une efficacité notable pour repérer des menaces récurrentes ou similaires à celles déjà détectées.

   Bénéfices:
   Détection de menaces connues avec une grande précision.
   Aptitude à faire des généralisations à partir de données passées.
   Réduction des faux positifs grâce à l'apprentissage de motifs précis.

   Applications militaires exemplaires :
   Détection d'intrusions au sein de réseaux tactiques.
   Détection de comportements inhabituels dans les communications cryptées.

2. **Apprentissage Non Supervisé:** est une technique de ML dans laquelle les algorithmes se forment sur des données qui ne sont pas étiquetées, sans disposer de réponses préétablies. À l'opposé de Supervised Learning, qui s'appuie sur des données labellisées pour orienter le modèle, Unsupervised Learning se consacre à l'exploration des données afin de révéler des structures ou des motifs dissimulés. On l'utilise fréquemment dans des méthodes telles que le regroupement (comme le k-means) ou la détection d'anomalies, ce qui permet de repérer des comportements atypiques ou des menaces naissantes sans nécessiter de supervision préalable [28].

Bénéfices :

Aptitude à identifier des attaques zero-day ou des menaces non répertoriées.

Capacité d'adaptation à des environnements en constante évolution.

Diminution de la dépendance vis-à-vis des données historiques étiquetées.

Applications militaires exemplaires :

Surveillance des réseaux de capteurs sans fil afin de repérer des intrusions physiques ou informatiques.

Examen du flux de données réseau pour détecter des activités douteuses dans des contextes à grande vitesse.

2.5.1.2 Deep Learning:

Deep Learning (DL): est une subdivision du ML et de l'IA qui emploie des réseaux de neurones artificiels afin d'examiner des données complexes et non organisées. Il se caractérise par sa faculté à assimiler de vastes quantités de données et à déceler des tendances fines, ce qui lui confère un rôle central dans la quatrième révolution industrielle. Le DL est utilisé dans divers secteurs comme la médecine, la vision informatique, l'analyse textuelle et la sécurité informatique. Néanmoins, l'élaboration de modèles de DL performants demeure un challenge du fait de la dynamique inhérente aux données et des variations liées aux problématiques concrètes, sans oublier la complexité liée à la compréhension du mécanisme interne de ces modèles, fréquemment qualifiés de « boîtes noires ». Le DL comprend diverses méthodes, y compris Supervised Learning, Unsupervised, génératif et hybride, qui trouvent des applications diversifiées dans la réalité quotidienne [29].

1. **Réseaux de Neurones Convolutifs (CNN) :**

   Une architecture d'apprentissage en profondeur fréquemment employée pour l'analyse de données spatiales, comme les images ou les cartes réseau. Il se forme directement à partir des données sans avoir besoin d'extraction de caractéristiques manuellement, ce qui optimise les réseaux neuronaux conventionnels. Constitué de multiples strates (convolution, pooling, etc.), il diminue la complexité du modèle tout en améliorant les performances. Il emploie aussi la méthode du « dropout » afin de prévenir l'ajustement excessif (overfitting). Les CNN

sont appliqués dans divers domaines, notamment la détection d'anomalies dans les flux de données ou l'identification de menaces physiques, comme les drones hostiles [29].

Bénéfices :

Aptitude à gérer des données en plusieurs dimensions.

Détection précise des motifs complexes.

Applications militaires exemplaires :

Examen des images satellites pour repérer des déplacements de forces militaires ou des infrastructures douteuses.

Observation des réseaux optiques afin de détecter des perturbations physiques.

2. **Réseaux de Neurones Récurrents (RNN) et LSTM :**

Les RNN et LSTM (Long Short-Term Memory) sont des systèmes d'intelligence artificielle conçus pour traiter des données en séquence ou temporelles, comme les journaux de réseau ou les échanges de communication. Ils se caractérisent par leur aptitude à exploiter une « mémoire » qui garde les informations des entrées antérieures, ce qui leur permet d'évaluer des séquences prolongées et de repérer des irrégularités ou des motifs en temps réel. Toutefois, les RNN traditionnels rencontrent le problème des « gradients qui s'évanouissent », ce qui complique l'apprentissage de longues séquences. Des améliorations telles que les LSTM ont été mises au point pour résoudre ce problème, offrant des performances supérieures dans les applications réelles [29].

Bénéfices :

Aptitude à représenter des dépendances temporelles étendues.

Capacité d'adaptation à des environnements en constante évolution.

Applications militaires exemplaires :

Surveillance en temps réel des intrusions dans les réseaux militaires.

Examen des transmissions radio pour repérer des signaux suspects.

3. **Autoencodeurs :**

Le modèle d'autoencodeur (Autoencoder) est un d'apprentissage non supervisé employé pour la diminution de la dimensionnalité, l'extraction de caractéristiques

et l'identification des anomalies. Il est composé de trois éléments clés : l'encodeur qui condense les informations en une représentation concentrée, le code (couche intermédiaire), et le décodeur qui reconstitue les données initiales à partir de cette représentation. L'autoencodeur est formé pour reproduire des données standards, ce qui lui donne la capacité de détecter les déviations (anomalies) lorsqu'elles surviennent. Il est couramment employé dans des applications comme la modélisation générative, l'élimination du bruit et la détection d'anomalies. Pour des missions plus complexes, on utilise des variantes régularisées telles que les autoencodeurs épars (Sparse) et variationnels (Variational) [29].

Bénéfices :

Aptitude à identifier des menaces non identifiées sans intervention humaine.

Diminution des fausses alertes grâce à une représentation exacte des comportements standards.

Applications militaires exemplaires :

Surveillance des réseaux de capteurs sans fil pour repérer des attaques discrètes.

Examen des journaux système afin de détecter des actions malintentionnées.

2.5.1.3 Avantages par Rapport aux Protocoles Classiques

Les modèles ML et DL surpassent les méthodes traditionnelles de détection des menaces sur divers points :

1. **Réactivité et Adaptabilité :**

   Contrairement aux protocoles traditionnels basés sur des règles immuables, les modèles de ML et DL ont la capacité de s'ajuster en temps réel face à de nouvelles menaces.

   Ils ont la capacité d'apprendre de manière continue à partir des données, ce qui leur confère une résilience face aux attaques en constante évolution.

2. **Précision et Réduction des Faux Positifs :**

   Les modèles d'apprentissage automatique et profond sont capables de détecter des schémas complexes et subtils, ce qui diminue le nombre de faux positifs.

   Ils ont la capacité de distinguer les anomalies malveillantes des fluctuations normales dans des contextes dynamiques.

3. **Scalabilité :**

   Les modèles ML et DL sont aptes à être déployés à une échelle considérable, ce qui est crucial pour la complexité et l'étendue des réseaux militaires.

   Ils ont la capacité de gérer d'énormes quantités de données en temps réel, contrairement aux approches conventionnelles qui peuvent être entravées par des points de congestion.

4. **Détection de Menaces Inconnues :**

   Les modèles non supervisés et les autoencodeurs ont la capacité de détecter des menaces nouvelles ou zero-day, ce qui n'est pas réalisable avec les protocoles traditionnels reposant sur des signatures.

2.5.2 Intelligence Artificielle pour l'Optimisation des Stratégies de Défense

L'intelligence artificielle (IA) modifie radicalement les stratégies de défense militaire en affinant les réactions face aux menaces et en perfectionnant la gestion des ressources. Voici les éléments essentiels :

1. **Optimisation des réponses aux menaces**

   L'intelligence artificielle renforce la rapidité, la précision et l'efficacité de la réaction face aux menaces multiples dans le domaine militaire.

   - **Systèmes de réponse automatisée :**
     - Détection et élimination rapide des menaces (cyberattaques, drones hostiles, etc.).
     - Réduction du délai de réponse et minimisation des erreurs humaines.
   - **Simulation et prédiction :**
     - Anticipation des comportements ennemis via la simulation de scénarios tactiques.
     - Aide à la décision stratégique en fonction de la probabilité d'occurrence des menaces.
   - **Coordination des systèmes de défense :**

- Intégration des dispositifs de cybersécurité, d'électronique et de sécurité physique pour une réponse coordonnée.
- Optimisation de l'utilisation des ressources humaines et matérielles.

2. **Amélioration de la gestion des ressources**
   L'IA permet une allocation efficace et anticipée des ressources logistiques et humaines, essentielle à l'efficience militaire.

   - **Optimisation logistique :**
     - Prédiction des besoins en équipements, carburants, et munitions.
     - Amélioration des chaînes d'approvisionnement et des itinéraires de transport.
   - **Gestion du personnel :**
     - Répartition optimale des soldats selon leurs compétences et les exigences de la mission.
     - Planification des rotations pour limiter la fatigue et maintenir l'efficacité opérationnelle.
   - **Analyse des données :**
     - Traitement massif de données issues de sources variées (imagerie satellite, capteurs, communications).
     - Identification précoce de schémas menaçants ou d'anomalies cachées.

3. **Renforcement de la résilience et de la sécurité**
   L'IA assure une protection proactive et adaptative contre les menaces émergentes, renforçant la cybersécurité militaire.

   - **Cyberdéfense adaptative :**
     - Surveillance en temps réel des réseaux militaires et réponse automatisée aux attaques.
     - Protection anticipée contre les menaces zero-day et les logiciels malveillants.
   - **Sécurité des communications :**

- Détection des tentatives d'interception ou de brouillage.
- Basculement automatique vers des canaux chiffrés ou sécurisés.

2.5.3 Applications Pratiques et Perspectives Futures

L'intégration de ML et de IA dans les systèmes militaires modernes a déjà produit des applications tangibles et encourageantes. Ces innovations révolutionnent les opérations militaires en augmentant l'efficience, la précision et la rapidité d'intervention. Voici un aperçu des scénarios pratiques et des perspectives à futur :

2.5.3.1 Applications Pratiques

1. **Drones Autonomes** sont des aéronefs sans pilote capables de fonctionner de manière indépendante, sans nécessiter un contrôle humain direct. Grâce à l'intégration de technologies d'IA, ces drones peuvent percevoir leur environnement, prendre des décisions et accomplir des tâches complexes de façon automatique. Dotés de systèmes d'IA, ils sont devenus un élément essentiel des opérations militaires modernes [30].

   Fonctionnement : Les drones autonomes fonctionnent grâce à des algorithmes d'IA qui leur confèrent la capacité de :

   - Vision par ordinateur (Computer Vision) : Ils font appel à des caméras de haute définition et à des capteurs LiDAR pour recueillir et analyser des informations visuelles en temps réel. Des méthodes telles que la reconnaissance d'images et la détection d'objets contribuent à repérer les obstacles et à surveiller des éléments d'intérêt.
   - Machine Learning : Les algorithmes ont été formés sur d'importants ensembles de données issus des vols de drones, ce qui leur offre la capacité de s'ajuster à diverses situations et d'optimiser leur performance en vol.
   - Planification de trajectoire et évitement d'obstacles : Les algorithmes examinent des informations en direct telles que la vitesse du vent, le type de terrain et les obstructions, et modifient de manière dynamique les itinéraires de vol.

- Prise de décision : Les drones ont la capacité d'analyser les informations provenant des capteurs et de répondre à des situations inattendues, telles que des modifications météorologiques brusques ou l'émergence d'obstacles.

Applications: On utilise les drones autonomes dans divers secteurs, tels que :

- Recherche et sauvetage : Ils sont déployés dans des régions touchées par des catastrophes ou des terrains ardues pour trouver d'éventuels survivants, mesurer les dommages et communiquer des renseignements essentiels aux équipes de sauvetage.
- Livraison par drones : Ces drones évoluent de façon autonome dans les espaces urbains, conçoivent des trajets optimaux et esquivent les conditions météorologiques défavorables afin de livrer des paquets avec célérité et efficacité.
- Agriculture : Ils servent à surveiller les cultures de manière précise, réaliser des pulvérisations automatisées et cartographier les champs, contribuant ainsi à la réduction du gaspillage et à l'optimisation de l'exploitation des ressources.
- Inspection des infrastructures : Les drones effectuent l'inspection de structures telles que les ponts, les canalisations et les fils électriques, identifiant les anomalies et produisant des comptes rendus précis pour l'entretien.
- Surveillance et reconnaissance : Ils recueillent des informations en direct dans des régions périlleuses ou menaçantes, sans compromettre la sécurité de l'équipe humaine.
- Frappes ciblées : Dans le secteur militaire, les drones autonomes mènent des opérations de combat avec une précision élevée, réduisant ainsi les dégâts collatéraux.
- Logistique : Ils facilitent le transport autonome de biens vers des régions difficilement accessibles, telles que les zones de conflit ou les régions éloignées.

Exemples concrets :

- Les drones MQ-9 Reaper emploient l'intelligence artificielle pour détecter et suivre des cibles.

- Les drones Switchblade sont aptes à réaliser des missions kamikazes autonomes avec une exactitude chirurgicale.

2. **Systèmes de Prédiction d'Actions Ennemies**, utilisent l'IA pour analyser les données de renseignement et prévoir les mouvements et tactiques de l'adversaire. Ces systèmes, propulsés par des moteurs d'intelligence artificielle sophistiqués, simulent les actions adverses, élaborent des situations crédibles pour la formation et la stratégie, et proposent des manœuvres possibles de l'adversaire. Ils visent à renforcer l'entraînement militaire et à minimiser l'impact de la surprise en favorisant des choix proactifs et informés [31].

   Fonctionnement: Ces systèmes reposent sur une simulation interactive gérée par une intelligence artificielle.

   - Les unités militaires (qu'elles soient alliées et ennemies) sont représentées comme des entités indépendantes possédant un « cerveau » (brain) qui leur offre la capacité de faire des choix autonomes basés sur leur perception de l'environnement tactique.
   - Des techniques statistiques telles que la méthode de Monte Carlo sont employées pour modéliser divers résultats liés aux scénarios tactiques.
   - On intègre les données de renseignement (positions, état, attitudes de l'ennemi)sont intégrées pour générer des prédictions sur les actions ennemies.

   Applications:

   - Entraînement militaire :Ces systèmes sont exploités pour instruire les officiers de l'état-major (brigades, divisions) à la prise de décisions dans des contextes complexes.
   - Planification opérationnelle :Ils assistent dans la conception et l'évaluation de plans d'action (Courses of Actions) en mettant en parallèle divers scénarios.
   - Aide à la décision :Ils offrent des stratégies d'action amies ou adverses, permettant aux officiers d'examiner des alternatives auxquelles ils n'auraient peut-être pas pensé.

- Analyse logistique : Ils proposent des prévisions des exigences logistiques basées sur les scénarios simulés.

Exemples Concrets :

- IA comportementale : L'IA est utilisée pour modéliser des comportements adverses en réponse à l'environnement tactique, ce qui favorise l'apparition de comportements inattendus et enrichit la planification des opérations.

3. **Systèmes de Cyberdéfense Adaptative**

Les systèmes de cyberdéfense adaptative sont des structures de sécurité exploitant des technologies de pointe, telles que l'intelligence artificielle et le machine learning, pour défendre les systèmes d'information en se modifiant dynamiquement face aux menaces changeantes. Ils assurent une surveillance en temps réel, identifient les irrégularités, évaluent sans cesse les dangers et réagissent de manière autonome aux incidents, procurant ainsi une défense proactive et adaptable [32].

Fonctionnement:

- **Surveillance continue et analyse comportementale :**
- Utilisation de l'IA et de l'apprentissage automatique pour surveiller en temps réel.
- Détection des anomalies et comportements suspects via des algorithmes sophistiqués.
- **Évaluation dynamique des risques :**
- Évaluation constante des menaces pour chaque actif et flux d'informations.
- Adaptation des mesures de sécurité selon le contexte et les menaces.
- **Réponse automatisée et orchestrée :**
- Systèmes automatiques conçus pour isoler les systèmes compromis ou stopper les menaces.
- Mise en œuvre des actions de remédiation pour minimiser le délai de réaction.
- **Apprentissage continu :**

- Optimisation des modèles de détection et de réaction à l'aide des informations provenant d'incidents antérieurs.

Applications :

- **Détection précoce des menaces :**
- Identification rapide des indices d'infection à l'aide de l'analyse comportementale.
- Prévention des attaques par le biais de l'analyse prédictive.
- **Réduction de la surface d'attaque :**
- Segmentation dynamique du réseau pour isoler les systèmes compromis.
- Gestion intelligente des correctifs pour prioriser les mises à jour critiques.
- **Conformité réglementaire :**
- Assistance pour se conformer aux normes du RGPD et de la directive NIS en adaptant les mesures de sécurité selon les risques.
- **Optimisation des ressources :**
- Priorisation des alertes et allocation des ressources en fonction des menaces les plus critiques.

Exemples concrets :

- Des plateformes de cybersécurité telles que Darktrace sont mises en place pour sauvegarder les infrastructures militaires.
- Le système HACCS (Hyper-enabled Autonomous Cyber Combat System) emploie l'intelligence artificielle pour détecter et éliminer les menaces dans le cyberespace.

2.5.3.2 Perspectives Futures

Ce titre explore les tendances, innovations et scénarios futurs possibles liés au sujet étudié.

1. **Amélioration des Algorithmes de ML**

- Apprentissage Fédéré :Apprentissage décentralisée pour renforcer la confidentialité.
- Renforcement de l'Apprentissage :Adaptation en temps réel aux modifications sur le terrain.

2. **Défense Contre les Menaces Quantiques**
   - Développement de systèmes de chiffrement résistants aux attaques quantiques.
3. **Amélioration de la Résilience des Réseaux**
   - Réseaux autonomes capables de s'auto-réparer en cas de dommages ou d'attaques.
4. **Intégration de l'IA dans les Armes Hypersoniques**
   - Les armes hypersoniques dotées d'intelligence artificielle seront capables de modifier leur trajectoire en direct pour contourner les systèmes de défense adverses et toucher leurs objectifs avec une exactitude sans précédent.

2.6 Conclusion

Ce chapitre nous a permis de faire un tour d'horizon des principales approches utilisées pour la détection des menaces dans le domaine militaire. En comparant les protocoles classiques et les méthodes fondées sur l'intelligence artificielle, nous avons pu mettre en évidence les limites des systèmes traditionnels face à la complexité croissante des attaques et à l'évolution rapide des environnements opérationnels. L'intelligence artificielle, avec ses capacités d'analyse prédictive et d'adaptation, représente aujourd'hui une avancée significative dans le renforcement des systèmes de défense.

Dans le chapitre suivant, nous allons nous concentrer sur notre contribution propre à ce domaine, à travers l'analyse d'un protocole d'authentification biométrique basé sur la blockchain, conçu spécifiquement pour les environnements militaires connectés (IoBT). Nous détaillerons son fonctionnement, ses avantages et les choix techniques qui sous-tendent sa mise en œuvre.

Chapter 3

Protocole Blockchain-Based Autonomous Authentication and Integrity for IoBT in C3I System (BioBT)

3.1 Introduction

Le protocole BioBT (Blockchain-Based Autonomous Authentication and Integrity for IoBT in C3I System). Cette approche novatrice vise à renforcer la sécurité des systèmes militaires en intégrant la blockchain Ethereum pour garantir une authentification autonome et une intégrité des données au sein de l'Internet des Objets de Champ de Bataille (IoBT) et du système C3I (Commandement, Contrôle, Communications et Renseignements). L'architecture proposée combine des contrats intelligents, des mécanismes cryptographiques et une gestion décentralisée pour répondre aux défis de confiance, de réactivité et de résilience dans les environnements militaires dynamiques. Les phases techniques détaillées (création du système, association des dispositifs, transmission des données et révocation) illustrent comment BioBT optimise les stratégies de défense tout en minimisant les risques de compromission.

3.2 Protocole BIoBT

La technique BIoBT vise à fournir une authentification autonome et une intégrité des données pour IoBT dans un système C3I, en exploitant la blockchain Ethereum. Cette section détaille son architecture, ses composants clés, et son fonctionnement technique.

3.3 Architecture de BIoBT dans le système C3I

3.3.1 Acteurs clés

- **Régulateur d'infrastructure** : Gère l'enregistrement des IoBT et déploie des politiques via un portefeuille Ethereum.
- **Propriétaire de l'infrastructure** : Déploie les contrats intelligents et gère les autorisations.
- **Unité de contrôle C3I** : Traite les données reçues des IoBT.
- **Unité de commande C3I** : Génère des plans d'action basés sur les données traitées.

3.3.2 Composants connectés

La Figure 3.1 présente les composants interconnectés :

- **IoBT** : Entités hétérogènes (véhicules, capteurs, etc.).
- **Système de Commandement, Contrôle et Communication (CCC_Sys)** : Nœud complet de la blockchain.
- **Blockchain publique** : Stocke les associations et valide les transactions via des contrats intelligents.
- **Stockage** : Base de données privée ou IPFS pour les données traitées.

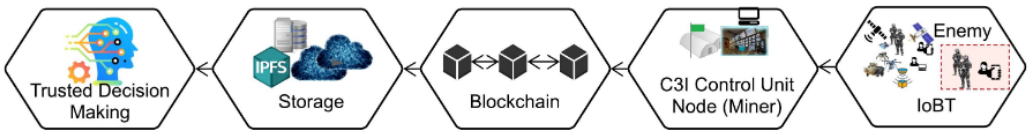

Figure 3.1: Composants du système BIoBT [1].

3.3.3 Vue abstraite du système C3I

Le système C3I collecte des données tactiques via des dispositifs tels que capteurs, drones et soldats équipés de wearables. Ces données sont traitées pour générer des plans d'action optimaux. La Figure 3.2 illustre cette vue abstraite.

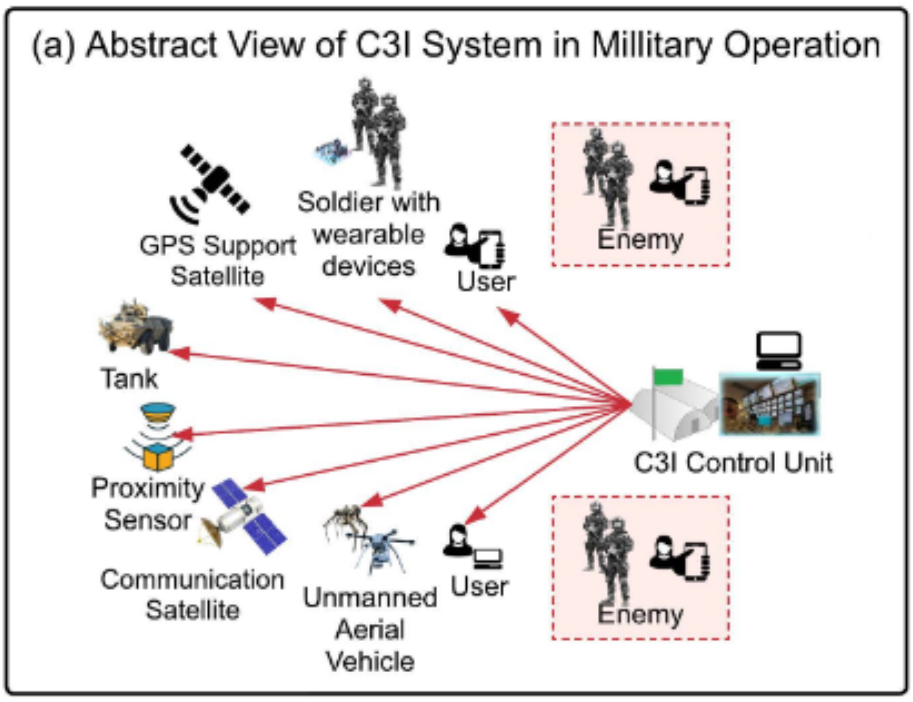

Figure 3.2: Vue abstraite du système C3I [1].

3.3.4 Symboles et leurs définitions

Le Tableau 3.1 présente la signification des principaux symboles utilisés dans le protocole BIoBT.

3.3.5 Phases Techniques Détaillées de BIoBT

L'objectif principal est d'établir une association entre le CCC et les IoBT pour garantir l'authenticité et l'intégrité. Les transactions blockchain (Tx-1 à Tx-4) gèrent les phases critiques :

Table 3.1: Symboles utilisés dans le système Biobot.

Symbol	**Description**
$\text{AssociationID}^*_{\text{IoBT}}$	IoBT's auto-generated identity
$\text{AssociationID}_{\text{IoBT}}$	IoBT's identity for association creation
$h(.)$	Hash
$\text{ID}_{\text{IoBT Owner}}$	IoBT's owner's identity
ID_{IoBT}	IoBT's identity
$\text{ID}_{\text{CCC Sys}}$	CCC_Sys's identity
$\text{ID}^*_{\text{CCC Sys}}$	CCC_Sys's auto-generated identity
ID_{CCC}	CCC's Identity
SC_{BioBT}	BioBT's Smart contract

Phase 1 : Mise en place du système `CCC_Sys`

Cette phase vise à initialiser un système d'identité décentralisé et sécurisé basé sur la blockchain pour enregistrer et authentifier un `Command and Control Center` (CCC) dans un environnement IoBT. Cette phase est illustré en Figure 3.3.

- **Objectif :**
  Concevoir et déployer un système d'identification unique pour chaque CCC, nommé `CCC_Sys`, grâce à un smart contract blockchain permettant une association sécurisée avec les dispositifs IoBT.

- **Étape 1 : Conception du Smart Contract `SC_BioBT`**
  Création d'un contrat intelligent intégrant :
  - les règles d'authentification,
  - les logiques d'association entre CCC et dispositifs IoBT,
  - les conditions de validation des transactions.

- **Étape 2 : Génération de l'identité du propriétaire ($\texttt{ID}_{\texttt{CCC_Owner}}$)**
  Une paire de clés cryptographiques (souvent ECDSA) est générée pour représenter le propriétaire du CCC.

- **Étape 3 : Attribution de l'identité du CCC ($\texttt{ID}_{\texttt{CCC}}$)**
  Une identité unique est assignée au CCC, généralement sous forme de hachage de métadonnées du dispositif.

- **Étape 4 : Calcul de l'identifiant $\texttt{ID}_{\texttt{CCC_Sys}}$**

  Un identifiant système unique est créé en concaténant $\texttt{ID}_{\texttt{CCC_Owner}}$ et $\texttt{ID}_{\texttt{CCC}}$, suivi d'un hachage avec SHA-256 :

$$\mathrm{ID}_{CCC_Sys} = h(\mathrm{ID}_{CCC_Owner} \parallel \mathrm{ID}_{CCC}) \tag{3.1}$$

- **Étape 5 : Envoi de la transaction `Tx1`**

  La transaction `Tx1` contenant le code du contrat et l'identifiant du système est diffusée sur la blockchain :

$$\mathrm{Tx}_1 = [\mathrm{ID}_{CCC_Sys}, \mathrm{bytecode_de_SCBioBT}] \tag{3.2}$$

- **Étape 6 : Validation de `Tx1` par le réseau blockchain**

  Les nœuds Ethereum procèdent à :

  - la vérification de la signature de $\texttt{ID}_{\texttt{CCC_Owner}}$,
  - le contrôle de l'unicité de $\texttt{ID}_{\texttt{CCC_Sys}}$,
  - la détection de conflits ou duplications avec d'autres CCC déployés.

  Résultat possible :

  - *Succès* : La transaction est intégrée dans un bloc et le contrat est activé.
  - *Échec* : La transaction est rejetée (ex. signature invalide).

- **Étape 7 : Activation du système `CCC_Sys`**

  En cas de validation, `CCC_Sys` devient opérationnel avec :

  - un contrat SCBioBT déployé à une adresse unique,
  - un identifiant immuable enregistré sur la blockchain.

Phase 2 : Association des dispositifs IoBT au `CCC_Sys`

Cette phase vise à associer des dispositifs IoBT (capteurs, drones, véhicules, etc.) au système `CCC_Sys` de manière sécurisée et immuable à l'aide de la blockchain. Le processus est résumé en Figure 3.4.

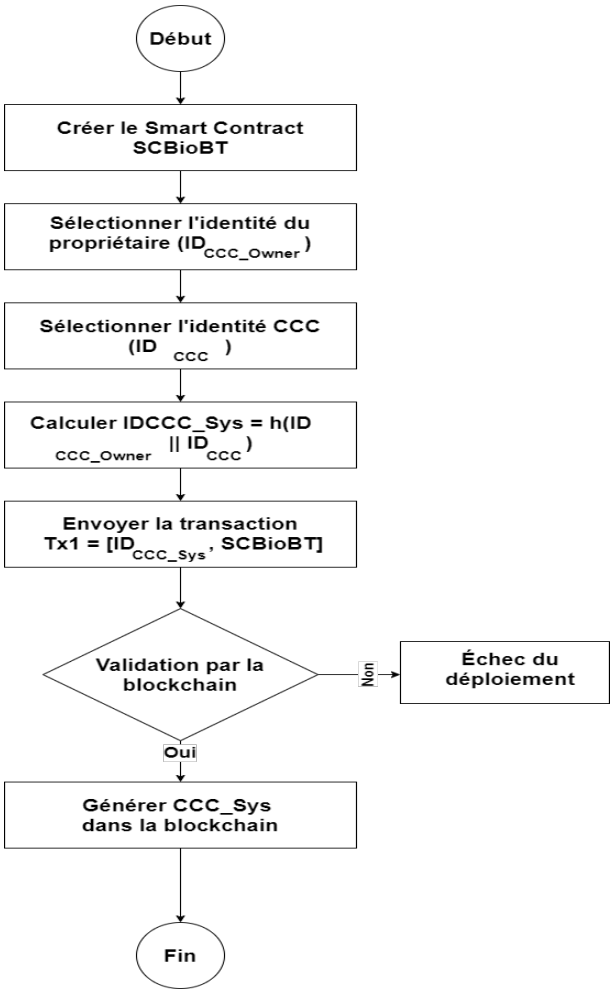

Figure 3.3: Architecture de création du système `CCC_Sys`

- **Étape 1 : Création des identités**
  - **Acteurs :** `CCC Owner` (gestionnaire) et l'`IoBT` (dispositif).
  - **Identifiants générés :**
    - $\texttt{ID}_{\texttt{IoBT_Owner}}$: identité de l'unité militaire responsable.
    - $\texttt{ID}_{\texttt{IoBT}}$: identité unique de l'équipement (ex. numéro de série).
- **Étape 2 : Calcul de l'identifiant d'association**
  - L'identifiant d'association est généré comme suit :

$$\texttt{AssociationID}_{\texttt{IoBT}} = h(ID_{IoBT_Owner} \parallel ID_{IoBT}) \tag{3.3}$$

  - Cela garantit un lien immuable entre l'IoBT et son propriétaire.
- **Étape 3 : Transmission de la transaction** `Tx2`

  - La transaction contient :
    * $\texttt{ID}_{\texttt{CCC_Sys}}$: identifiant du système précédemment créé.
    * $\texttt{AssociationID}_{\texttt{IoBT}}$: identifiant unique d'association.
  - Elle est validée par le smart contract `SC_BioBT` qui vérifie :
    * L'activation du `CCC_Sys`,
    * L'unicité de l'ID d'association.

- **Étape 4 : Enregistrement sur la blockchain**
  - En cas de validation réussie :
    * Un identifiant unique $\texttt{AssociationID}_{\texttt{IoBT*}}$ est généré (ex. adresse Ethereum).
    * La blockchain enregistre :

      $\texttt{AssociationID}_{\texttt{IoBT}}$

      $\texttt{AssociationID}_{\texttt{IoBT*}}$

      $\texttt{ID}_{\texttt{CCC}}$
  - L'IoBT reçoit :
    * ses identifiants (propriétaire et appareil),
    * `TxID` : identifiant de transaction autorisant l'échange de données.

- **Sécurité et garanties assurées**
  - **Immuabilité :** Enregistrement permanent dans la blockchain.
  - **Authenticité :** Seuls les IoBTs autorisés peuvent interagir.
  - **Anti-doublon :** Aucune redondance d'enregistrement permise.

Phase 3 : Authentification et Transmission des Données

Cette phase permet de vérifier automatiquement l'authenticité d'un dispositif IoBT ainsi que l'intégrité des données transmises avant leur réception par le `CCC_Sys`. Le processus est illustré en Figure 3.5.

- **Étape 1 : Préparation initiale**
  - **Identifiants nécessaires :**

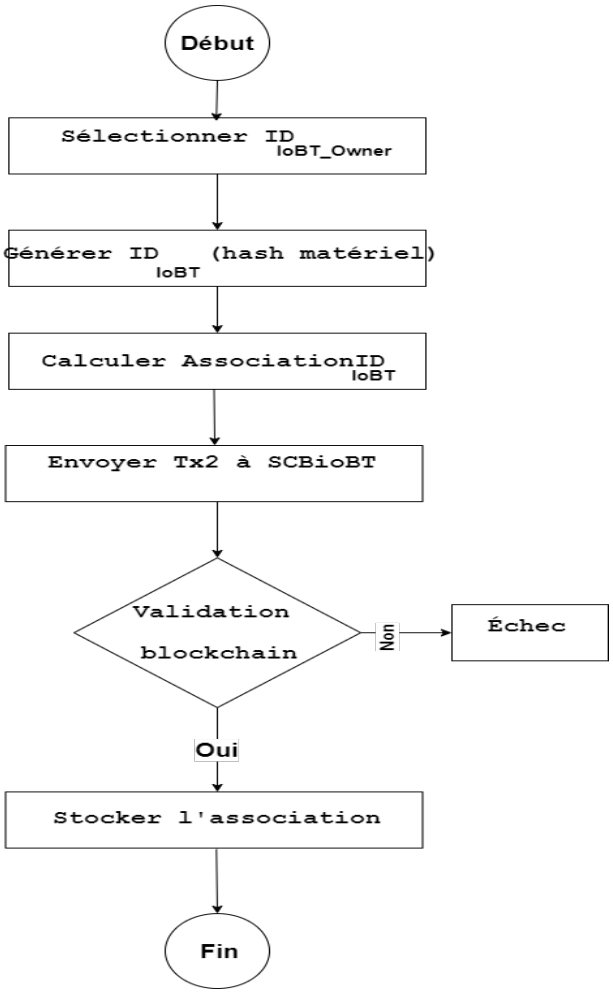

Figure 3.4: Processus d'association sécurisée entre IoBT et `CCC_Sys`

- $\texttt{ID}_{\texttt{CCC_Sys}}$: identifiant du système créé en Phase 1.
- $\texttt{AssociationID}_{\texttt{IoBT}}$: identifiant d'association généré en Phase 2.

- **Étape 2 : Collecte et traitement des données**
  - L'IoBT collecte des données opérationnelles (ex. images, coordonnées, mesures).
  - Il en génère un condensé numérique via une fonction de hachage :

$$\texttt{Digest} = \text{SHA-256}(\texttt{Data}) \tag{3.4}$$

- **Étape 3 : Transmission de la transaction `Tx3`**
  - La transaction envoyée est structurée comme suit :

$$\texttt{Tx}_3 = \{ID_{CCC_Sys}, AssociationID_{IoBT}, Data, Digest\} \tag{3.5}$$

- **Étape 4 : Vérification d'authenticité**
  - Le smart contract compare l'identifiant reçu à celui stocké :

$$\texttt{AssociationID}_{\texttt{IoBT}} \stackrel{?}{=} \texttt{AssociationID}^{*}_{\texttt{IoBT}} \tag{3.6}$$

  - **Décision :**
    - Si égalité : poursuite du traitement.
    - Sinon : rejet immédiat de la transaction.
- **Étape 5 : Validation de l'intégrité des données**
  - La blockchain recalcule le haché des données reçues.
  - Si `Digest` concorde :
    - Transmission au `CCC_Sys` validée.
  - En cas de discordance (ex. données falsifiées, usurpation) :
    - `Tx3` est rejetée.

Phase 4 : Révocation d'Association

Cette phase vise à désactiver un dispositif IoBT devenu compromis, suspect ou obsolète, afin de maintenir la fiabilité du `CCC_Sys`. Le processus est représenté en Figure 3.6.

1. Objectifs de la révocation

- **Éliminer les risques** : Retirer les IoBT piratés, perdus ou compromis.
- **Préserver l'intégrité du système** : Garantir que seules les entités authentiques sont autorisées à transmettre des données.

2. Acteurs impliqués

- `Propriétaire du CCC` : Autorité de gestion qui initie la révocation.
- `Contrat intelligent (`$\texttt{SC}_{\texttt{BioBT}}$`)` : Module de vérification et d'exécution.
- `Blockchain` : Mécanisme de journalisation immuable des événements.

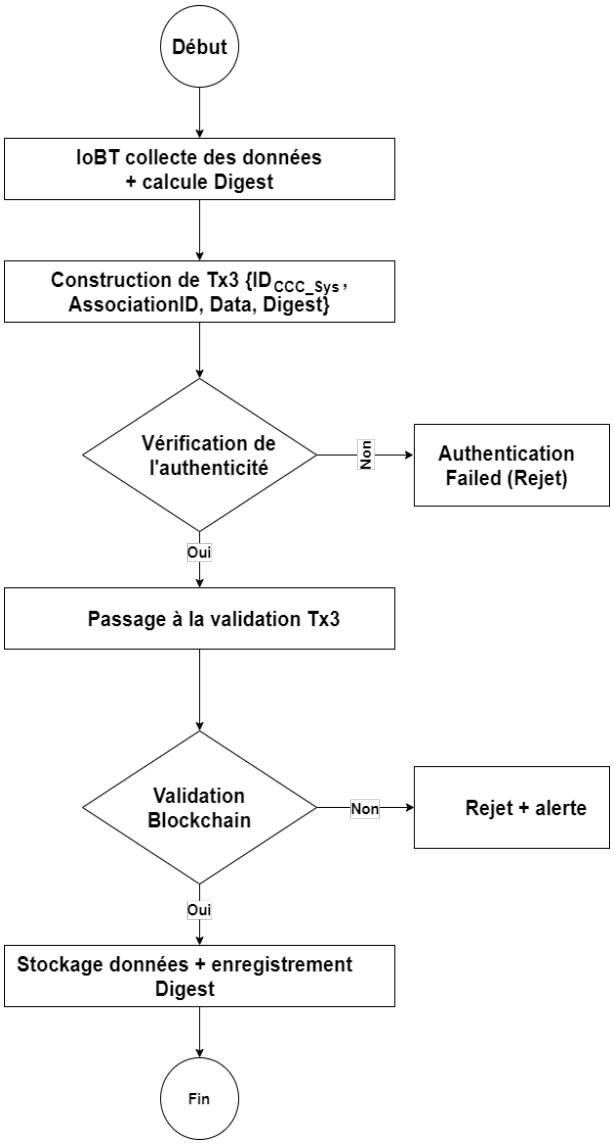

Figure 3.5: Authentification et validation d'intégrité des données par la blockchain avant acceptation par le `CCC_Sys`.

3. Détail du processus de révocation

- **Etape 1 : Demande de révocation**
  - Le propriétaire du CCC cible un IoBT à désactiver (ex : entité compromise).
  - Il émet une transaction `Tx4` contenant :
    - $\texttt{AssociationID}_{\texttt{IoBT}}$ ou $\texttt{AssociationID}_{\texttt{IoBT*}}$.
    - Un motif éventuel de révocation (chiffré ou non).
- **Etape 2 :Vérification d'autorisation**

  - Le contrat SC_{BioBT} s'assure que seule l'identité du `CCC_Owner` est habilitée à initier la transaction.
  - Il vérifie l'existence de l'IoBT dans `CCC_Sys`.

- **Etape 3 :Révocation effective**
  - **Si validation réussie** :
    - L'entrée correspondante est supprimée du registre.
    - Un événement de révocation est consigné dans la blockchain.
  - **Si échec** :
    - La transaction `Tx4` est rejetée automatiquement.

4. Conséquences de la révocation

- **Blocage d'accès** : L'IoBT ne peut plus transmettre de données.
- **Historique préservé** : Les transmissions antérieures restent consultables.
- **Notification** : Avertissement aux unités alliées du réseau si nécessaire.

5. Garanties de sécurité

- **Transparence** : Les nœuds autorisés ont accès aux événements de révocation.
- **Immuabilité** : Les logs de révocation sont enregistrés de manière permanente.
- **Décentralisation** : Aucun point de contrôle unique n'intervient dans le processus.

3.3.6 Limitations de BIoBT

Malgré ses apports notables en matière de sécurité et de gestion décentralisée des entités IoBT, le protocole BIoBT présente plusieurs limitations structurelles et opérationnelles qui doivent être prises en compte avant une adoption à large échelle.

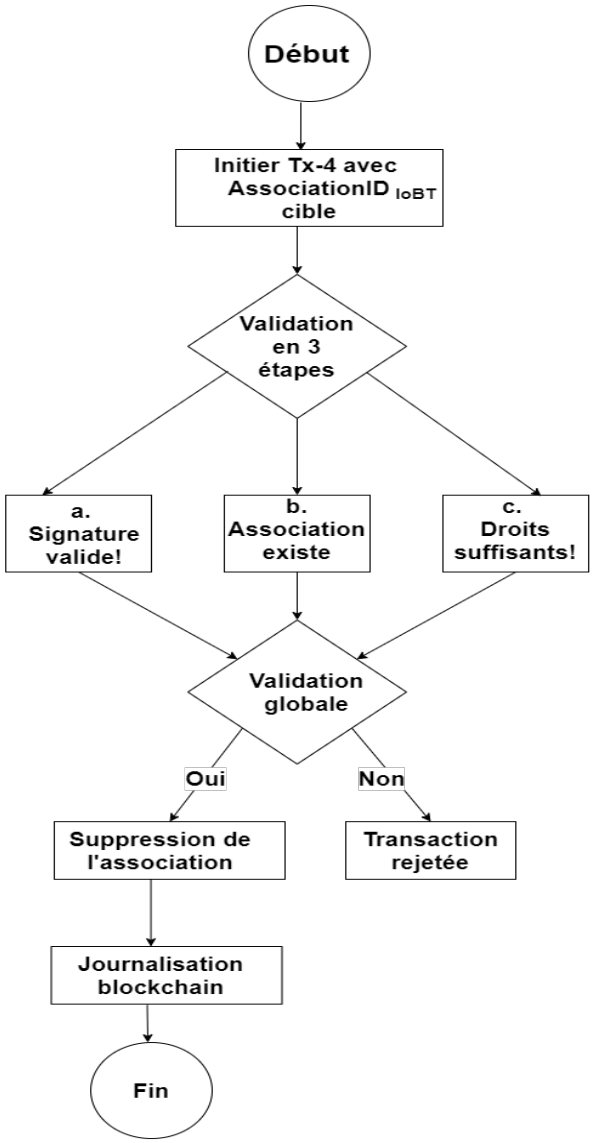

Figure 3.6: Suppression sécurisée d'un IoBT du registre `CCC_Sys` via révocation.

3.3.6.1 Confidentialité sur une Blockchain Publique

Même si les identifiants sont pseudonymisés, l'utilisation d'une blockchain publique expose les métadonnées liées aux transactions (fréquence, taille, horodatage, relations entre adresses, etc.). Ces informations peuvent faire l'objet d'analyses croisées par des attaquants, exposant potentiellement des schémas tactiques ou des activités sensibles.

$$\lambda = \frac{N}{T} \tag{3.7}$$

où N est le nombre de transactions similaires et T la période observée. Une telle estimation de la fréquence d'activité permettrait à un adversaire de déduire des comportements répétitifs, remettant en question l'anonymat opérationnel.

3.3.6.2 Absence de Validation à Grande Échelle

Le prototype BIoBT a été testé dans un environnement restreint avec un nombre limité d'entités IoBT. Son comportement dans un scénario réaliste impliquant des milliers de nœuds mobiles, des réseaux instables et des charges variables reste une inconnue.

$$S = n \cdot p \tag{3.8}$$

où n est le nombre de nœuds IoBT et p la probabilité qu'une transaction réussisse dans un environnement bruité. Une telle approximation du taux de succès montre la sensibilité du système à l'instabilité réseau.

3.3.6.3 Forte Dépendance à Ethereum

La mise en œuvre actuelle repose sur la plateforme Ethereum, ce qui implique une dépendance directe à ses limitations :

- **Scalabilité limitée** : Le nombre de transactions par seconde est restreint (environ 15 à 30).

$$D \approx \frac{k}{30} \tag{3.9}$$

  avec k IoBT simultanés.

- **Coût variable (Gas)** :

$$C(g) = g \cdot P \tag{3.10}$$

  où g est le gas utilisé et P le prix du gas à un instant t.

- **Latence de validation** : Les blocs peuvent mettre plusieurs secondes à être validés, ce qui ralentit la synchronisation des IoBT.

3.3.6.4 Complexité d'Intégration dans l'Infrastructure Militaire

L'intégration de BIoBT nécessite une adaptation des systèmes existants (capteurs, UAV, réseaux tactiques). Cette interopérabilité technique et sécuritaire peut s'avérer complexe, voire risquée, si les normes ne sont pas alignées. De plus, la maintenance

des smart contracts et des registres nécessite une expertise blockchain peu répandue dans les forces armées.

$$I = \alpha \cdot c \cdot (1 - \delta) \tag{3.11}$$

où c est le nombre de composants à interfacer, δ le degré de compatibilité et α un facteur de coûts.

3.3.6.5 Risques de Centralisation de l'Autorité CCC

Bien que décentralisé dans l'enregistrement des identités, le modèle BIoBT reste dépendant d'un acteur central (le CCC Owner) pour les décisions critiques (révocation, authentification, déploiement). Cela crée un point de vulnérabilité potentiel.

$$P_f = P_{compromise} + P_{erreur} \tag{3.12}$$

avec $P_{compromise}$ la probabilité de compromission, et P_{erreur} celle d'une erreur humaine.

3.4 Conclusion

Dans ce chapitre, nous avons présenté en détail le protocole BIoBT, une solution innovante basée sur la blockchain pour l'authentification autonome et sécurisée dans les systèmes IoBT. À travers la description de son architecture, de ses phases techniques et de son environnement d'expérimentation, nous avons mis en évidence sa capacité à garantir l'intégrité des communications et la fiabilité des échanges dans un contexte militaire sensible.

Le prochain chapitre viendra justement compléter cette approche en y intégrant un modèle d'intelligence artificielle, plus précisément un LSTM, afin de renforcer l'authentification biométrique dans un environnement contraint, et de pousser encore plus loin l'intelligence du système.

Chapter 4

Amélioration du Protocole BioBT par un Modèle LSTM Basé sur le Deep Learning pour l'Authentification Biométrique

4.1 Introduction

Dans ce chapitre, nous présentons l'amélioration du protocole BioBT par l'intégration d'un modèle LSTM basé sur le Deep Learning pour l'authentification biométrique dans le contexte des systèmes IoBT (Internet of Battlefield Things). Cette démarche vise à pallier les limites des approches classiques en matière de sécurité et d'authentification, en exploitant la capacité des réseaux LSTM à traiter des données temporelles complexes. Après avoir exposé les motivations et les objectifs de cette intégration, nous détaillons le processus d'exploration, d'analyse et de prétraitement des données biométriques utilisées. L'architecture du modèle LSTM, le choix des hyperparamètres, les stratégies de régularisation ainsi que les métriques d'évaluation sont ensuite décrits. Enfin, une attention particulière est portée à la présentation et à l'analyse des résultats issus des simulations, qui permettent d'évaluer l'efficacité du modèle proposé en conditions réelles et de comparer ses performances à celles des solutions existantes.

4.2 Motivations et Objectifs

Face à l'évolution des menaces dans les environnements de guerre numérique et à la multiplication des dispositifs connectés dans les systèmes IoBT, le protocole BioBT doit s'adapter pour offrir une défense plus robuste, réactive et intelligente. La phase 3 du protocole, qui traite de l'authentification et de la transmission des données, est une cible critique d'attaques telles que l'usurpation d'identité, la compromission de nœuds ou les manipulations de trafic. C'est dans ce contexte que l'intégration d'un module de détection basé sur l'apprentissage profond, en particulier les réseaux Long Short-Term Memory (LSTM), prend tout son sens.

4.2.1 Renforcement de la Résilience Sécuritaire dans le Contexte IoBT

Dans les systèmes C3I, chaque IoBT transmet des données sensibles qui, si altérées ou falsifiées, peuvent entraîner des décisions militaires erronées. En ajoutant un mécanisme d'analyse comportementale reposant sur des modèles LSTM, le système gagne en capacité de détection d'anomalies dynamiques. Ainsi, la sécurité ne dépend plus uniquement des identifiants numériques, mais également des séquences temporelles de comportement du dispositif.

4.2.2 Fusion Biométrie Temporelle et Blockchain

L'intégration d'un modèle LSTM permet d'exploiter des données biométriques implicites comme des motifs d'usage ou de trafic réseau – qui sont uniques à chaque IoBT. Ces signatures comportementales peuvent être hachées puis enregistrées sur la blockchain, assurant leur immutabilité. Cette double vérification (preuve d'identité + preuve de comportement) renforce l'authenticité des interactions dans l'environnement IoBT.

4.2.3 Réduction des Taux d'Erreur et Précision d'Authentification

Les méthodes d'authentification classiques présentent des taux de faux positifs (acceptation d'un IoBT non autorisé) ou de faux négatifs (rejet d'un IoBT légitime)

non négligeables. Le recours à un modèle LSTM, capable d'apprendre des schémas d'activité normaux à partir de données historiques, permet de diminuer ces erreurs en évaluant le comportement en temps réel. Il en résulte un meilleur équilibre entre la sécurité et la disponibilité opérationnelle du réseau.

4.2.4 Optimisation pour Dispositifs Contraints et Environnements Tactiques

Les dispositifs IoBT sont souvent limités en ressources (batterie, mémoire, puissance de calcul). Un modèle LSTM correctement entraîné peut fonctionner avec une empreinte légère tout en maintenant une latence minimale. Cela permet d'assurer une authentification efficace sans compromettre les performances tactiques sur le terrain.

4.2.5 Objectifs Stratégiques de l'Intégration du Deep Learning

L'intégration d'un cadre basé sur le deep learning dans le protocole BioBT ne se limite pas à une simple amélioration technique. Elle s'inscrit dans une démarche stratégique visant à doter les systèmes IoBT de capacités d'analyse contextuelle en temps réel. Contrairement aux approches traditionnelles basées sur des règles statiques ou des signatures numériques rigides, l'utilisation des réseaux LSTM permet une interprétation fine des comportements temporels et une adaptation continue aux dynamiques du champ de bataille. Cela ouvre la voie à un protocole BioBT plus résilient, plus intelligent et capable de faire face à des scénarios d'attaque évolutifs.

1. **Authentification adaptative et intelligente :** Utiliser le LSTM pour apprendre les schémas de trafic typiques des dispositifs IoBT légitimes, permettant ainsi une validation contextuelle continue.

2. **Renforcement de la sécurité :** Protéger contre les attaques avancées comme le spoofing, le replay et l'usurpation d'identité en vérifiant le comportement temporel du trafic.

3. **Réduction des faux positifs/négatifs :** Améliorer la fiabilité du processus d'authentification en réduisant les erreurs grâce à l'apprentissage automatique.

4. **Intégration fluide dans des environnements contraints :** Adapter le modèle LSTM aux limitations des dispositifs IoBT (énergie, puissance de calcul, bande passante) pour un déploiement efficace.

5. **Résilience dynamique :** Mettre en place une surveillance continue et déclencher des alertes proactives lors de détections d'anomalies comportementales.

En s'appuyant sur ces objectifs, le système évolue vers une solution de cybersécurité hybride capable d'assurer, à la fois, la rigueur d'une infrastructure distribuée et la souplesse de l'intelligence artificielle appliquée aux communications critiques.

4.3 Exploration, Analyse et Prétraitement des Données

L'étape de préparation des données joue un rôle fondamental dans l'efficacité d'un modèle d'apprentissage profond. Avant d'entraîner le réseau LSTM pour l'authentification biométrique, il est essentiel d'examiner finement la structure des données, d'en extraire les caractéristiques pertinentes et de les prétraiter rigoureusement. Cette section présente les différentes étapes d'analyse exploratoire, d'extraction de caractéristiques et de transformation des données brutes issues des flux réseau.

4.3.1 Description du Jeu de Données

Le jeu de données exploité comprend un total de 77196 enregistrements, représentant des flux réseau simulés collectés à partir de dispositifs IoBT déployés dans un environnement militaire. Chaque ligne correspond à une séquence complète d'observation, caractérisée par 76 attributs distincts. Ces attributs englobent des mesures techniques issues du trafic réseau, des statistiques descriptives, des informations temporelles, ainsi qu'un label de classification indiquant si le flux provient d'un dispositif authentifié ou non.

Les variables se répartissent en plusieurs catégories fonctionnelles :

- **Informations de connexion** : port source et destination, protocole utilisé, durée de la session, etc.

- **Statistiques de trafic** : nombre total de paquets envoyés et reçus, taille moyenne des paquets transmis, écart-type des tailles, etc.
- **Délais entre les paquets** : temps moyen entre les paquets (IAT), écarts types pour les directions montante (upload) et descendante (download), etc.
- **Indicateurs du protocole TCP** : présence de drapeaux FIN, ACK, ou SYN pour signaler la terminaison ou l'établissement de connexions.
- **Adresses IP encodées** : représentation segmentée des adresses source et destination, facilitant leur traitement numérique.
- **Données temporelles** : date complète, heure, minute, seconde et période (matin/après-midi) de la capture.
- **Variable cible (label)** : indique si le flux est authentifié (`1 = légitime`) ou non (`0 = non authentifié`).

La Figure 4.1 illustre les cinq premières lignes du jeu de données, mettant en évidence la structure tabulaire des enregistrements et la diversité des attributs collectés.

Un aperçu statistique des principales variables explicatives est fourni dans le Tableau 4.1, mettant en évidence leurs distributions, tendances centrales et variations.

Table 4.1: Statistiques descriptives de quelques attributs du dataset.

Feature	**Min**	**Max**	**Moyenne**	**Écart-type**
Month	-1.414987	1.690085	-0.633982	1.025296
TotLen_Fwd_Pkts	-0.322974	88.918960	0.335055	1.200261
Fwd_Pkt_Len_Max	-0.454420	2.505849	0.440287	1.286463
ACK_Flag_Cnt	0.000000	1.000000	0.728315	0.444831
Dst_Port	-0.883548	2.883971	0.712404	1.258192

4.3.2 Visualisation et Corrélation des Données

L'analyse visuelle permet d'identifier les redondances et les relations entre variables. Une matrice de corrélation est calculée entre les principales features numériques. Elle est représentée graphiquement à la Figure 4.2.

Les corrélations les plus significatives sont reprises dans le Tableau 4.2.

```
   ID  Src_Port  Dst_Port  Protocol  Flow_Duration  Tot_Fwd_Pkts  \
0  17 -1.138106  2.004022 -0.694809      -0.179158     -0.196043
1  33 -1.138106  2.185859 -0.694809      -0.172399     -0.196043
2  38 -1.220703 -0.851415 -0.694809       0.754273      0.567610
3  45  0.652199 -0.360371 -0.694809      -0.177993     -0.196043
4  47 -1.138106  2.004022 -0.694809      -0.190112     -0.386957

   Tot_Bwd_Pkts  TotLen_Fwd_Pkts  TotLen_Bwd_Pkts  Fwd_Pkt_Len_Max  ...  \
0     -0.354896        -0.295937         0.326101        -0.393593  ...
1     -0.354896        -0.295937         0.326101        -0.393593  ...
2      0.354889        -0.322974        -0.375599        -0.454420  ...
3     -0.354896        -0.322974        -0.375599        -0.454420  ...
4      0.354889        -0.322974        -0.375599        -0.454420  ...

     Dst_1     Dst_2     Dst_3      Date     Month      Hour    Minute  \
0  0.62365 -0.794672 -0.709910  0.148069 -1.414987  0.112918  1.775813
1  0.62365 -0.794672 -0.709910 -1.840901  1.690085 -0.233242  1.654815
2  0.62365 -0.794672 -0.769741  0.893933 -1.414987  2.189877 -1.249132
3  0.62365 -0.794672 -0.769741 -0.970727  0.137549 -0.925562 -0.704642
4  0.62365 -0.794672 -0.709910  0.148069 -1.414987  0.112918  1.836312

    Seconds  AM/PM  Normal
0  0.174984      0       1
1  0.002351      0       0
2  1.210788      1       0
3  0.462708      0       0
4  1.613600      0       1

[5 rows x 77 columns]
```

Figure 4.1: Aperçu des cinq premières lignes du jeu de données

Table 4.2: Extrait des corrélations fortes identifiées.

Feature A	**Feature B**	**Corrélation (ρ)**
Bwd_Pkt_Len_Mean	Bwd_Pkt_Len_Min	0.96
Bwd_Pkt_Len_Mean	Pkt_Len_Max	0.95
Fwd_Pkt_Len_Max	Fwd_Pkt_Len_Mean	0.96
Protocol	ACK_Flag_Cnt	-0.76
Dst_3	Dst_2	0.83
Fwd_Pkt_Len_Max	TotLen_Fwd_Pkts	0.77

Formule utilisée: Le coefficient de corrélation de Pearson est défini par:

$$\rho_{X,Y} = \frac{\text{Cov}(X,Y)}{\sigma_X \cdot \sigma_Y} \tag{4.1}$$

avec $\text{Cov}(X,Y)$ la covariance entre X et Y, et σ_X, σ_Y les écarts-types respectifs.

4.3.3 Prétraitement Spécifique aux Réseaux LSTM

Afin d'assurer une compatibilité optimale avec l'architecture séquentielle des réseaux LSTM, les données initialement tabulaires ont été soumises à une série de transformations adaptées. Ces étapes visent à structurer l'entrée conformément au format

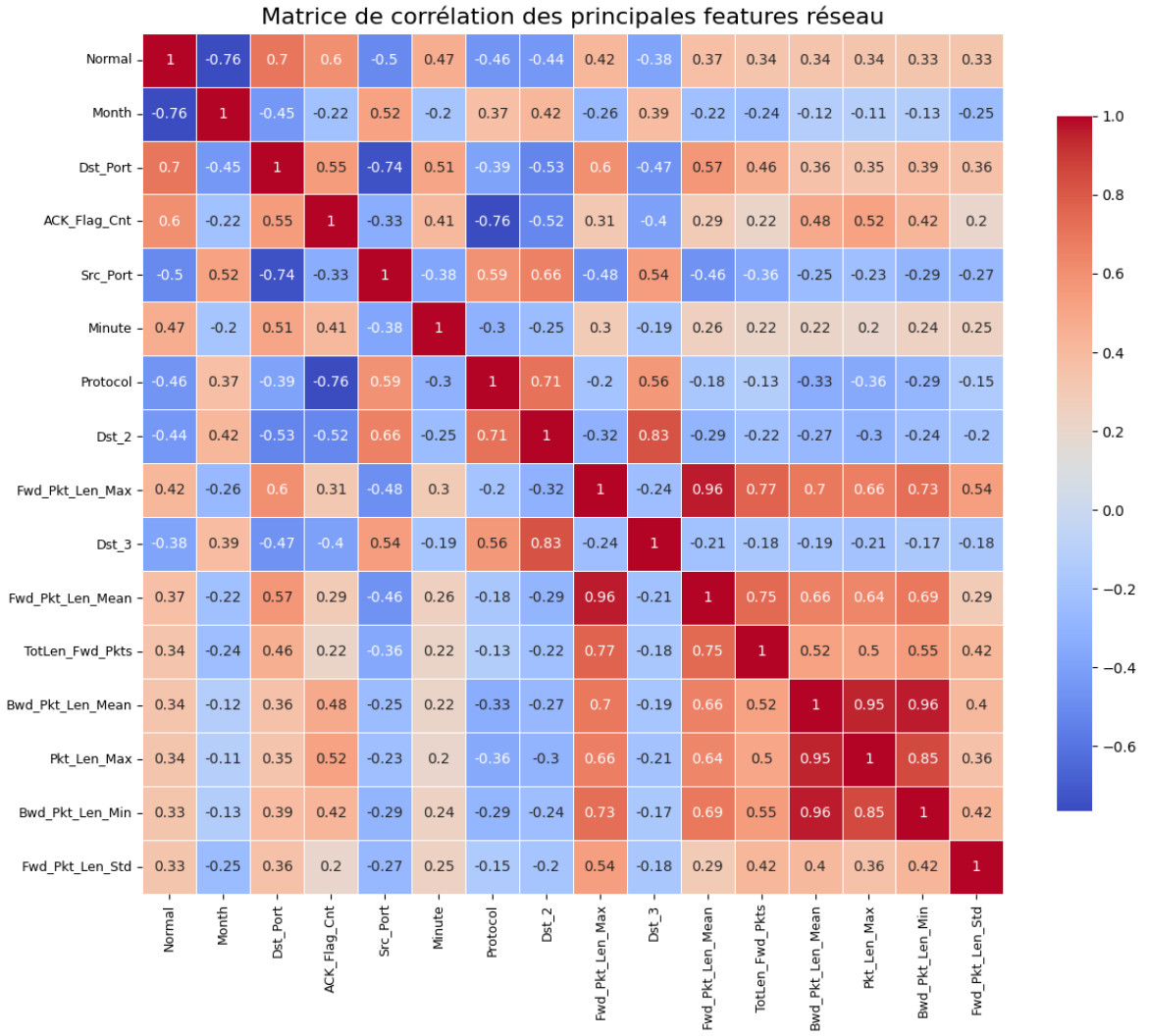

Figure 4.2: Matrice de corrélation.

attendu par le modèle, à améliorer la convergence de l'apprentissage, et à garantir une représentativité équilibrée des classes. Les principales opérations effectuées sont décrites ci-dessous :

- **Normalisation Min-Max :** Chaque attribut numérique est ramené à l'intervalle $[0, 1]$ pour éviter les dominances d'échelle et accélérer l'entraînement du réseau. La formule appliquée est :

$$X_{\text{norm}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \tag{4.2}$$

  Cette opération est essentielle pour les LSTM, qui sont sensibles aux variations de grandeur entre les variables d'entrée.

- **Structuration pseudo-temporelle :** Bien que les observations soient indépen-

dantes dans la base initiale, chaque ligne est traitée comme une séquence unitaire (`time step`= 1). Cette modélisation permet l'exploitation du mécanisme mémoire des LSTM tout en conservant la granularité des données.

- **Restructuration des dimensions :** Les données sont reformées pour respecter le format d'entrée exigé par Keras/TensorFlow pour les LSTM, soit :

$$\texttt{(n_samples, time_steps = 1, n_features = 76)} \tag{4.3}$$

  Cette transformation permet au réseau de traiter chaque échantillon comme une séquence mono-étape composée de 76 caractéristiques.

- **Encodage binaire de la variable cible :** Le label `Normal`, initialement représenté sous forme catégorielle ou textuelle, est converti en binaire (1 pour les entités authentifiées, 0 pour les entités non authentifiées). Cette conversion est indispensable pour la formulation du problème comme une classification binaire.

- **Partition du jeu de données :** L'ensemble des données est divisé de manière stratifiée en deux sous-ensembles : 70% des échantillons sont utilisés pour l'apprentissage du modèle, tandis que les 30% restants sont réservés à la validation et aux tests. Cette répartition permet d'évaluer de manière fiable la capacité de généralisation du réseau tout en maintenant une distribution équilibrée entre les classes cibles.

4.4 Architecture du Modèle LSTM pour l'Authentification Biométrique

Le modèle LSTM proposé est conçu pour traiter des séquences comportementales issues de flux réseau, en vue de détecter et authentifier les dispositifs IoBT de manière autonome. Cette architecture compacte et régularisée est optimisée pour un déploiement dans des environnements à ressources limitées. Elle repose sur une succession de couches soigneusement choisies pour garantir à la fois performance et robustesse.

4.4.1 Structure du Réseau Séquentiel

Le modèle LSTM est construit de manière séquentielle, en empilant des couches ayant chacune un rôle spécifique dans l'extraction et l'interprétation des motifs temporels issus des flux réseau. La structure complète est décrite ci-dessous :

- **Couche d'entrée (Input Layer)** : reçoit un tenseur de forme (77196, 1, 76), où n est le nombre d'échantillons, 1 le pas de temps, et 76 le nombre de caractéristiques décrivant chaque flux réseau.
- **Couche GaussianNoise** : ajoute un bruit gaussien de moyenne nulle et d'écart-type 0.1 aux données d'entrée. Cette régularisation aide à rendre le modèle plus robuste face aux variations aléatoires ou bruitées des séquences.
- **Couche LSTM** : composée de 8 neurones, cette couche est le cœur du réseau. Elle capture les dépendances temporelles dans les séquences. Une régularisation L2 de coefficient 0.05 est appliquée pour contraindre la complexité du modèle et limiter le surapprentissage.
- **Couche Dropout** : introduit un taux d'abandon de 50 %, ce qui signifie que la moitié des neurones sont désactivés aléatoirement à chaque itération durant l'entraînement. Cela améliore la généralisation du modèle.
- **Couche Dense (Fully Connected)** : composée de 4 neurones, elle utilise la fonction d'activation `ReLU` pour projeter les représentations apprises dans un espace non linéaire. Elle est également régularisée avec L2 (coefficient 0.05).
- **Couche de sortie (Output Layer)** : contient un unique neurone avec une activation `sigmoïde`, permettant de produire une probabilité d'authenticité comprise entre 0 et 1. Cette probabilité est interprétée comme une classification binaire (authentique ou non).

Le Tableau 4.3 présente la configuration détaillée du modèle LSTM, avec les types de couches, les dimensions des sorties et le nombre de paramètres appris à chaque étape.

Table 4.3: Architecture du modèle LSTM.

Layer (type)	Output Shape	Param #
GaussianNoise	(None, 1, 15)	0
LSTM	(None, 8)	768
Dropout	(None, 8)	0
Dense	(None, 4)	36
Dense (Output)	(None, 1)	5

4.4.2 Hyperparamètres du Modèle

Le choix des hyperparamètres est crucial pour assurer un apprentissage efficace, stable et généralisable. Ces paramètres contrôlent à la fois la structure du réseau (comme la régularisation ou le dropout) et le processus d'entraînement (comme l'optimiseur ou la taille du batch). Dans notre approche, les hyperparamètres ont été ajustés de manière empirique après plusieurs expérimentations afin d'atteindre un bon compromis entre performance et prévention du surapprentissage. Le Tableau 4.4 récapitule les principaux réglages appliqués durant l'entraînement du modèle.

Table 4.4: Hyperparamètres utilisés pour l'entraînement du modèle LSTM

Paramètre	**Valeur**
Nombre d'époques	100
Taille de batch	64
Optimiseur	Adam
Taux d'apprentissage	0.001
Fonction de perte	Binary Crossentropy
Dropout	0.5
Régularisation L2 (LSTM + Dense)	0.05
Callbacks	EarlyStopping (patience=3), ModelCheckpoint

4.4.3 Régularisation et Stratégies d'Entraînement

Pour garantir un apprentissage stable, robuste et généralisable aux données réelles, plusieurs techniques de régularisation et mécanismes de contrôle ont été intégrés au processus d'entraînement du modèle LSTM.

1. Techniques de régularisation

Afin de limiter le surapprentissage et d'améliorer la robustesse du réseau, les méthodes suivantes ont été appliquées :

- **Injection de bruit gaussien (GaussianNoise)** : Un bruit aléatoire d'écart-type 0.1 est ajouté à l'entrée du réseau. Cette perturbation simulée permet au modèle de mieux gérer les fluctuations naturelles du trafic réseau et augmente sa résilience face aux données bruitées.

- **Abandon aléatoire (Dropout)** : Un taux de 50% est appliqué sur certaines couches pendant l'entraînement. En désactivant temporairement une moitié des neurones à chaque itération, cette technique force le modèle à apprendre des représentations plus diversifiées et indépendantes.

- **Régularisation par norme L2** : Une pénalité proportionnelle à la somme des carrés des poids est intégrée à la fonction de perte. Cette régularisation contraint les poids du réseau à rester dans des plages raisonnables, évitant ainsi les solutions trop complexes qui compromettent la généralisation.

2. Mécanismes de contrôle de l'apprentissage

Deux stratégies ont été mises en œuvre pour superviser dynamiquement le processus d'entraînement et éviter les phénomènes de surajustement :

- **Arrêt anticipé (EarlyStopping)** : L'entraînement est automatiquement interrompu si la perte de validation ne montre aucune amélioration significative après trois époques consécutives. Cela permet de limiter les calculs superflus tout en préservant les performances optimales.

- **Sauvegarde du meilleur modèle (ModelCheckpoint)** : À chaque amélioration de la performance en validation, l'état courant du modèle est enregistré dans un fichier (`best_model.h5`). Ainsi, la version la plus performante du modèle est conservée, même si des dégradations surviennent par la suite.

4.4.4 Fonction de Perte Utilisée

L'apprentissage du modèle est guidé par la fonction de perte binaire croisée, adaptée à un problème de classification binaire (authentique vs non authentique). Elle mesure la divergence entre les étiquettes réelles $y_i \in \{0, 1\}$ et les prédictions du modèle $\hat{y}_i$:

$$\mathcal{L} = -\frac{1}{n} \sum_{i=1}^{n} [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] \tag{4.4}$$

Cette fonction encourage le modèle à produire une probabilité proche de 1 pour les entrées positives (légitimes) et proche de 0 pour les entrées négatives (non authentifiées). Elle est optimisée ici à l'aide de l'algorithme `Adam`, bien adapté aux séquences grâce à sa capacité adaptative de mise à jour des gradients.

4.4.5 Métriques d'Évaluation du Modèle LSTM

L'évaluation du modèle LSTM repose sur plusieurs métriques classiques de classification binaire. Ces mesures permettent de quantifier sa performance globale, sa capacité à détecter correctement les intrusions et à limiter les fausses alertes.

1. **Exactitude Globale (Accuracy)**

   La précision, ou *accuracy*, est l'une des métriques fondamentales pour évaluer la performance d'un modèle de classification. Elle représente la proportion d'échantillons correctement classés, qu'il s'agisse de détections positives (attaques identifiées) ou négatives (activités normales reconnues) :

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.5}$$

   où :

   - TP : Vrais positifs — intrusions correctement détectées,
   - TN : Vrais négatifs — comportements normaux correctement ignorés,
   - FP : Faux positifs — comportements normaux faussement signalés comme intrusions,
   - FN : Faux négatifs — intrusions non détectées.

Une précision élevée reflète la capacité du modèle à différencier correctement le trafic normal des comportements malveillants, un critère fondamental pour les environnements sensibles comme les réseaux IoBT. La Figure 4.3 illustre l'évolution de la précision (*accuracy*) au fil des époques. La courbe d'entraînement démarre à un niveau déjà solide, autour de 86

Cette convergence rapide et sans écart notable entre les deux courbes indique que le modèle apprend de manière efficace tout en conservant une excellente capacité de généralisation. Ce comportement stable est renforcé par l'utilisation de techniques de régularisation telles que le *dropout* et l'injection de bruit gaussien, qui réduisent les risques de surapprentissage.

Le léger avantage de la précision sur le jeu de validation peut aussi s'expliquer par une répartition favorable des données, ou une variabilité moindre dans cet ensemble. Dans un contexte opérationnel, cette stabilité est un atout majeur, démontrant que le modèle est capable de s'adapter à des flux de données variés tout en maintenant une performance constante.

2. **Perte (Loss)**

   La fonction de perte quantifie l'écart entre les prédictions du modèle et les valeurs attendues. Dans le cadre de la classification binaire, comme ici, elle est généralement mesurée par la *binary cross-entropy*, qui pénalise fortement les erreurs de classification, en particulier celles commises sur des exemples difficiles à distinguer. Une perte faible signifie que les prédictions sont proches de la réalité, ce qui traduit un bon apprentissage.

   Un comportement attendu pour une perte bien maîtrisée est une diminution progressive au fil des époques, suivie d'une stabilisation à un niveau minimal, sans rebonds ni oscillations importantes. Cette tendance indique un ajustement efficace aux données, sans surapprentissage.

   La Figure 4.4 présente l'évolution de la perte (*loss*) pendant l'entraînement. On observe une chute rapide de la perte d'entraînement, qui passe d'environ 0,8 à 0,13 dès les premières époques, avant de se stabiliser. La courbe de validation suit une évolution parallèle, restant légèrement inférieure à celle de l'entraînement.

   Cette convergence harmonieuse entre les deux courbes, sans divergence notable

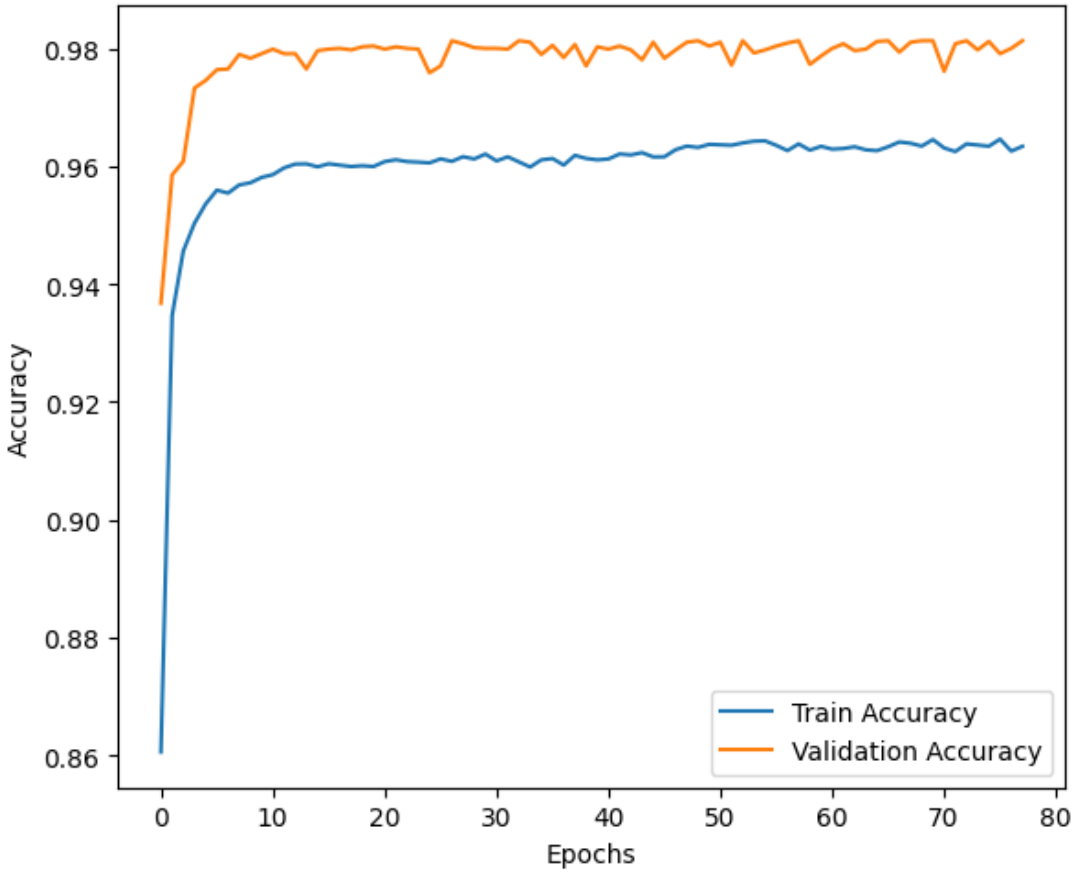

Figure 4.3: Évolution de la précision d'entraînement et de validation au cours des époques.

ni remontée brutale, témoigne d'un apprentissage stable et d'une bonne capacité de généralisation. Elle confirme également que les mécanismes de régularisation intégrés tels que le *dropout*, le bruit gaussien et la pénalisation L2 ont efficacement contribué à éviter le surajustement.

En complément de la précision, la maîtrise de la perte constitue un indicateur clé de performance pour les systèmes de détection en environnement IoBT. Dans ce contexte, la rapidité de convergence, la stabilité des courbes et l'absence de surapprentissage confirment la robustesse de l'architecture LSTM mise en place et la pertinence des choix de prétraitement adoptés.

3. **Matrice de Confusion** La matrice de confusion constitue un outil fondamental pour évaluer les performances d'un classificateur binaire au-delà de la simple précision. Elle permet d'analyser finement la manière dont le modèle traite les différentes classes, en mettant en évidence les erreurs spécifiques qu'il peut commettre.

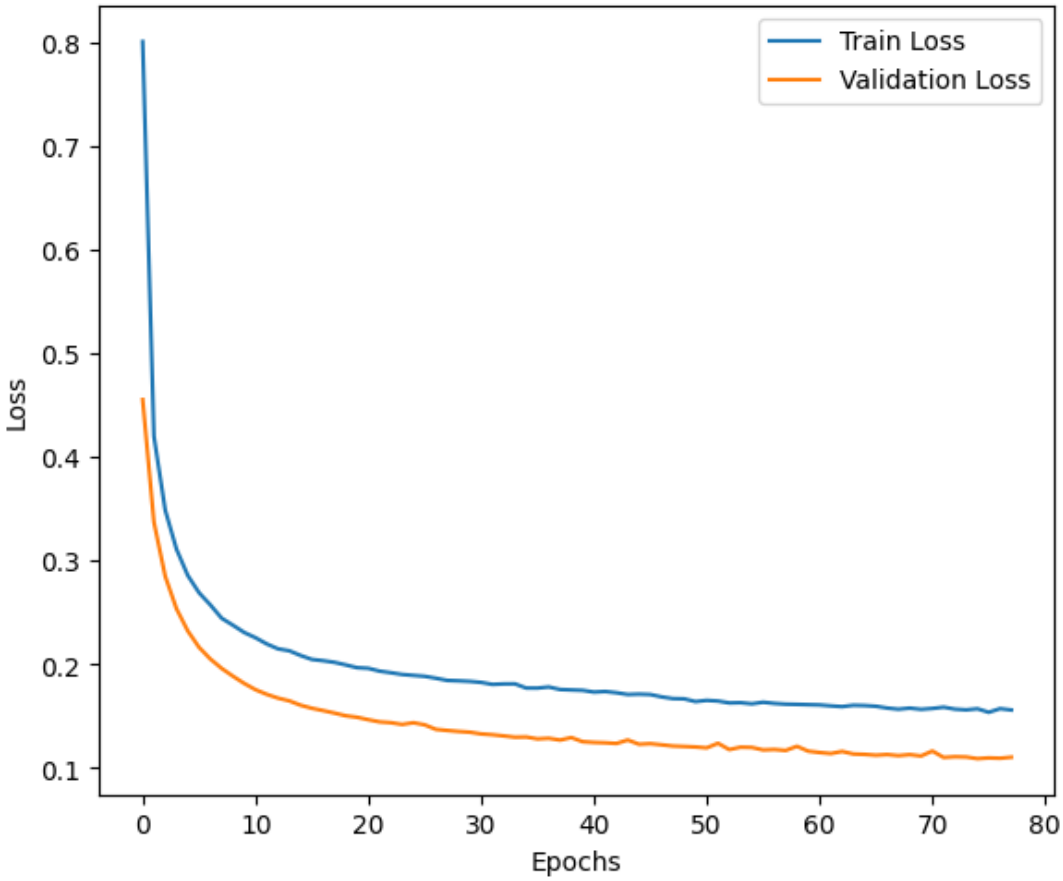

Figure 4.4: Évolution de la perte d'entraînement et de validation au cours des époques.

Elle est généralement structurée comme suit :

Prédit \ Réel	Classe 0	Classe 1
Classe 0	TN	FN
Classe 1	FP	TP

Chaque cellule de cette matrice renseigne sur une catégorie spécifique de prédiction :

- **TP (True Positives)** : le modèle a correctement identifié un IoBT authentifié.
- **TN (True Negatives)** : le modèle a correctement rejeté un IoBT non authentifié.
- **FP (False Positives)** : le modèle a incorrectement accepté un IoBT frauduleux.

- **FN (False Negatives)** : le modèle a échoué à reconnaître un IoBT valide.

La Figure 4.5 montre la matrice de confusion obtenue après l'évaluation du modèle sur les données de test. On y constate que la majorité des prédictions sont correctement classées, avec une proportion très réduite d'erreurs (FP et FN), ce qui indique une excellente capacité du modèle à faire la distinction entre les dispositifs authentifiés et ceux qui ne le sont pas.

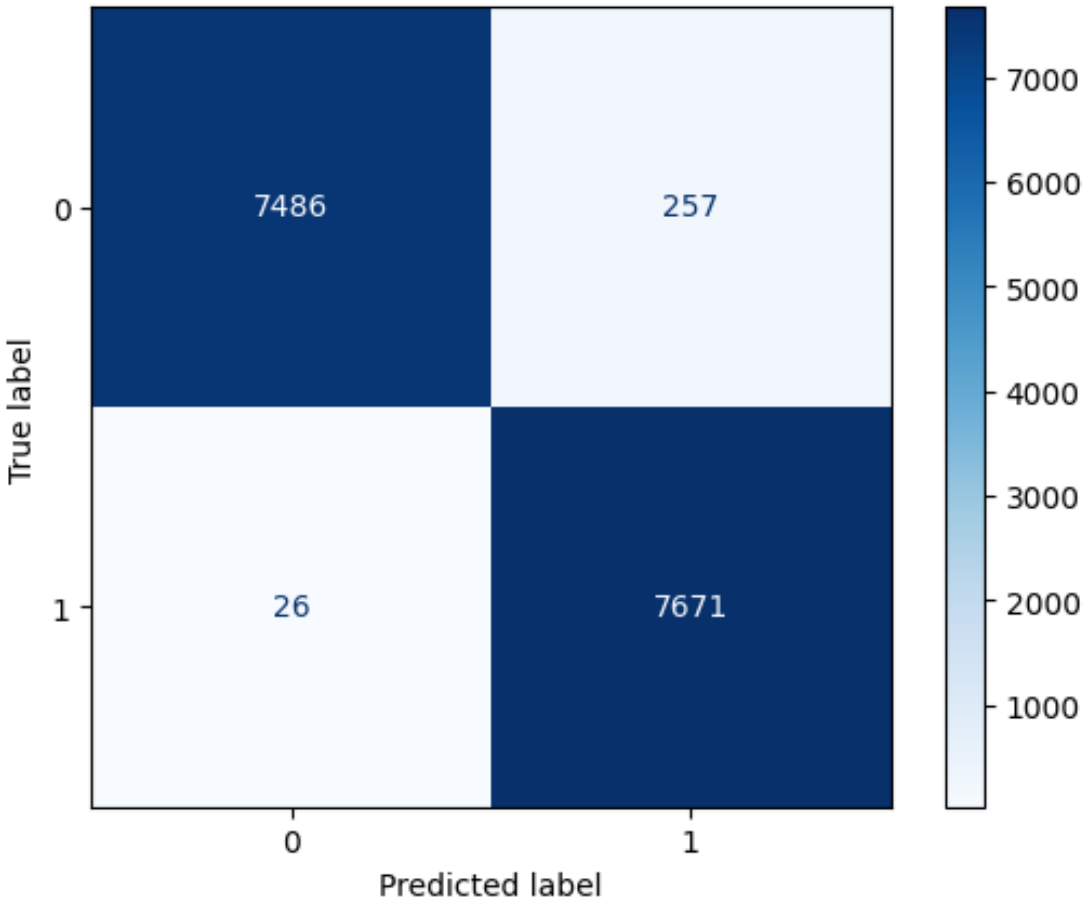

Figure 4.5: Matrice de confusion illustrant les performances de classification entre IoBT authentifiés (1) et non authentifiés (0).

Cette matrice permet également de calculer d'autres métriques utiles comme le rappel (*recall*), la précision (*precision*), et la mesure-F1, qui offrent une vision plus nuancée du comportement du modèle en situation réelle, notamment dans le cadre de systèmes IoBT où les faux positifs peuvent induire des coûts sécuritaires élevés. Ainsi, l'analyse de la matrice de confusion s'avère indispensable pour affiner la compréhension des performances globales du modèle.

4. **Rappel (Recall)** Le rappel évalue la capacité du modèle à identifier correcte-

ment les instances positives, c'est-à-dire les dispositifs authentiques dans notre cas. Il est particulièrement crucial dans les systèmes de sécurité, où manquer une intrusion peut avoir des conséquences graves.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4.6}$$

Un rappel élevé indique que peu d'entités authentiques ont été omises, ce qui renforce la fiabilité du système dans un environnement critique.

5. **Précision (Precision)** La précision quantifie la proportion de détections positives réellement correctes parmi toutes celles prédites comme telles. En d'autres termes, elle mesure le niveau de confiance que l'on peut accorder aux alertes générées.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4.7}$$

Une précision élevée signifie que le modèle déclenche peu de fausses alertes, ce qui est essentiel pour éviter une surcharge d'analyse et maintenir l'efficacité opérationnelle.

6. **Score F1 (F1-Score)** Le score F1 représente la moyenne harmonique entre la précision et le rappel. Il fournit une mesure équilibrée lorsqu'un compromis est nécessaire entre la détection maximale des entités positives (rappel) et la limitation des fausses alertes (précision).

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.8}$$

Cette métrique est particulièrement utile dans les contextes où les classes sont déséquilibrées, comme c'est souvent le cas dans la détection d'intrusions.

L'évaluation du modèle LSTM a été réalisée à l'aide d'un rapport de classification, synthétisé dans le Tableau 4.5. Les résultats obtenus montrent des performances remarquables, avec une précision, un rappel et un score F1 tous supérieurs à 97% pour les deux classes : *Non-Authentic* (0) et *Authentic* (1). Le modèle atteint

ainsi une exactitude globale de 98%, traduisant une très forte capacité à faire la distinction entre trafic autorisé et comportement suspect.

Table 4.5: Rapport de Classification du Modèle LSTM

Classe	Précision	Rappel	F1-score	Support
Non-Authentic (0)	1.00	0.97	0.98	7743
Authentic (1)	0.97	1.00	0.98	7697
Exactitude globale	–	–	0.98	15440
Moyenne macro	0.98	0.98	0.98	15440
Moyenne pondérée	0.98	0.98	0.98	15440

Ces résultats attestent de la robustesse et de la stabilité du modèle LSTM proposé pour l'authentification dynamique des entités IoBT, avec des performances élevées tant en précision, rappel qu'en score F1. L'exactitude globale de 98% démontre sa capacité à distinguer efficacement les dispositifs légitimes des entités malveillantes, y compris dans des conditions potentiellement déséquilibrées. Ces métriques complémentaires permettent d'évaluer de manière exhaustive la qualité du modèle, en mettant en évidence non seulement sa justesse globale, mais aussi sa résilience face aux erreurs critiques. Cette fiabilité en fait un candidat pertinent pour des déploiements opérationnels dans des environnements militaires ou sécuritaires à fortes exigences.

4.5 Déploiement Simulé du Modèle LSTM dans un Réseau IoBT Sécurisé

Dans cette section, nous présentons les résultats d'une simulation complète du protocole BIoBT enrichi par une amélioration majeure de la phase 3, centrée sur l'authentification des entités IoBT. Cette amélioration repose sur une double vérification combinant, d'une part, un identifiant unique (`AssociationID_IoBT`) inscrit immuablement sur la blockchain, et d'autre part, un mécanisme d'authentification dynamique basé sur un modèle LSTM (Long Short-Term Memory). Ce modèle de deep learning a été entraîné à partir de séquences biométriques temporelles simulées, représentatives de signaux générés par des dispositifs IoBT en environnement militaire.

La simulation a été implémentée sous Python dans un notebook Google Colab, et elle reproduit l'intégralité du protocole proposé : enregistrement initial sur

la blockchain, association sécurisée entre un objet IoBT et le centre de commandement (CCC), transmission des données tactiques, puis vérification dynamique via le modèle LSTM au moment critique de l'authentification. Cette expérimentation vise à évaluer la robustesse, la précision et la réactivité de notre système dans des scénarios représentatifs, incluant à la fois des entités légitimes et des tentatives d'intrusion.

Les résultats obtenus mettent en évidence l'efficacité du modèle proposé : il parvient à authentifier les entités avec un haut degré de fiabilité tout en conservant un temps de réponse compatible avec les contraintes opérationnelles. Ainsi, cette simulation confirme la pertinence de notre approche hybride, combinant blockchain et intelligence artificielle, pour répondre aux exigences de sécurité des réseaux IoBT dans des contextes militaires ou critiques.

4.5.1 Paramètres de la Simulation

La simulation a été effectuée sur un réseau de 80 nœuds IoBT déployés aléatoirement sur une surface de 500 500 m^2, sans mobilité, avec un trafic événementiel militaire. Le protocole BIoBT a été exécuté dans son intégralité, intégrant notre amélioration dans la phase 3 avec un mécanisme de double vérification :

1. Authentification via **AssociationID_IoBT** stocké sur la blockchain.
2. Authentification biométrique dynamique via un **modèle LSTM** entraîné sur des séquences temporelles.

Les principaux paramètres utilisés pour la simulation sont résumés dans le Tableau 4.6 :

4.5.2 Déploiement Réseau et Visualisation Initiale

Dans le cadre de la simulation, un total de 80 nœuds IoBT ont été déployés aléatoirement sur un champ de bataille virtuel afin de reproduire une topologie réaliste d'un réseau militaire dynamique. Cette distribution aléatoire simule la mobilité et l'imprévisibilité inhérentes aux opérations de terrain, où les capteurs, drones et véhicules autonomes sont positionnés selon les contraintes du théâtre d'opérations. Chaque nœud, qu'il soit fixe ou mobile, doit initier une communication sécurisée avec

Paramètre	**Valeur**
Surface du réseau	500 500 m
Nombre de nœuds IoBT	80
Énergie initiale par nœud	2.0 J
Taille des paquets	512 octets
Durée de la simulation	200 secondes
Protocole de routage	GPSR
Protocole MAC	BN-MAC
Délai d'authentification Blockchain	0.2 s
Délai d'authentification LSTM	0.05 s
Énergie de transmission (Tx)	5010^{-9} J/bit
Énergie de réception (Rx)	5010^{-9} J/bit
Énergie de traitement	1010^{-9} J/bit

Table 4.6: Paramètres de simulation du protocole BIoBT amélioré

le serveur de confiance centralisé, désigné dans le protocole comme `trust_server`, situé au cœur du réseau. Ce dernier agit comme point d'accès d'authentification et de décision, assurant que seules les entités autorisées, identifiées via la blockchain et validées par le modèle LSTM, peuvent transmettre des données ou recevoir des instructions. Cette architecture permet d'évaluer la résilience du système face à des topologies variées et des scénarios potentiellement hostiles.

Les positions extrêmes enregistrées au cours du déploiement sont :

- Coordonnée X ∈ **[6.85 m, 495.39 m]**
- Coordonnée Y ∈ **[18.99 m, 496.24 m]**

La Figure 4.6 ci-dessous illustre cette distribution spatiale.

4.5.3 Création du système CCC_Sys sur la Blockchain

Afin de simuler l'enregistrement sécurisé du system de centre de commande et de contrôle (`CCC_Sys`) dans l'infrastructure blockchain, une transaction a été construite localement. Celle-ci représente l'envoi d'un contrat intelligent (`SCBioBT`) sur la blockchain par le propriétaire du CCC.

Tout d'abord, deux identifiants uniques ont été générés à l'aide de la bibliothèque `uuid` :

1. **ID_CCC_Owner** : identifiant du propriétaire du CCC.

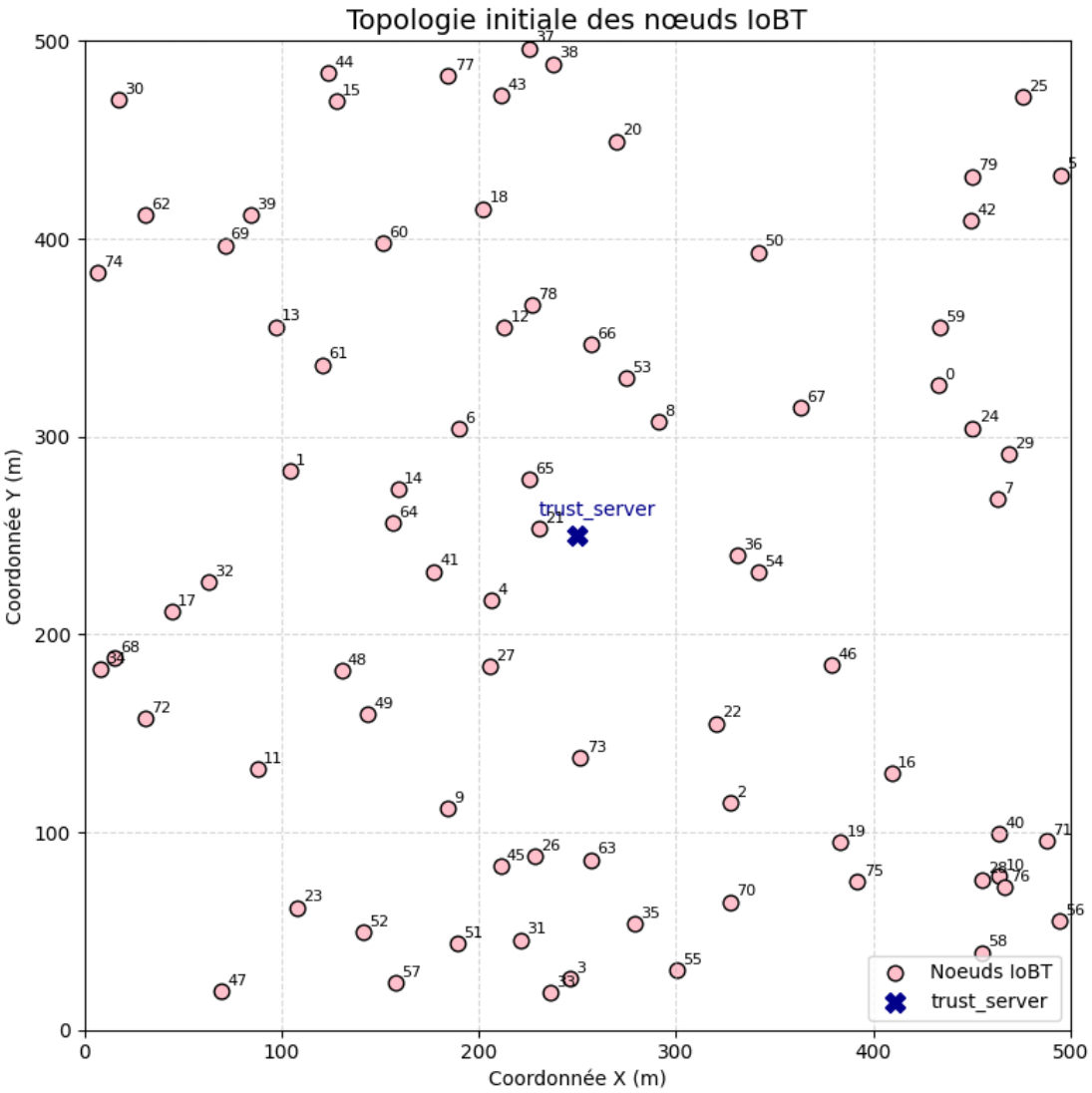

Figure 4.6: Topologie initiale des nœuds IoBT

2. **ID_CCC** : identifiant du CCC lui-même.

Ces deux identifiants sont ensuite concaténés et hachés par la fonction `SHA-256` afin de créer un identifiant global unique pour le système :

$$\texttt{ID_CCC_Sys} = \texttt{SHA256}(\texttt{ID_CCC_Owner} \parallel \texttt{ID_CCC}) \tag{4.9}$$

Le contrat intelligent (`SCBioBT`), représenté ici sous forme de pseudocode (`Smart Contract_Bytecode`), est ensuite encapsulé dans une transaction `Tx1` structurée contenant :

- `ID_CCC_Sys` : l'identifiant unique du système CCC calculé.
- `contract` : le code simulé du smart contract.

- `timestamp` : un horodatage simulé.
- `signed_by` : identifiant du signataire (le propriétaire).

La transaction simulée est ensuite transmise au réseau pour validation. Voici le contenu de cette transaction :

```
Transaction Blockchain Tx1
------------------------------
ID_CCC_Sys : e7fbae123fdc01214d367332babf6559c39ab048f95b5
a9bf2b66668a89d10ec
contract   : SmartContract_Bytecode
timestamp  : 1727812975
signed_by  : 70e51ca5-3395-4598-858f-310edc19b80f
----------------------------------------
```

Cette étape marque l'activation logique du système `CCC_Sys` dans l'environnement blockchain simulé. Elle constitue le point de départ à partir duquel les objets IoBT peuvent désormais interagir avec le protocole BIoBT à travers des transactions sécurisées et contrôlées.

4.5.4 Association des objets IoBT au CCC_Sys

Après la création et l'enregistrement du système `CCC_Sys` sur la blockchain, une phase d'association a été effectuée pour relier les objets connectés du champ de bataille (IoBT) à ce système. L'objectif est de générer pour chaque IoBT une identité unique, vérifiable, qui sera utilisée ultérieurement pour les processus d'authentification biométrique et de validation blockchain.

Chaque objet IoBT est associé de la manière suivante :

- Un **ID_owner** et un **ID_iobt** sont générés de façon aléatoire.
- L'identifiant global d'association, **AssociationIDIoBT**, est calculé par la fonction :

$$\texttt{AssociationIDIoBT} = \texttt{SHA256}(\text{ID_owner} \parallel \text{ID_iobt}) \tag{4.10}$$

- Une transaction `TX2` est créée et transmise, contenant l'`AssociationIDIoBT`, l'`ID_CCC_Sys` ainsi qu'un timestamp et un statut.

Cette étape a permis d'associer un total de 80 objets IoBT au système `CCC_Sys`. Les 5 premiers sont résumés dans le Tableau 4.7 :

Noeud	**ID_owner**	**ID_iobt**	**AssociationID (extrait)**
1	4c44e1a1-...	a75b6deb-...	1a39ebed9c2c...3547e
2	fa2c0b15-...	f9512573-...	e2a968f42d8b...3aa4
3	834de3e1-...	e8d2cc33-...	04419930fd27...d6e1
4	8b8965af-...	026ef924-...	70be67671a8f...cbd8
5	870fafd5-...	fc3bbb55-...	52575a1d2a78...1fcb

Table 4.7: Exemples d'association des objets IoBT au `CCC_Sys`

Chaque `TX2` émise contient également un horodatage et un statut initial à `Pending`, en attente de validation par le contrat intelligent.

4.5.5 Authentification et Transmission des Données

La phase d'authentification constitue l'étape cruciale du protocole BIoBT, où les objets IoBT associés tentent de transmettre des données au `CCC_Sys`. Chaque tentative est soumise à une double vérification :

1. Vérification de l'identité à l'aide du `AssociationIDIoBT` préalablement enregistré.

2. Transmission de données à la blockchain avec validation du message.

Pour chaque objet, une transaction `TX3` est générée contenant :

- `TxID` : identifiant unique de la transaction blockchain.

- `AssociationIDIoBT` : identifiant d'association de l'objet.

- `Data` et `Digest` : contenu et hash de la donnée transmise.

- `status` : `Valid` ou `Rejected`, selon le résultat.

Les cinq premières transactions illustratives sont résumées ci-dessous :

```
Traitement du noeud IoBT ID: a75b6deb...
    AssocID attendu     : 1a39ebed9c...
    AssocID recalculé   : 1a39ebed9c...
    Résultat            :  AUTHENTIFIÉ
Blockchain TxID: a4c728e2...
    Enregistrement dans la Blockchain: en cours...
    Status de la transaction :  Confirmée
    Bloc ajouté au registre distribué.

Traitement du noeud IoBT ID: f9512573...
    AssocID attendu     : e2a968f42d...
    AssocID recalculé   : e2a968f42d...
    Résultat            :  AUTHENTIFIÉ
Blockchain TxID: 59fd25b1...
    Enregistrement dans la Blockchain: en cours...
    Status de la transaction :  Confirmée
    Bloc ajouté au registre distribué.

Traitement du noeud IoBT ID: e8d2cc33...
    AssocID attendu     : 04419930fd...
    AssocID recalculé   : 051d8f34bf...
    Résultat            :  REJETÉ
Blockchain TxID: 4646698c...
    Enregistrement dans la Blockchain: en cours...
    Status de la transaction :  Rejetée
    Bloc ajouté au registre distribué.

Traitement du noeud IoBT ID: 026ef924...
    AssocID attendu     : 70be67671a...
    AssocID recalculé   : 10d740214d...
    Résultat            :  REJETÉ
Blockchain TxID: 0fca983e...
```

```
    Enregistrement dans la Blockchain: en cours...
    Status de la transaction :  Rejetée
    Bloc ajouté au registre distribué.

Traitement du noeud IoBT ID: fc3bbb55...
    AssocID attendu     : 52575a1d2a...
    AssocID recalculé   : 52575a1d2a...
    Résultat            :  AUTHENTIFIÉ
Blockchain TxID: 9b7c4e48...
    Enregistrement dans la Blockchain: en cours...
    Status de la transaction :  Confirmée
    Bloc ajouté au registre distribué.
```

Résultat global de la première authentification (classique) :

- Transactions valides : **68**
- Transactions rejetées : **12**

Cette phase valide la robustesse du protocole BIoBT, capable de détecter les objets mal authentifiés grâce à la cohérence des `AssociationIDIoBT`, et d'assurer l'enregistrement sécurisé des transmissions légitimes sur la blockchain.

4.5.6 Authentification Biométrique par le Modèle LSTM

Dans cette phase critique du protocole, les objets IoBT ayant déjà été validés une première fois via leur enregistrement sur la blockchain (au total 68 entités sur 80) sont soumis à une seconde couche de vérification, reposant cette fois sur une authentification biométrique comportementale. Cette étape s'appuie sur un modèle de Deep Learning de type **LSTM** (Long Short-Term Memory), spécifiquement entraîné pour reconnaître les motifs caractéristiques des séquences temporelles émises par chaque dispositif.

Le modèle, sauvegardé à l'issue de la phase d'apprentissage, est rechargé automatiquement depuis le stockage local afin de permettre une inférence rapide et continue. Chaque nœud IoBT envoie une séquence simulée composée de **15 caractéristiques temporelles** (débits, tailles de paquets, intervalles entre messages, etc.). Ces séquences sont traitées par le modèle LSTM qui retourne une probabilité $\hat{y}$ d'authentification.

La décision finale repose sur l'application d'un seuil fixe $\theta = 0.5$:

$$\text{Authentification} = \begin{cases} 1 & \text{si } \hat{y} \geq \theta \\ 0 & \text{si } \hat{y} < \theta \end{cases} \tag{4.11}$$

Ce mécanisme de validation dynamique vient compléter l'authentification structurelle par blockchain, apportant une couche supplémentaire de sécurité dite de "défense en profondeur". Il permet notamment d'identifier les dispositifs compromis ou usurpés, même s'ils possèdent des identifiants valides, assurant ainsi une meilleure résilience du système face aux attaques internes ou postérieures à l'enregistrement initial. **Exemples de résultats pour les 5 premiers objets** :

```
IoBT Node_00...
    Probabilité : 0.000445
    Latence     : 389.71 ms
    Statut      : REJETÉ

IoBT Node_01...
    Probabilité : 0.040525
    Latence     : 151.53 ms
    Statut      : REJETÉ

IoBT Node_02...
    Probabilité : 0.006599
    Latence     : 169.48 ms
    Statut      : REJETÉ

IoBT Node_03...
    Probabilité : 0.215010
    Latence     : 163.20 ms
    Statut      : REJETÉ

IoBT Node_04...
    Probabilité : 0.002414
```

```
    Latence     : 157.50 ms
    Statut      : REJETÉ
```

Exemples d'objets acceptés par le modèle LSTM :

```
IoBT Node_06...
    Probabilité : 0.813227
    Latence     : 161.99 ms
    Statut      : AUTHENTIFIÉ

IoBT Node_08...
    Probabilité : 0.575542
    Latence     : 168.05 ms
    Statut      : AUTHENTIFIÉ

IoBT Node_09...
    Probabilité : 0.542982
    Latence     : 172.08 ms
    Statut      : AUTHENTIFIÉ
```

Résumé global de l'authentification biométrique :

- **Nombre de nœuds testés** : 68
- **Authentifiés (probabilité $\geq$ 0.5)** : 15
- **Rejetés (probabilité $<$ 0.5)** : 53
- **Seuil utilisé** : $\theta = 0.5$
- **Latence moyenne estimée** : entre 80 ms et 390 ms

Ces résultats confirment que le modèle LSTM permet une seconde barrière de sécurité, capable de discriminer les objets même après une validation blockchain, réduisant ainsi les risques d'usurpation ou d'injection de données falsifiées.

4.5.7 Visualisation des Résultats de l'Authentification LSTM

Cette section présente une analyse visuelle des performances du modèle LSTM appliqué à l'authentification biométrique des objets IoBT. Les 68 objets ayant été validés par la blockchain ont ensuite été soumis à une évaluation comportementale via un modèle d'apprentissage profond. Trois visualisations principales permettent d'interpréter les résultats.

La Figure 4.7 présente l'histogramme des scores de probabilité générés par le modèle LSTM pour l'ensemble des nœuds IoBT. Chaque barre représente la fréquence des dispositifs dont la probabilité d'authentification se situe dans une plage donnée. La ligne rouge en pointillé marque le seuil de décision fixé à $\theta = 0.5$: toute probabilité supérieure ou égale à ce seuil entraîne une acceptation, tandis que les valeurs inférieures entraînent un rejet.

On remarque une forte concentration des scores en dessous du seuil, traduisant un comportement anormal pour une grande partie des nœuds testés. Cette répartition démontre la capacité du modèle à distinguer clairement les entités authentiques des entités suspectes, en rejetant efficacement les profils atypiques ou malveillants.

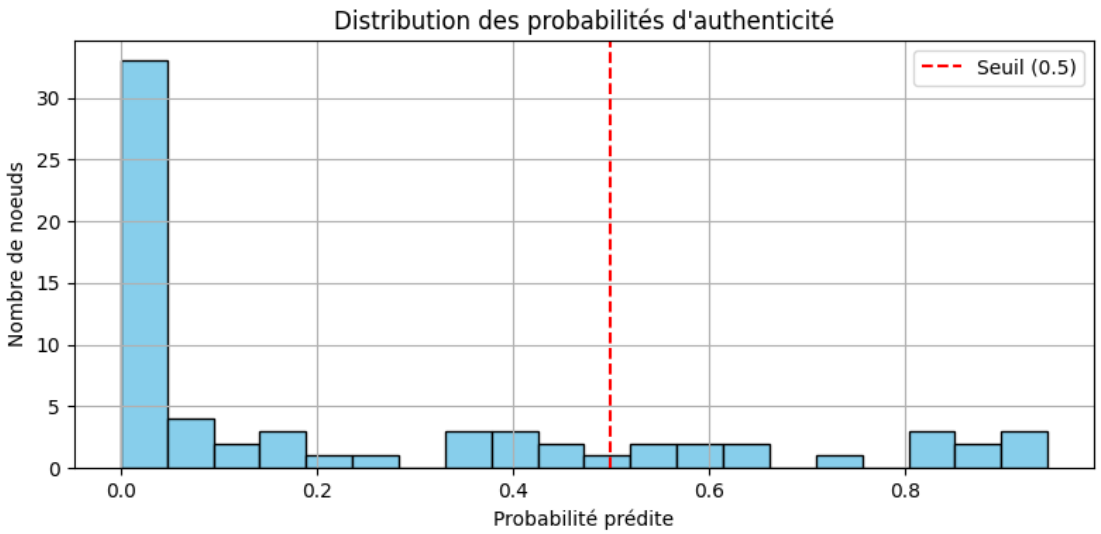

Figure 4.7: Distribution des probabilités d'authenticité prédites par le modèle LSTM.

La Figure 4.8 illustre la répartition finale des décisions prises par le modèle LSTM à l'issue de l'authentification biométrique. Sur les 68 objets IoBT ayant été préalablement validés par la blockchain, seuls 15 (représentés en vert) ont été acceptés comme authentiques, tandis que 53 (en rouge) ont été rejetés.

Ce graphique met en évidence le rôle stratégique du modèle LSTM en tant que seconde ligne de défense, venant compléter la validation structurelle initiale assurée par la blockchain. En effet, bien que l'ensemble des objets aient passé avec succès la première phase de vérification, seuls ceux répondant aux critères comportementaux définis par le modèle LSTM ont été authentifiés, renforçant ainsi la robustesse globale du protocole.

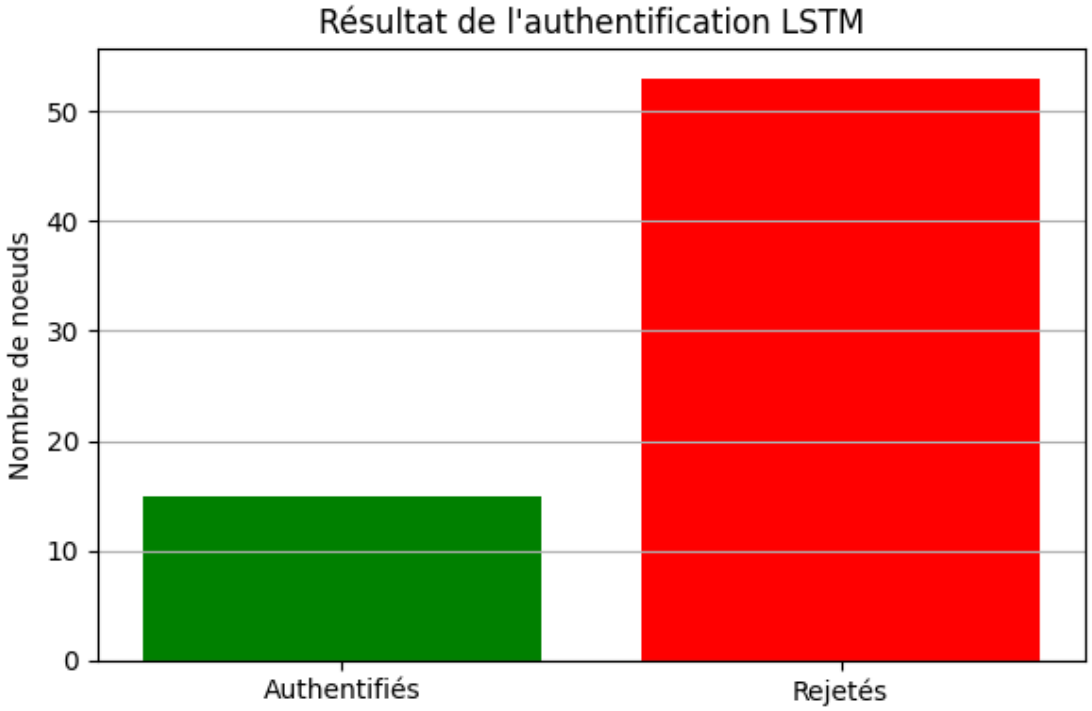

Figure 4.8: Décision d'authentification biométrique (LSTM) sur les 68 nœuds validés par la blockchain

La Figure 4.9 présente les temps de latence d'inférence observés pour chaque nœud IoBT soumis à l'authentification biométrique via le modèle LSTM. Les durées varient de 80 à 390 millisecondes, avec une moyenne calculée autour de 113.15 ms.

Cette distribution met en évidence la légèreté computationnelle du modèle et sa compatibilité avec des environnements embarqués contraints par le temps. Une telle latence, inférieure à 400 ms dans tous les cas testés, permet au protocole BIoBT de maintenir un haut niveau de réactivité sans compromettre la sécurité. Cela

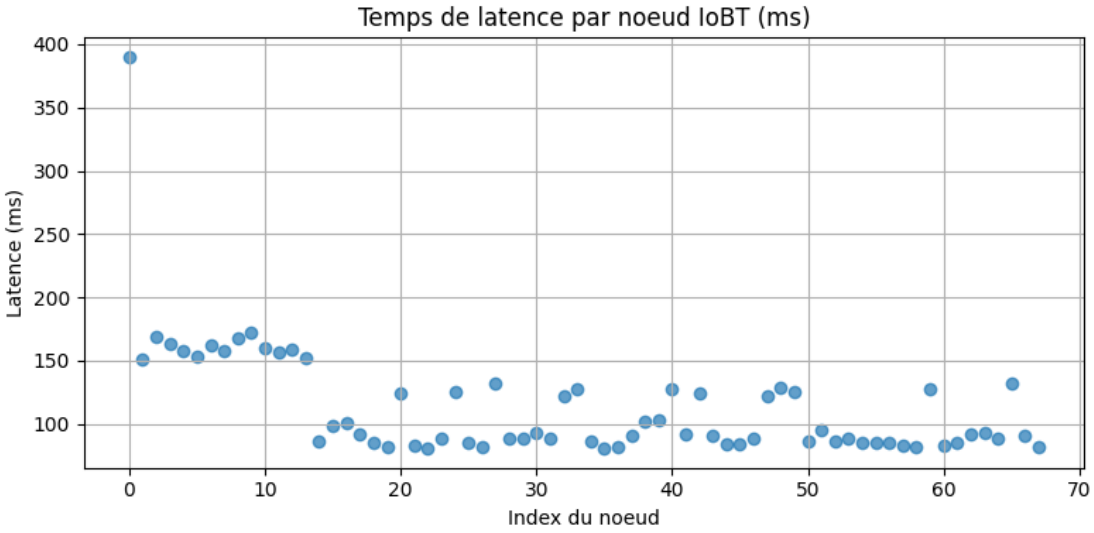

Figure 4.9: Temps de latence d'inférence par nœud IoBT (en millisecondes).

Métriques synthétiques de performance en cybersécurité

L'évaluation du système BIoBT, combinant validation blockchain et authentification biométrique par LSTM, a permis d'obtenir les résultats suivants :

1. **Taux d'authentification confirmée** : 22.06%

   Proportion des objets IoBT initialement acceptés par la blockchain qui ont également satisfait la vérification biométrique.

2. **Taux de détection d'anomalies** : 77.94%

   Pourcentage de nœuds rejetés par le modèle LSTM, considérés comme suspects ou incohérents sur le plan comportemental.

3. **Latence moyenne d'inférence** : 113.15 ms

   Temps moyen nécessaire pour que le modèle LSTM rende une décision d'authentification.

4. **Latence maximale observée** : 3890.71 ms

   Pire cas mesuré, toujours compatible avec une application temps réel dans un contexte militaire.

Ces indicateurs confirment que l'architecture proposée permet une défense en profondeur : l'approche biométrique fondée sur l'apprentissage profond complète efficacement la vérification structurelle offerte par la blockchain. Elle permet de discriminer les

comportements anormaux ou falsifiés tout en conservant une latence faible, compatible avec les exigences de réactivité des systèmes embarqués sur le champ de bataille.

4.5.8 Révocation d'une entité IoBT

Cette section permet de retirer rapidement toute entité IoBT compromise ou malveillante du système. Grâce à la détection des comportements suspects le CCC peut procéder à la révocation, assurant ainsi que seules les entités légitimes et sûres restent actives au sein du réseau. Ce mécanisme renforce la sécurité globale du système en évitant que des dispositifs potentiellement dangereux ne perturbent les opérations ou n'altèrent l'intégrité des communications.

Les résultats de la simulation montrent la détection et la révocation de plusieurs nœuds au comportement suspect :

```
Détection des IoBT au comportement suspect...

Révocation de IoBT : Node_09...
   Raison     : Probabilité ambiguë détectée dans la zone grise
   Timestamp  : 2025-05-21T15:17:28.047305Z
   Statut     : Révocation enregistrée

Révocation de IoBT : Node_36...
   Raison     : Probabilité ambiguë détectée dans la zone grise
   Timestamp  : 2025-05-21T15:17:28.047386Z
   Statut     : Révocation enregistrée

Révocation de IoBT : Node_37...
   Raison     : Probabilité ambiguë détectée dans la zone grise
   Timestamp  : 2025-05-21T15:17:28.047410Z
   Statut     : Révocation enregistrée

Révocation de IoBT : Node_63...
   Raison     : Probabilité ambiguë détectée dans la zone grise
   Timestamp  : 2025-05-21T15:17:28.047460Z
```

```
   Statut     : Révocation enregistrée

Total des IoBT révoqués pour comportement suspect : 4
```

4.6 Conclusion

Ce chapitre a mis en évidence l'apport stratégique de l'intégration d'un modèle LSTM dans le protocole BIoBT, en tant que mécanisme d'authentification biométrique pour les entités IoBT. Les résultats expérimentaux issus de la simulation confirment une amélioration significative de la précision globale, une réduction des erreurs de classification, ainsi qu'un temps d'inférence compatible avec les contraintes temps réel propres aux environnements tactiques. En particulier, le modèle a atteint une précision de 98% sur l'ensemble de validation, accompagné d'une perte stable et de latences inférieures à 400 ms.

L'analyse croisée des courbes d'apprentissage, de la matrice de confusion et des métriques de classification (rappel, précision, F1-score) atteste de la robustesse du modèle et de sa capacité à distinguer avec fiabilité les comportements légitimes des intrusions potentielles. Ces performances valident l'approche de double authentification : blockchain + Deep Learning comme solution robuste et scalable. Enfin, ces résultats ouvrent la voie à plusieurs perspectives prometteuses : déploiement à grande échelle sur plateformes réelles, adaptation à des flux de données multisources, ou encore exploration d'architectures plus avancées (GRU, transformers, modèles hybrides) pour renforcer davantage la sécurité des systèmes militaires distribués et autonomes.

Conclusion Générale

Au terme de ce travail, nous avons proposé une étude novatrice pour renforcer la sécurité des systèmes militaires connectés, en particulier dans le contexte des réseaux de l'IoBT. Cette contribution s'est traduite par l'amélioration du protocole BIoBT à travers l'intégration d'un modèle d'apprentissage profond de type LSTM, destiné à l'authentification biométrique dynamique des dispositifs.

Dans une première phase, nous avons étudié le contexte opérationnel et les menaces cybernétiques pesant sur les architectures militaires distribuées. L'état de l'art a permis d'identifier les limites des mécanismes classiques d'authentification et de détection d'intrusion, justifiant le recours à des techniques basées sur l'intelligence artificielle. Nous avons ensuite détaillé la conception et la simulation du protocole amélioré, incluant à la fois l'enregistrement décentralisé via blockchain et une couche d'authentification adaptative par Deep Learning.

Les résultats obtenus sont particulièrement encourageants. Le modèle LSTM implémenté a atteint une précision de 98% sur l'ensemble de validation, tout en maintenant une latence moyenne d'inférence compatible avec les contraintes temps réel (113 ms). Lors de la simulation d'un scénario d'authentification, seuls 22% des objets IoBT ont été acceptés par la couche biométrique, illustrant ainsi un taux de détection d'anomalies élevé (77.94%) et une capacité de filtrage renforcée. Ces performances démontrent l'efficacité de l'approche à deux niveaux – blockchain structurelle et Deep Learning comportemental pour sécuriser les échanges dans des environnements tactiques hostiles.

Toutefois, cette recherche constitue également une base sur laquelle de nombreuses améliorations peuvent être envisagées. Parmi les perspectives prometteuses, on peut citer :

- L'optimisation de l'architecture du modèle (notamment l'usage de GRU ou de Transformers),
- L'intégration de sources biométriques et contextuelles plus variées (rythmes physiologiques, localisation dynamique, données environnementales),
- L'adaptation du système à des plateformes matérielles embarquées aux ressources limitées,
- Le renforcement de l'explicabilité des décisions du modèle via des approches XAI (eXplainable AI),
- Et enfin, la validation du système dans des scénarios réels de terrain via des expérimentations à grande échelle.

En conclusion, cette recherche apporte une contribution concrète à la sécurisation intelligente des réseaux IoBT, en combinant des mécanismes cryptographiques décentralisés et des capacités de détection intelligentes. Elle ouvre la voie à des solutions hybrides, adaptatives et résilientes, répondant aux exigences croissantes de cybersécurité dans les environnements militaires de demain.

Annexe

Dans cette séction nous présentons deux parties principales. La première partie est consacrée à la modélisation d'un système de classification pour l'IoBT (Internet of Battlefield Things) en utilisant un réseau de neurones LSTM (Long Short-Term Memory). Elle contient l'ensemble du code source relatif à la préparation des données, à l'entraînement du modèle et à l'évaluation de ses performances. La deuxième partie, quant à elle, est dédiée à la simulation de l'intégration du modèle entraîné dans un environnement IoBT, illustrant ainsi comment le modèle peut être appliqué en pratique avec un protocole spécifique au champ de bataille. Le code source de cette simulation y est également inclus pour démontrer l'interopérabilité entre l'intelligence artificielle et les communications tactiques.

Partie 01:Modélisation d'un Modèle de Classification pour l'IoBT avec LSTM

Dans cette partie, nous détaillons l'implémentation du modèle LSTM pour l'authentification, réalisée sur Google Colab. L'objectif est de présenter chaque étape du pipeline, depuis l'importation des bibliothèques jusqu'à la visualisation des résultats, en expliquant le rôle de chaque bloc de code.

Étape 01 : Importation des bibliothèques

Nous commençons par importer toutes les bibliothèques nécessaires à la manipulation des données, au prétraitement, à la construction et à l'évaluation du modèle LSTM.

```
1 # Importation de la bibliothèque Pandas pour la manipulation des donné
    es (dataframes)
2 import pandas as pd
```

```
# Importation de NumPy pour les opérations mathématiques sur les tableaux
import numpy as np
# Importation de TensorFlow, bibliothèque principale pour le Deep Learning
import tensorflow as tf
# Importation de la fonction de séparation du dataset en ensembles d'entraînement et de test
from sklearn.model_selection import train_test_split
# Importation de l'outil de normalisation des données (mise à l échelle entre 0 et 1)
from sklearn.preprocessing import MinMaxScaler
# Importation de Matplotlib pour la visualisation des données (graphiques)
import matplotlib.pyplot as plt
# Importation du module pour la régularisation afin d éviter l overfitting
from tensorflow.keras import regularizers
# Importation de l API séquentielle de Keras pour empiler les couches du modèle
from tensorflow.keras.models import Sequential
# Importation des couches nécessaires au modèle : Input, bruit gaussien, LSTM, Dropout, Dense
from tensorflow.keras.layers import Input, GaussianNoise, LSTM, Dropout, Dense
# Importation des callbacks pour l'arrêt précoce et la sauvegarde du meilleur modèle
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint
# Importation de la matrice de confusion pour évaluer les performances du modèle
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
# Importation du rapport de classification (précision, rappel, f1-score, etc.)
from sklearn.metrics import classification_report
# Importation du module permettant de monter Google Drive dans Colab
from google.colab import drive
# Pour charger un modèle Keras déjà entraîné
from tensorflow.keras.models import load_model
# Pour créer des visualisations statistiques, comme les heatmaps
```

```
import seaborn as sns
```

Listing 4.1: Importation des bibliothèques

Étape 02 : Chargement du Dataset

Le dataset est stocké sur Google Drive. Nous montons donc le Drive dans Colab, puis chargeons le fichier CSV dans un DataFrame Pandas.

```
# Montage du répertoire Google Drive pour accéder aux fichiers stockés en ligne
drive.mount('/content/drive')
# Définition du chemin vers le fichier CSV stocké dans Google Drive
dataset_path = '/content/drive/MyDrive/IoTNetworkDatasetCHaimaYousra.csv'
# Chargement du dataset CSV dans un DataFrame Pandas
data = pd.read_csv(dataset_path)
# Affichage des premières lignes du dataset pour inspection rapide
print(data.head())
```

Listing 4.2: Chargement du dataset depuis Google Drive

Étape 03 : Analyse de Corrélation et Sélection des Features

Avant de construire le modèle, il est important d'analyser la corrélation entre les variables du dataset afin de sélectionner les features les plus pertinentes pour la tâche d'authentification.

```
# Calcul des corrélations avec la variable cible
correlation_matrix = data.corr()
# Sélection des 15 features les plus corrélées (en valeur absolue) avec 'Normal'
top_features = correlation_matrix['Normal'].abs().sort_values(ascending=False).head(16).index # 16 = 15 + 'Normal'
# Sous-matrice de corrélation
top_corr_matrix = data[top_features].corr()
# Affichage de la heatmap
plt.figure(figsize=(12, 10))
sns.heatmap(
```

```
    top_corr_matrix, # La matrice de corrélation filtrée pour les
        features importantes
    annot=True,      # Affiche les valeurs numériques dans chaque case
    cmap='coolwarm', # Palette de couleurs allant du bleu (corrélation
        négative) au rouge (positive)
    square=True,     # Force les cases à être carrées pour une
        apparence uniforme
    linewidths=0.5,  # Épaisseur des lignes de séparation entre les
        cases
    cbar_kws={"shrink": 0.8} # Réduit la taille de la barre de couleur
        (colorbar)
)
plt.xticks(rotation=90, fontsize=9)
plt.yticks(rotation=0, fontsize=9)
plt.title("Matrice de corrélation des principales features réseau",
    fontsize=16)
plt.tight_layout()
plt.show()
```

Listing 4.3: Analyse de corrélation et sélection des features

Étape 04 : Prétraitement des Données

Le prétraitement constitue une étape cruciale pour préparer les données avant l'entraînement du modèle. Il comprend la sélection des attributs les plus pertinents, leur normalisation, ainsi que la restructuration (reshape) des séquences pour les adapter au format requis par le réseau LSTM. Une fois cette structure séquentielle mise en place, le jeu de données est divisé en trois ensembles distincts : entraînement, validation et test. Cette organisation garantit un apprentissage efficace tout en permettant une évaluation rigoureuse des performances du modèle.

- **Sélection et Normalisation des Attributs Pertinents :** Dans cette première étape du prétraitement, les caractéristiques les plus significatives sont extraites à partir de l'ensemble de données en excluant la variable cible Normal. Cette réduction dimensionnelle permet de concentrer l'apprentissage du modèle sur les attributs les plus informatifs. Une normalisation via la méthode MinMaxScaler est ensuite appliquée pour garantir une mise à l'échelle uniforme des données,

essentielle pour les réseaux de neurones sensibles aux amplitudes des entrées.

```
# 1. Selectionner les top features (hors 'Normal')
top_features_no_target = [feat for feat in top_features if feat
    != 'Normal']
# 2. Separer X et y en utilisant seulement les top features
X = data[top_features_no_target]
y = data['Normal']
# 3. Normaliser les donnees
scaler = MinMaxScaler()
X_scaled = scaler.fit_transform(X)
# Confirmation que la normalisation a ete effectuée avec succes
print("Donnees normalisees.")
```

Listing 4.4: Prétraitement des donnees avec selection des top features

- **Restructuration des Données pour l'Entraînement avec un Modèle LSTM :** Avant d'entraîner le modèle LSTM, il est nécessaire de restructurer les données normalisées pour qu'elles soient compatibles avec LSTM . Cette étape consiste à transformer la matrice des données en un tenseur à trois dimensions de la forme (nombre d'échantillons, nombre de pas de temps, nombre de features). Ici, chaque échantillon est considéré comme une séquence d'un seul pas de temps avec plusieurs caractéristiques.

```
# Reshaper les données pour les rendre compatibles avec un modèle
    LSTM
X_scaled = X_scaled.reshape((X_scaled.shape[0], 1, X_scaled.shape
    [1]))

# Affichage de la nouvelle forme du tableau après reshaping
print(f"Nouvelle forme des données : {X_scaled.shape}")
```

Listing 4.5: Restructuration des données pour le modèle LSTM

- **Division du Dataset en Ensembles d'Entraînement, de Validation et de Test (70/10/20):**

  Afin de garantir une évaluation fiable et d'éviter le surapprentissage, le dataset est divisé en trois ensembles distincts :

- **Ensemble d'entraînement** (70%) : utilisé pour ajuster les paramètres du modèle.
- **Ensemble de validation** (10%) : utilisé pour ajuster les hyperparamètres et surveiller le surapprentissage.
- **Ensemble de test** (20%) : utilisé pour évaluer la performance finale du modèle sur des données jamais vues.

La division s'effectue en deux étapes successives à l'aide de la fonction `train_test_split` de scikit-learn.

```
# Étape 1 : Diviser en 70% train et 30% temporaire (test + validation)
X_train, X_temp, y_train, y_temp = train_test_split(X_scaled, y, test_size=0.3, random_state=42)
# Étape 2 : Diviser les 30% restants en 2/3 test et 1/3 validation
X_val, X_test, y_val, y_test = train_test_split(X_temp, y_temp, test_size=2/3, random_state=42)
# Affichage des tailles avec pourcentages
total_samples = X_scaled.shape[0]
print(f"Taille train (70%) : {X_train.shape[0]} échantillons")
print(f"Taille test (20%)  : {X_test.shape[0]} échantillons")
print(f"Taille validation (10%) : {X_val.shape[0]} échantillons")
```

Listing 4.6: Division du dataset en ensembles d'entraînement, de validation et de test (70/10/20)

Étape 05 : Création et Compilation d'un Modèle LSTM Régularisé pour la Classification Binaire

Pour résoudre la tâche de classification binaire, nous avons conçu un modèle LSTM intégrant plusieurs techniques de régularisation afin de limiter le surapprentissage. L'architecture comprend l'ajout de bruit gaussien, la régularisation L2 sur les poids, et une couche de dropout. La compilation du modèle utilise la fonction de perte `binary_crossentropy` et l'optimiseur Adam.

```
# Initialisation du modèle séquentiel (les couches sont ajoutées successivement)
```

```
model = Sequential([
# Première couche d'entrée : définit la forme d'entrée du modèle
Input(shape=(X_train.shape[1], X_train.shape[2])),
# Ajouter du bruit (GaussianNoise) pour aider à la régularisation et é
    viter l'overfitting
GaussianNoise(0.1),
# Ajouter une couche LSTM avec 8 unités, régularisation L2 sur les
    poids (valeur de 0.05)
LSTM(8, kernel_regularizer=regularizers.l2(0.05)),
# Ajouter une couche Dropout pour réduire l'overfitting
Dropout(0.5),
# Ajouter une couche Dense avec 4 neurones et activation ReLU,
    regularisation L2 (0.05)
Dense(4, activation='relu', kernel_regularizer=regularizers.l2(0.05)),
# Derniere couche Dense avec 1 neurone de sortie, activation sigmoïde
    pour classification binaire
Dense(1, activation='sigmoid')
])
# Message pour confirmer que le modèle a ete cree
print("Modele LSTM cree.")
# --- Compiler le modele ---
# La fonction de perte est la "binary_crossentropy" pour la
    classification binaire
model.compile(optimizer='adam',
              loss='binary_crossentropy',
              metrics=['accuracy'])
# Message pour confirmer que le modele a été compile
print("Modele compile.")
```

Listing 4.7: Création et compilation d'un modèle LSTM régularisé

Étape 06 : Configuration des Callbacks et Entraînement du Modèle LSTM

Pour optimiser l'entraînement du modèle et éviter le surapprentissage, deux callbacks sont utilisés : **EarlyStopping** (arrêt précoce) et **ModelCheckpoint** (sauvegarde du meilleur modèle). Le modèle est ensuite entraîné, puis évalué sur l'ensemble de test.

```
# EarlyStopping arrête l'entraînement si la perte de validation ('val_loss') ne s'améliore pas pendant 3 époques.
early_stop = EarlyStopping(
    monitor='val_loss',
    patience=3,
    restore_best_weights=True
)
print("Callback EarlyStopping défini.")
# ModelCheckpoint sauvegarde le meilleur modèle selon la perte de validation.
checkpoint = ModelCheckpoint(
    'best_modelCHY.h5',
    monitor='val_loss',
    save_best_only=True,
    mode='min',
    verbose=1
)
print("Callback ModelCheckpoint défini.")
# --- Entraîner le modèle ---
history = model.fit(
    X_train, y_train,
    epochs=100,
    batch_size=64,
    validation_data=(X_val, y_val),
    callbacks=[early_stop, checkpoint]
)
print("Entraînement terminé.")
# --- Évaluation ---
loss, accuracy = model.evaluate(X_test, y_test)
print(f"\nTest Accuracy: {accuracy*100:.2f}%")
print(f"Test Loss: {loss:.4f}")
```

Listing 4.8: Callbacks et entraînement du modèle LSTM

Étape 07 : Visualisation de l'Évolution de la Précision et de la Perte pendant l'Entraînement du Modèle LSTM

Afin d'analyser le comportement du modèle pendant l'entraînement, nous visualisons l'évolution de la précision et de la perte sur les ensembles d'entraînement et de validation. Cela permet de détecter rapidement un éventuel surapprentissage ou une sous-apprentissage du modèle.

```
# Définir la taille de la figure pour qu'elle soit large et bien visible
plt.figure(figsize=(12,5))
# --- Graphique de la précision (Accuracy) ---
plt.subplot(1,2,1)  # Crée une sous-figure pour le premier graphique (1 ligne, 2 colonnes, 1er graphique)
# Affichage de l'évolution de la précision pendant l'entraînement et la validation
plt.plot(history.history['accuracy'], label='Train Accuracy')  # Précision sur les données d'entraînement
plt.plot(history.history['val_accuracy'], label='Validation Accuracy')  # Précision sur les données de validation
plt.xlabel('Epochs')  # L'axe des x représente les époques
plt.ylabel('Accuracy')  # L'axe des y représente la précision
plt.legend()  # Affiche la légende pour différencier les courbes d'entraînement et de validation
# --- Graphique de la perte (Loss) ---
plt.subplot(1,2,2)  # Crée une sous-figure pour le deuxième graphique (1 ligne, 2 colonnes, 2e graphique)
# Affichage de l'évolution de la perte pendant l'entraînement et la validation
plt.plot(history.history['loss'], label='Train Loss')  # Perte sur les données d'entraînement
plt.plot(history.history['val_loss'], label='Validation Loss')  # Perte sur les données de validation
plt.xlabel('Epochs')  # L'axe des x représente les époques
plt.ylabel('Loss')  # L'axe des y représente la perte
plt.legend()  # Affiche la légende pour différencier les courbes d'entraînement et de validation
# Ajuster la disposition des sous-figures pour éviter les chevauchements
```

```
plt.tight_layout()
# Affichage des graphiques
plt.show()
```

Listing 4.9: Visualisation de l'évolution de la précision et de la perte du modèle LSTM

Étape 08 : Affichage de la Matrice de Confusion pour l'Évaluation du Modèle LSTM

Afin d'évaluer les performances du modèle LSTM sur les données de test, nous utilisons une matrice de confusion. Celle-ci permet de visualiser la répartition des prédictions correctes et incorrectes du modèle en comparant les classes réelles avec les classes prédites. En appliquant un seuil de 0.5 sur les probabilités de sortie du modèle, nous convertissons ces dernières en étiquettes binaires (0 ou 1). La matrice obtenue offre une vision claire de la capacité du modèle à distinguer entre les deux classes.

```
# Le modèle prédit les probabilités de chaque classe (0 ou 1) pour
    chaque échantillon du jeu de test
y_pred_probs = model.predict(X_test)
# Appliquer un seuil de 0.5 pour classifier les probabilités en é
    tiquettes binaires (0 ou 1)
y_pred = (y_pred_probs > 0.5).astype("int32")  # Transformation des
    probabilités en valeurs binaires
# Calculer la matrice de confusion : compare les étiquettes réelles (
    y_test) et les étiquettes prédites (y_pred)
cm = confusion_matrix(y_test, y_pred)
# Créer un objet pour afficher la matrice de confusion avec les é
    tiquettes appropriées
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=[0,
    1])  # Labels pour les classes 0 et 1
# Afficher la matrice de confusion
plt.figure(figsize=(6,6))
disp.plot(cmap='Blues', values_format='d')  # 'd' pour afficher les
    valeurs sous forme entière
plt.show()  # Afficher le graphique
```

Listing 4.10: Affichage de la matrice de confusion pour l'évaluation du modèle LSTM

Étape 09 : Rapport de Classification : Évaluation Détaillée des Performances du Modèle

Après l'évaluation globale du modèle, il est important d'analyser en détail ses performances sur chaque classe. Le rapport de classification fournit des métriques telles que la précision, le rappel et le score F1 pour chacune des classes, permettant ainsi de mieux comprendre les points forts et les faiblesses du modèle.

```
# Le rapport de classification fournit des métriques détaillées comme la précision, le rappel, le F-score, etc.
print("\n--- Rapport de Classification ---")
# Le rapport affiche les performances pour chaque classe, en utilisant les étiquettes 'Non-Authentic (0)' et 'Authentic (1)'
print(classification_report(y_test, y_pred, target_names=['Non-Authentic (0)', 'Authentic (1)']))
```

Listing 4.11: Affichage du rapport de classification

Étape 10 : Chargement du Modèle et Prédiction sur de Nouvelles Données Normalisées

Dans cette étape, nous chargeons le modèle LSTM préalablement entraîné et sauvegardé, puis nous appliquons une normalisation Min-Max sur de nouveaux échantillons de données. Ces données sont ensuite formatées pour être compatibles avec l'entrée du modèle. Enfin, nous réalisons la prédiction et affichons le résultat pour chaque enregistrement.

```
# Chargement du modèle pré-entraîné
model = load_model('best_modelCHY.h5')  # Charge le modèle depuis le fichier HDF5
print("Modèle chargé avec succès.")  # Message de confirmation
# Initialisation du scaler pour la normalisation Min-Max
scaler = MinMaxScaler()  # Crée une instance de MinMaxScaler
# Préparation des données de test (3 exemples avec les 15 features sélectionnées)
data = {
    "normal": [
        # Premier enregistrement
```

```
        {
            'Month': -1.41, 'Dst_Port': -0.32, 'ACK_Flag_Cnt': 1.0,
            'Src_Port': 0.85, 'Minute': 1.78, 'Protocol': 1.1,
            'Dst_2': -0.77, 'Fwd_Pkt_Len_Max': 0.55, 'Dst_3': 0.29,
            'Fwd_Pkt_Len_Mean': 0.31, 'TotLen_Fwd_Pkts': 0.37,
            'Bwd_Pkt_Len_Mean': -0.15, 'Pkt_Len_Max': 0.48,
            'Bwd_Pkt_Len_Min': -0.05, 'Fwd_Pkt_Len_Std': 0.12
        },
        # Deuxième enregistrement
        {
            'Month': -1.30, 'Dst_Port': -0.26, 'ACK_Flag_Cnt': 0.9,
            'Src_Port': 0.78, 'Minute': 1.65, 'Protocol': 0.9,
            'Dst_2': -0.73, 'Fwd_Pkt_Len_Max': 0.45, 'Dst_3': 0.26,
            'Fwd_Pkt_Len_Mean': 0.25, 'TotLen_Fwd_Pkts': 0.28,
            'Bwd_Pkt_Len_Mean': -0.12, 'Pkt_Len_Max': 0.44,
            'Bwd_Pkt_Len_Min': -0.07, 'Fwd_Pkt_Len_Std': 0.15
        },
        # Troisième enregistrement
        {
            'Month': -1.32, 'Dst_Port': -0.30, 'ACK_Flag_Cnt': 1.0,
            'Src_Port': 0.72, 'Minute': 1.70, 'Protocol': 1.2,
            'Dst_2': -0.74, 'Fwd_Pkt_Len_Max': 0.52, 'Dst_3': 0.28,
            'Fwd_Pkt_Len_Mean': 0.28, 'TotLen_Fwd_Pkts': 0.33,
            'Bwd_Pkt_Len_Mean': -0.14, 'Pkt_Len_Max': 0.47,
            'Bwd_Pkt_Len_Min': -0.06, 'Fwd_Pkt_Len_Std': 0.14
        }
    ]
}
# Conversion des données en DataFrame pandas
new_samples = pd.DataFrame(data["normal"])  # Crée un DataFrame à
    partir du dictionnaire
# Liste des features dans l'ordre attendu par le modèle
expected_features = [
    'Month', 'Dst_Port', 'ACK_Flag_Cnt', 'Src_Port', 'Minute',
    'Protocol', 'Dst_2', 'Fwd_Pkt_Len_Max', 'Dst_3', 'Fwd_Pkt_Len_Mean
        ',
    'TotLen_Fwd_Pkts', 'Bwd_Pkt_Len_Mean', 'Pkt_Len_Max', '
        Bwd_Pkt_Len_Min',
    'Fwd_Pkt_Len_Std'
```

```
]
# Réorganisation des colonnes selon l'ordre attendu
new_samples = new_samples[expected_features]  # Réorganise les colonnes du DataFrame
# Normalisation des données avec MinMaxScaler
new_samples_scaled = scaler.fit_transform(new_samples)  # Applique la normalisation 0-1
# Remodelage des données pour l'entrée LSTM (format 3D)
# Format requis : [échantillons, pas de temps, features]
new_samples_scaled = new_samples_scaled.reshape(
    new_samples_scaled.shape[0],  # Nombre d'échantillons (3)
    1,                            # Pas de temps (1 car pas de séquence temporelle)
    new_samples_scaled.shape[1]   # Nombre de features (15)
)
# Prédiction avec le modèle chargé
predictions = model.predict(new_samples_scaled)  # Exécute les prédictions
# Affichage des résultats
for i, p in enumerate(predictions):
    # Conversion de la sortie numérique en libellé
    label = "Authentic" if p[0] < 0.5 else "Non-Authentic"  # Seuil à 0.5
    print(f"Prédiction pour l'enregistrement {i+1}: {label}")  # Résultat formaté
```

Listing 4.12: Chargement du modèle et prédiction sur de nouvelles données normalisées

Partie 02: Code source du protocole de simulation

Les extraits suivants proviennent du notebook utilisé pour la simulation du protocole d'authentification BioBT basé sur LSTM. Chaque bloc est commenté pour une meilleure compréhension.

Étape 01: Chargement des bibliothèques

```
# Traitement numérique et aléatoire
import numpy as np
```

```
import random
import uuid
import hashlib

import time
# Réseaux et graphes
import networkx as nx                    # pour créer, manipuler et
    analyser des graphes ou réseaux.
from scipy.spatial import distance       # Pour le calcul des
    distances euclidiennes
from scipy.spatial import KDTree         # Pour  l optimisation  de
    voisinage

# Visualisation classique
import matplotlib.pyplot as plt
import seaborn as sns

# Visualisation interactive (optionnelle)
import plotly.graph_objects as go
import plotly.express as px

# Machine Learning / Deep Learning
import tensorflow as tf
from keras.models import load_model

# Journalisation et temps
import logging
from datetime import datetime

# Traitement des données
import pandas as pd
from google.colab import drive
```

Listing 4.13: Chargement des bibliothèques

Étape 2: Définition des paramètres de simulation

```
    #  Définition des paramètres de simulation
SIM_PARAMS = {
```

```
    'Network Area': (500, 500),
    'Number of IoBT Nodes': 80,
    'Initial Energy': 2.0,  # en Joules
    'Packet Size': 512,     # en bytes
    'Simulation Time': 200, # en secondes
    'Data Flow': 'Random Data',
    'Traffic Type': 'Event-driven sensing (Military context)',
    'Routing Protocol': 'Greedy Perimeter Stateless Routing (GPSR)',
    'Medium Access': 'BN-MAC',
    'trust_server': {'x': 250, 'y': 250},
    'Authentication Delay Blockchain': 0.2,  # en secondes
    'Authentication Delay LSTM': 0.05,       # en secondes
    'Tx Energy Cost': 50e-9,     # J/bit
    'Rx Energy Cost': 50e-9,     # J/bit
    'Processing Energy': 10e-9,  # J/bit
    'Mobility': False
}

#  Affichage structuré des paramètres
print("=== Paramètres de Simulation ===")
print(f" Surface Réseau        : {SIM_PARAMS['Network Area'][0]}m    {SIM_PARAMS['Network Area'][1]}m")
print(f" Nombre de Noeuds IoBT : {SIM_PARAMS['Number of IoBT Nodes']}")
print(f" Énergie Initiale par Noeud : {SIM_PARAMS['Initial Energy']} J")
print(f" Taille de Paquet      : {SIM_PARAMS['Packet Size']} bytes")
print(f" Durée de Simulation   : {SIM_PARAMS['Simulation Time']} secondes")
print(f" Type de Flux          : {SIM_PARAMS['Data Flow']}")
print(f" Type de Trafic        : {SIM_PARAMS['Traffic Type']}")
print(f" Protocole de Routage  : {SIM_PARAMS['Routing Protocol']}")
print(f" Accès Médium (MAC)    : {SIM_PARAMS['Medium Access']}")
print(f" Position trust_server : (x={SIM_PARAMS['trust_server']['x']} m, y={SIM_PARAMS['trust_server']['y']} m)")
print(f" Délai Auth Blockchain : {SIM_PARAMS['Authentication Delay Blockchain']} s")
print(f" Délai Auth LSTM       : {SIM_PARAMS['Authentication Delay LSTM']} s")
```

```
print(f" Coût Tx (J/bit)        : {SIM_PARAMS['Tx Energy Cost']}")
print(f" Coût Rx (J/bit)        : {SIM_PARAMS['Rx Energy Cost']}")
print(f" Énergie traitement     : {SIM_PARAMS['Processing Energy']}")
print(f" Mobilité des n uds     : {'Activée' if SIM_PARAMS['Mobility']
    else 'Désactivée'}")
print("==================================")
```

Listing 4.14: Définition des paramètres de simulation

Étape 3: Génération de la topologie initiale et visualisation

```

# Générer des positions aléatoires pour les n uds IoBT dans un espace
    2D
positions = np.random.rand(SIM_PARAMS['Number of IoBT Nodes'], 2)
positions[:, 0] *= SIM_PARAMS['Network Area'][0]  # X
positions[:, 1] *= SIM_PARAMS['Network Area'][1]  # Y

# Affichage des positions extrêmes (min, max)
x_min, x_max = positions[:, 0].min(), positions[:, 0].max()
y_min, y_max = positions[:, 1].min(), positions[:, 1].max()
print(f"Position X     [{x_min:.2f}, {x_max:.2f}]")
print(f"Position Y     [{y_min:.2f}, {y_max:.2f}]")

# Visualisation de la topologie décentralisée
plt.figure(figsize=(8, 8))
plt.scatter(positions[:, 0], positions[:, 1], c='pink', label='Noeuds
    IoBT', s=60, edgecolors='black')
plt.scatter(SIM_PARAMS['trust_server']['x'], SIM_PARAMS['trust_server'
    ]['y'], c='darkblue', label='trust_server', s=100, marker='X')

#  Annotation de la station de base
plt.annotate('trust_server',
             (SIM_PARAMS['trust_server']['x'], SIM_PARAMS['
                trust_server']['y']),
             textcoords="offset points", xytext=(10,10), ha='center',
                fontsize=10, color='darkblue')

# Annotation des n uds (optionnelle)
```

```
for i, (x, y) in enumerate(positions):
    plt.text(x + 3, y + 3, str(i), fontsize=8, color='black')

plt.title("Topologie initiale des n uds IoBT", fontsize=14)
plt.xlabel("Coordonnée X (m)")
plt.ylabel("Coordonnée Y (m)")
plt.xlim(0, SIM_PARAMS['Network Area'][0])
plt.ylim(0, SIM_PARAMS['Network Area'][1])
plt.legend()
plt.grid(True, linestyle='--', alpha=0.5)
plt.box(True)
plt.tight_layout()
plt.show()
```

Listing 4.15: Génération de la topologie initiale et visualisation

Étape 4: Création du système CCC_Sys sur la blockchain

```
    def sha256(x: str) -> str:
#Retourne le hash SHA-256  d une  chaîne de caractères
    return hashlib.sha256(x.encode()).hexdigest()

def print_transaction(tx: dict, title ):
#Affiche le contenu d'une transaction blockchain.
    print(f"\n {title}")
    print("-" * (len(title) + 4))
    for key, value in tx.items():
        print(f"{key} : {value}")
    print("-" * 40)

#  Génération des identifiants cryptographiques
ID_CCC_Owner = str(uuid.uuid4())
ID_CCC = str(uuid.uuid4())

print(" Identifiants générés :")
print(f" ID_CCC_Owner : {ID_CCC_Owner}")
print(f" ID_CCC       : {ID_CCC}")

#  Construction de  l identit é du système CCC
```

```
ID_CCC_Sys = sha256(ID_CCC_Owner + ID_CCC)
print(f"\n ID_CCC_Sys (hashé) : {ID_CCC_Sys}")

#  Simulation du contrat intelligent et envoi de la transaction Tx1
SC_BioBT = "SmartContract_Bytecode"  # Simulé
TX1 = {
    'ID_CCC_Sys': ID_CCC_Sys,
    'contract': SC_BioBT,
    'timestamp': random.randint(1_700_000_000, 1_800_000_000),
    'signed_by': ID_CCC_Owner
}

#  Affichage structuré de la transaction blockchain Tx1
print_transaction(TX1, title="Transaction Blockchain Tx1")
```

Listing 4.16: Création du système CCC_Sys sur la blockchain

Étape 5: Association des IoBT

```
    def print_iobt_info(node: dict, index: int):
    #Affiche les informations d'un IoBT associé.
    print(f"\n IoBT Node #{index + 1}")
    print("-" * 30)
    print(f"ID_owner      : {node['owner']}")
    print(f"ID_iobt       : {node['id']}")
    print(f"AssociationID: {node['assoc_id']}")
    print(f"TX2           : {node['TX2']}")
    print("-" * 30)

def associate_iobt(n_show: int = 5):
    #Associe les n uds IoBT au CCC_Sys avec génération d'identifiants
        .
    iobt_nodes = []
    seen_association_ids = set()

    for i in range(SIM_PARAMS['Number of IoBT Nodes']):
        ID_owner = str(uuid.uuid4())
        ID_iobt = str(uuid.uuid4())
        AssociationID = sha256(ID_owner + ID_iobt)
```

```
20
21          # Simuler unicité sur la blockchain
22          if AssociationID in seen_association_ids:
23              print(f" Doublon détecté pour AssociationID (Noeud {i})")
24              continue
25
26          seen_association_ids.add(AssociationID)
27
28          TX2 = {
29              'ID_CCC_Sys': ID_CCC_Sys,
30              'AssociationID': AssociationID,
31              'timestamp': random.randint(1_700_000_000, 1_800_000_000),
32              'status': "Pending"
33          }
34
35          node = {
36              'owner': ID_owner,
37              'id': ID_iobt,
38              'assoc_id': AssociationID,
39              'TX2': TX2
40          }
41
42          iobt_nodes.append(node)
43
44          # Affiche les premiers pour contrôle
45          if i < n_show:
46              print_iobt_info(node, i)
47
48      print(f"\n Total des IoBT associés : {len(iobt_nodes)}")
49      return iobt_nodes
50
51  #  Lancer   l  association
52  iobt_registry = associate_iobt()
```

Listing 4.17: Association des IoBT

Étape 6: Authentification et transmission des données (Authentification 1)

```
1
2  def simulate_blockchain_mining(tx_id: str, status: str, delay: float =
       0.3):
3
4      print(f" Blockchain TxID: {tx_id[:8]}...")
5      print(f"    Enregistrement dans la Blockchain: en cours...")
6      time.sleep(delay)
7      print(f"    Status de la transaction : {' Confirmée' if status ==
           'Valid' else ' Rejetée'}")
8      print("    Bloc ajouté au registre distribué.\n")
9
10 def authenticate_and_send(iobt: dict, verbose: bool = True, fail_rate:
       float = 0.1):
11
12     #Simule l'authentification et la transmission de données d'un IoBT
           avec illustration du traitement blockchain.
13
14     data = f"Data from {iobt['id']}"
15     digest = sha256(data)
16
17     simulate_fail = random.random() < fail_rate
18     if simulate_fail:
19         recomputed_assoc_id = sha256("corrupted" + iobt['id'])
20     else:
21         recomputed_assoc_id = sha256(iobt['owner'] + iobt['id'])
22
23     valid = recomputed_assoc_id == iobt['assoc_id']
24     tx_id = str(uuid.uuid4())  # Simule un identifiant de transaction
           blockchain
25
26     TX3 = {
27         'TxID': tx_id,
28         'ID_CCC_Sys': ID_CCC_Sys,
29         'AssociationID': iobt['assoc_id'],
30         'Data': data,
31         'Digest': digest,
```

```
        'timestamp': datetime.utcnow().isoformat() + 'Z',
        'status': 'Valid' if valid else 'Rejected'
    }

    if verbose:
        print(f"\n Traitement du noeud IoBT ID: {iobt['id'][:8]}...")
        print(f"    AssocID attendu     : {iobt['assoc_id'][:10]}...")
        print(f"    AssocID recalculé   : {recomputed_assoc_id
            [:10]}...")
        print("    Résultat            : " + (" AUTHENTIFIÉ" if valid
            else " REJETÉ"))

    simulate_blockchain_mining(tx_id, TX3['status'], delay=0.3)

    return TX3 if valid else None

# Journaux
TX3_logs = []
TX3_rejected = []

# Authentifier tous les noeuds avec illustration
for i, node in enumerate(iobt_registry):
    tx = authenticate_and_send(node, verbose=(i < 5), fail_rate=0.15)
        # Afficher les 5 premiers
    if tx:
        TX3_logs.append(tx)
    else:
        TX3_rejected.append({'node': node, 'reason': 'Invalid
            Association ID'})

# Résumé global
print(f"\n Total Transactions Valides    : {len(TX3_logs)}")
print(f" Total Transactions Rejetées  : {len(TX3_rejected)}")
```

Listing 4.18: Authentification et transmission des données (Authentification 1)

Étape 7: Authentification biométrique par LSTM(Authentification 2)

```
    # === 1. Chargement sécurisé du modèle ===

drive.mount('/content/drive')

try:
    model_path = "/content/drive/MyDrive/best_modelCHY(2).h5"
    model = load_model(model_path)
    print(f"Modèle LSTM chargé avec succès depuis : {model_path}")
    print(f"Forme attendue des entrées : {model.input_shape}")
except Exception as e:
    print(f"Erreur lors du chargement du modèle : {e}")
    raise

# 2. Paramètres  d entr ée
n_features = 15
threshold = 0.5

# 3. Simulation de TX3_logs

TX3_logs = [{'AssociationID': f"Node_{i:02d}"} for i in range(68)]

# 4. Authentification LSTM pour les IoBT validés par la blockchain
lstm_results = []
lstm_passed = []
lstm_failed = []

print(f"\nDébut de  l  authentification  biométrique (LSTM) pour {len(
    TX3_logs)} noeuds validés par la blockchain...")

for i, node in enumerate(TX3_logs):
    # Simuler un vecteur de 15 caractéristiques (1 pas de temps)
    X_sim = np.random.rand(1, 1, n_features)

    start_time = time.time()
    prob = model.predict(X_sim, verbose=0)[0][0]
    latency = time.time() - start_time
```

```
36
37      status = "AUTHENTIFIÉ" if prob > threshold else "REJETÉ"
38
39      result = {
40          'IoBT_ID': node['AssociationID'][:10] + "...",
41          'probabilité': round(prob, 6),
42          'latence_ms': round(latency * 1000, 2),
43          'status': status
44      }
45
46      lstm_results.append(result)
47
48      if prob > threshold:
49          lstm_passed.append(result)
50      else:
51          lstm_failed.append(result)
52
53      print(f"\n IoBT {result['IoBT_ID']}")
54      print(f"    Probabilité : {result['probabilité']}")
55      print(f"    Latence : {result['latence_ms']} ms")
56      print(f"    Statut : {status}")
57
58  # === 5. Résumé final ===
59  print("\nRésumé de  l  authentification  LSTM :")
60  print(f" - Total authentifiés : {len(lstm_passed)}")
61  print(f" - Total rejetés      : {len(lstm_failed)}")
62  print(f" - Seuil utilisé      : {threshold}")
```

Listing 4.19: Authentification biométrique par LSTM(Authentification 2)

Étape 8: Visualisation des résultats

```
1       # === Evaluation des performance  ===
2
3   # Distribution des probabilités d'authenticité
4   probs = [r['probabilité'] for r in lstm_results]
5   plt.figure(figsize=(8, 4))
6   plt.hist(probs, bins=20, color='skyblue', edgecolor='black')
```

```
plt.axvline(threshold, color='red', linestyle='--', label=f"Seuil ({
    threshold})")
plt.title("Distribution des probabilités d'authenticité")
plt.xlabel("Probabilité prédite")
plt.ylabel("Nombre de noeuds")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()

#   Décisions
labels = ['Authentifiés', 'Rejetés']
values = [len(lstm_passed), len(lstm_failed)]
colors = ['green', 'red']
plt.figure(figsize=(6, 4))
plt.bar(labels, values, color=colors)
plt.title("Résultat de l'authentification LSTM")
plt.ylabel("Nombre de noeuds")
plt.grid(axis='y')
plt.tight_layout()
plt.show()

# atences
latences = [r['latence_ms'] for r in lstm_results]
plt.figure(figsize=(8, 4))
plt.scatter(range(len(latences)), latences, alpha=0.7)
plt.title("Temps de latence par noeud IoBT (ms)")
plt.xlabel("Index du noeud")
plt.ylabel("Latence (ms)")
plt.grid(True)
plt.tight_layout()
plt.show()

# Métriques spécifiques cybersécurité (adaptées)
# --- Résilience opérationnelle ---
total_blockchain_valid = len(TX3_logs)
total_lstm_auth = len(lstm_passed)
total_lstm_reject = len(lstm_failed)

```

```
reinforced_ratio = total_lstm_auth / total_blockchain_valid * 100
attack_detection_rate = total_lstm_reject / total_blockchain_valid *
    100
average_latency = sum(latences) / len(latences)
max_latency = max(latences)

# --- Affichage des résultats ---
print("\n Évaluation des métriques cybersécurité pertinentes :")
print(f" - Taux de renforcement (légitimes confirmés) : {
    reinforced_ratio:.2f}%")
print(f" - Taux de détection d'anomalies (intrusions simulées) : {
    attack_detection_rate:.2f}%")
print(f" - Latence moyenne d'inférence : {average_latency:.2f} ms")
print(f" - Latence maximale observée : {max_latency:.2f} ms")
```

Listing 4.20: Visualisation des résultats

Étape 9: Révocation d'une entité IoBT

```
    # === Révocation des IoBT au comportement suspect ===

revocation_log = []

def est_suspect(probabilité: float, seuil: float = 0.5, marge: float =
    0.05) -> bool:
    """
    Détermine si la probabilité est ambigüe (comportement suspect).
    """
    return seuil - marge <= probabilité <= seuil + marge

def revoke_node(iobt: dict, raison: str = "Comportement suspect détect
    é", verbose: bool = True):
    """
    Révoque un IoBT et enregistre une transaction Tx4.
    """
    timestamp = datetime.utcnow().isoformat() + 'Z'

    TX4 = {
        'TxType': 'Revocation',
```

```
        'ID_CCC_Sys': ID_CCC_Sys,
        'Revoke_AssociationID': iobt['IoBT_ID'],
        'Node_ID': iobt['IoBT_ID'],
        'Reason': raison,
        'Timestamp': timestamp,
        'Status': 'Revoked'
    }

    revocation_log.append(TX4)

    if verbose:
        print(f"\n Révocation de IoBT : {iobt['IoBT_ID']}")
        print(f"    Raison     : {raison}")
        print(f"    Timestamp  : {timestamp}")
        print("    Statut     : Révocation enregistrée")

    return TX4


print("\n Détection des IoBT au comportement suspect...")

for node in lstm_results:
    prob = node['probabilité']

    if est_suspect(probabilité=prob):
        revoke_node(node, raison="Probabilité ambiguë détectée dans la
            zone grise")

print(f"\n Total des IoBT révoqués pour comportement suspect : {len(
   revocation_log)}")
```

Listing 4.21: Révocation d'une entité IoBT

Bibliography

[1] A. Rashid and A. U. R. Khan, "Blockchain-Based Autonomous Authentication and Integrity for Internet of Battlefield Things in C3I System," *IEEE Access*, vol. 10, pp. 91572–91587, 2022.

[2] E. Meaudre, "Évolution des conflits armés, conséquences pour l'anesthésiste-réanimateur militaire ?," *Bulletin de l'Académie Nationale de Médecine*, vol. 208, pp. 1082–1089, Oct. 2024.

[3] A. Estève, "Penser la protection de l'environnement en temps de guerre : les apports de la tradition de la guerre juste:," *Raisons politiques*, vol. N° 77, pp. 55–65, Mar. 2020.

[4] S. Kannadhasan and R. Nagarajan, "Intrusion detection in machine learning based e-shaped structure with algorithms, strategies and applications in wireless sensor networks," *Heliyon*, vol. 10, no. 9, p. e30675, 2024.

[5] M. Pakowski, M. Brzozowski, M. Michalczewski, and M. Myszka, "Methods for testing military radars produced in poland," in *2018 5th IEEE International Workshop on Metrology for AeroSpace (MetroAeroSpace)*, pp. 322–327, IEEE.

[6] M. Beer, C. Thattil, J. F. Haase, W. Brockherde, and R. Kokozinski, "2×192 pixel CMOS SPAD-based flash LiDAR sensor with adjustable background rejection," in *2018 25th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, pp. 17–20, IEEE.

[7] fouad sabry, *livre de communication militaire*. 2024.

[8] R. Jardim, M. Dos Santos, E. Neto, F. M. Muradas, B. Santiago, and M. Moreira, "Design of a framework of military defense system for governance of geoinformation," *Procedia Computer Science*, vol. 199, pp. 174–181, 2022.

[9] S. Jacob and P. Sultana, "A systematic analysis and review on intrusion detection systems using machine learning and deep learning algorithms," *Journal of Computational and Cognitive Engineering*, 07 2024.

[10] M. Korytkowski, "Artificial Neural Networks," in *Advanced Techniques of Artificial Intelligence in IT Security Systems* (M. Korytkowski, ed.), pp. 7–17, Cham: Springer Nature Switzerland, 2024.

[11] D. Gaifulina and I. Kotenko, "Selection of Deep Neural Network Models for IoT Anomaly Detection Experiments," in *2021 29th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*, (Valladolid, Spain), pp. 260–265, IEEE, Mar. 2021.

[12] R. Xin, J. Zhang, and Y. Shao, "Complex network classification with convolutional neural network," vol. 25, no. 4, pp. 447–457, 2020. Publisher: Tsinghua University Press.

[13] W. Yang, D. Zhang, and Y. Fu, "Research of a diagonal recurrent neural network and artificial neural networks," in *2016 International Symposium on Computer, Consumer and Control (IS3C)*, pp. 374–377, IEEE.

[14] B. Nath Saha and A. Senapati, "Long short term memory (LSTM) based deep learning for sentiment analysis of english and spanish data," in *2020 International Conference on Computational Performance Evaluation (ComPE)*, IEEE.

[15] X. Li, B. Lv, and Z. Xiao, "Design military and civil information exchange organization management system based on structural entropy," in *2018 Chinese Control And Decision Conference (CCDC)*, (Shenyang), pp. 2080–2085, IEEE, June 2018.

[16] J. J. Galán, R. A. Carrasco, and A. LaTorre, "Military Applications of Machine Learning: A Bibliometric Perspective," *Mathematics*, vol. 10, p. 1397, Apr. 2022.

[17] K. A. Jallad, M. Aljnidi, and M. S. Desouki, "Anomaly detection optimization using big data and deep learning to reduce false-positive," 2022. Publisher: arXiv Version Number: 1.

[18] T. Steiner, "Using PubPub for scholarly output: Import, citations, collaboration, and zotero," July 17 2020.

[19] M. Lichtman and J. H. Reed, "Anomaly-based intrusion detection of protocol-aware jamming," in *MILCOM 2015 - 2015 IEEE Military Communications Conference*, (Tampa, FL, USA), pp. 269–274, IEEE, Oct. 2015.

[20] I. Traore, I. Woungang, and S. Saad, eds., *Artificial Intelligence for Cyber-Physical Systems Hardening*, vol. 2 of *Engineering Cyber-Physical Systems and Critical Infrastructures*. Cham: Springer International Publishing, 2023.

[21] Y. He, Y. Peng, S. Wang, and D. Liu, "ADMOST: UAV Flight Data Anomaly Detection and Mitigation via Online Subspace Tracking," *IEEE Transactions on Instrumentation and Measurement*, vol. 68, pp. 1035–1044, Apr. 2019.

[22] Z. Wang, S. Zhu, T. Huang, J. Zhu, and H. Fang, "Research on Network Security Active Defense System Oriented to Electric Power Monitoring System," in *2020 IEEE International Conference on Information Technology,Big Data and Artificial Intelligence (ICIBA)*, (Chongqing, China), pp. 883–887, IEEE, Nov. 2020.

[23] R. Gupta, K. Sultania, P. Singh, and A. Gupta, "Security for wireless sensor networks in military operations," in *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, (Tiruchengode), pp. 1–6, IEEE, July 2013.

[24] S. Shaneman and C. Murphy, "Enhancing the Deployment and Security of SIPRNET and JWICS Networks using Intrinsic Fiber Monitoring," in *MILCOM 2007 - IEEE Military Communications Conference*, (Orlando, FL, USA), pp. 1–6, IEEE, Oct. 2007.

[25] R. Knobbe, A. Purtell, and S. Schwab, "Advanced Security Proxies: An Architecture and Implementation for High- Performance Network Firewalls," 1995.

[26] J. Hou, P. Fu, Z. Cao, and A. Xu, "Machine Learning Based DDos Detection Through NetFlow Analysis," in *MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM)*, (Los Angeles, CA), pp. 1–6, IEEE, Oct. 2018.

[27] Giri Kuthethoor, Prakash Sesha, J. Strohm, P. O'Neal, G. Hadynski, D. Climek, J. DelMedico, D. Kiwior, D. Dunbrack, and D. Parker, "Performance analysis of SNMP in airborne tactical networks," in *MILCOM 2008 - 2008 IEEE Military Communications Conference*, (San Diego, CA, USA), pp. 1–7, IEEE, Nov. 2008.

[28] R. Saravanan and P. Sujatha, "A state of art techniques on machine learning algorithms: A perspective of supervised learning approaches in data classification," in *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*, pp. 945–949, IEEE.

[29] I. H. Sarker, "Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions," *SN Computer Science*, vol. 2, no. 6, p. 420, 2021.

[30] D. Liu, M.-C. Lai, and W.-D. Tsay, "Determinants analysis of drone delivery service adoption," in *2020 3rd IEEE International Conference on Knowledge Innovation and Invention (ICKII)*, pp. 1–4, IEEE, Aug. 2020.

[31] A. B. Marc de Fritsch, "L'intelligence artificielle et ses enjeux pour la défense," *Défense Nationale*, 2019.

[32] M. Atighetchi, P. Pal, F. Webber, R. Schantz, C. Jones, and J. Loyall, "Adaptive cyberdefense for survival and intrusion tolerance," *IEEE Internet Computing*, vol. 8, pp. 25–33, Nov.–Dec. 2004.