الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي والبحث العلمي

جامعة سعيدة د. مولاي الطاهر

كلية الرياضيات و الإعلام الآلي و الاتصالات السلكية و اللاسلكية

قسم: الإعلام الآلي

Mémoire de Master en informatique

Spécialité : MICR

T h è m e

SecTalk : système intelligent qui explique les alertes de sécurité en langage naturel

▪ **Présenté par :**

BOUAMOUD Yasmine

▪ **Dirigé par :**

Dr.BOUARARA Hadj Ahmed

Année universitaire 2025-2026

Remerciements

*Avant tout, je remercie **Allah**, le Tout-Puissant, pour m'avoir accordé la santé, la patience et la force nécessaires afin de mener à bien ce travail.*

*Je tiens à adresser mes sincères remerciements à mon encadrant, **M. Bouarara Hadj Ahmed**, pour son accompagnement, ses orientations précieuses, sa disponibilité constante ainsi que la qualité de ses conseils tout au long de l'élaboration de ce mémoire. Son encadrement rigoureux et bienveillant a grandement contribué à la réalisation de ce travail.*

J'exprime également ma reconnaissance à l'ensemble de mes professeurs, pour la qualité de leur enseignement, leur engagement pédagogique et les connaissances fondamentales qu'ils m'ont transmises tout au long de mon cursus universitaire.

Enfin, je remercie toutes les personnes qui, de près ou de loin, ont contribué à la réalisation de ce travail.

Dédicace

*Je dédie ce travail à **ma famille**, qui représente pour moi bien plus qu'un soutien : un refuge, une force et une raison de toujours avancer, même dans les moments les plus difficiles.*

*À ma chère **mère**, que Dieu lui fasse miséricorde. Son absence est une blessure silencieuse, mais son amour, ses valeurs et son souvenir continuent de m'accompagner chaque jour et de guider mes pas. Ce travail est aussi une prière pour elle et un témoignage de gratitude éternelle.*

*Particulièrement à **mon père**, dont le courage, la dignité et les sacrifices silencieux ont façonné mon parcours et m'ont appris la valeur du travail et de la persévérance. Ta présence dans ma vie est une bénédiction inestimable.*

*À mon frère **Mohamed**, pour sa loyauté, son soutien discret mais constant, et pour avoir toujours cru en moi, même lorsque le doute m'envahissait.*

*À ma sœur **Kawter**, pour son affection, sa bienveillance et surtout pour ses encouragements constants tout au long de mon parcours. Ta motivation, ton soutien dans les moments de travail difficiles et ta présence rassurante m'ont donné la force de continuer et de ne jamais abandonner.*

*Et surtout à ma petite et merveilleuse **Nour El Houda**, la lumière pure de ma vie, dont le sourire innocent et le regard rempli de joie ont le pouvoir d'effacer toutes les fatigues. Tu es mon bonheur, mon espoir et la plus belle raison de continuer à avancer. Ce travail t'est dédié avec tout mon amour.*

Liste des figures

Liste des tableaux

Liste des acronymes

BERT	**B**idirectional **E**ncoder **R**epresentations from **T**ransformers
DUIP	**D**ynamic **U**ser **I**ntent **P**rediction
FAISS	**F**acebook **AI** **S**imilarity **S**earch
LLMs	**L**arge **L**anguage **M**odel**s**
LSTM	**L**ong **S**hort-**T**erm **M**emory
NLP	**N**atural **L**anguage **P**rocessing
IDS	**I**ntrusion **D**etection **S**ystem
SIEM	**S**ecurity **I**nformation and **E**vent **M**anagement
SOC	**S**ecurity **O**perations **C**enter
GAN	**G**enerative **A**dversarial **N**etworks
GenAI	**Gen**erative **A**rtificial **I**ntelligence
AGI	**A**rtificial **G**eneral **I**ntelligence
XAI	e**X**plainable **A**rtificial **I**ntelligence
RAG	**R**etrieval-**A**ugmented **G**eneration

Introduction générale

Introduction générale

L'évolution rapide des technologies de l'information et de la communication a profondément transformé les systèmes informatiques et les réseaux. Cependant, cette transformation s'accompagne d'une augmentation significative des cybermenaces, devenues de plus en plus fréquentes, complexes et sophistiquées. Les organisations sont ainsi confrontées à des attaques capables de compromettre la confidentialité, l'intégrité et la disponibilité de leurs données.

Afin de faire face à ces menaces, plusieurs solutions de sécurité ont été développées, notamment les systèmes de détection d'intrusion (IDS), les pare-feux et les plateformes de gestion des informations et des événements de sécurité (SIEM). Ces outils permettent de surveiller les activités du réseau et de générer des alertes en cas de comportement suspect. Toutefois, ces alertes sont souvent nombreuses, techniques et difficiles à interpréter, ce qui complique leur exploitation par les analystes de sécurité.

Par ailleurs, les avancées récentes en intelligence artificielle, en particulier dans les domaines de l'apprentissage automatique et du traitement automatique du langage naturel, offrent de nouvelles opportunités pour améliorer l'analyse des données de sécurité. Les modèles de langage de grande taille permettent désormais de transformer des informations complexes en explications compréhensibles, facilitant ainsi l'interprétation des alertes.

Dans ce contexte, ce mémoire propose le développement d'un système intelligent, nommé **SecTalk**, visant à améliorer la détection et la compréhension des alertes de sécurité, tout en assistant les utilisateurs dans la prise de décision face aux incidents.

Contexte

La transformation numérique des organisations et la généralisation des systèmes connectés ont considérablement augmenté la surface d'attaque des infrastructures informatiques. Les entreprises et les institutions dépendent fortement de leurs systèmes d'information, ce qui les rend particulièrement vulnérables aux cyberattaques.

Pour renforcer leur sécurité, ces organisations déploient des outils tels que les IDS, les pare-feux et les SIEM, qui collectent et analysent en continu les données issues des réseaux et des systèmes. Cependant, ces solutions génèrent un volume très important d'alertes, rendant leur gestion complexe et exigeante pour les analystes de sécurité.

Dans les centres opérationnels de sécurité (SOC), les analystes doivent traiter ces alertes en temps réel, ce qui représente une charge de travail élevée et nécessite une expertise technique importante. Cette situation met en évidence le besoin de solutions capables de simplifier l'analyse des alertes et d'améliorer leur compréhension.

Motivation

L'analyse des alertes de sécurité constitue aujourd'hui un défi majeur pour les organisations, en raison du volume élevé d'informations à traiter et de la complexité des données générées. Les analystes de sécurité passent une grande partie de leur temps à interpréter des alertes techniques, ce qui peut entraîner une surcharge de travail, des erreurs d'analyse et un retard dans la prise de décision.

De plus, les utilisateurs non experts rencontrent des difficultés à comprendre les informations issues des systèmes de sécurité, ce qui limite leur capacité à réagir efficacement face aux incidents.

Dans ce contexte, il apparaît nécessaire de développer des solutions intelligentes capables non seulement d'automatiser l'analyse des alertes, mais aussi de rendre les résultats plus accessibles et compréhensibles. L'intégration des techniques d'intelligence artificielle et des modèles de langage offre une opportunité prometteuse pour répondre à ces besoins.

Problématique

Malgré les progrès réalisés dans le domaine de la cybersécurité, plusieurs limites persistent dans les systèmes actuels. Les alertes générées sont souvent techniques, difficiles à interpréter et nécessitent une intervention humaine importante pour être analysées et exploitées.

Par ailleurs, ces systèmes manquent généralement de capacités d'explication et d'interaction, ce qui complique la compréhension des incidents et la prise de décision, en particulier pour les utilisateurs non spécialisés.

Dans ce contexte, la problématique principale de ce mémoire peut être formulée comme suit :

Comment concevoir un système intelligent capable de détecter automatiquement les attaques, de transformer les alertes techniques en explications compréhensibles et d'assister les utilisateurs dans la gestion des incidents de sécurité ?

Objectif et structure du mémoire

L'objectif de ce travail est de concevoir un système intelligent capable d'améliorer l'analyse des alertes de sécurité, en combinant des techniques d'apprentissage automatique et d'intelligence artificielle générative. Le système proposé, nommé **SecTalk**, vise à détecter automatiquement les attaques à partir des données de sécurité, puis à générer des explications claires en langage naturel afin de faciliter la compréhension des alertes et d'assister les utilisateurs dans la prise de décision.

Dans ce cadre, ce mémoire s'intéresse à l'étude des approches existantes en cybersécurité, notamment celles basées sur la machine learning et le traitement automatique du langage

naturel, afin de répondre aux défis liés à la complexité des alertes et à la surcharge d'informations dans les systèmes de sécurité.

La structure de ce mémoire est organisée comme suit :

Le **premier chapitre** présente les concepts fondamentaux de l'intelligence artificielle générative, en mettant en évidence les principes des modèles de langage et leurs applications dans le traitement des données textuelles.

Le **deuxième chapitre** est consacré à l'état de l'art et à l'analyse des travaux existants dans le domaine de la cybersécurité, en particulier les systèmes de gestion des alertes, les approches d'analyse intelligente ainsi que les limites des solutions actuelles.

Le **troisième chapitre** décrit le système proposé **SecTalk**, dédié à la détection et à l'explication automatique des alertes de sécurité, en présentant son architecture générale et les différentes étapes de traitement, l'implémentation du système, les expérimentations réalisées ainsi que l'analyse et la discussion des résultats obtenus.

Chapitre I :

IA générative

I.1 Introduction

L'intelligence artificielle générative constitue aujourd'hui l'une des avancées majeures du domaine de l'intelligence artificielle, en permettant non seulement l'analyse des données mais également la production automatique de contenus nouveaux. Dans le contexte du traitement automatique du langage naturel, ces approches reposent sur des modèles de langage avancés capables de comprendre, représenter et générer du texte de manière cohérente.

Ce chapitre présente les fondements théoriques nécessaires à la compréhension de ces technologies. Il introduit d'abord la définition de l'intelligence artificielle générative, puis distingue les modèles d'IA génératifs et discriminants. Il aborde ensuite les notions de NLP, de modèles de langage, ainsi que l'architecture Transformer et les grands modèles de langage qui constituent la base des systèmes génératifs modernes. Enfin, il présente l'importance de l'IA générative dans les applications actuelles, ainsi que les limites et défis associés à son utilisation.

I.2 Définition d'IA générative :

L'intelligence artificielle (IA) a été définie comme « la capacité d'un système à interpréter correctement des données externes, à apprendre de ces données et à utiliser ces apprentissages pour atteindre des objectifs et accomplir des tâches spécifiques grâce à une adaptation flexible » [1].

I.2.1 Distinguer les modèles d'IA génératifs et discriminants

I.2.1.1 Modèles génératifs

Les modèles génératifs sont une catégorie de modèles d'intelligence artificielle capables d'apprendre la distribution des données afin de produire de nouvelles instances similaires à celles observées durant l'apprentissage. Contrairement aux modèles discriminatifs qui se limitent à prédire une étiquette, les modèles génératifs cherchent à capturer la structure sous-jacente des données et à reproduire leur comportement.

D'un point de vue probabiliste, ces modèles génératifs apprennent à modéliser la distribution de probabilité conjointe (P(X, Y)), où (X) représente les caractéristiques d'entrée et (Y) les étiquettes. En modélisant la distribution conjointe, ces modèles peuvent générer de nouveaux échantillons de données similaires aux données d'apprentissage. En substance, ils "comprennent" comment les données sont distribuées et peuvent créer de nouvelles instances qui leur ressemblent [22].

Parmi les principaux modèles génératifs, on peut citer :

- **Naive Bayes :** Suppose l'indépendance des caractéristiques pour modéliser la distribution des données.

- **Modèles de mélange gaussien (GMM) :** Modélise les données comme un mélange de distributions gaussiennes.
- **Autoencodeurs variationnels (VAE) :** Apprendre les représentations latentes pour générer de nouvelles données.
- **Réseaux adverbiaux génératifs (GAN) :** Utiliser un générateur et un discriminateur pour créer des données réalistes.
- **Les grands modèles de langage (LLM)**, tels que GPT, Llama ou Mistral, capables de générer du texte en langage naturel à partir d'un contexte donné.

Les modèles génératifs sont largement utilisés dans la génération de contenu, la création de données synthétiques, la détection d'anomalies, l'imputation de valeurs manquantes ainsi que les systèmes conversationnels intelligents.

I.2.1.2 Modèles discriminants

Les modèles discriminatifs, quant à eux, se concentrent sur la modélisation de la probabilité conditionnelle (P(Y|X)), qui prédit directement l'étiquette (Y) compte tenu des caractéristiques d'entrée (X). Ces modèles sont conçus pour trouver la limite de décision qui sépare le mieux les classes sans modéliser explicitement la distribution sous-jacente des données.

Voici quelques exemples de modèles discriminants :

- **Régression logistique :** Prévoit les probabilités pour la classification binaire ou multiclasse.
- **Machines à vecteurs de support (SVM) :** Trouve l'hyperplan optimal pour séparer les classes.
- **Arbres de décision et forêts aléatoires :** Utiliser des structures arborescentes pour la classification ou la régression.
- **Réseaux neuronaux (par exemple, CNN, RNN) :** Apprendre les limites de décision complexes pour diverses tâches.

Les modèles discriminatifs excellent dans les tâches où l'objectif est une prédiction ou une classification précise, comme la détection de spam ou la classification d'images.

Afin de mieux comprendre les différences entre les modèles génératifs et discriminatifs, le tableau suivant présente une comparaison de leurs principales caractéristiques, notamment leurs objectifs, leurs principes de fonctionnement, leurs applications et leurs avantages respectifs.

Critère	Modèles génératifs	Modèles discriminatifs
Objectif principal	Apprendre la distribution des données afin de générer de nouvelles instances ou de modéliser leur structure sous-jacente	Apprendre la relation entre les données d'entrée et les classes afin de prédire les étiquettes
Principe de fonctionnement	Modélisent la distribution P(X) ou la distribution conjointe P(X, Y)	Modélisent directement la probabilité conditionnelle P(Y|X) ou la frontière de décision
Capacité de génération	Peuvent produire de nouveaux contenus (texte, images, audio, données synthétiques)	Ne génèrent pas de nouvelles données ; ils effectuent uniquement des prédictions
Applications principales	Génération de texte, d'images, de code, création de données synthétiques, détection d'anomalies	Classification, régression, détection de spam, reconnaissance d'images, détection d'intrusion
Exemples de modèles	GPT, Llama, Gemini, GAN	Régression logistique, SVM, Random Forest, arbres de décision, CNN de classification
Avantages	Grande flexibilité, capacité de création de contenu, apprentissage de représentations riches	Haute précision, rapidité d'exécution, bonnes performances en classification
Limites	Entraînement coûteux, forte consommation de ressources, résultats parfois imprévisibles	Incapacité à générer du contenu, dépendance aux données annotées

Tableau I.1 : Comparaison entre modèles génératifs et discriminatifs

I.3 Mécanismes de génération de données dans l'IA générative

Les modèles d'IA génératifs créent de nouvelles données grâce à des processus sophistiqués qui impliquent l'apprentissage à partir de vastes ensembles de données et l'exploitation d'architectures avancées d'apprentissage automatique.

I.3.1 Apprendre à partir de données étendues

Les modèles d'IA génératifs sont équipés de vastes ensembles de données et de conceptions complexes qui leur permettent de créer un contenu nouveau et diversifié

Ces modèles sont aptes à modéliser des distributions de probabilité de grande dimension de données, telles que le langage ou les images, à partir de domaines spécifiques ou généraux [2]. Cette capacité à comprendre les modèles et les structures sous-jacents des données est cruciale pour générer un nouveau contenu réaliste.

I.3.2 Tirer parti des architectures avancées

L'émergence de modèles d'IA génératifs révolutionnaires a ouvert une nouvelle ère dans la synthèse et la manipulation de contenus numériques [2]. Ces modèles démontrent des capacités sans précédent en matière de synthèse d'images, d'audio, de texte et d'autres modalités de données.

Ils y parviennent en tirant parti des prouesses de l'apprentissage en profondeur et des architectures de transformateurs [1]. D'autres modèles innovants tels que les réseaux antagonistes génératifs (GAN) et les autoencodeurs variationnels contribuent également à leur capacité à capturer la complexité des données [2].

I.3.3 Transformation et synthèse du contenu

L'IA générative peut produire des formats multimédias réellement différents, tels que la vidéo, l'audio ou le texte, à partir de différents formats d'entrée [2].

En complétant les modèles génératifs par des techniques supplémentaires qui mappent l'espace latent de haute dimension des données à des représentations multimédias, il devient possible de transformer n'importe quel format d'entrée en une variété de formats de sortie. Cette polyvalence permet une conversion fluide entre les formats multimédia, ce qui rend les modèles génératifs inestimables dans de nombreuses applications.

Essentiellement, les modèles d'IA génératifs génèrent de nouvelles données en apprenant les propriétés statistiques complexes de leurs données d'entraînement à l'aide de réseaux neuronaux profonds et d'architectures spécialisées, puis en synthétisant de nouvelles sorties qui adhèrent à ces modèles appris, transformant souvent les entrées selon différentes modalités [2].

I.4 Les modèles génératifs en traitement du langage naturel (NLP)

I.4.1 Définition de NLP

Le NLP désigne les techniques qui permettent aux ordinateurs de comprendre, analyser ou produire du langage humain. Il sert à automatiser des tâches linguistiques pour faciliter la communication, l'accès à l'information et les interactions homme-machine.

Le Traitement Automatique des Langues est une branche de l'intelligence artificielle qui vise la compréhension automatique du langage humain par des ordinateurs. [3]

Il englobe aussi l'analyse du texte, la détection de similarités sémantiques et la traduction entre langues.

Le but principal du NLP est d'automatiser des tâches linguistiques pour rendre les textes et la parole plus exploitables par des machines et par des utilisateurs (extraction d'information, traduction, réponses automatiques, etc.) [4]

I.4.2 IA générative appliquée au NLP

L'émergence récente de l'intelligence artificielle générative a profondément transformé le domaine du NLP. Traditionnellement, les systèmes de NLP étaient principalement orientés vers l'analyse du texte, par exemple pour identifier le sentiment d'un message ou extraire des informations spécifiques. Cependant, les modèles génératifs ont introduit une nouvelle capacité : celle de produire automatiquement du langage naturel cohérent.

Ces modèles sont entraînés sur de très grandes quantités de données textuelles et apprennent les régularités statistiques du langage. Ils peuvent ensuite générer des phrases plausibles, répondre à des questions, reformuler des textes ou produire des explications détaillées.

Les architectures modernes basées sur les transformateurs permettent aux modèles d'apprendre des représentations profondes du langage, facilitant ainsi la génération de textes fluents et contextuellement pertinents [5].

Grâce à ces avancées, le NLP ne se limite plus à l'analyse automatique du texte, mais inclut désormais la production automatique de contenu linguistique, ouvrant la voie à de nombreuses applications innovantes.

I.4.3 Génération de texte en NLP

La génération de texte peut être définie comme le processus qui transforme une représentation abstraite (symbolique, statistique ou neuronale) en une séquence linguistique cohérente et grammaticalement correcte. L'enjeu principal est double :

- Assurer la correction linguistique (syntaxe, morphologie, lexique).
- Garantir la pertinence pragmatique et sémantique afin que le texte soit compréhensible et adapté au contexte d'utilisation [6].

I.4.3.1 Étapes du processus

La littérature distingue généralement deux étapes fondamentales :

- Planification du contenu : déterminer *quoi dire*, c'est-à-dire sélectionner les informations pertinentes à transmettre.
- Réalisation linguistique : décider *comment le dire*, en transformant les concepts choisis en structures syntaxiques et en unités lexicales adaptées [7].

I.4.3.2 Approches méthodologiques

- **Approches symboliques** : basées sur des règles linguistiques et des grammaires formelles.
- **Approches statistiques** : exploitant des modèles probabilistes tels que les n-grammes.
- **Approches neuronales** : utilisant des architectures de réseaux de neurones, notamment les modèles séquentiels (seq2seq) et les Transformers (GPT, BERT).
- **Approches hybrides** : combinant règles linguistiques et apprentissage automatique [8].

I.4.4 Applications du NLP génératif

Les techniques de NLP génératif sont aujourd'hui utilisées dans de nombreux domaines. Elles permettent notamment :

I.4.4.1 Assistants conversationnels

Les systèmes de dialogue (chatbots, assistants vocaux) exploitent le NLP pour comprendre les requêtes des utilisateurs et générer des réponses adaptées. Ils sont utilisés dans le support client, l'éducation et la santé, offrant une interaction fluide et personnalisée.

I.4.4.2 Traduction automatique

La traduction neuronale, fondée sur les modèles seq2seq et Transformers, a considérablement amélioré la qualité des traductions multilingues. Des outils comme DeepL ou Google Translate illustrent l'efficacité de ces modèles.

I.4.4.3 Résumé automatique de documents

Le NLP permet de condenser des textes volumineux en résumés pertinents, facilitant la veille documentaire et la recherche scientifique.

I.4.4.4 Analyse des sentiments et opinions

Les techniques de NLP sont mobilisées pour détecter les émotions et opinions exprimées dans les textes, notamment sur les réseaux sociaux. Elles sont utilisées en marketing, en politique et dans la gestion de la réputation des entreprises.

I.4.4.5 Génération de rapports automatisés

Dans des secteurs comme la finance, la météorologie ou la santé, le NLP est utilisé pour transformer des données structurées en textes compréhensibles, permettant une diffusion rapide et accessible de l'information [9] [10].

I.5 Les modèles de langage

I.5.1 Définition

Un modèle de langage est un système informatique conçu pour estimer la probabilité d'apparition d'une séquence de mots dans une langue donnée. Autrement dit, il apprend les régularités statistiques du langage afin de déterminer si une phrase est probable ou non et de générer du texte cohérent.

Un modèle de langage attribue une probabilité à une séquence de mots w1,w2,…,wn en modélisant la distribution [11]:

$$P(\omega_1, \omega_2, \dots, \omega_n)$$

Cette capacité permet aux modèles de langage d'être utilisés dans de nombreuses tâches du TALN, telles que la traduction automatique, la reconnaissance vocale, la génération de texte ou les systèmes conversationnels.

I.5.2 Types de modèles de langage

- **Modèles statistiques classiques** (ex. n-grammes) : basés sur des probabilités de séquences de mots.
- **Modèles neuronaux** (ex. RNN, LSTM) : utilisent des réseaux de neurones pour capturer des dépendances complexes.
- **Modèles à base de transformers** (ex. GPT, BERT, Llama) : s'appuient sur des mécanismes d'attention pour traiter des contextes longs et des relations à longue distance [12].

I.5.3 Principe de prédiction du mot suivant dans les modèles de langage

Le principe fondamental d'un modèle de langage consiste à estimer la probabilité du mot suivant à partir du contexte précédent.

Ainsi, au lieu de modéliser directement toute la phrase, le modèle décompose la probabilité globale en probabilités conditionnelles :

$$P(\omega_1, \dots, \omega_n) = \prod_{i=1}^{n} P(\omega_i | \omega_1, \dots, \omega_{i-1})$$

Dans cette formulation, chaque mot est prédit en fonction des mots précédents. Les modèles modernes apprennent ces probabilités en analysant de grandes quantités de texte et en identifiant les dépendances entre les mots, ce qui leur permet de générer des phrases grammaticalement correctes et sémantiquement plausibles.

La prédiction du mot suivant constitue le cœur des modèles probabilistes du langage et sert de base aux systèmes actuels de génération de texte [13].

I.5.4 Fonctionnement général des modèles de langage

I.5.4.1 Entraînement sur de grands corpus de données

- Les modèles de langage sont entraînés sur d'énormes ensembles de données textuelles, souvent appelés corpus. Ces corpus peuvent inclure des livres, des articles de journaux, des sites web, des conversations, du code source, et bien plus encore.
- L'objectif de cet entraînement est de permettre au modèle d'apprendre **les patterns** (schémas) et les structures du langage. Il apprend la grammaire, le vocabulaire, la syntaxe, et même des nuances sémantiques et contextuelles.
- Plus le corpus d'entraînement est grand et diversifié, plus le modèle sera performant et capable de généraliser à de nouveaux textes.

I.5.4.2 Modélisation probabiliste du langage

- Au cœur d'un modèle de langage se trouve la notion de **probabilité**. Le modèle calcule la probabilité qu'un certain mot ou une séquence de mots apparaisse dans un contexte donné.
- Par exemple, si vous tapez « Le ciel est… », un modèle de langage va calculer la probabilité des mots qui pourraient logiquement suivre « Le ciel est… ». Il pourrait déterminer que « bleu », « clair », « nuageux », « étoilé » sont des mots hautement probables, tandis que « banane » ou « voiture » sont extrêmement improbables.
- Il utilise les statistiques apprises pendant l'entraînement pour faire ces prédictions de probabilité.

I.5.4.2 Génération de texte et compréhension du contexte

- **Génération de texte** : grâce à sa capacité à prédire les mots suivants, un modèle de langage peut générer du texte. En commençant par une phrase ou un mot initial, il peut continuer à prédire le mot suivant, puis le suivant, et ainsi de suite, créant ainsi un texte potentiellement long et cohérent. C'est ainsi que les modèles de langage peuvent écrire des articles, des poèmes, répondre à des questions, etc.
- **Compréhension du langage (limitée)** : bien qu'on dise qu'ils « comprennent » le langage, il est important de noter que leur « compréhension » est différente de la compréhension humaine. Ils ne comprennent pas le sens profond ou l'intention derrière les mots de la même manière qu'un humain. Leur compréhension est basée sur des patterns statistiques et des relations entre les mots qu'ils ont appris. Cependant, cette « compréhension » statistique est suffisamment puissante pour de nombreuses applications [12].

I.5.5 Différence entre modèles n-grammes et modèles neuronaux

Caractéristique	Modèles n-grammes (Statistiques)	Modèles Neuronaux (LLM)
Représentation	Discrète : Basée sur des identifiants uniques et des fréquences de mots.	Continue : Utilise des plongements vectoriels (*embeddings*) dans un espace sémantique.
Gestion du contexte	Limitée : Fenêtre fixe et courte (souvent 2 à 5 mots).	Étendue : Mécanismes d'attention permettant de traiter des milliers de tokens.
Généralisation	Faible : Incapable de prédire une séquence jamais vue précisément dans les données.	Élevée : Prédit par similarité sémantique même si la séquence exacte est inédite.
Paramètres	Statique : Table de probabilités calculée une seule fois par comptage.	Dynamique : Milliards de poids ajustés par rétropropagation du gradient.
Complexité	Explosion combinatoire : La taille de la table croît exponentiellement avec n.	Scalabilité : Gère de vastes vocabulaires et contextes grâce à la parallélisation (Transformers).
Mémoire	Lissage (Smoothing) : Nécessaire pour éviter les probabilités nulles.	Régularisation : Utilise le *dropout* et la normalisation pour stabiliser l'apprentissage.
Cas d'usage	Autocomplétion simple, correcteurs orthographiques légers.	ChatGPT, traduction complexe, génération de code, raisonnement.

Tableau I.2 : Différence entre modèles n-grammes et modèles neuronaux [14] [15] [16]

Ce tableau montre que les modèles neuronaux offrent une meilleure capacité de généralisation et de représentation du contexte par rapport aux modèles n-grammes traditionnels, qui restent limités par la taille du contexte et la sparsité des données.

I.6 L'architecture Transformer

I.6.1 Définition de l'architecture du transformateur

L'architecture Transformer, initialement développée pour des tâches de séquence à séquence telles que la traduction automatique, est fondamentalement basée sur un framework encodeur-décodeur. Cette conception est largement utilisée dans le traitement du langage naturel (NLP) pour convertir une séquence de mots d'entrée d'une langue à une autre [17]. L'architecture comprend deux composants principaux :

- **Encodeur** : Ce composant est chargé de transformer une séquence de jetons d'entrée en une série de vecteurs d'intégration, souvent appelés état ou contexte caché. Il traite l'entrée pour créer des représentations numériques riches adaptées à des tâches telles que la classification de texte ou la reconnaissance d'entités nommées [17].

- **Décodeur** : Le décodeur utilise l'état caché généré par l'encodeur pour produire de manière itérative une séquence de sortie de jetons, un jeton à la fois. Ce processus itératif se poursuit jusqu'à ce qu'un jeton spécial de fin de séquence (EOS) soit atteint ou qu'une longueur maximale soit atteinte [17].

L'encodeur et le décodeur sont eux-mêmes construits à partir de plusieurs éléments constitutifs, ce qui permet une conception modulaire et évolutive.

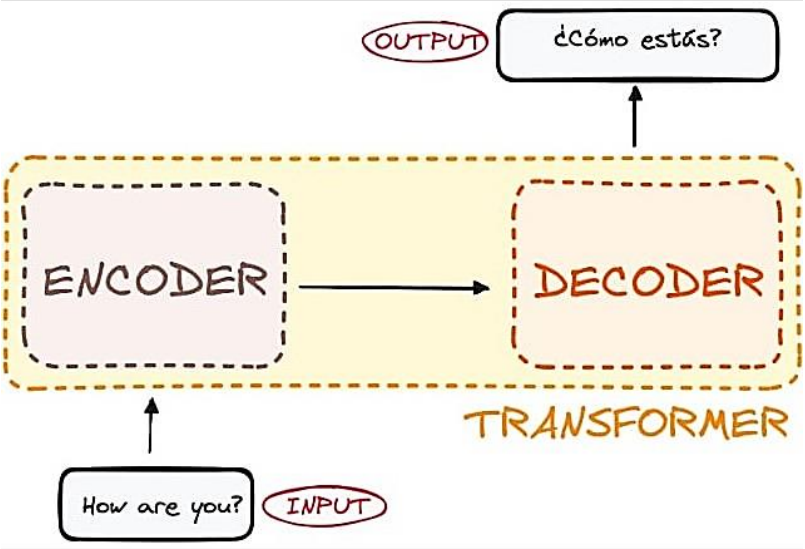

Figure I.1 : Structure globale du codeur-décodeur [23].

I.6.2 Architecture et fonction du transformateur

L'architecture Transformer traite le texte d'entrée en plusieurs étapes pour générer des représentations et des prédictions contextualisées :

- **Tokénisation et intégration** : le texte saisi est d'abord tokenisé puis converti en jetons intégrés. Pour tenir compte de la nature séquentielle du texte, ces intégrations de jetons sont combinées à des intégrations positionnelles, car le mécanisme d'attention lui-même n'est pas intrinsèquement conscient de la position des jetons [17]. Cela garantit que le modèle peut comprendre l'ordre des mots.
- **Structure de l'encodeur** : L'encodeur se compose de plusieurs couches de codeur empilées, ou « blocs », chacune recevant une séquence d'intégrations. Chaque couche d'encodeur comprend une couche d'auto-attention à plusieurs têtes et une couche de feedback entièrement connectée. Le rôle principal de cette pile est d'affiner l'intégration des entrées pour intégrer des informations contextuelles, permettant aux mots d'adapter leur signification en fonction de leur environnement (par exemple, « pomme » devenant « semblable à une entreprise » lorsqu'il est proche de « keynote » ou « téléphone ») **[**17**]**.
- **Structure du décodeur** : Le décodeur comporte également une pile de couches de décodeur. Sa sortie est introduite dans chaque couche de décodeur suivante, qui prédit

ensuite le jeton suivant le plus probable. Ce jeton prédit est renvoyé dans le décodeur, jusqu'à ce que la séquence soit terminée.

- **Variantes architecturales** : Alors que le Transformer original est un modèle d'encodeur et de décodeur, des architectures autonomes à encodeur et à décodeur uniquement ont vu le jour. Les modèles utilisant uniquement des encodeurs (par exemple, BERT, Roberta) sont adaptés aux tâches nécessitant un contexte bidirectionnel, comme la classification de texte. Les modèles utilisant uniquement des décodeurs (par exemple, la famille GPT) excellent dans les tâches génératives, prédisant les jetons suivants en fonction du contexte de gauche. Les modèles encodeurs-décodeurs (par exemple, BART, T5) sont utilisés pour des mappages séquence-séquence complexes tels que la traduction automatique et la synthèse **[**17**]**.

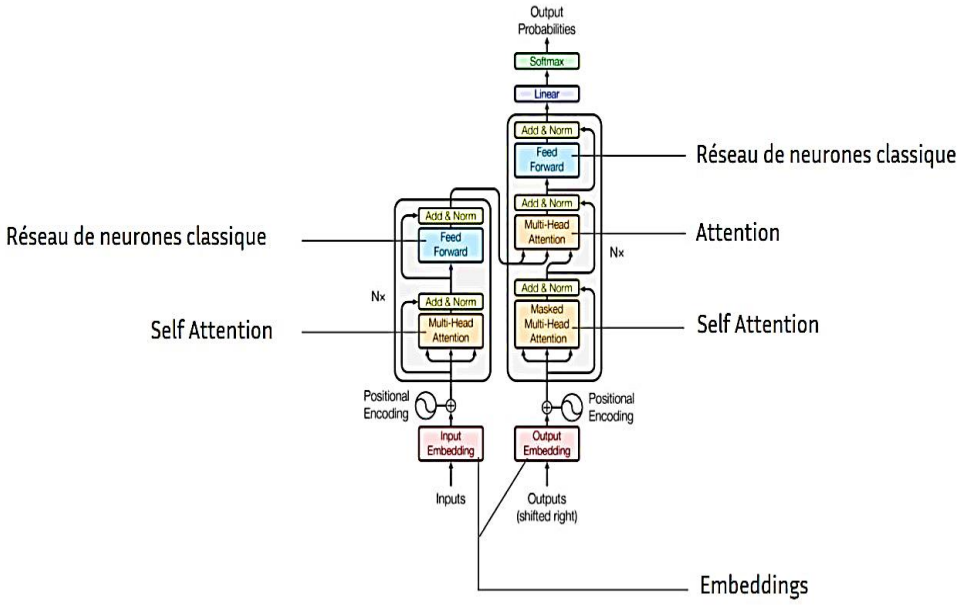

Figure I.2 : L'architecture d'un Transformer [23].

I.6.3 Le mécanisme de l'attention

L'attention est un mécanisme central des réseaux de transformateurs, qui permet au modèle d'attribuer divers degrés d'importance, ou « poids », à différents éléments d'une séquence d'entrée [14]. Pour le texte, ces éléments sont des intégrations de jetons, où chaque jeton est mappé à un vecteur de dimension fixe [17].

- **Auto-attention** : Le « soi » dans l'auto-attention indique que ces poids sont calculés pour tous les états cachés d'un même ensemble, tels que tous les états cachés de l'encodeur. Cela contraste avec l'attention portée aux modèles récurrents, qui calculent la pertinence entre les états cachés du codeur et du décodeur à des étapes de décodage spécifiques.
- **Intégrations contextualisées** : L'idée fondamentale de l'attention personnelle est de calculer une moyenne pondérée de toutes les intégrations de jetons dans une séquence,

plutôt que d'utiliser une intégration fixe pour chaque jeton. Ce processus génère des « intégrations contextualisées », dans lesquelles la représentation d'un jeton est informée par le contexte qui l'entoure. Par exemple, la signification de « mouches » peut être clarifiée en attribuant des poids plus élevés au « temps » et à la « flèche » dans la phrase « le temps passe comme une flèche ».

- **Scaled Dot-Product Attention** : Il s'agit de la méthode la plus courante pour mettre en œuvre l'attention personnelle. Cela implique de projeter chaque jeton intégré dans trois vecteurs : requête, clé et valeur. Les scores d'attention sont calculés en prenant le produit scalaire de la requête et des vecteurs clés, puis mis à l'échelle et normalisés à l'aide d'une fonction softmax. Enfin, ces poids d'attention sont multipliés par les vecteurs de valeur pour mettre à jour les intégrations de jetons.

I.6.4 Fonctionnement du mécanisme d'auto-attention

Le processus d'attention personnelle implique plusieurs étapes clés pour déterminer les relations entre les jetons et mettre à jour leurs représentations [17] :

- **Projections de requête, de clé et de valeur** : l'intégration de chaque jeton est projetée dans trois vecteurs distincts : une requête (Q), une clé (K) et une valeur (V). Ces projections permettent à la couche d'auto-attention d'apprendre différents aspects sémantiques de la séquence.
- **Calcul du score d'attention** : La similitude entre les vecteurs de requête et les vecteurs clés est calculée, généralement à l'aide d'un produit scalaire. Ces scores indiquent dans quelle mesure un jeton de requête doit accorder à d'autres jetons clés. Les scores sont ensuite redimensionnés pour éviter que de grands nombres ne déstabilisent l'entraînement et normalisés à l'aide d'une fonction softmax pour produire des poids d'attention dont la somme est égale à un.
- **Somme pondérée des valeurs** : Les poids d'attention normalisés sont ensuite multipliés par leurs vecteurs de valeurs correspondants et additionnés. Cette somme pondérée constitue la nouvelle représentation contextualisée de chaque jeton.
- **Attention multidirecte** : Pour capturer diverses relations, le Transformer utilise plusieurs « têtes d'attention ». Chaque tête apprend différentes projections linéaires pour Q, K et V, ce qui permet au modèle de se concentrer simultanément sur divers aspects de la similitude (par exemple, les interactions sujet-verbe, les adjectifs). Les sorties de ces têtes sont ensuite concaténées et transmises à travers une couche linéaire finale.
- **Intégrations positionnelles** : Pour remédier au manque inhérent de conscience positionnelle dans l'attention personnelle, des intégrations positionnelles sont ajoutées aux incrustations de jetons. Ces modèles dépendant de la position permettent au modèle d'intégrer des informations séquentielles.

I.6.5 Avantages des Transformers par rapport aux RNN

Les transformateurs offrent des avantages significatifs par rapport aux réseaux neuronaux récurrents (RNN) traditionnels, notamment en ce qui concerne leur capacité à gérer les dépendances à longue distance et à tirer parti de la parallélisation [17] :

- **Parallélisation** : Contrairement aux RNN, qui traitent les séquences de manière séquentielle, le mécanisme d'auto-attention de Transformers permet de calculer en parallèle les relations entre tous les jetons d'une séquence. Cela accélère considérablement la formation, en particulier sur le matériel moderne comme les GPU.
- **Dépendances à longue durée** : l'attention personnelle calcule directement les relations entre deux jetons d'une séquence, quelle que soit leur distance. Cela contraste avec les RNN, où les informations provenant de jetons distants peuvent diminuer sur de nombreux intervalles de temps, ce qui rend plus difficile la capture des dépendances à long terme.
- **Efficacité informatique** : alors que le mécanisme naïf d'auto-attention évolue de manière quadratique en fonction de la longueur de la séquence (O(n2)$O(n2)$), les recherches ont cherché à le rendre plus efficace grâce à des techniques telles que l'attention clairsemée ou l'attention linéarisée, qui peuvent réduire la complexité à O(n)$O(n)$ ou O(n logn)$O(n\ logn)$. Cela permet aux Transformers de traiter des séquences beaucoup plus longues que les modèles traditionnels.
- **Évolutivité** : La conception modulaire de Transformers, avec des couches d'encodeur et de décodeur empilées, permet une mise à l'échelle facile en termes de taille de modèle et de taille de jeu de données. Cela a conduit au développement de modèles extrêmement grands et puissants qui atteignent des performances de pointe pour diverses tâches de PNL.

En conclusion, l'architecture Transformer, avec son mécanisme d'attention innovant et sa structure modulaire encodeur-décodeur, a révolutionné le NLP en permettant un traitement parallèle efficace, une gestion supérieure des dépendances à longue portée et une évolutivité remarquable, ce qui a permis de créer des modèles très efficaces pour un large éventail de tâches linguistiques.

I.7 Les grands modèles de langage LLM

I.7.1 Définition des grands modèles de langage (LLM)

Les grands modèles de langage (LLM) sont des modèles d'intelligence artificielle avancés, généralement basés sur l'architecture Transformer, conçus pour traiter et générer du texte semblable à celui d'un humain. Ils se caractérisent par leur grand nombre de paramètres, qui se chiffrent souvent en milliards, voire en milliers de milliards, et sont entraînés sur d'énormes ensembles de données de texte et de code. Cette formation approfondie leur permet de comprendre le contexte, de générer des réponses cohérentes et pertinentes et d'effectuer un large éventail de tâches de traitement du langage naturel (NLP).

I.7.2 Principe de fonctionnement des LLM

La plupart des LLM sont construits à l'aide d'une architecture de type transformeur. Ils fonctionnent en divisant le texte d'entrée en jetons (unités de sous-mots), en intégrant ces jetons dans des vecteurs numériques et en utilisant des mécanismes d'attention pour comprendre les relations dans l'ensemble de l'entrée. Ils prédisent ensuite le jeton suivant dans une séquence afin de générer des sorties cohérentes [18].

I.7.3 Préformation des LLM

La pré-formation est une phase critique pour les LLM, où les modèles apprennent la compréhension générale du langage à partir de grands corpus de textes non étiquetés. Ce processus implique souvent des tâches d'apprentissage autosupervisées :

- Modélisation du langage masqué (MLM) : dans cette tâche, une partie des jetons d'entrée est masquée et le modèle est entraîné pour prédire les jetons masqués d'origine en fonction de leur contexte. Cela force le modèle à apprendre les relations bidirectionnelles entre les mots [17]
- Prévision du prochain jeton (modélisation autorégressive) : des modèles tels que la famille GPT sont pré-entraînés pour prédire le mot suivant d'une séquence en fonction des mots précédents. Cette attention causale ou autorégressive se concentre uniquement sur le contexte gauche. Cet objectif est particulièrement efficace pour les tâches génératives.
- Lois d'évolution des LLM : L'efficacité des grands modèles de langage dépend étroitement de trois facteurs principaux : la puissance de calcul disponible, la taille des ensembles de données et la dimension du modèle. Les études récentes montrent que la perte sur les données de test suit une relation de type loi de puissance par rapport à ces paramètres, ce qui indique que l'augmentation simultanée de ces trois éléments constitue une voie prometteuse pour améliorer les performances des modèles. Cette approche a conduit à la conception de modèles tels que GPT-3, dont le nombre de paramètres a été multiplié par 100 par rapport à GPT-2, atteignant entre 175 milliards de paramètres [17].

I.7.4 Exemples de grands modèles de langage

Plusieurs modèles illustrent cette évolution du NLP :

- GPT (Generative Pre-trained Transformer) – génération de texte
- BERT – compréhension du langage et classification
- T5 – modèle unifié pour différentes tâches linguistiques
- modèles récents multimodaux combinant texte et images

Ces systèmes démontrent que l'augmentation de la taille des modèles et des données d'entraînement améliore fortement leurs performances sur de nombreuses tâches linguistiques.

I.7.5 Importance des LLM dans le NLP moderne

Les grands modèles de langage (LLM) jouent aujourd'hui un rôle central dans le traitement automatique du langage naturel (NLP). Leur capacité à apprendre des représentations profondes et contextuelles du langage leur permet de comprendre, générer et manipuler le texte avec une précision auparavant inatteignable. Contrairement aux modèles traditionnels, qui se limitaient à des règles ou à des séquences statistiques comme les n-grammes, les LLM peuvent capturer des dépendances à long terme et intégrer des informations sémantiques complexes dans leurs prédictions.

Cette puissance se traduit par des applications concrètes et variées : traduction automatique, résumé de texte, question-réponse, génération de dialogue et extraction d'information. Les LLM ont également permis de réduire la dépendance à des systèmes spécifiques à une tâche, ouvrant la voie à des modèles polyvalents et adaptables capables de s'appliquer à plusieurs tâches de NLP avec un ajustement minimal [19].

I.8 IA générative et cybersécurité

L'IA générative a transformé la cybersécurité en renforçant détection des menaces et réponse aux incidents En analysant de vastes quantités de données en temps réel, les organisations peuvent identifier et atténuer les menaces émergentes plus rapidement et avec une plus grande précision grâce aux techniques d'apprentissage automatique et d'apprentissage profond [20].

Les outils basés sur l'IA, tels que les plateformes d'automatisation et les systèmes de détection d'intrusion (IDS), utilisent l'IA générative pour prédire et prévenir les failles de sécurité. De nombreux cas d'usage de l'IA contribuent aujourd'hui à relever les défis de la cybersécurité. Ces cas d'usage vont de l'automatisation des processus de cybersécurité à l'amélioration de la prise de décision dans des environnements de sécurité complexes. La capacité d'adaptation et d'apprentissage de l'IA en fait un outil essentiel pour les équipes de sécurité, leur permettant d'opérer plus efficacement et de faire face à l'évolution constante des cybermenaces.

Bien que l'IA générative recèle un immense potentiel pour améliorer la sécurité, elle soulève également de nouveaux défis. L'un des risques majeurs réside dans la menace d'attaques adverses, où des acteurs malveillants manipulent les modèles d'IA pour échapper à la détection ou perturber les systèmes de sécurité. Ce type d'utilisation malveillante de l'IA peut compromettre la fiabilité des solutions de sécurité basées sur l'IA et rendre les organisations vulnérables aux cyberattaques sophistiquées [20].

Par ailleurs, l'utilisation de modèles génératifs permet de produire automatiquement des explications claires et accessibles des alertes de sécurité. Cette capacité facilite la communication entre les systèmes de détection et les équipes humaines, rendant les menaces plus lisibles et accélérant la prise de décision face aux incidents.

I.9 Limites et défis de l'IA générative

Limites et défis de l'intelligence artificielle générative

L'intelligence artificielle générative (GenAI), malgré ses capacités impressionnantes à produire du texte, des images, de la musique et du code plausibles, est confrontée à plusieurs limites et défis importants dans les domaines techniques, éthiques et opérationnels. Ces problèmes soulèvent des questions quant à sa fiabilité à long terme, à son efficacité et à son adéquation aux applications critiques.

I.9.1 Limites techniques

- **Hallucinations et incohérences :** Les algorithmes GenAI produisent fréquemment des résultats plausibles mais manifestement faux, un phénomène connu sous le nom d'« hallucinations ». Ils peuvent également générer des résultats incompatibles avec la logique, les mathématiques ou la compréhension du monde réel. Cela provient de leur conception, qui peut confondre le « bruit » des données d'entraînement avec des modèles significatifs, ce qui entraîne des résultats étranges lorsqu'il est appliqué à de nouvelles données **[**21**]**.
- **Biais problématiques et manque d'explicabilité :** Les modèles GenAI ont tendance à régurgiter les biais problématiques présents dans leurs données d'entraînement et ne peuvent pas fournir d'explication fiable de la façon dont leurs résultats ont été dérivés. Ce manque de transparence, où « l'IA pure troque la transparence du « raisonnement » contre le pouvoir », rend difficile la confiance dans les résultats, en particulier dans les contextes nécessitant une supervision et une confiance humaines **[**21**]**.
- **Limites de l'approche axée sur les données :** La génération actuelle de GenAI repose largement sur les relations statistiques au sein des données d'entraînement, créant ainsi une carte multidimensionnelle de « jetons ». Bien que cela permette de générer de nouvelles combinaisons cohérentes, cela ne fait pratiquement aucune hypothèse quant à la forme de ces relations, ce qui contribue aux problèmes susmentionnés. L'augmentation des données et de la puissance de traitement ou l'ajout de processus de réglage externes sont proposées par les « optimistes GENAI » pour répondre aux problèmes de précision et de traçabilité, mais les « pessimistes GENAI » suggèrent que les solutions mettent du temps à émerger, ce qui indique que l'approche atteint peut-être ses limites [21**]**.

I.9.2 Défis opérationnels et en matière de ressources

- **Consommation d'énergie et de ressources élevée :** La formation et l'exploitation de ces applications GenAI basées sur la « force brute » et pilotées par les données nécessitent une infrastructure importante, ce qui entraîne une consommation d'énergie et de ressources très élevée. Cela exerce une pression sur des approvisionnements déjà limités, mettant en évidence un défi opérationnel important.

- **Faiblesses de conception pour les utilisations critiques :** Les faiblesses de conception inhérentes, telles que la sensibilité au bruit et le manque de transparence, limitent fondamentalement l'adéquation de GenAI aux applications critiques. La simple mise à l'échelle de GenAI ne résout pas ces problèmes et exacerbe au contraire l'utilisation des ressources **[**21**]**.

I.9.3 Défis éthiques et politiques

- **Fiabilité pour une IA digne de confiance :** Les problèmes de fiabilité susmentionnés remettent en question la capacité de GenAI à fournir une « IA fiable et centrée sur l'humain », essentielle à la croissance économique et au respect des droits et principes fondamentaux **[**21**]**.
- **Mettre l'accent sur le GenAI par rapport aux alternatives :** Les initiatives politiques actuelles, telles que l'accent mis par la Commission européenne sur la mise en œuvre de l'IA GenAI plutôt que sur l'exploration d'approches alternatives ou complémentaires en matière d'IA, peuvent avoir une incidence négative sur la recherche de solutions plus efficientes et efficaces. Un investissement important est investi dans la course géopolitiquement stratégique à l'intelligence générale artificielle (AGI), ce qui soulève des questions quant aux limites de l'IA GenAI et à la nécessité d'approches alternatives [21].

I.10 Conclusion

Ce chapitre a exposé les fondements théoriques de l'intelligence artificielle générative et son articulation avec le traitement automatique du langage naturel. Après avoir défini l'IA générative et distingué ses approches des modèles discriminatifs, nous avons montré comment elle vise à modéliser la structure des données pour produire de nouveaux contenus. L'évolution des modèles, des méthodes statistiques aux architectures neuronales modernes, a conduit à l'émergence des Transformers, désormais au cœur des grands modèles de langage. Nous avons également mis en évidence leur rôle central dans les systèmes contemporains de NLP, qu'il s'agisse de la génération de texte, de l'assistance intelligente ou de l'explication automatique d'informations complexes.

Enfin, les principaux défis liés à ces modèles hallucinations, biais, manque d'explicabilité et coûts computationnels ont été discutés. L'ensemble de ces éléments constitue le cadre conceptuel nécessaire pour aborder le chapitre suivant, consacré à l'état de l'art des travaux sur l'explication automatique des alertes de sécurité, domaine dans lequel s'inscrit le projet de ce mémoire.

Chapitre II :

État de l’art et analyse des travaux existants

II.1 Introduction

Ce chapitre a pour objectif de présenter une étude approfondie des travaux existants relatifs à l'analyse automatique des alertes de sécurité et à l'utilisation de l'intelligence artificielle dans le domaine de la cybersécurité.

Face à l'augmentation continue des cyberattaques et à la complexité croissante des infrastructures informatiques, les systèmes de sécurité, tels que les systèmes de détection d'intrusion, les pare-feux et les plateformes de gestion des événements (SIEM), génèrent aujourd'hui un volume considérable d'alertes. Cependant, ces alertes restent souvent techniques, difficiles à interpréter et nécessitent une expertise avancée pour être exploitées efficacement.

Dans ce contexte, plusieurs approches ont été proposées pour améliorer l'analyse des alertes, notamment les méthodes basées sur des règles, les techniques d'apprentissage automatique ainsi que les approches récentes fondées sur le traitement automatique du langage naturel et les grands modèles de langage. Ces solutions visent à automatiser la détection et l'interprétation des événements de sécurité, à réduire la surcharge d'alertes et à faciliter la prise de décision.

Ainsi, ce chapitre présente d'abord les systèmes classiques de gestion des alertes, puis les différentes approches intelligentes proposées dans la littérature. Une analyse comparative des solutions existantes est ensuite réalisée afin d'identifier leurs limites et de mettre en évidence le besoin de systèmes plus intelligents, capables non seulement de détecter les menaces, mais aussi de fournir des explications compréhensibles et d'assister les utilisateurs dans leur prise de décision. Cette analyse permettra enfin de positionner clairement la contribution du projet proposé dans ce mémoire.

II.2 Les systèmes classiques de gestion d'alertes de sécurité

II.2.1 IDS traditionnels

Les systèmes de détection d'intrusion (IDS) sont un composant essentiel de la cybersécurité, conçus pour identifier et alerter en cas de cyberattaques potentielles en surveillant les événements de sécurité. Ces systèmes peuvent fonctionner en utilisant soit la détection des abus, qui compare les données surveillées à des signatures d'attaques connues, soit la détection des anomalies, qui émet des alertes en fonction des écarts par rapport à une base de référence normale [24].

Au fil du temps, les IDS sont devenus de plus en plus sophistiqués et surveillent un large éventail de sources d'événements de sécurité hétérogènes, notamment le trafic réseau, les journaux du système hôte, les processus, les fichiers et des protocoles spécifiques.

L'évolution des IDS vise à fournir des alertes précoces, à minimiser les dommages et à améliorer la connaissance globale de la situation en corrélant les événements de sécurité provenant de différentes sources [24].

Ces solutions permettent de signaler rapidement des incidents potentiels, mais elles produisent souvent des alertes techniques difficiles à interpréter pour les analystes.

II.2.2 Les systèmes SIEM

Les systèmes de gestion des informations et des événements de sécurité (SIEM) représentent une approche globale de la cybersécurité, distincte des systèmes de détection d'intrusion (IDS) traditionnels. Ces systèmes sont conçus pour fournir une vision globale de la sécurité informatique d'une organisation en agrégeant et en corrélant les événements de sécurité provenant de diverses sources hétérogènes, telles que les réseaux, les appareils des utilisateurs finaux, les serveurs, les pare-feux et les systèmes de prévention des intrusions.

L'une des principales fonctions des SIEM est de surveiller en permanence les incidents, de corréler les événements et d'émettre des notifications d'alerte, améliorant ainsi la connaissance de la situation [24].

Cependant, même si les SIEM améliorent la détection des incidents, ils restent fortement dépendants de règles prédéfinies et génèrent souvent des messages techniques complexes.

II.2.3 Limites des alertes techniques

Les alertes produites par Les systèmes de détection d'intrusion (IDS) et les solutions de gestion des événements et informations de sécurité (SIEM) sont généralement formulées sous forme technique, incluant des identifiants d'événements, des signatures, des ports réseau ou des adresses IP.

Cette représentation, bien qu'utile pour des experts en sécurité, peut être difficile à comprendre pour des utilisateurs non spécialisés ou pour des décideurs.

De plus, ces alertes ne fournissent pas toujours d'explication claire sur la gravité réelle de la menace, son origine ou les actions à entreprendre, ce qui complique l'analyse des incidents.

II.2.4 Le problème de surcharge d'alertes

Les systèmes de surveillance de la cybersécurité, tels que les systèmes de détection d'intrusion (IDS), génèrent fréquemment un volume énorme d'alertes techniques, ce qui entraîne un problème important de surcharge d'alertes pour le personnel de sécurité informatique [24].

Ce problème est aggravé par le fait qu'un seul IDS peut déclencher de nombreuses alertes, et cette complexité augmente lorsqu'il s'agit de sources hétérogènes ou de plusieurs IDS. La quantité d'alertes entraîne souvent une surabondance d'informations ambiguës ou de fausses

alarmes, ce qui complique la gestion efficace pour les administrateurs de sécurité. Par exemple, les IDS peuvent générer régulièrement des milliers d'alertes par jour, dont jusqu'à 99 % sont des faux positifs, ce qui peut désensibiliser les analystes de sécurité au flux constant d'alarmes inutiles. Ce volume impressionnant empêche le personnel de discerner les menaces réelles à partir du bruit de fond, ce qui a un impact sur l'efficacité globale des efforts de surveillance de la sécurité [24].

Ce phénomène est largement documenté dans les études sur les centres SOC, où la priorisation et l'explication des alertes constituent des besoins essentiels pour améliorer la gestion des incidents [25].

II.3 Approches d'analyse intelligente des alertes

L'analyse des alertes de sécurité est devenue un enjeu essentiel face à la complexité et au volume croissant des données générées par les systèmes tels que les IDS et les pare-feu. Les approches intelligentes permettent d'améliorer l'interprétation des alertes, de réduire les faux positifs et d'identifier efficacement les menaces.

Cette section présente les principales méthodes d'analyse intelligente des alertes, notamment les approches basées sur des règles, l'apprentissage automatique (classification, détection d'anomalies, clustering) ainsi que le transfer learning. Elle se termine par une discussion des limites de ces approches.

II.3.1 Approches basées sur des règles

Les approches d'analyse des alertes de cybersécurité s'appuient souvent sur des règles et des bases de connaissances prédéfinies pour interpréter et corréler les incidents de sécurité. Ces méthodes sont fondamentales dans les systèmes conçus pour traiter les alertes provenant de divers systèmes de détection d'intrusion (IDS) afin de réduire les faux positifs, d'identifier des modèles d'attaque complexes et de déterminer les causes profondes [26].

Les systèmes basés sur des règles, en particulier ceux classés comme des algorithmes basés sur la connaissance, fonctionnent sur la base de définitions ou de scénarios d'attaque définis. Par exemple, les algorithmes « Prérequisites/Conséquences » enchaînent les incidents en utilisant des combinaisons de conjonctions et de disjonctions, créant ainsi un réseau d'attaques possibles sur la base de conditions et de résultats prédéfinis. De même, les algorithmes « Scénario » détectent les attaques en plusieurs étapes en comparant les alertes de bas niveau à des étapes d'intrusion prédéfinies stockées dans une base de connaissances, en corrélant des séquences d'alertes liées à une attaque spécifique [26].

Ces approches soulignent l'importance d'une compréhension structurée des comportements d'attaque et des configurations des systèmes pour analyser efficacement les menaces de sécurité et y répondre [26].

II.3.1.1 Corrélation d'alertes

Les systèmes de corrélation d'alertes basés sur des règles constituent une approche fondamentale de la cybersécurité pour analyser et relier les alertes de sécurité. Ces systèmes partent du principe que les attaques sophistiquées comportent souvent plusieurs phases, chacune comportant des prérequis et des conséquences spécifiques. En utilisant des règles prédéfinies stockées dans une base de connaissances, ces systèmes peuvent agréger les alertes associées, reconstruire des scénarios d'attaque et identifier des modèles d'attaque de haut niveau. Ce processus permet de réduire le volume impressionnant d'alertes brutes générées par les systèmes de détection d'intrusion (IDS) et d'augmenter la pertinence et le niveau d'abstraction des rapports de sécurité. L'une des limites de cette approche est toutefois l'effort important requis pour définir un grand nombre de règles et la difficulté inhérente à la corrélation de modèles d'attaque nouveaux ou inédits. Ces systèmes contribuent également à réduire les faux positifs en se concentrant sur des événements corrélés qui forment un récit d'attaque cohérent, plutôt que sur des alertes isolées [27].

II.3.1.2 Systèmes experts

Les systèmes experts, en particulier ceux qui utilisent une approche basée sur des règles, sont utilisés en cybersécurité pour la corrélation des alertes, afin d'analyser et de relier les évènements de sécurité. Ces systèmes s'appuient sur une base de connaissances contenant des règles prédéfinies, qui sont essentiellement issues des connaissances d'experts, pour identifier les relations entre les différentes alertes. Les règles spécifient les préconditions et les conséquences des différentes étapes de l'attaque, permettant au système d'agréger les alertes associées et de reconstituer les scénarios d'attaque cohérents. Cette approche permet de résumer les alertes brutes en modèles d'attaque plus significatifs et de réduire le volume global des alertes, bien qu'elle demande des efforts considérables pour définir des règles et lutter contre de nouveaux modèles d'attaque [27].

II.3.2 Approches basées sur l'apprentissage automatique

Face aux limites des systèmes déterministes, les chercheurs ont introduit des techniques d'apprentissage automatique afin d'automatiser l'analyse des alertes et d'améliorer la détection des comportements malveillants.

Ces approches permettent de modéliser les comportements normaux et anormaux à partir des données historiques.

II.3.2.1 Classification supervisée

L'apprentissage supervisé est une approche fondamentale des systèmes de détection d'intrusion (IDS), où les modèles sont entraînés à l'aide de données étiquetées pour différencier le comportement normal du réseau des activités malveillantes. Cette méthode consiste à fournir au modèle des exemples historiques qui sont explicitement étiquetés comme un type d'attaque spécifique ou un trafic bénin **[**28**]**.

En s'appuyant sur ces exemples étiquetés, des algorithmes supervisés, tels que les arbres de décision, les machines à vecteurs de support, les forêts aléatoires, les K-voisins les plus proches et Naïve Bayes, développent la capacité de reconnaître des modèles d'attaque connus. Bien que très efficace pour identifier les menaces rencontrées précédemment, le principal défi réside dans la difficulté d'obtenir des données étiquetées complètes dans des scénarios réels, en particulier compte tenu de l'émergence continue de types d'attaques nouveaux et inconnus **[**28**]**.

II.3.2.2 Détection d'anomalies

La détection d'anomalies repose sur l'identification de comportements déviant du fonctionnement normal d'un système ou d'un réseau, en s'appuyant sur un modèle de référence appris à partir des données observées. Contrairement aux approches supervisées, les systèmes de détection d'intrusion (IDS) basés sur les anomalies ne nécessitent pas de données préalablement étiquetées, ce qui leur permet de découvrir automatiquement des structures ou des schémas inhabituels. Cette capacité rend ces méthodes particulièrement adaptées à la détection d'attaques inconnues, notamment les attaques de type « jour zéro ». Des techniques telles que le clustering K-means, les autoencodeurs ou l'analyse en composantes principales (ACP) sont couramment utilisées dans ce contexte. Toutefois, cette approche peut entraîner un taux élevé de faux positifs lorsque le modèle du comportement normal est mal défini ou insuffisamment représentatif. [28]

II.3.2.3 Clustering

Le clustering joue un rôle crucial dans l'analyse des alertes de sécurité en regroupant des points de données similaires, permettant ainsi d'identifier les comportements suspects sans s'appuyer sur des données d'attaque pré-étiquetées. Cette approche est fondamentale pour les méthodes d'apprentissage non supervisées, qui visent à découvrir des modèles, des structures ou des anomalies inhérentes aux données sans connaissance préalable de ce qui constitue une attaque. En appliquant des algorithmes tels que le clustering K-means, les systèmes de sécurité peuvent modéliser le comportement normal d'un réseau ou d'un système. Tous les points de données qui s'écartent de manière significative de ces groupes établis d'activité normale sont ensuite signalés comme des anomalies potentielles, indiquant des actions inhabituelles ou malveillantes **[**28**]**.

Cette fonctionnalité est particulièrement utile pour détecter les nouvelles attaques et les attaques de type « jour zéro », car elle ne nécessite pas la signature de menaces connues, mais permet plutôt d'identifier les comportements qui ne correspondent pas au niveau de normalité appris **[**28**]**.

II.3.3 Limites des approches existantes

Malgré les progrès réalisés, les approches actuelles présentent encore plusieurs limites :

- Difficulté d'interprétation des modèles complexes ;
- Manque d'explicabilité des décisions ;
- Dépendance aux données d'apprentissage ;

- Incapacité à fournir des explications compréhensibles pour les analystes humains.

Ces limites justifient le développement de nouvelles solutions capables non seulement de détecter les incidents, mais également de produire des **explications automatiques et compréhensibles des alertes de sécurité**, ce qui constitue l'objectif du projet SecTalk.

II.4 Utilisation du NLP et des LLM en cybersécurité

Les progrès rapides du traitement du langage naturel (NLP) ont influencé de manière significative le développement de grands modèles de langage (LLM).

Ces modèles fondamentaux sont capables d'apprendre des représentations linguistiques riches et peuvent être adaptés à diverses applications en aval avec un minimum de réglages. Dans le domaine de la cybersécurité, les LLM jouent un rôle de plus en plus important dans les systèmes intelligents capables de détecter automatiquement les vulnérabilités, d'analyser les malwares et de répondre aux attaques, d'autant plus que les cybermenaces augmentent en volume et en sophistication.

Les chercheurs tirent parti des capacités de compréhension du langage des LLM pour identifier et analyser les anomalies dans les données de sécurité, démontrant ainsi des performances et une interprétabilité supérieure à celles des méthodes traditionnelles d'apprentissage profond [30].

Cette intégration du NLP et des LLM est cruciale pour améliorer l'efficience et l'efficacité des pratiques de cybersécurité en traitant et en extrayant des informations à partir de grandes quantités de texte non structuré et de modèles d'apprentissage à partir de vastes ensembles de données [30].

II.4.1 Analyse automatique des logs

Les journaux système sont indispensables à la cybersécurité, car ils constituent des enregistrements critiques des informations d'exécution qui permettent aux ingénieurs d'évaluer l'état du système et d'identifier les problèmes. Cependant, le volume et la complexité de ces journaux rendent l'analyse manuelle fastidieuse et sujette à des erreurs, nécessitant des solutions automatisées [31].

Les grands modèles de langage (LLM), tels que LogLLM, permettent de résoudre ce problème en capturant efficacement la signification sémantique intégrée dans les données de log, que les méthodes traditionnelles d'apprentissage en profondeur ont souvent du mal à gérer [31].

Cette compréhension sémantique permet aux LLM de détecter automatiquement les anomalies et les comportements anormaux, améliorant ainsi de manière significative la fiabilité des systèmes logiciels et renforçant la sécurité en rationalisant l'identification des problèmes potentiels [31].

II.4.2 Résumé automatique d'incidents

Les centres opérationnels de sécurité (SOC) sont confrontés à un défi de taille en raison de l'immense volume d'alertes et de rapports d'incidents générés par les systèmes logiciels, qui constituent des enregistrements critiques des informations d'exécution. Cette énorme quantité de données inclut des journaux de console qui enregistrent l'état du système et les événements d'exécution critiques, ce qui rend les analyses manuelles fastidieuses et sujettes à des erreurs [31].

Pour y remédier, des techniques de traitement du langage naturel (NLP), en particulier celles reposant sur des modèles de langage large (LLM) sont utilisées pour résumer automatiquement ces messages de journal complexes. Ces modèles avancés excellent dans la capture de la signification sémantique intégrée dans les journaux en langage naturel, une tâche à laquelle les méthodes traditionnelles d'apprentissage en profondeur ont souvent du mal.

Les avantages d'une telle synthèse automatisée incluent des gains de temps importants, une meilleure compréhension des incidents et une meilleure aide à la décision, ce qui se traduit en fin de compte par des systèmes logiciels plus fiables et une meilleure posture de sécurité en facilitant l'identification des anomalies et des comportements anormaux [31].

II.4.3 Génération d'explications en langage naturel

Dans le domaine de la cybersécurité, la transformation des alertes techniques provenant des systèmes de détection d'intrusion (IDS) ou de la gestion des informations et des événements de sécurité (SIEM) en explications en langage naturel est cruciale pour permettre aux utilisateurs non techniques de comprendre et de faire confiance aux mécanismes de défense. L'intelligence artificielle explicable (XAI) joue un rôle central en améliorant la transparence et l'interprétabilité de ces systèmes, en allant au-delà des modèles opaques de type « boîte noire » [32].

Les méthodes XAI, telles que celles qui génèrent des explications basées sur du texte, sont essentielles pour transmettre des informations précises et rendre les décisions en matière d'IA plus intelligibles.

Cela permet aux experts en sécurité et aux autres parties prenantes de comprendre les raisons qui sous-tendent les conclusions du système, favorisant ainsi la confiance et une gestion efficace des cybermenaces [32].

2.4.4 Chatbots de cybersécurité

Les chatbots intelligents, alimentés par le traitement du langage naturel (NLP) et les grands modèles de langage (LLM) sont en train de devenir des outils essentiels pour aider les analystes de sécurité et les utilisateurs non techniques dans les opérations de cybersécurité. Ces chatbots

peuvent traiter des messages de journal complexes et des alertes techniques, en les transformant en explications compréhensibles en langage naturel [31].

Leurs fonctionnalités s'étendent à la fourniture d'informations sur les incidents de sécurité, à l'explication des méthodologies d'attaque, à la formulation de recommandations d'atténuation et à l'amélioration de la sensibilisation générale à la sécurité. Les capacités avancées de compréhension linguistique des LLM, préentraînées sur de vastes ensembles de données, améliorent considérablement la précision et la pertinence contextuelle des réponses fournies par ces chatbots. Cette approche est très pertinente pour des projets tels que SecTalk, qui visent à expliquer automatiquement les alertes de sécurité, comblant ainsi le fossé entre les données techniques et la compréhension humaine afin de faciliter la prise de décisions plus éclairées et d'améliorer la posture de sécurité globale [31].

II.5 Outils et solutions existantes dans le domaine

Face au volume croissant d'alertes générées par les systèmes de sécurité, plusieurs solutions technologiques ont été développées pour aider les centres opérationnels de sécurité (SOC) à mieux gérer les incidents. Ces outils visent à centraliser les données, automatiser l'analyse et rendre les informations plus accessibles

II.5.1 Plateformes SIEM intelligentes

Les systèmes de gestion des informations et des événements de sécurité (SIEM) servent essentiellement de plateformes centrales pour les centres d'opérations de sécurité modernes. Ils collectent, agrégent, stockent et corrélent les événements générés par diverses infrastructures gérées, telles que les systèmes de détection d'intrusion, les logiciels antivirus et les pare-feux **[**33**]**.

Ces systèmes ont évolué de manière significative pour devenir des outils complets offrant une large visibilité permettant d'identifier les zones à haut risque et de mettre en œuvre de manière proactive des stratégies d'atténuation, réduisant ainsi les coûts et les délais de réponse aux incidents [33].

La prochaine génération de SIEM intègre des fonctionnalités avancées telles que la détection en temps réel des anomalies comportementales, la gestion améliorée des incidents et la visualisation intelligente des interconnexions réseau, évoluant vers l'automatisation dans la sélection et le déploiement de contre-mesures. Cette évolution permet d'améliorer la corrélation des événements, de détecter les incidents plus efficacement et de centraliser les données de sécurité, afin de relever le défi des cyberattaques sophistiquées et dangereuses [33].

II.5.2 Outils d'analyse automatique d'alertes

Les systèmes de sécurité, tels que les systèmes de détection d'intrusion, les logiciels antivirus et les pare-feux, génèrent un grand nombre d'événements et d'alertes provenant de diverses infrastructures, qui sont collectés et agrégés par des plateformes telles que les systèmes de gestion des informations et des événements de sécurité (SIEM). Cet immense volume de données représente un défi de taille pour les analystes de sécurité, car l'interprétation et la réponse manuelles sont lentes, complexes et sujettes à des erreurs. Pour y remédier, des outils d'analyse automatisés, notamment des plateformes SIEM intelligentes, sont de plus en plus utilisés pour traiter et hiérarchiser ces alertes [33].

Ces outils tirent parti des technologies d'automatisation, d'intelligence artificielle (IA) et d'apprentissage automatique (ML) pour analyser de grands ensembles de données, identifier des modèles suspects et détecter des anomalies comportementales en temps réel [33].

Cette intégration d'analyses avancées et d'automatisation permet d'améliorer les capacités de détection, de répondre plus efficacement aux incidents, de réduire le nombre de faux positifs et de prendre des décisions plus rapides et plus précises face à des cybermenaces sophistiquées [33].

II.5.3 Solutions de génération automatique de rapports

La génération de rapports est une étape cruciale de la gestion des incidents de cybersécurité, qui aide à documenter les événements, identifier les causes sous-jacentes et enregistrer les mesures correctives. Cependant, avec la multiplication des données produites par les systèmes et les SIEM, la création de rapports est une tâche qui prend de plus en plus de temps pour les analystes. Les solutions de reporting automatisé facilitent ce processus en synthétisant les rapports et en utilisant les technologies du traitement du langage naturel pour transformer les données techniques en explications compréhensibles. L'utilisation de ces services accélère la création des rapports, établissent la piste d'audit pour les incidents et facilitent la communication interne entre les équipes techniques et la direction des risques [33].

II.5.4 Systèmes d'explication assistée

Le volume et la complexité croissants des alertes de sécurité générées par divers systèmes à partir de diverses infrastructures constituent un défi de taille pour les analystes de sécurité, car l'interprétation et les réponses manuelles sont lentes, complexes et sujettes aux erreurs [33].

Les systèmes actuels ne fournissent souvent pas d'explications claires et compréhensibles pour expliquer le grand nombre d'événements, ce qui nuit à l'efficacité de la prise de décisions [33].

Les systèmes d'explication assistée, en particulier les futurs SIEM, visent à y remédier en améliorant les extensions de visualisation et d'analyse, en fournissant aux utilisateurs des informations plus approfondies sur les situations de sécurité et des capacités de prise de décision

plus efficaces. Ces systèmes intègrent des analyses et une automatisation avancée, tirant potentiellement parti de l'intelligence artificielle explicable (XAI) et de l'apprentissage automatique, pour rendre leurs processus plus transparents et la justification des alertes ou des décisions plus compréhensibles pour les opérateurs humains [33].

Cette approche permet de mieux comprendre les incidents, de réduire le nombre de faux positifs et d'erreurs, et d'améliorer globalement l'efficacité des analystes et leurs capacités de réponse aux incidents [33].

II.6 Comparaison des approches existantes

Afin de synthétiser les différentes méthodes présentées (IDS, SIEM, systèmes experts, apprentissage automatique, NLP/LLM), le tableau suivant compare leurs points forts, leurs limites et leur niveau d'explicabilité :

Approche	Points forts	Limites	Explicabilité	Automatisation
IDS traditionnels	Détection rapide des signatures connues	Beaucoup de faux positifs, alertes techniques	Faible	Partielle
SIEM	Corrélation multi-sources, vision globale	Dépendance aux règles, messages complexes	Moyenne	Bonne
Systèmes experts (règles)	Explications explicites (IF–THEN)	Maintenance lourde, pas adapté aux attaques nouvelles	Bonne	Faible
Apprentissage automatique (ML)	Classification efficace, détection anomalies	Données annotées nécessaires, modèles boîte noire	Faible à moyenne	Bonne
NLP / LLM	Génération d'explications en langage naturel	Risque d'hallucinations, coût computationnel	Élevée	Très bonne

Tableau II.3 : Comparaison des approches existantes

Ce tableau montre que chaque approche apporte des avantages spécifiques, mais aucune ne parvient à fournir une explication claire, automatique et compréhensible des alertes de sécurité. C'est précisément ce manque que notre projet SecTalk vise à combler, en combinant classification automatique et génération en langage naturel.

II.7 Limites des travaux existants

Malgré les progrès importants réalisés dans le domaine de la cybersécurité et de l'intelligence artificielle, les solutions actuelles de gestion et d'analyse des alertes présentent encore plusieurs limitations. Les plateformes de sécurité modernes, telles que les systèmes SIEM et les outils de détection d'intrusion, permettent de collecter et d'analyser de grandes quantités de données. Cependant, leur capacité à fournir des explications compréhensibles et exploitables par les analystes reste limitée. Cette section présente les principales limites identifiées dans les travaux existants.

II.7.1 Manque d'explications compréhensibles pour l'humain

Dans le domaine de la cybersécurité, les systèmes actuels génèrent fréquemment des alertes techniques et des résultats complexes difficiles à comprendre pour les utilisateurs humains, en particulier pour les non-experts.

Cette difficulté provient d'un manque fondamental d'interprétabilité dans de nombreux systèmes d'apprentissage automatique (ML), qui donnent souvent la priorité à la performance par rapport à la compréhension humaine **[**34**]**.

Alors que les systèmes de machine learning sont de plus en plus omniprésents et peuvent surpasser les humains dans des tâches spécifiques, leur raisonnement reste souvent opaque, ce qui rend difficile la vérification du bien-fondé de leurs décisions, en particulier dans des applications critiques telles que la sécurité **[**34**]**.

Cette lacune met en évidence le besoin urgent d'une intelligence artificielle explicable (XAI), car la capacité d'expliquer ou de présenter des informations en termes compréhensibles à un humain est cruciale pour garantir la sécurité et la confiance dans les systèmes de machine learning opérant dans des domaines complexes tels que la cybersécurité. Sans explications claires, les utilisateurs ont du mal à convertir les alertes techniques en informations exploitables, ce qui constitue un obstacle important à une prise de décision efficace et à la validation du système **[**34**]**.

II.7.2 Systèmes souvent très techniques

Les systèmes de cybersécurité modernes présentent souvent une complexité technique importante, générant des alertes et des résultats qu'il est souvent difficile pour les utilisateurs humains, en particulier les non-experts, de comprendre pleinement ou d'agir en conséquence. Cette complexité inhérente est due au fait que de nombreux systèmes d'apprentissage automatique (ML), bien que puissants et capables de surpasser les humains dans des tâches spécifiques, fonctionnent souvent comme des « boîtes noires » où leur raisonnement interne est opaque **[**34**]**.

Un tel manque d'interprétabilité rend difficile pour les utilisateurs de comprendre pourquoi un système a pris une décision particulière, de vérifier le bien-fondé de sa logique ou de convertir les alertes techniques en informations de sécurité exploitables **[**34**]**.

La difficulté d'interprétation des résultats constitue un obstacle majeur, car les mécanismes sous-jacents de ces systèmes ne sont pas facilement accessibles ou compréhensibles, ce qui limite leur déploiement et leur validation efficaces dans des applications de sécurité réelles **[**34**]**. Cela met en évidence un défi plus vaste sur le terrain, où la recherche de performances éclipse souvent le besoin crucial d'explications compréhensibles par l'homme.

II.7.3 Manque d'automatisation complète

Malgré l'intégration croissante des techniques d'apprentissage automatique dans les systèmes de cybersécurité, un manque d'automatisation complète persiste, ce qui impose une charge importante aux analystes humains [34].

Les systèmes actuels produisent souvent des alertes techniques et des résultats qui nécessitent une intervention humaine importante pour l'interprétation et la validation, au lieu de fournir des informations facilement exploitables. Cette limitation provient de la complexité inhérente et du manque d'interprétabilité de nombreux modèles de machine learning, qui, bien que puissants, fonctionnent comme des « boîtes noires » dont le raisonnement est opaque pour les utilisateurs humains **[**34**]**.

Par conséquent, les utilisateurs non experts ont du mal à comprendre les décisions du système, à vérifier leur solidité ou à convertir les résultats techniques en mesures de sécurité pratiques, entravant ainsi la prise de décisions et le déploiement efficaces du système **[**34**]**. Cela met en évidence le besoin critique d'une automatisation intelligente qui non seulement exécute les tâches, mais explique également son raisonnement en termes compréhensibles, réduisant ainsi l'effort manuel requis de la part des professionnels de la sécurité et renforçant la confiance dans les systèmes automatisés.

II.7.4 Absence d'assistance interactive

Les systèmes de cybersécurité actuels présentent souvent un manque d'assistance interactive, ce qui pose des défis aux opérateurs humains. Ces systèmes génèrent fréquemment des alertes techniques et des résultats difficiles à interpréter pour les utilisateurs, ce qui rend difficile la compréhension des raisons qui sous-tendent les décisions du système. Cette limitation s'étend à divers domaines où la complexité technique inhérente aux outils de sécurité empêche leur utilisation efficace par des utilisateurs non experts **[**35**]**.

L'absence d'outils conversationnels aggrave encore ce problème, car les opérateurs humains ont du mal à interagir avec des systèmes qui ne peuvent pas fournir d'explications transparentes et compréhensibles **[**35**]**.

Par conséquent, il existe un besoin évident d'automatisation plus intelligente et de systèmes interactifs, tels que des chatbots avancés et des assistants pilotés par l'IA, pour combler le fossé entre les technologies de sécurité complexes et la compréhension humaine, permettant ainsi des opérations de cybersécurité plus intuitives et plus efficaces [35].

Cependant, ces systèmes interactifs représentent une perspective intéressante pour des travaux futurs et ne sont pas directement implémentés dans le cadre de ce projet, qui se concentre principalement sur l'explication automatique des alertes de sécurité.

Ainsi, l'analyse des travaux existants montre que, malgré les avancées significatives dans le domaine de la cybersécurité et de l'intelligence artificielle, plusieurs défis persistent, notamment en matière d'explicabilité des alertes, d'accessibilité des informations et d'automatisation de leur analyse.

Ces limitations justifient le développement de nouvelles approches capables de transformer les alertes techniques en explications claires et compréhensibles, comme le propose le système SecTalk.

Afin de mieux résumer les limites identifiées dans les solutions actuelles de cybersécurité, le tableau suivant présente une synthèse des principaux problèmes observés et de leurs impacts sur l'analyse des alertes de sécurité.

Limite identifiée	Description	Impact sur l'analyse des alertes
Manque d'explications compréhensibles	Les alertes générées par les IDS et les SIEM sont souvent présentées sous forme de messages techniques contenant des informations telles que des adresses IP, des ports ou des signatures d'attaque, sans fournir d'explication claire du contexte.	Difficulté pour les analystes à comprendre rapidement la nature de la menace et à prendre une décision appropriée.
Systèmes très techniques	Les interfaces et les messages générés par les systèmes de cybersécurité utilisent un vocabulaire technique complexe, principalement destiné à des experts en sécurité informatique.	Risque de mauvaise interprétation des alertes et difficulté d'utilisation pour les utilisateurs moins expérimentés.
Manque d'automatisation complète	Une grande partie du processus d'analyse des alertes reste manuelle, nécessitant l'intervention des analystes pour corréler les informations provenant de différentes sources.	Augmentation de la charge de travail dans les centres opérationnels de sécurité (SOC) et ralentissement de la réponse aux incidents.
Absence d'assistance interactive	Les systèmes actuels fournissent généralement des alertes statiques sans permettre aux utilisateurs d'interagir avec le système pour obtenir des explications supplémentaires.	Difficulté pour les analystes à approfondir l'analyse des alertes et à obtenir rapidement des informations complémentaires.

Tableau II.4 : Synthèse des limites des travaux existants

Le tableau ci-dessus met en évidence les principales limites des solutions actuelles de gestion des alertes de sécurité. Ces limitations montrent la nécessité de développer des systèmes capables d'améliorer la compréhension des alertes, d'automatiser leur interprétation et de fournir des explications claires aux utilisateurs. Dans ce contexte, le projet **SecTalk** vise à proposer une approche basée sur l'intelligence artificielle générative afin de transformer les alertes techniques en explications compréhensibles en langage naturel.

II.8 Positionnement de notre travail

L'analyse des travaux existants et des solutions actuelles en cybersécurité met en évidence plusieurs limites, notamment le manque d'explications compréhensibles pour les utilisateurs, la complexité technique des outils, le faible niveau d'automatisation dans l'analyse des alertes ainsi que l'absence d'assistance interactive. Ces insuffisances compliquent la compréhension rapide des incidents et ralentissent la prise de décision au sein des centres opérationnels de sécurité (SOC).

Dans ce contexte, le projet SecTalk propose une approche visant à améliorer à la fois la détection et la compréhension des alertes de sécurité. Contrairement aux solutions traditionnelles, souvent limitées à la détection ou à la corrélation d'événements, SecTalk adopte une approche intégrée permettant de traiter les alertes de manière plus intelligente et accessible.

L'objectif principal du système est de transformer les alertes techniques en explications claires et compréhensibles en langage naturel, afin de faciliter leur interprétation par les utilisateurs, qu'ils soient experts ou non. En complément, le système vise à fournir une assistance permettant d'aider à la prise de décision face aux incidents de sécurité.

Ainsi, SecTalk s'inscrit dans une démarche visant à détecter, comprendre et aider à réagir face aux cybermenaces. En intégrant des techniques d'intelligence artificielle et de traitement du langage naturel, il permet de combler le fossé entre la complexité des données de sécurité et la compréhension humaine.

Par conséquent, ce travail se positionne à l'intersection de la cybersécurité et de l'intelligence artificielle, avec pour objectif d'améliorer l'efficacité des analystes, de réduire leur charge de travail et de faciliter la prise de décision dans des environnements de sécurité complexes.

Afin de mieux situer notre approche par rapport aux solutions existantes, le tableau suivant présente une comparaison entre les systèmes traditionnels et le système proposé SecTalk.

Critère	Solutions existantes (SIEM, IDS, outils d'analyse)	Système SecTalk
Type d'alertes	Alertes techniques basées sur des logs, signatures et événements réseau	Transformation des alertes en explications en langage naturel
Compréhension	Nécessite une expertise technique avancée	Explications claires et accessibles aux experts et non-experts
Automatisation	Automatisation limitée à la détection et à la corrélation	Automatisation de l'analyse et de l'explication
Objectif principal	Détection et gestion des incidents	Détecter + Comprendre + Aider à réagir

Tableau II.5 : Comparaison entre les solutions existantes et le système SecTalk

Cette comparaison met en évidence l'apport de SecTalk, qui ne se limite pas à la détection des incidents, mais vise également à améliorer leur compréhension et à assister les utilisateurs dans la prise de décision.

II.9 Conclusion

Ce chapitre a présenté une analyse des principaux travaux et solutions existants dans le domaine de la cybersécurité et de l'analyse des alertes de sécurité. Dans un premier temps, les concepts liés aux sources de données de sécurité, notamment les journaux d'événements, les systèmes de détection d'intrusion et les plateformes SIEM, ont été abordés. Ensuite, l'apport de l'intelligence artificielle dans la détection et l'analyse des menaces a été présenté, en mettant en évidence le rôle des techniques d'apprentissage automatique et des méthodes d'analyse des données.

Par ailleurs, l'évolution récente du traitement automatique du langage naturel (NLP) et l'émergence des modèles de langage de grande taille ont ouvert de nouvelles perspectives pour l'analyse et l'interprétation des données de sécurité. Ces technologies permettent notamment d'automatiser l'analyse des logs, de produire des résumés d'incidents et de générer des explications en langage naturel afin de faciliter la compréhension des alertes.

Cependant, l'étude des solutions existantes a également mis en évidence plusieurs limitations importantes. Les systèmes actuels génèrent souvent des alertes techniques difficiles à interpréter, nécessitent une expertise élevée et reposent encore largement sur l'intervention humaine. De plus, les mécanismes d'explication automatisée restent encore limités dans les outils actuels.

Face à ces limitations, il apparaît nécessaire de développer des approches capables non seulement d'améliorer la compréhension des alertes, mais aussi d'automatiser davantage leur analyse. Dans ce contexte, le projet SecTalk propose une approche intégrée visant à détecter les menaces, générer des explications compréhensibles en langage naturel et assister les utilisateurs dans la prise de décision.

Chapitre III :

Conception et réalisation du système intelligent SecTalk

III.1 Introduction

Avec l'évolution rapide des cybermenaces et l'augmentation continue du trafic réseau, les systèmes de détection d'intrusion (IDS) sont devenus indispensables pour assurer la sécurité des infrastructures informatiques. Toutefois, bien qu'efficaces pour identifier les activités suspectes, ces systèmes génèrent des alertes souvent complexes et difficiles à interpréter, ce qui limite leur exploitation par les utilisateurs.

Dans ce contexte, les techniques d'apprentissage automatique permettent d'améliorer la détection automatique des attaques réseau à partir des données de trafic. Le système proposé, **SecTalk**, repose sur une approche de classification appliquée au dataset **CIC-IDS2017**, visant à identifier différents types d'attaques.

En complément de la phase de détection, le système intègre un module d'explication basé sur des modèles de langage exécutés via **Ollama**, permettant de transformer les alertes techniques en explications simples et compréhensibles. Un chatbot de cybersécurité permet à l'utilisateur de saisir une alerte de sécurité et d'interagir avec celle-ci. Ce chatbot fournit des explications claires ainsi que des recommandations adaptées au contexte de chaque incident.

Par ailleurs, ce chapitre inclut une phase d'expérimentation et d'analyse des résultats, portant à la fois sur les modèles de détection et sur les modèles utilisés pour la génération des explications.

Ce chapitre présente ainsi l'environnement de développement, le dataset utilisé, les étapes de prétraitement des données, l'architecture générale du système SecTalk, ainsi que les résultats obtenus et leur analyse.

III.2 Environnement de développement et technologies utilisées

Le développement du système SecTalk repose sur plusieurs technologies et outils logiciels utilisés pour le traitement des données, l'entraînement des modèles de machine learning, la génération des explications intelligentes ainsi que le développement de l'interface utilisateur. Cette section présente les principales technologies utilisées dans la réalisation du système.

III.2.1 Langage de programmation : Python

Le système SecTalk a été développé principalement avec le langage de programmation **Python** grâce à sa simplicité, sa flexibilité et sa large utilisation dans les domaines du machine learning, de l'intelligence artificielle et de la cybersécurité. Python dispose également de nombreuses bibliothèques spécialisées facilitant le traitement des données, l'analyse du trafic réseau et le développement des applications intelligentes.

Dans ce projet, Python a été utilisé pour le prétraitement des données, l'entraînement du modèle de détection, l'intégration du chatbot intelligent ainsi que la génération des explications en langage naturel.

III.2.2 Bibliothèques et frameworks utilisés

Plusieurs bibliothèques et frameworks Python ont été utilisés dans le développement du système SecTalk afin d'assurer le traitement des données, la détection des attaques et la génération des explications intelligentes.

- **Pandas** : utilisé pour la manipulation, le nettoyage et l'analyse des données du dataset CIC-IDS2017.
- **NumPy** : utilisé pour les opérations numériques et le traitement des tableaux de données.
- **Scikit-learn** : utilisé pour le prétraitement des données, l'encodage des labels, la division du dataset ainsi que l'évaluation des performances du modèle.
- **LightGBM** : utilisé pour l'implémentation du modèle de machine learning chargé de la détection des attaques réseau.
- **Joblib** : utilisé pour sauvegarder et charger le modèle entraîné ainsi que les fichiers nécessaires au système.
- **Ollama** : utilisé pour l'exécution locale des modèles de langage génératifs afin de produire des explications automatiques des alertes de sécurité.
- **Plotly** : utilisé pour la visualisation graphique et l'affichage interactif des résultats.
- **FastAPI** : utilisé pour développer le backend du système et assurer la communication entre l'interface utilisateur, le modèle de détection et le chatbot intelligent.

III.2.3 Technologies d'intelligence artificielle générative

Afin d'ajouter une capacité d'explication intelligente au système SecTalk, plusieurs technologies d'intelligence artificielle générative ont été intégrées. Ces technologies permettent de générer des réponses automatiques en langage naturel afin d'aider les utilisateurs à mieux comprendre les alertes de sécurité et les attaques détectées.

III.2.3.1 Ollama

Ollama est une plateforme permettant l'exécution locale des modèles de langage génératifs (LLM). Dans le système SecTalk, Ollama est utilisé pour générer automatiquement des explications en langage naturel à partir des alertes détectées ou des événements de sécurité saisis par l'utilisateur.

L'utilisation d'Ollama permet de garantir une meilleure confidentialité des données tout en offrant une exécution rapide des modèles sans dépendre des services cloud externes.

III.2.3.2 Modèle de langage utilisé

Le système SecTalk utilise un modèle de langage génératif tel que Phi3 afin de produire des réponses explicatives en langage naturel. Ces modèles sont capables d'analyser les informations fournies par le système et de générer des explications détaillées concernant les alertes détectées.

III.2.3.3 Architecture RAG (Retrieval-Augmented Generation)

Le système SecTalk intègre une approche **RAG (Retrieval-Augmented Generation)** afin d'améliorer la qualité des réponses générées par le modèle de langage.

Cette approche consiste à récupérer des informations pertinentes depuis une base de connaissances contenant des descriptions d'attaques, des impacts et des recommandations de sécurité, puis à intégrer ces informations dans le prompt envoyé au modèle génératif.

L'utilisation du RAG permet de produire des explications plus précises, contextualisées et adaptées aux alertes analysées.

III.2.4 Outils de développement

Plusieurs outils de développement ont été utilisés dans la réalisation du système SecTalk afin de faciliter la programmation.

- ✓ Anaconda **Jupyter Notebook** : utilisé principalement durant la phase d'entraînement et d'expérimentation des modèles de machine learning. Cet environnement a permis de réaliser le prétraitement des données, l'analyse du dataset CIC-IDS2017, l'entraînement des modèles de détection ainsi que l'évaluation des performances des algorithmes.
- ✓ **Visual Studio Code (VS Code)** : utilisé pour le développement de l'application finale du système SecTalk. Il a servi à l'implémentation du module d'explication intelligent, de l'intégration avec Ollama, du système RAG ainsi que de l'interface utilisateur interactive et du chatbot de cybersécurité.

III.2.5 Configuration matérielle

Le développement et les expérimentations du système SecTalk ont été réalisés sur une machine disposant des caractéristiques matérielles suivantes :

- **Processeur** : Intel Core i5
- **Mémoire RAM** : 8 Go
- **Système d'exploitation** : Windows 11

Cette configuration a permis d'exécuter les traitements de données, le modèle de détection Light GBM ainsi que les modèles de langage utilisés dans le système.

III.3 Présentation du dataset et des données de sécurité

Dans le cadre du développement du système SecTalk, le dataset **CIC-IDS2017** a été utilisé afin d'entraîner et d'évaluer le modèle de détection des attaques réseau. Le choix du dataset représente une étape importante dans la conception d'un système IDS performant, car la qualité et la diversité des données influencent directement les performances du modèle de machine learning.

Le dataset CIC-IDS2017 a été choisi en raison de sa richesse, de la diversité des attaques qu'il contient ainsi que de sa large utilisation dans les travaux de recherche en cybersécurité et en détection d'intrusion.

III.3.1 Présentation du dataset CIC-IDS2017

Le dataset **CIC-IDS2017** a été développé par le *Canadian Institute for Cybersecurity (CIC)* de l'Université du Nouveau-Brunswick. Il contient un ensemble de données réseau représentant des scénarios réalistes de trafic normal et de trafic malveillant générés dans un environnement contrôlé.

Ce dataset inclut plusieurs types d'attaques réseau modernes ainsi que du trafic normal appelé *Benign*. Les données sont organisées sous forme de fichiers CSV contenant différentes caractéristiques réseau (features) utilisées pour l'analyse et la classification des attaques.

Le dataset CIC-IDS2017 est réparti sur plusieurs jours de capture du trafic réseau. Chaque jour contient des types spécifiques d'activités normales et d'attaques réseau permettant de simuler différents scénarios de cybersécurité. Cette organisation par jours permet d'obtenir une grande diversité de trafic réseau et améliore la capacité des modèles de machine learning à détecter différents comportements malveillants dans des environnements variés.

Le dataset CIC-IDS2017 est largement utilisé dans les travaux de recherche liés aux systèmes de détection d'intrusion et à la machine learning grâce à sa diversité, sa taille importante et la présence de plusieurs catégories d'attaques réelles.

III.3.2 Types de trafic réseau

Le dataset utilisé contient deux grandes catégories de trafic réseau :

- **Trafic normal (Benign)** : représente les communications réseau légitimes et les activités normales des utilisateurs sans comportement malveillant.
- **Trafic malveillant** : représente les activités suspectes ou les attaques réalisées contre le réseau afin de perturber les services, voler des informations ou compromettre les systèmes.

Cette diversité de trafic permet d'entraîner les modèles de machine learning à distinguer les comportements normaux des comportements malveillants.

III.3.3 Types d'attaques utilisées

Le système SecTalk a été entraîné sur plusieurs types d'attaques présentes dans le dataset CIC-IDS2017, notamment :

Attaques	Description
DDoS (Distributed Denial of Service)	Attaque visant à saturer un serveur ou un réseau avec un grand volume de trafic.
DoS (Denial of Service)	Attaque destinée à rendre un service indisponible.
Botnet	Réseau de machines compromises contrôlées par un attaquant.
PortScan	Technique utilisée pour analyser les ports ouverts d'une machine cible.
WebAttack	Attaques ciblant les applications web, comme les injections SQL ou XSS.
Brute Force	Attaque consistant à tester plusieurs combinaisons de mots de passe afin d'obtenir un accès non autorisé.
Heartbleed	Vulnérabilité affectant le protocole OpenSSL permettant la fuite d'informations sensibles.
Infiltration	Attaque où un attaquant pénètre progressivement dans un système afin d'exécuter des actions malveillantes.

Tableau III.6 : Les attaques et les descriptions

III.3.4 Structure des données

Les données du dataset CIC-IDS2017 sont organisées sous forme de fichiers CSV contenant plusieurs colonnes représentant les caractéristiques du trafic réseau.

Chaque ligne correspond à une connexion réseau ou un flux de communication analysé. Les colonnes du dataset contiennent différentes informations statistiques et réseau telles que :

- La durée des connexions ;
- Le nombre de paquets transmis ;
- Le volume de données échangées ;
- Les statistiques des flux réseau ;
- Les informations liées aux protocoles et aux ports ;
- Les indicateurs des drapeaux TCP (*Flags*).

Le dataset contient également une colonne nommée **Label**, représentant la catégorie du trafic réseau, soit un trafic normal (*Benign*), soit une attaque spécifique.

III.3.5 Fichiers CSV utilisés

Dans ce projet, plusieurs fichiers CSV du dataset CIC-IDS2017 ont été utilisés afin de construire le dataset final destiné à l'entraînement du modèle de détection.

Les fichiers exploités sont :

- ✓ tuesday.csv
- ✓ wednesday.csv
- ✓ thursday.csv
- ✓ friday.csv

Ces fichiers regroupent différentes catégories de trafic réseau normal et malveillant, couvrant plusieurs scénarios d'attaques. Leur utilisation permet d'obtenir un ensemble de données varié et représentatif des comportements observés dans un environnement réseau réel.

III.4 Préparation et prétraitement des données

Avant l'entraînement des modèles de machine learning, plusieurs étapes de préparation et de prétraitement ont été réalisées afin d'améliorer la qualité des données et d'optimiser les performances du système de détection. Ces étapes permettent de nettoyer les données, sélectionner les caractéristiques importantes et préparer le dataset pour les modèles de classification.

III.4.1 Fusion des fichiers CSV

Les différents fichiers sélectionnés ont été fusionnés à l'aide de la fonction **pd.concat()** de la bibliothèque Pandas afin de constituer un unique dataset d'apprentissage.

Cette opération permet de regrouper l'ensemble des observations dans une seule structure de données et d'augmenter la diversité des exemples disponibles pour l'entraînement. Le dataset obtenu contient ainsi plusieurs catégories d'attaques ainsi que du trafic bénin, ce qui favorise une meilleure capacité de généralisation du modèle de détection.

III.4.2 Nettoyage des données

Après la fusion des fichiers CSV, plusieurs opérations de nettoyage ont été réalisées afin de supprimer les données incorrectes ou inutiles pouvant affecter les performances du modèle.

III.4.2.1 Valeurs manquantes

Certaines lignes du dataset contenaient des valeurs manquantes (*NaN*). Afin d'éviter les erreurs durant l'entraînement, les valeurs manquantes ont été remplacées par `0` à l'aide de la fonction `fillna(0)` de Pandas.

III.4.2.2 Valeurs infinies

Le dataset contenait également des valeurs infinies (`inf` et `-inf`) générées par certaines opérations statistiques liées au trafic réseau. Ces valeurs ont été remplacées par des valeurs nulles (`NaN`) grâce à NumPy avant le traitement final des données.

III.4.2.3 Colonnes supprimées

Certaines colonnes ont été supprimées car elles ne sont pas utiles pour la classification ou peuvent affecter les performances du modèle. Les colonnes supprimées sont :

- Src IP
- Dst IP
- Timestamp
- Attempted Category

La suppression de ces colonnes permet de réduire la complexité des données et de conserver uniquement les informations pertinentes pour la détection des attaques.

III.4.3 Sélection des caractéristiques

La sélection des caractéristiques (*Feature Selection*) représente une étape importante dans le développement du système de détection.

Dans ce projet, seules les colonnes numériques de type `int64` et `float64` ont été conservées à l'aide de la fonction `select_dtypes()` afin d'assurer la compatibilité avec le modèle LightGBM.

III.4.4 Encodage des labels

Les labels représentant les différentes catégories d'attaques ont été transformés en valeurs numériques grâce à la technique `LabelEncoder` de Scikit-learn.

Cette étape est nécessaire car les modèles de machine learning ne peuvent pas traiter directement les labels textuels. Chaque type d'attaque a donc été converti en une valeur numérique utilisée durant l'apprentissage du modèle.

III.4.5 Division des données

Après les étapes de prétraitement, le dataset final a été divisé en deux ensembles à l'aide de la fonction `train_test_split()` :

- **80 %** des données ont été utilisées pour l'entraînement ;
- **20 %** des données ont été utilisées pour les tests.

La division a été réalisée avec le paramètre `stratify=y` afin de conserver une distribution équilibrée des différentes classes d'attaques dans les ensembles d'entraînement et de test.

Le paramètre `random_state=42` a également été utilisé afin de garantir la reproductibilité des résultats expérimentaux.

III.5 Conception générale du système SecTalk

La conception du système SecTalk repose sur une architecture modulaire permettant de séparer les différentes étapes du processus de détection et d'explication des attaques réseau. Chaque module joue un rôle précis dans le traitement des données, depuis l'acquisition des informations jusqu'à la génération des réponses intelligentes destinées aux utilisateurs. Cette organisation facilite la maintenance du système et améliore sa flexibilité.

III.5.1 Architecture globale du système

L'architecture globale du système SecTalk repose sur un pipeline intelligent combinant la détection des attaques réseau, l'explication des alertes et la génération de recommandations de sécurité. Le système fonctionne selon deux modes principaux.

Dans le premier mode, l'utilisateur charge un fichier CSV contenant des données de trafic réseau issues du dataset CIC-IDS2017. Ces données passent d'abord par une étape de prétraitement afin de nettoyer et préparer les informations avant leur analyse. Les données sont ensuite envoyées au modèle de machine learning LightGBM chargé de détecter automatiquement le type de trafic réseau et d'identifier les différentes catégories d'attaques.

Après la phase de classification, le système génère automatiquement une explication en langage naturel décrivant l'attaque détectée, ses impacts possibles ainsi que des recommandations pratiques de sécurité permettant d'aider l'utilisateur à mieux comprendre la situation analysée.

Dans le second mode, l'utilisateur interagit directement avec un chatbot de cybersécurité. L'utilisateur peut saisir une alerte de sécurité ou un événement suspect sous forme textuelle, puis le chatbot génère automatiquement une réponse explicative en langage naturel. Le chatbot fournit également des recommandations adaptées afin d'aider les utilisateurs débutants ou non experts à réagir correctement face aux incidents de sécurité.

Le système utilise également une approche RAG (Retrieval-Augmented Generation) simplifiée basée sur une base de connaissances spécialisée contenant des informations sur les attaques réseau, les alertes de sécurité et les solutions recommandées. Cette approche permet d'améliorer la pertinence et la précision des réponses générées par le modèle de langage.

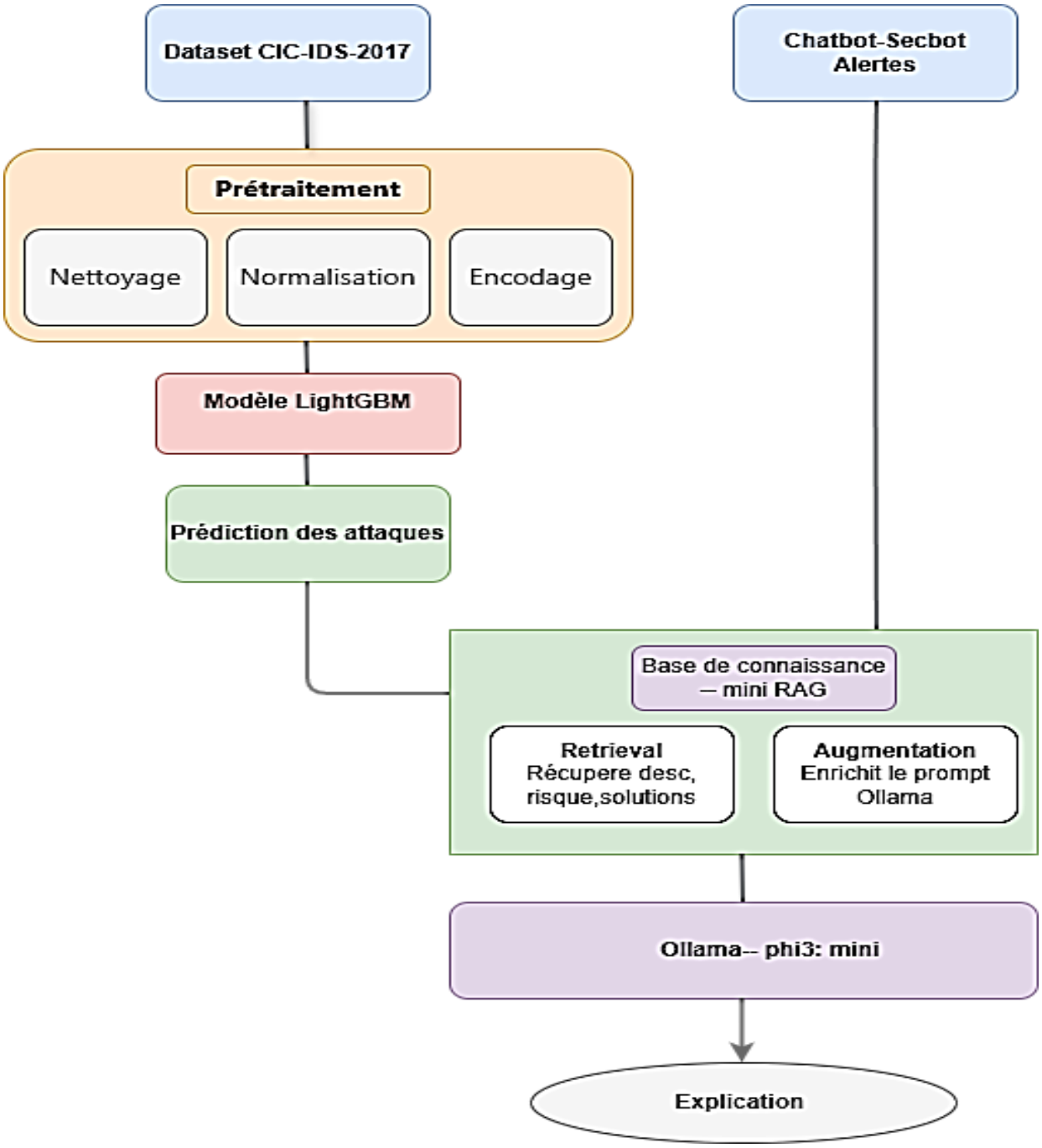

Figure III.3 : Pipeline de fonctionnement du système SecTalk

III.5.2 Description des modules principaux

Le système SecTalk est composé de plusieurs modules interconnectés, chacun ayant un rôle spécifique dans le processus global de détection et d'explication des attaques réseau.

III.5.2.1 Module d'acquisition des données

Ce module permet l'importation des fichiers CSV contenant les données de trafic réseau ainsi que la réception des alertes de sécurité saisies par l'utilisateur. Il constitue le point d'entrée principal du système.

III.5.2.2 Module de prétraitement

Ce module assure le nettoyage et la préparation des données avant leur utilisation par le modèle de machine learning. Il effectue notamment la gestion des valeurs manquantes, la suppression des colonnes inutiles et la sélection des caractéristiques numériques.

III.5.2.3 Module de détection ML

Le module de détection utilise l'algorithme LightGBM afin de classifier automatiquement le trafic réseau et identifier les différentes catégories d'attaques présentes dans les données analysées.

III.5.2.4 Module RAG et base de connaissances

Ce module contient une base de connaissances spécialisée sur les attaques et les alertes de cybersécurité. Il utilise l'approche RAG pour récupérer les informations pertinentes et enrichir les réponses générées par le modèle de langage.

III.5.2.5 Module d'explication intelligent avec Ollama

Ce module est responsable de la génération des explications automatiques en langage naturel. Il utilise Ollama ainsi qu'un modèle de langage génératif afin d'analyser les résultats de détection ou les alertes saisies par l'utilisateur.

Le module produit des explications détaillées concernant les attaques détectées, leurs impacts possibles ainsi que des recommandations pratiques de sécurité adaptées au contexte analysé. Grâce à l'approche RAG, les réponses générées sont enrichies par les informations provenant de la base de connaissances du système.

III.5.2.6 Module chatbot intelligent

Le chatbot intelligent permet une interaction directe avec l'utilisateur. Il analyse les alertes de sécurité saisies et génère des explications détaillées ainsi que des recommandations pratiques en langage naturel afin d'aider les utilisateurs à comprendre les incidents détectés.

III.5.2.7 Interface utilisateur

L'interface utilisateur a été développée avec HTML, CSS et FastAPI. Elle permet de charger les fichiers CSV, visualiser les résultats de détection, consulter les explications générées et interagir avec le chatbot de cybersécurité de manière simple et interactive.

III.6 Implémentation et optimisation des modèles de détection d'intrusion

Afin d'assurer la détection automatique des attaques réseau, plusieurs algorithmes de machine learning ont été implémentés et évalués dans le système SecTalk. L'objectif principal est d'identifier le modèle offrant les meilleures performances pour la classification des données du dataset CIC-IDS2017.

Dans ce projet, trois modèles de classification supervisée ont été utilisés :

- Random Forest
- Regression Logistic
- LightGBM

Ces modèles ont été entraînés et comparés à l'aide de différentes métriques d'évaluation afin de sélectionner l'algorithme le plus performant pour le système final de détection d'intrusion.

III.6.1 Modèle Random Forest

III.6.1.1 Principe de fonctionnement

Le modèle **Random Forest** est un algorithme d'apprentissage supervisé basé sur plusieurs arbres de décision (*Decision Trees*). Son principe consiste à construire un ensemble d'arbres entraînés sur différentes parties aléatoires du dataset, puis à combiner leurs prédictions afin d'obtenir un résultat final plus stable et plus précis.

Chaque arbre prend une décision indépendamment des autres, puis le système applique un mécanisme de vote majoritaire pour déterminer la classe finale prédite.

Dans le système SecTalk, Random Forest a été utilisé pour analyser les caractéristiques du trafic réseau et détecter les différentes catégories d'attaques présentes dans le dataset CIC-IDS2017.

III.6.1.2 Paramètres utilisés

Plusieurs paramètres ont été définis afin de configurer le modèle Random Forest, notamment :

- n_estimators=150,
- class_weight='balanced',

- random_state=42,
- n_jobs=-1

Ces paramètres influencent directement la précision et la capacité de généralisation du modèle.

III.6.2 Modèle de Régression Logistique

III.6.2.1 Principe de fonctionnement

La régression logistique est un algorithme de classification supervisée utilisé pour estimer la probabilité qu'une observation appartienne à une classe donnée. Elle repose sur l'application d'une fonction sigmoïde permettant de transformer une combinaison linéaire des variables d'entrée en une probabilité comprise entre 0 et 1.

Dans un contexte multi-classes, la régression logistique utilise généralement la stratégie *One-vs-Rest* pour distinguer les différentes catégories d'attaques.

Dans le système SecTalk, ce modèle a été utilisé comme baseline afin de comparer ses performances avec des algorithmes plus complexes comme Random Forest et LightGBM dans la détection des intrusions réseau.

III.6.2.2 Paramètres utilisés

Le modèle de régression logistique a été configuré avec les hyperparamètres suivants :

- **max_iter = 1000** : nombre maximal d'itérations permettant d'assurer la convergence du modèle ;
- **class_weight = 'balanced'** : gestion du déséquilibre des classes dans le dataset, en attribuant des poids inverses à la fréquence des classes ;
- **random_state = 42** : garantit la reproductibilité des résultats expérimentaux ;
- **n_jobs = -1** : permet d'utiliser tous les cœurs du processeur pour accélérer l'entraînement du modèle.

Ces paramètres améliorent la stabilité du modèle ainsi que ses performances sur des données déséquilibrées, comme celles rencontrées dans les datasets de détection d'intrusion.

III.6.3 Modèle LightGBM

III.6.3.1 Principe de fonctionnement

Le modèle **LightGBM (Light Gradient Boosting Machine)** est un algorithme de machine learning basé sur la technique du *Gradient Boosting*. Son fonctionnement repose sur la création progressive de plusieurs arbres de décision afin de corriger les erreurs produites par les arbres précédents.

Contrairement aux modèles classiques basés sur un seul arbre de décision, LightGBM améliore progressivement les performances de classification grâce à un apprentissage itératif.

Cet algorithme est particulièrement adapté aux grands datasets grâce à sa rapidité d'exécution, sa faible consommation de mémoire et sa capacité à gérer efficacement les données complexes.

III.6.3.2 Paramètres utilisés

Le modèle LightGBM a été configuré avec les paramètres suivants :

- n_estimators = 300
- learning_rate = 0.05
- num_leaves = 31
- class_weight = 'balanced'
- random_state = 42

Le paramètre class_weight='balanced' permet d'améliorer la détection des classes minoritaires présentes dans le dataset.

III.6.4 Validation des modèles

Afin d'évaluer les performances des différents modèles de détection, plusieurs techniques de validation ont été utilisées durant les expérimentations.

III.6.4.1 Validation croisée

La validation croisée (*Cross-Validation*) est une méthode permettant d'évaluer la robustesse et la capacité de généralisation des modèles de machine learning.

Cette technique consiste à diviser les données en plusieurs parties afin d'entraîner et tester le modèle plusieurs fois sur différentes combinaisons du dataset.

La validation croisée permet d'obtenir une estimation plus fiable des performances réelles des modèles et réduit le risque de surapprentissage (*Overfitting*).

III.6.4.2 StratifiedKFold

Dans ce projet, la méthode **StratifiedKFold** de Scikit-learn a été utilisée pour réaliser la validation croisée.

Les paramètres suivants ont été utilisés :

- n_splits = 3
- shuffle = True
- random_state = 42

Cette méthode garantit une distribution équilibrée des différentes classes d'attaques dans chaque division du dataset, ce qui est particulièrement important dans les systèmes IDS où certaines attaques sont moins représentées que d'autres.

III.6.4.3 Cross-validation avec F1-score

La fonction cross_val_score() a été utilisée afin d'évaluer les performances des modèles sur plusieurs divisions du dataset.

L'évaluation a été réalisée à l'aide de la métrique f1_macro, qui permet de calculer un score moyen équilibré entre toutes les classes d'attaques.

Cette étape a permis de comparer les performances des modèles Random Forest, Régression Logistique et LightGBM afin d'identifier l'algorithme le plus performant pour le système SecTalk.

III.7. Implémentation du système d'explication intelligent basé sur RAG simplifiée

III.7.1 Base de connaissances des attaques et alertes

La base de connaissances constitue un élément central du système d'explication intelligent. Elle contient des informations structurées concernant plusieurs types d'attaques réseau.

Cette base permet d'enrichir les réponses générées par le modèle de langage et d'améliorer la pertinence des explications produites par le système SecTalk.

III.7.1.1 Structure de la base de connaissances

La base de connaissances a été implémentée sous forme d'une structure Python contenant les informations principales associées à chaque type d'attaque détectée.

Chaque entrée contient :

- Une description de l'attaque ;
- Un niveau de risque ;
- Des solutions proposées.

Un extrait simplifié de la structure utilisée dans le système est présenté ci-dessous :

```
attack_knowledge = {
    "DDoS": {"description": "Un grand nombre d'ordinateurs envoient du trafic pour bloquer un serveur.",
             "risk": "Très élevé",
             "solutions": ["Limiter le trafic entrant","Utiliser une protection anti-DDoS","Surveiller le réseau"]},

    "DoS": {"description": "Un attaquant surcharge un service pour le rendre indisponible.",
            "risk": "Élevé",
            "solutions": ["Bloquer l'adresse suspecte","Limiter les requêtes"]},

    "Botnet": {"description": "Un réseau de machines infectées lance des attaques.",
               "risk": "Très élevé",
               "solutions": ["Scanner les machines","Isoler les postes infectés"]},

    "PortScan": {"description": "Tentative de découverte des ports ouverts.",
                 "risk": "Moyen",
                 "solutions": ["Fermer les ports inutiles","Surveiller le trafic"]},

    "BruteForce_FTP": {"description": "Tentative de deviner un mot de passe FTP.",
                       "risk": "Élevé",
                       "solutions": ["Utiliser des mots de passe forts","Limiter les tentatives"]},

    "BruteForce_SSH": {"description": "Tentative répétée de connexion SSH.",
                       "risk": "Élevé",
                       "solutions": ["Utiliser des clés SSH","Bloquer après plusieurs essais"]}
```

Figure III.4 : base de connaissance des attaques

Cette structure est utilisée par le module RAG afin de récupérer les informations pertinentes avant la génération des réponses avec Ollama.

III.7.2 Principe du RAG dans SecTalk

Le système SecTalk utilise une approche basée sur le **RAG (Retrieval-Augmented Generation)** afin d'améliorer la qualité des explications générées.

Le principe du RAG consiste à enrichir les réponses du modèle LLM à l'aide d'informations récupérées depuis une base de connaissances spécialisée en cybersécurité.

Lorsqu'une attaque ou une alerte est détectée, le système recherche automatiquement les informations correspondantes dans la base, notamment :

- La description de l'attaque (alertes)
- Le niveau de risque
- Les impacts possibles
- Les recommandations de sécurité.

Ces informations sont ensuite utilisées pour enrichir le contexte envoyé au modèle de langage afin de produire des réponses plus précises et adaptées au domaine de la cybersécurité.

Cette approche permet également de réduire les erreurs et les hallucinations du modèle génératif.

III.7.3 Génération des explications avec Ollama

Le système SecTalk utilise Ollama afin d'exécuter localement un modèle de langage génératif chargé de produire des explications automatiques des attaques et alertes de sécurité.

Après la détection d'une attaque, le système construit automatiquement un prompt contenant :

- Les attaques détectées ;
- Le contexte de sécurité ;
- Les informations récupérées depuis la base de connaissances.

Le modèle LLM génère ensuite une réponse en langage naturel expliquant :

- La signification de l'attaque ;
- Les risques associés ;
- Les recommandations de sécurité adaptées.

Le système utilise également un prompt système définissant le rôle du modèle comme un expert SOC capable de répondre en français simple et compréhensible.

En cas d'indisponibilité du modèle Ollama, le système utilise automatiquement des réponses locales générées à partir de la base de connaissances.

La figure suivante présente un exemple du prompt utilisé dans le système SecTalk.

```
186  summary_prompt = (
187      f"En 2 phrases simples en français, résume la situation pour ces attaques
188      f"Parle à un non-informaticien. Sois direct."
189  )
190  try:
191      response = ollama.chat(
192          model="phi3:mini",
193          messages=[
194              {"role": "system", "content": "Tu es un expert SOC. Réponds en 2 p
195              {"role": "user",   "content": summary_prompt},
196          ],
```

Figure III.5 : Exemple de prompt utilisé pour l'explication des attaques détectées

```
SYSTEM_PROMPT = """Tu es SecTalk, un assistant SOC en cybersecurite. Tu analyses des logs

Regles STRICTES :
- Explique uniquement ce qui est present dans le log. N'invente rien.
- Ne suppose pas une attaque ou compromission sauf si explicitement confirme.
- N'exagere pas la menace. Sois neutre et factuel.
- Un scan = reconnaissance uniquement, PAS une attaque.
- Un echec de connexion = suspect, PAS une compromission.
- Un compte cree = action admin, PAS malveillant par defaut.
- Un malware en quarantaine = deja contenu, focalise sur l'investigation.
- Si une information manque, ecris : "Non indique dans le log"

Classification de severite :
- Faible : evenement informatif, action admin normale
- Moyen : activite suspecte, scan, echec connexion
- Eleve : malware detecte, attaque repetee, indicateur fort
- Critique : intrusion active explicitement confirmee

Reponds TOUJOURS en francais. Suis ce format exactement :

EXPLICATION : [description claire basee uniquement sur le log]
DANGER : [Faible / Moyen / Eleve / Critique]
CAUSES POSSIBLES : [causes plausibles basees sur le log]
ACTION 1 : [etape d'investigation non destructive]
ACTION 2 : [surveillance ou verification complementaire]
CONCLUSION : [resume neutre - suspect / normal / a investiguer]"""
```

Figure III.6 : Exemple de prompt utilisé par le chatbot intelligent

La figure présente un exemple de prompt envoyé au modèle de langage. Celui-ci contient les informations relatives à l'alerte analysée ainsi que les instructions permettant de guider la génération de la réponse.

III.7.4 Fonctionnement du chatbot intelligent

Le chatbot intelligent intégré au système SecTalk permet aux utilisateurs d'obtenir des explications automatiques concernant les alertes et les incidents de cybersécurité.

L'utilisateur peut saisir une alerte de sécurité ou un message suspect directement dans l'interface du système. Cette entrée est d'abord analysée afin d'identifier les informations pertinentes associées à l'incident. Le système exploite ensuite une base de connaissances spécialisée en cybersécurité pour récupérer des informations contextuelles relatives à l'alerte fournie.

Les informations extraites sont intégrées au prompt envoyé au modèle de langage exécuté via Ollama, permettant ainsi de générer des réponses plus précises et mieux adaptées au contexte de l'incident.

Le chatbot fournit principalement :

- Une explication simplifiée de l'alerte ;
- Les risques potentiels associés ;
- Des recommandations de sécurité adaptées.

Cette fonctionnalité permet d'aider les utilisateurs à mieux comprendre les incidents de cybersécurité et à identifier les actions appropriées à entreprendre.

III.8 Développement de l'interface utilisateur du système

L'interface utilisateur du système SecTalk a été développée afin de faciliter l'analyse des alertes réseau et l'interaction avec le chatbot intelligent. Elle permet aux utilisateurs de charger des fichiers CSV, visualiser les attaques détectées et obtenir des explications automatiques en langage naturel.

III.8.1 Technologies utilisées

L'interface du système a été développée à l'aide des technologies suivantes :

- **HTML** : structure des pages web ;
- **CSS** : mise en forme et design de l'interface ;
- **FastAPI** : communication entre l'interface utilisateur et les modules d'analyse développés en Python.

Ces technologies permettent de créer une interface légère, rapide et facilement utilisable.

III.8.2 Interface principale du système

La figure suivante présente l'interface principale de SecTalk. Elle regroupe les différentes fonctionnalités du système

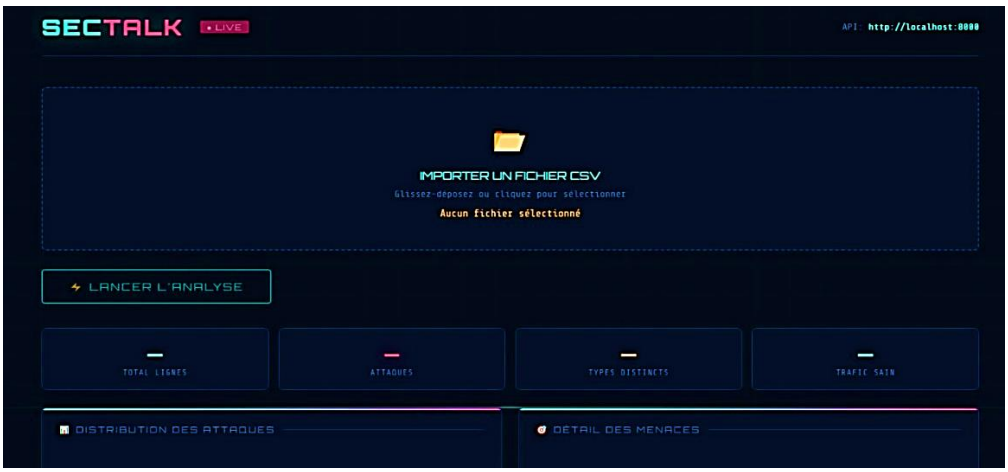

Figure III.7 : Interface principale du système SecTalk

III.8.3 Module d'analyse des fichiers CSV

Le module d'analyse permet à l'utilisateur de charger un fichier CSV contenant les caractéristiques du trafic réseau. Une fois le fichier importé, le système applique automatiquement le modèle LightGBM afin d'identifier les attaques présentes dans les données.

Figure III.8 : Importation d'un fichier CSV

Après l'analyse, les résultats de détection sont affichés directement dans l'interface.

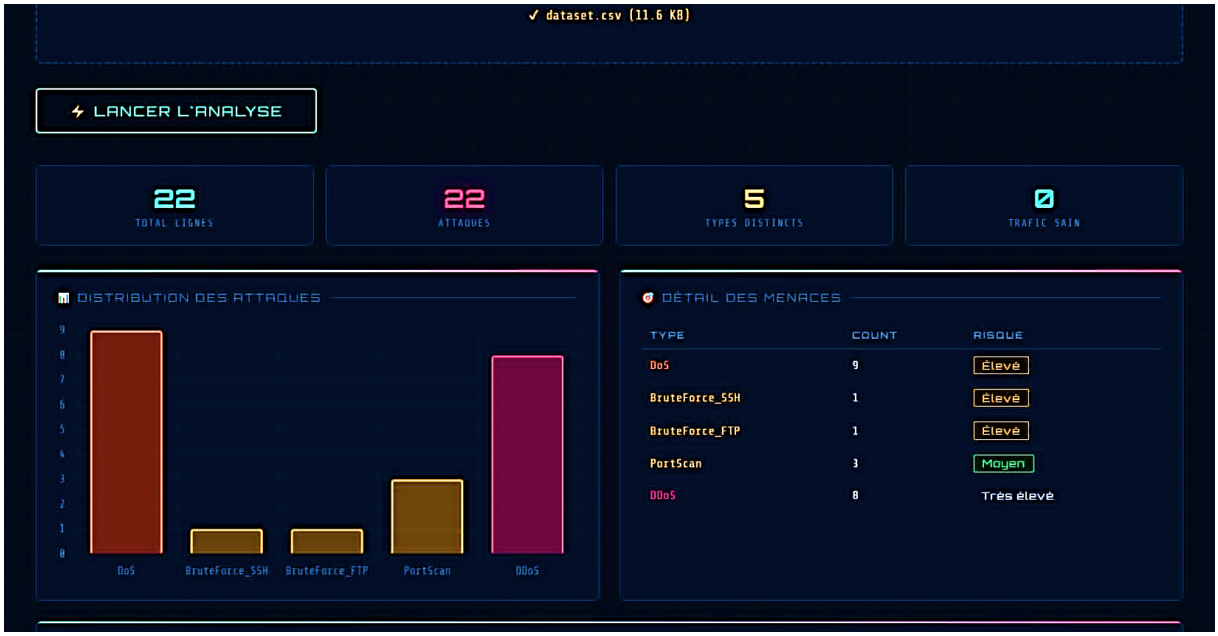

Figure III.9 : Résultats de détection des attaques

III.8.4 Génération des explications intelligentes

Lorsque des attaques sont détectées, le système génère automatiquement une explication en langage naturel à l'aide du modèle Phi3-mini exécuté via Ollama.

Les explications fournissent des informations sur la nature des attaques détectées ainsi que des recommandations de sécurité adaptées.

Figure III.10 : Explication automatique des attaques détectées

III.8.5 Chatbot intelligent de cybersécurité

SecTalk intègre également un chatbot permettant d'interpréter des alertes ou des événements de sécurité saisis par l'utilisateur.

Le chatbot analyse la requête puis génère une réponse explicative afin d'aider à comprendre l'incident signalé.

Figure III.11 : Interface du chatbot intelligent

Figure III.12 : Exemple de réponse générée par le chatbot

III.9. Résultats expérimentaux

Cette section présente les résultats obtenus après l'entraînement et l'évaluation des différents modèles de détection utilisés dans le système SecTalk. Les performances des modèles ont été analysées à l'aide de plusieurs métriques d'évaluation afin de comparer leur efficacité dans la détection des attaques réseau.

III.9.1 Métriques d'évaluation

Afin d'évaluer les performances des modèles de classification, plusieurs métriques couramment utilisées en apprentissage automatique ont été retenues.

***Accuracy* :** représente la proportion de prédictions correctes parmi l'ensemble des observations.

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

Où :

- **TP (True Positives)** : attaques correctement détectées
- **TN (True Negatives)** : trafics normaux correctement identifiés
- **FP (False Positives)** : trafics normaux classés comme attaques
- **FN (False Negatives)** : attaques non détectées.

***Precision* :** mesure la capacité du modèle à limiter les fausses alertes. Elle correspond à la proportion des prédictions positives réellement correctes.

$$Precision = \frac{TP}{TP + FP}$$

Une valeur élevée indique que la majorité des attaques signalées par le modèle sont effectivement des attaques.

Recall : mesure la capacité du modèle à détecter les attaques présentes dans les données.

$$Recall = \frac{TP}{TP + FN}$$

Un Recall élevé signifie que peu d'attaques échappent au système de détection.

***F1-score* :** est la moyenne harmonique de la Precision et du Recall.

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

Cette métrique permet d'obtenir une évaluation équilibrée des performances du modèle, notamment lorsque les classes sont déséquilibrées.

Dans ce travail, le **F1-score macro** a été privilégié, car il accorde la même importance à toutes les classes du dataset CIC-IDS2017, indépendamment de leur fréquence.

III.9.2 Résultats du modèle Regression Logistic

La régression logistique a été utilisée comme modèle de référence pour la classification des attaques réseau.

Les résultats obtenus montrent une accuracy élevée de 99 %, mais un F1-score macro de 79,57 %. Cette différence s'explique par le déséquilibre des classes dans le dataset, certaines attaques étant beaucoup moins représentées que d'autres.

Le score moyen obtenu lors de la validation croisée est de :

CV F1 Macro = 0,7957

Ces résultats montrent que la régression logistique présente des limites pour la détection de certaines classes minoritaires.

```
=== CLASSIFICATION REPORT ===
              precision    recall  f1-score   support

           0       1.00      0.99      1.00    242188
           1       0.43      0.98      0.60       147
           2       1.00      1.00      1.00       814
           3       1.00      1.00      1.00     19029
           4       0.98      1.00      0.99      1513
           5       0.50      1.00      0.67        16
           6       1.00      1.00      1.00     31694
           7       0.88      0.99      0.93       116
           8       0.97      0.99      0.98       348
           9       1.00      1.00      1.00       674
          10       0.93      1.00      0.96       772
          11       0.99      0.99      0.99       369
          12       1.00      1.00      1.00       794
          13       0.06      0.50      0.11         2
          14       1.00      1.00      1.00         2
          15       0.02      1.00      0.04         7
          16       1.00      1.00      1.00         9
          17       0.95      0.98      0.96     14354
          18       0.99      1.00      1.00     31813
          19       0.99      1.00      0.99       592
          20       0.83      1.00      0.91         5
          21       0.79      1.00      0.88        15
          22       0.65      0.56      0.60       258
          23       0.11      1.00      0.20         3
          24       0.00      0.00      0.00         1
          25       1.00      1.00      1.00         4
          26       0.41      0.66      0.51       131

    accuracy                           0.99    345670
   macro avg       0.76      0.91      0.79    345670
weighted avg       1.00      0.99      0.99    345670
```

Figure III.13 : Rapport de classification de modèle Regression Logistic

III.9.3 Résultats du modèle Random Forest

Le modèle Random Forest a permis d'améliorer significativement les performances de détection.

Les résultats obtenus montrent une excellente capacité de classification avec un score moyen de validation croisée égal à :

CV F1 Macro = 0,9154

Le modèle détecte correctement la majorité des attaques et offre une meilleure robustesse face au déséquilibre des données.

```
============================================================
MODEL: Random Forest
============================================================
CV F1 Macro Mean: 0.9154
Model trained ✓
Prediction done ✓

=== CLASSIFICATION REPORT ===
              precision    recall  f1-score   support

           0       1.00      1.00      1.00    242188
           1       1.00      1.00      1.00       147
           2       1.00      1.00      1.00       814
           3       1.00      1.00      1.00     19029
           4       1.00      1.00      1.00      1513
           5       1.00      0.94      0.97        16
           6       1.00      1.00      1.00     31694
           7       1.00      0.99      1.00       116
           8       1.00      0.99      1.00       348
           9       1.00      1.00      1.00       674
          10       1.00      1.00      1.00       772
          11       1.00      0.99      0.99       369
          12       1.00      1.00      1.00       794
          13       1.00      1.00      1.00         2
          14       1.00      1.00      1.00         2
          15       1.00      1.00      1.00         7
          16       1.00      0.78      0.88         9
          17       1.00      1.00      1.00     14354
          18       1.00      1.00      1.00     31813
          19       1.00      1.00      1.00       592
          20       1.00      1.00      1.00         5
          21       1.00      0.93      0.97        15
          22       0.80      0.90      0.85       258
          23       1.00      0.67      0.80         3
          24       0.00      0.00      0.00         1
          25       1.00      1.00      1.00         4
          26       0.76      0.60      0.67       131

    accuracy                           1.00    345670
   macro avg       0.95      0.92      0.93    345670
weighted avg       1.00      1.00      1.00    345670
```

Figure III.14 : Rapport de classification de modèle Random Forest

III.9.4 Résultats du modèle LightGBM

Le modèle LightGBM a obtenu les meilleures performances parmi les algorithmes testés.

La validation croisée a permis d'obtenir un score moyen de :

CV F1 Macro = 0,9444

Les résultats montrent une excellente capacité de détection pour la majorité des classes ainsi qu'une meilleure prise en compte des attaques rares.

```
=== CLASSIFICATION REPORT ===
              precision    recall  f1-score   support

           0       1.00      1.00      1.00    242188
           1       1.00      1.00      1.00       147
           2       1.00      1.00      1.00       814
           3       1.00      1.00      1.00     19029
           4       1.00      1.00      1.00      1513
           5       1.00      0.94      0.97        16
           6       1.00      1.00      1.00     31694
           7       1.00      0.99      1.00       116
           8       1.00      0.99      1.00       348
           9       1.00      1.00      1.00       674
          10       1.00      1.00      1.00       772
          11       1.00      0.99      0.99       369
          12       1.00      1.00      1.00       794
          13       1.00      1.00      1.00         2
          14       1.00      1.00      1.00         2
          15       1.00      1.00      1.00         7
          16       1.00      1.00      1.00         9
          17       1.00      1.00      1.00     14354
          18       1.00      1.00      1.00     31813
          19       1.00      1.00      1.00       592
          20       1.00      1.00      1.00         5
          21       1.00      1.00      1.00        15
          22       0.85      0.88      0.86       258
          23       0.75      1.00      0.86         3
          24       0.00      0.00      0.00         1
          25       1.00      1.00      1.00         4
          26       0.74      0.73      0.73       131

    accuracy                           1.00    345670
   macro avg       0.94      0.94      0.94    345670
weighted avg       1.00      1.00      1.00    345670
```

Figure III.15 : Rapport de classification de modèle LightGBM

III.9.5 Comparaison des performances des modèles

Le tableau suivant résume les performances obtenues par les différents modèles.

Modèle	CV F1 Macro
Regression Logistic	0,7957
Random Forest	0,9154
LightGBM	0.9444

Tableau III.7 : Comparaison des performances des modèles

Les résultats montrent que le modèle LightGBM obtient les meilleures performances globales. Il dépasse Random Forest de près de 3 points de F1-score macro et surpasse largement la régression logistique.

Cette comparaison confirme que les méthodes de boosting sont particulièrement adaptées à la détection des intrusions dans le dataset CIC-IDS2017.

Pour cette raison, LightGBM a été retenu comme modèle principal du système SecTalk.

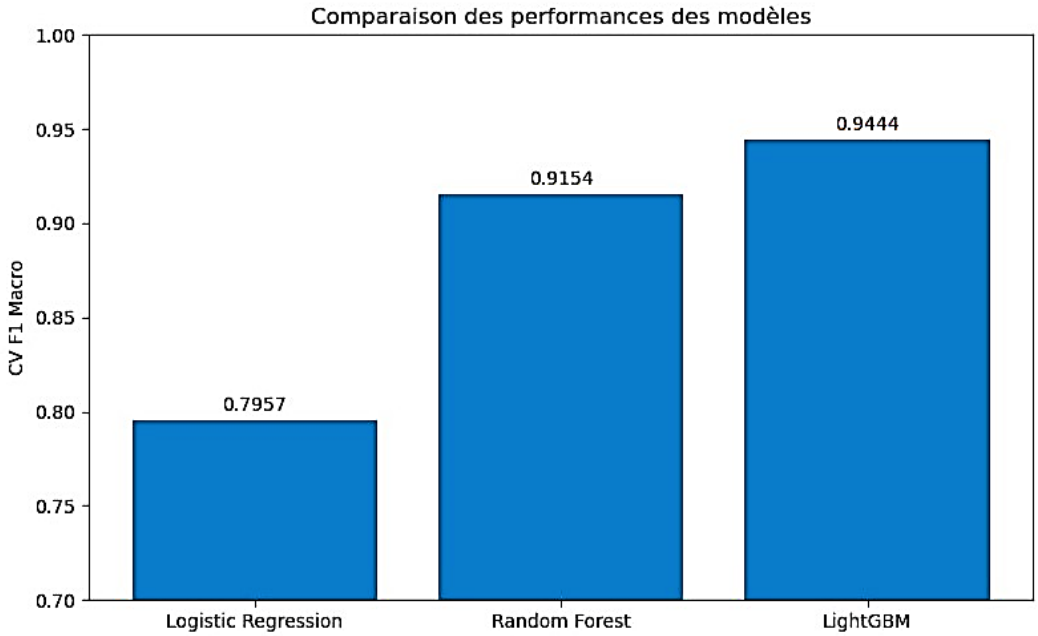

Figure III.16 : Comparaison des performances des modèles Regression Logistic, Random Forest et LightGBM selon le score F1 Macro obtenu lors de la validation croisée.

III.9.6 Optimisation du modèle LightGBM

Après la sélection de LightGBM comme modèle principal, plusieurs expérimentations ont été réalisées afin d'identifier la configuration offrant les meilleures performances.

III.9.6.1 Ajustement des hyperparamètres

Afin d'améliorer les performances du modèle LightGBM, plusieurs expérimentations ont été réalisées en faisant varier certains hyperparamètres influençant directement l'apprentissage du modèle. Les paramètres étudiés sont :

- n_estimators : nombre d'arbres de décision ;
- learning_rate : taux d'apprentissage ;
- num_leaves : nombre maximal de feuilles par arbre.

Le paramètre class_weight a été fixé à la valeur **balanced** afin de prendre en compte le déséquilibre des classes présent dans le dataset CIC-IDS2017.

III.9.6.2 Tests des différentes configurations

Plusieurs configurations ont été évaluées afin d'analyser leur impact sur les performances du modèle.

Configuration	n_estimators	learning_rate	num_leaves	CV F1 Macro
Config 1	100	0,10	31	0.945058
Config 2	200	0,05	31	0.947022
Config 3	300	0,05	31	0.944363
Config 4	400	0,03	50	0.942023

Tableau III.8 : Tests des différentes configurations

III.9.6.3 Sélection de la meilleure configuration

Les résultats montrent que la **Configuration 2** obtient le meilleur score de validation croisée avec un **F1-score macro de 94,70 %**.

Cette configuration correspond aux paramètres suivants :

- n_estimators = 200
- learning_rate = 0,05
- num_leaves = 31
- class_weight = balanced

Elle offre le meilleur compromis entre performance et complexité du modèle. Pour cette raison, elle a été retenue pour les expérimentations finales du système SecTalk.

III.9.7 Évaluation et comparaison des modèles de langage

Afin de sélectionner le modèle de langage le plus adapté au module d'explication intelligent de SecTalk, plusieurs modèles exécutés localement via Ollama ont été évalués dans les mêmes conditions matérielles, sur une machine fonctionnant uniquement avec le processeur (CPU), sans accélération GPU.

Les modèles testés sont :

- **Phi-3.**
- **Phi-3 Mini.**
- **TinyLlama**.

L'évaluation a été réalisée sur deux scénarios représentatifs du système :

- L'explication des attaques détectées lors de l'analyse d'un fichier CSV.
- L'explication d'alertes de sécurité via le chatbot intelligent.

Les critères retenus sont le temps d'exécution, la quantité de texte générée et la qualité globale des réponses produites.

Résultats expérimentaux

Modèle	Analyse Dataset	Chatbot	Mots générés	Qualité
Phi3	55.2 s	39.3 s	~64 mots	Très élevée
Phi3-mini	41.5 s	42,9 s	~67 mots	Élevée
TinyLlama	9.3 s	8,2 s	~24 mots	Limitée

Tableau III.9 : Comparaison des temps d'exécution des modèles de langage testés.

Les résultats montrent que **TinyLlama** est le modèle le plus rapide, avec un temps de réponse inférieur à **dix secondes** dans les deux scénarios. Cependant, cette rapidité s'accompagne d'une diminution notable de la qualité et de la richesse des réponses générées.

À l'inverse, Phi3 et Phi3-mini nécessitent davantage de temps de calcul mais produisent des explications plus détaillées, plus cohérentes et mieux adaptées au contexte de la cybersécurité.

L'analyse détaillée des réponses générées et la justification du choix final du modèle sont présentées dans la section 3.10.2.

```python
import ollama
import time

# ==========================
# PROMPT DE TEST
# ==========================
TEST_PROMPT_DATASET = """Résume en 2 phrases simples en français pourquoi ces attaques réseau sont dangereuses : DDoS, DoS, BruteForce_SSH, PortScan. Parle à quelqu'un qui ne connaît pas l'informatique."""

TEST_PROMPT_CHATBOT = """Alerte : EventID=4672. IP=185.23.44.12. Port=4444. Message=Suspicious outbound connection detected.
Analyse en français : danger, explication simple, 2 actions."""

MODELS = ["phi3", "phi3:mini", "tinyllama"]


# ==========================
# FONCTION DE MESURE
# ==========================
def benchmark(model, prompt, options, label):
    print(f"\n{'='*55}")
    print(f"  Modèle : {model}  |  Test : {label}")
    print(f"{'='*55}")

    try:
        start = time.time()
        response = ollama.chat(
            model=model,
            messages=[
                {"role": "system", "content": "Tu es un expert SOC. Réponds en français. Sois concis."},
                {"role": "user",   "content": prompt}
```

Figure III.17 : Prompt utilisé pour l'évaluation comparative des modèles Phi3, Phi3-mini et TinyLlama

III.9.8 Choix du modèle de langage final

Les résultats obtenus montrent que Phi3 et Phi3-mini offrent des performances très proches en matière de génération de texte. Les deux modèles produisent des explications cohérentes, structurées et adaptées au contexte de la cybersécurité, tout en respectant les consignes définies dans les prompts du système.

Bien que Phi3 fournisse des réponses légèrement plus détaillées, l'amélioration observée reste limitée au regard des ressources supplémentaires nécessaires à son exécution. En effet, Phi3 consomme davantage de mémoire et présente un temps d'exécution légèrement supérieur à celui de Phi3-mini.

De son côté, TinyLlama offre les meilleurs temps de réponse mais présente plusieurs limitations importantes. Les tests réalisés ont montré que ce modèle ne respecte pas toujours les instructions du prompt et génère parfois des réponses incomplètes ou peu pertinentes. Ces limitations réduisent son intérêt dans un contexte d'assistance à la cybersécurité où la précision des explications est essentielle.

Au terme de cette comparaison, **Phi3-mini** a été retenu comme modèle de langage du système SecTalk. Ce choix est justifié par :

- Sa capacité à générer des explications claires et cohérentes ;
- Son respect des consignes définies dans les prompts ;
- Son temps d'exécution acceptable sur CPU ;
- Son bon compromis entre qualité de génération et performances.

Ainsi, Phi3-mini représente le meilleur compromis entre qualité des explications, performances d'exécution et ressources matérielles disponibles. Il a donc été intégré dans la version finale du système SecTalk pour la génération automatique d'explications et le fonctionnement du chatbot intelligent.

III.10. Analyse et discussion des résultats

Cette section présente une analyse critique des résultats obtenus lors de l'évaluation du système SecTalk. L'objectif est d'interpréter les performances du module de détection d'intrusion ainsi que celles du module d'explication intelligent afin de justifier les choix technologiques retenus dans la version finale du système.

III.10.1 Analyse des performances des modèles de détection

III.10.1.1 Comparaison des modèles évalués

Afin d'identifier le modèle de détection le plus performant, trois algorithmes de machine learning ont été évalués sur le dataset CIC-IDS2017 : Logistic Regression, Random Forest et LightGBM.

Les résultats expérimentaux montrent que les trois modèles atteignent une excellente accuracy globale. Toutefois, l'analyse du F1-score macro met en évidence des différences significatives entre les algorithmes.

Le modèle Logistic Regression obtient un F1-score macro de 0,7957. Bien qu'il fournisse de bonnes performances sur les classes majoritaires, ses résultats sont moins satisfaisants pour certaines classes minoritaires du dataset.

Le modèle Random Forest améliore considérablement les performances avec un F1-score macro de 0,9154. Cette amélioration s'explique par sa capacité à exploiter des relations non linéaires entre les caractéristiques du trafic réseau.

Le meilleur résultat est obtenu par LightGBM avec un F1-score macro de 0,9444. Ce modèle démontre une meilleure capacité de généralisation ainsi qu'une meilleure prise en compte des classes rares présentes dans le dataset.

Ces résultats confirment la supériorité des méthodes basées sur les arbres de décision pour les tâches de détection d'intrusion.

III.10.1.2 Comparaison des algorithmes

La comparaison des modèles évalués met en évidence des différences significatives en termes de capacité de détection des attaques réseau.

Le modèle Logistic Regression présente les performances les plus faibles parmi les modèles testés. Cette limitation s'explique principalement par sa nature linéaire, qui ne lui permet pas de capturer efficacement les relations complexes existant entre les différentes caractéristiques du trafic réseau.

Le modèle Random Forest offre de meilleures performances grâce à l'utilisation d'un ensemble d'arbres de décision construits de manière aléatoire. Cette approche permet d'améliorer la robustesse du modèle et de réduire les risques de surapprentissage.

Le modèle LightGBM obtient les meilleurs résultats grâce à son mécanisme de Gradient Boosting. Contrairement à Random Forest, qui construit les arbres de manière indépendante, LightGBM crée progressivement chaque nouvel arbre afin de corriger les erreurs commises par les arbres précédents. Cette stratégie permet d'améliorer la précision globale du modèle et d'optimiser la détection des différentes catégories d'attaques.

Les résultats obtenus confirment ainsi que LightGBM constitue l'algorithme le plus performant pour le contexte étudié.

III.10.1.3 Influence de l'optimisation de LightGBM

Après avoir identifié LightGBM comme le modèle le plus performant, plusieurs configurations ont été testées afin d'optimiser ses hyperparamètres.

Les paramètres évalués concernent principalement :

- Le nombre d'arbres (*n_estimators*) ;
- Le taux d'apprentissage (*learning_rate*) ;
- Le nombre maximal de feuilles (*num_leaves*).

Les expérimentations ont montré que la configuration suivante offre les meilleurs résultats :

- n_estimators = 200 ;
- learning_rate = 0.05 ;

- num_leaves = 31.

Cette configuration a obtenu un F1-score macro de 0,9470 lors de la validation croisée, démontrant un bon équilibre entre précision et capacité de généralisation.

L'optimisation des hyperparamètres a ainsi permis d'identifier la configuration la plus adaptée aux caractéristiques du dataset étudié.

III.10.1.4 Justification du choix de LightGBM

Au regard des résultats obtenus, LightGBM a été retenu comme moteur de détection du système SecTalk.

Ce choix repose sur plusieurs facteurs :

- Meilleur F1-score macro parmi les modèles évalués.
- Bonne gestion du déséquilibre des classes.

Ainsi, LightGBM constitue la solution la plus adaptée pour assurer une détection fiable des attaques réseau dans le cadre de ce projet.

III.10.2 Analyse des modèles de langage

III.10.2.1 Comparaison de Phi3, Phi3-mini et TinyLlama

Afin de sélectionner le modèle de langage le plus adapté au système d'explication intelligent, plusieurs modèles exécutés localement via Ollama ont été évalués.

Les résultats obtenus montrent que Phi3 et Phi3-mini produisent des réponses de qualité similaire avec des temps d'exécution relativement proches.

À l'inverse, TinyLlama génère des réponses plus rapidement mais avec un niveau de qualité inférieur. Les explications obtenues sont généralement plus courtes, moins détaillées et parfois moins pertinentes pour le contexte de cybersécurité étudié.

Ces observations mettent en évidence l'existence d'un compromis entre vitesse d'exécution et qualité des réponses générées.

III.10.2.2 Justification du choix de Phi3-mini

Bien que Phi3 fournisse des réponses légèrement plus détaillées, les écarts observés restent limités dans le cadre de notre application.

Phi3-mini présente plusieurs avantages :

- ✓ Qualité de génération satisfaisante ;

- ✓ Consommation mémoire plus faible ;
- ✓ Temps d'exécution acceptable sur CPU ;
- ✓ Bonne cohérence des explications produites.

Pour cette raison, il a été retenu comme modèle de langage final du système SecTalk.

III.10.3 Limites du système

Malgré les résultats encourageants obtenus, certaines limites doivent être prises en considération.

Premièrement, les performances du module de détection dépendent fortement de la qualité et de la diversité des données utilisées lors de l'apprentissage. L'apparition de nouvelles formes d'attaques peut affecter la capacité du modèle à généraliser correctement.

Deuxièmement, le temps nécessaire à la génération des explications reste relativement élevé lorsque les modèles de langage sont exécutés uniquement sur processeur.

Troisièmement, la qualité des explications associées aux attaques détectées dépend directement des informations présentes dans la base de connaissances utilisée par le système.

Enfin, la version actuelle de SecTalk réalise l'analyse de fichiers CSV et ne prend pas encore en charge l'analyse temps réel des flux réseau.

III.10.4 Discussion générale

Les résultats obtenus démontrent la pertinence de l'approche adoptée dans le cadre du système SecTalk. L'intégration d'un modèle de détection d'intrusion basé sur la machine learning avec un module d'explication intelligent permet d'améliorer à la fois la précision de détection et la compréhension des résultats générés.

Les expérimentations ont montré que le modèle LightGBM offre d'excellentes performances pour la classification des attaques réseau, tandis que le modèle de langage Phi3-mini permet de fournir des explications claires et des recommandations adaptées aux événements détectés.

Contrairement aux systèmes IDS traditionnels qui se limitent à signaler la présence d'une menace, SecTalk propose également une interprétation des attaques détectées ainsi que des conseils pratiques permettant à l'utilisateur de mieux comprendre la situation et d'adopter les mesures de sécurité appropriées.

Cette combinaison entre détection automatique et génération d'explications en langage naturel constitue l'une des principales contributions de ce travail et contribue à rendre les systèmes de cybersécurité plus accessibles aux utilisateurs débutants.

III.11 Conclusion

Dans ce chapitre, nous avons présenté la conception et la réalisation du système **SecTalk**. Après la préparation du dataset CIC-IDS2017, plusieurs modèles de machine learning ont été évalués afin d'identifier la solution la plus performante pour la détection des intrusions.

Les résultats expérimentaux ont montré que **LightGBM** obtient les meilleures performances comparativement à Logistic Regression et Random Forest. Ce modèle a donc été retenu pour assurer la détection des attaques au sein du système.

Nous avons également présenté le module d'explication intelligent basé sur un modèle de langage exécuté via Ollama. Après comparaison de plusieurs modèles, **Phi3-mini** a été sélectionné en raison de son bon compromis entre qualité des réponses et temps d'exécution.

Enfin, l'évaluation globale du système a montré qu'il est capable de détecter efficacement les attaques réseau tout en fournissant des explications et des recommandations compréhensibles pour les utilisateurs. Ces résultats confirment la pertinence de l'approche proposée et ouvrent la voie à plusieurs perspectives d'amélioration qui seront présentées dans la conclusion générale.

Conclusion générale

Conclusion générale

L'objectif principal de ce mémoire était de concevoir et de développer un système intelligent capable de détecter les attaques réseau et de fournir des explications compréhensibles des alertes de sécurité. Pour répondre à cet objectif, nous avons proposé **SecTalk**, une solution hybride combinant des techniques d'apprentissage automatique pour la détection d'intrusions et des modèles de langage pour la génération automatique d'explications en langage naturel.

Dans un premier temps, une étude approfondie des concepts fondamentaux relatifs aux systèmes de détection d'intrusion, au machine learning et aux modèles de langage a été réalisée. Cette analyse a permis de mieux comprendre les limites des approches classiques et d'identifier les méthodes les plus adaptées à la problématique de détection et d'explication des cyberattaques.

Le système a été développé et évalué sur le dataset **CIC-IDS2017**, contenant différents types d'attaques ainsi que du trafic réseau légitime. Après les étapes de préparation et de prétraitement des données, plusieurs algorithmes de classification ont été expérimentés, notamment la **Logistic Regression**, le **Random Forest** et **LightGBM**. Les résultats obtenus ont montré que **LightGBM** offre les meilleures performances, ce qui a justifié son adoption comme modèle principal de détection.

Par ailleurs, un module d'explication intelligent a été intégré afin de rendre les résultats du système plus interprétables. Ce module repose sur une base de connaissances spécialisée dans les attaques réseau ainsi que sur un modèle de langage local déployé via **Ollama**. L'évaluation comparative de plusieurs modèles (Phi3, Phi3-mini et TinyLlama) a montré que **Phi3-mini** constitue le meilleur compromis entre qualité des réponses, ressources utilisées et temps d'exécution, et a été retenu pour la version finale du système.

Les expérimentations réalisées ont confirmé que SecTalk permet à la fois une détection efficace des intrusions et une génération automatique d'explications claires et accessibles. Cette combinaison entre détection automatique et explicabilité constitue une contribution importante de ce travail, en améliorant significativement la compréhension des alertes de sécurité et en facilitant l'aide à la décision.

Perspectives

Plusieurs pistes d'amélioration peuvent être envisagées afin de renforcer les performances et l'évolutivité du système SecTalk.

L'intégration d'un mécanisme d'analyse des flux réseau en temps réel constituerait une évolution majeure, permettant une détection plus réactive des attaques et une meilleure adaptation aux environnements dynamiques.

L'enrichissement de la base de connaissances dédiée aux attaques de cybersécurité permettrait d'élargir la couverture des scénarios de sécurité et d'améliorer la précision des explications générées.

De plus, l'utilisation de jeux de données plus récents et plus diversifiés contribuerait à améliorer la capacité de généralisation du modèle de détection face à l'évolution des menaces.

Concernant le module d'explication, l'approche de type **Retrieval-Augmented Generation (RAG) simplifiée**, basée sur une base de connaissances interne, pourrait être améliorée en adoptant des techniques de récupération sémantique plus avancées (embeddings, bases vectorielles, indexation optimisée). Cela permettrait de réduire davantage les hallucinations et d'améliorer la pertinence des réponses générées.

Bibliographie

Bibliographie

[1] Nah, F. F.-H., Zheng, R., Cai, J., Siau, K., & Chen, L. (2023). *Generative AI and ChatGPT: Applications, challenges, and AI-human collaboration*. *Journal of Information Technology Case and Application Research, 25*(3), 277–304. https://doi.org/10.1080/15228053.2023.2233814

[2] ResearchGate. (2023). *A survey of generative AI applications*. https://www.researchgate.net/publication/371311926_A_survey_of_Generative_AI_Applications

[3] Nadkarni, P. M., Ohno-Machado, L., & Chapman, W. W. (2011). Natural language processing: An introduction. *Journal of the American Medical Informatics Association, 18*(5), 544–551. https://doi.org/10.1136/amiajnl-2011-000464

[4] Pestian, J., Deléger, L., Savova, G., Dexheimer, J. W., & Solti, I. (s.d.). *Natural language processing: The basics*.

[5] Tunstall, L., von Werra, L., & Wolf, T. (2022). *Natural language processing with transformers: Building language applications with Hugging Face*. O'Reilly Media.

[6] Université de Paris. (2020). *Chapitre 12 : Génération automatique de textes*. Dans *Introduction au traitement automatique du langage naturel*. Presses Universitaires de France.

[7] Association pour le Traitement Automatique des Langues (ATALA). (1997–2019). *Corpus TALN : Recueil des actes des conférences TALN et RECITAL*. ATALA.

[8] JEP-TALN-RECITAL. (2024). *Actes de la conférence conjointe JEP-TALN-RECITAL*. ATALA.

[9] Jurafsky, D., & Martin, J. H. (2023). *Speech and language processing* (3rd ed., draft). Stanford University.

[10] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems, 30*.

[11] Jurafsky, D., & Martin, J. H. (2023). *Speech and language processing*. Stanford University.

[12] ORSYS. (s.d.). *Modèle de langage*. https://www.orsys.fr/orsys-lemag/Glossaire/modele-de-langage/

[13] Manning, C. D., & Schütze, H. (1999). *Foundations of statistical natural language processing*. MIT Press.

[14] Hassanpour, R. (2024). *From statistical models to LLMs*.

[15] Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003). A neural probabilistic language model. *Journal of Machine Learning Research, 3*, 1137–1155.

[16] Chu, Q. (2024). *History, development, and principles of LLMs*.

[17] Tunstall, L., von Werra, L., & Wolf, T. (2022). *Natural language processing with transformers: Building language applications with Hugging Face*. O'Reilly Media.

[18] Databricks. (s.d.). *What are large language models?* https://www.databricks.com/fr/blog/what-are-large-language-models

[19] Qin, L., Chen, Q., Feng, X., Wu, Y., Zhang, Y., Li, Y., Li, M., Che, W., & Yu, P. S. (2026). Large language models meet NLP: A survey. *Frontiers of Computer Science, 20*, 2011361.

[20] Swimlane. (s.d.). *Comment l'IA générative peut-elle être utilisée en cybersécurité ?* https://swimlane.com/fr/blog/comment-lia-generative-peut-elle-etre-utilisee-en-cybersecurite/

[21] Kemp, D. (2025). *What if generative AI is reaching its limits?* European Parliamentary Research Service (EPRS), Scientific Foresight Unit (STOA), PE 774.701.

[22] Carmatec. (s.d.). *Modèles génératifs et modèles discriminatifs : lequel utiliser ?* https://www.carmatec.com/fr_fr/blog/modeles-generatifs-et-modeles-discriminatifs-lequel-utiliser/

[23] DataCamp. (s.d.). *How transformers work*. https://www.datacamp.com/fr/tutorial/how-transformers-work

[24] Zuech, R., Khoshgoftaar, T. M., & Wald, R. (2015). Intrusion detection and big heterogeneous data: A survey. *Journal of Big Data, 2*(3).

[25] Ahmad, A., Hadgkiss, J., & Ruighaver, A. (s.d.). Incident response teams: Challenges in alert management. *Computers & Security*.

[26] Mirheidari, S. A., Arshad, S., & Jalili, R. (2018). *Alert correlation algorithms: A survey and taxonomy*. arXiv.

[27] Bateni, M., & Baraani, A. (2014). An architecture for alert correlation inspired by a comprehensive model of human immune system. *International Journal of Computer Network and Information Security, 6*(12), 47–57.

[28] Ogunbadejo, M., Ayilara-Adewale, O. A., & Alade, O. (2025). Machine learning methods for intrusion detection: A comprehensive survey. *International Journal of Scientific Research and Management, 13*(7), 2446–2456.

[29] Salvio, A. (2021). *Natural-scalaron inflation*. arXiv.

[30] Xu, H., Wang, S., Li, N., Wang, K., Zhao, Y., Chen, K., Yu, T., Liu, Y., & Wang, H. (2024). *Large language models for cyber security: A systematic literature review*. arXiv.

[31] Guan, W., Cao, J., Qian, S., Gao, J., & Ouyang, C. (2023). *LogLLM: Log-based anomaly detection using large language models*. arXiv.

[32] Zhang, Z., Al Hamadi, H., Damiani, E., Yeun, C. Y., & Taher, F. (2022). Explainable artificial intelligence applications in cyber security: State-of-the-art in research. *IEEE Access, 10*, 123624–123647.

[33] González-Granadillo, G., González-Zarzosa, S., & Diaz, R. (2021). Security information and event management (SIEM): Analysis, trends, and usage in critical infrastructures. *Sensors, 21*(14), 4759. https://doi.org/10.3390/s21144759

[34] Doshi-Velez, F., & Kim, B. (2017). *Towards a rigorous science of interpretable machine learning*. arXiv. https://arxiv.org/abs/1702.08608

[35] Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., Du, Y., Yang, C., Chen, Y., Chen, Z., Jiang, J., Ren, R., Li, Y., Tang, X., Liu, Z., Liu, P., Nie, J.-Y., & Wen, J.-R. (2023). *A survey of large language models*. arXiv. https://arxiv.org/abs/2303.18223

ملخص

أصبحت أنظمة الأمن المعلوماتي، مع التزايد المستمر للتهديدات السيبرانية والتعقيد المتزايد للبنى التحتية الرقمية، أمراً بالغ الأهمية. إلا أن هذه التنبيهات غالباً ما تكون تقنية ومعقدة، مما يجعل فهمها واستغلالها أمراً صعباً بالنسبة للمستخدمين. وقد يهدف تحسين عملية كشف الهجمات وفهم التنبيهات الأمنية إلى تحليل سريع ودقيق للتنبيهات التي تتطلب تحليلاً سريعاً. في هذا السياق، يقدم هذا العمل نظاماً ذكياً يسمى SecTalk، يعتمد على دمج تقنيات التعلم الآلي للكشف عن الهجمات الشبكية مع تقنيات الذكاء الاصطناعي التوليدي لإنتاج شروحات بلغة طبيعية تساعد على فهم التنبيهات بشكل أفضل.

تعتمد المقاربة المقترحة على توليد تفسيرات سياقية انطلاقاً من قاعدة معرفة متخصصة في الأمن السيبراني. يعتمد النظام على سلسلة من المراحل تشمل معالجة البيانات وتصنيف الأحداث الأمنية وتوليد تفسيرات سياقية. وتساهم هذه المقاربة في تحويل التنبيهات التقنية إلى معلومات أكثر وضوحاً وسهولة في الفهم بالنسبة للمستخدمين، مما يسهل إدارة التنبيهات وتحسين تفسيرها. يبرز هذا العمل أهمية الجمع بين التعلم الآلي والنماذج اللغوية في مجال الأمن السيبراني من أجل تحسين تفسير التنبيهات الأمنية.

الكلمات المفتاحية: الأمن السيبراني، كشف التسلل، الذكاء الاصطناعي التوليدي، التعلم الآلي، النماذج اللغوية، تفسير التنبيهات، الحوادث الأمنية، RAG

Abstract

With the continuous growth of cyber threats and the increasing complexity of digital infrastructures, security systems generate a large number of alerts that require rapid and accurate analysis. However, these alerts are often highly technical and difficult to interpret, making their exploitation and decision-making processes more challenging for users. In this context, this thesis presents SecTalk, an intelligent system designed to improve both the detection and understanding of security alerts. The proposed approach combines machine learning techniques for network attack detection with generative artificial intelligence techniques for producing natural language explanations. The system relies on a complete process that includes data preparation, security event classification, and the generation of contextualized explanations based on a cybersecurity knowledge base. This approach transforms technical alerts into more accessible and understandable information for users. This work highlights the value of combining machine learning and language models in cybersecurity to enhance alert interpretation and facilitate security incident management.

Keywords: Cybersecurity, Intrusion Detection, Generative Artificial Intelligence, Machine Learning, Language Models, Alert Explanation, RAG.

Résumé

Face à l'augmentation constante des cybermenaces et à la complexité croissante des infrastructures numériques, les systèmes de sécurité génèrent un volume important d'alertes qui nécessitent une analyse rapide et précise. Cependant, ces alertes sont souvent techniques et difficiles à interpréter, ce qui complique leur exploitation et la prise de décision par les utilisateurs. Dans ce contexte, ce mémoire propose SecTalk, un système intelligent visant à améliorer la détection et la compréhension des alertes de sécurité. L'approche adoptée combine des techniques d'apprentissage automatique pour l'identification des attaques réseau et des techniques d'intelligence artificielle générative pour la production d'explications en langage naturel. Le système s'appuie sur un processus complet incluant la préparation des données, la classification des événements de sécurité et la génération d'explications contextualisées à partir d'une base de connaissances spécialisée en cybersécurité. Cette approche permet de transformer des alertes techniques en informations plus accessibles et compréhensibles pour les utilisateurs. Ce travail met ainsi en évidence l'intérêt de l'intégration conjointe du machine learning et des modèles de langage dans le domaine de la cybersécurité, afin d'améliorer l'interprétation des alertes et de faciliter la gestion des incidents de sécurité.

Mots-clés : Cybersécurité, Détection d'intrusion, Intelligence Artificielle Générative, Machine Learning, Modèles de Langage, Explication des alertes, RAG.