UQAC
Université du Québec
à Chicoutimi

ADAPTATION D'UN SYSTÈME DE CYBERSÉCURITÉ SELON LE PROFIL AFFECTIF DE L'UTILISATEUR

PAR BRICE CHABI

MÉMOIRE PRÉSENTÉ À L'UNIVERSITÉ DU QUÉBEC À CHICOUTIMI EN VUE DE L'OBTENTION DU GRADE DE MAÎTRISE ÈS SCIENCES (M. SC.) EN INFORMATIQUE

QUÉBEC, CANADA

RÉSUMÉ

Ce mémoire de recherche s'inscrit dans le domaine de la cybersécurité adaptative et examine la possibilité d'intégrer le facteur humain comme variable active dans le processus de décision d'un système de protection. Dans la plupart des architectures de sécurité actuelles, les mécanismes de défense reposent sur des politiques statiques qui s'appliquent de manière uniforme, indépendamment de l'état de vigilance de l'utilisateur. Cette approche présente une limite importante, car elle néglige le fait qu'un opérateur humain peut voir sa capacité de discernement diminuer sous l'effet du stress, de la fatigue ou d'une surcharge cognitive. Or, dans des environnements sensibles, une simple erreur de manipulation peut suffire à compromettre la stabilité d'une infrastructure ou à déclencher un incident de sécurité. Ce mémoire part donc du constat que la sécurité informatique ne peut plus être pensée uniquement comme une affaire de contrôle d'accès ou de filtrage réseau, mais qu'elle doit aussi intégrer l'état cognitif de l'utilisateur comme dimension pertinente de l'évaluation du risque.

L'objectif général du sujet de recherche est de concevoir et de formaliser une architecture logicielle capable d'adapter son comportement en fonction d'un profil affectif simulé, afin de soutenir l'opérateur dans des situations où sa vigilance est susceptible de varier. Pour répondre à cet objectif, le travail propose un prototype fondé sur une logique de simulation d'états cognitifs, sur une base de connaissances structurée et sur un mécanisme de génération de réponses contextualisées. L'architecture vise à distinguer plusieurs niveaux d'état de l'utilisateur et à associer à chacun d'eux une stratégie d'assistance différente. Dans les situations de vigilance relâchée, le système peut chercher à stimuler l'attention par des alertes plus directes. À l'inverse, lorsque le système estime que l'utilisateur se trouve dans un état de surcharge ou de fragilité cognitive, il privilégie une posture de soutien, de simplification de l'interface et de réduction de la pression décisionnelle.

L'approche retenue ne repose pas sur des capteurs biométriques réels, mais sur un simulateur d'états cognitifs conçu pour tester l'architecture sans imposer les contraintes matérielles, éthiques et organisationnelles liées à l'acquisition de données physiologiques. Ce choix méthodologique permet de concentrer l'analyse sur la logique de décision, la cohérence de la chaîne logicielle et la capacité du système à produire des recommandations adaptées à un contexte donné. Le mémoire examine également la contribution d'un mécanisme de génération

augmentée par récupération, utilisé pour limiter les incohérences du modèle de langage et encadrer ses réponses par une base documentaire spécialisée. L'évaluation porte enfin sur plusieurs scénarios de simulation afin de vérifier la cohérence du comportement adaptatif, la pertinence des réponses générées et la robustesse générale de l'architecture proposée.

En somme, ce mémoire montre qu'il est possible de concevoir une architecture de cybersécurité qui ne se limite pas à protéger l'infrastructure contre les menaces externes, mais qui tente aussi de réduire le risque humain à la source. L'apport principal de ce travail réside dans l'articulation entre cybersécurité, adaptation contextuelle et prise en compte d'un état cognitif simulé dans une boucle logicielle cohérente.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES ABRÉVIATIONS

RAG	Retrieval-Augmented Generation
EEG	Électroencéphalogramme
LLM	Large Language Model
SOC	Security Operations Center
IVF-PQ	Inverted File Index + Product Quantization
UEBA	User and Entity Behavior Analytics
SPOF	Single Point of Failure
UDP	User Datagram Protocol
HTTP	HyperText Transfer Protocol
JSON	JavaScript Object Notation
UQAC	Université du Québec à Chicoutimi
DIM	Département d'Informatique et de Mathématique
SQL	Structured Query Language
ECG	Électrocardiogramme
PPG	Photopléthysmogramme
LSTM	Long Short-Term Memory
RNN	Recurrent Neural Networks
SVM	Support Vector Machine
HRV	Heart Rate Variability

DÉDICACES

À mes directeurs de recherche, les Professeurs Fehmi Jaafar et Hamdi Ben Abdessalem, pour leur gentillesse, disponibilité, patience, encadrement, rigueur et accompagnement durant ce travail.

À mes parents, Elie Chabi et Micheline Merrelose, pour leurs prières.

À mon épouse, Fifame Chabi née Zossou Yehouenou, À ma famille, Isaac Chabi, Harold Chabi, Rodrigue Chabi, Priva Agossou, Sandra Agossou, Anne-Marie Chabi, Prince Zossou Yehouenou, Lumière Zossou Yehouenou, Christian Zossou Yehouenou, Serge Zossou Yehouenou, Françoise Zossou Yehouenou née Zannou, Miguel Atrokpo, Stéphanie Dogba, Lyam, Luna et Light Chabi pour leurs prières et soutien indéfectible.

À l'Université du Québec À Chicoutimi (UQAC), Aux gouvernements du Canada et du Québec, pour l'opportunité d'entamer une étude sur le territoire canadien .

À mon professeur, Pierre-Claver Montcho (alias Mr PC).

REMERCIEMENTS

En premier lieu, mes sincères remerciements vont à l'endroit de mes directeurs de recherche ; les professeurs Fehmi Jaafar et Hamdi Ben Abdessalem pour leur patience, leur dévouement, leur disponibilité et rigueur. Merci de m'avoir accepté comme votre étudiant. Vous m'avez appris à être rigoureux, minutieux et surtout à penser en dehors de la boîte.

En deuxième lieu, je remercie tout le Département d'Informatique et de Mathématique de l'Université du Québec à Chicoutimi (UQAC), pour m'avoir offert un cadre favorable à l'aboutissement de ce sujet de recherche.

En troisième lieu, je tiens à remercier mes parents (Elie Chabi et Micheline Merrelose), ma famille et amis pour leur soutien qui a été très important dans ce processus.

Enfin, mes remerciements vont à ma chère épouse, Fifame Chabi née Zossou Yehouenou.

AVANT-PROPOS

Ce mémoire est l'aboutissement d'un parcours de recherche au sein du Département d'Informatique et de Mathématique de l'Université du Québec à Chicoutimi. Un intérêt pour ce sujet est né après un constat flagrant. L'être humain demeure un facteur particulièrement vulnérable du point de vue de la cybersécurité, pendant que la plupart des systèmes de protection et de surveillance se concentre sur le renforcement de leurs outils.

Ce sujet de recherche était un défi. Il s'agissait de créer un pont entre la cybersécurité et l'informatique affective. Le plus intéressant durant ce travail est la possibilité de prendre en charge le problème de la croissance des attaques informatiques, directement à leur source : l'erreur humaine. L'objectif est alors de chercher à contrôler les différents états d'un travailleur (exemple : un opérateur en cybersécurité) afin de maintenir son taux de vigilance pour éviter qu'il fasse des erreurs coûteuses, faute d'un état cognitif défavorable (stress élevé par exemple).

La réalisation de ce sujet de mémoire a exigé une immersion profonde dans les technologies les plus récentes, telles que l'intelligence artificielle à titre illustratif. Plus précisément, l'intégration de modèles de langage comme Llama3 associée à l'usage de la technologie de Génération Augmentée par Récupération (RAG) ont été des étapes décisives, marquées par plusieurs ajustements afin d'avoir un résultat fiable.

CHAPITRE I

INTRODUCTION

Le domaine de la cybersécurité est resté concentré durant de nombreuses décennies sur le développement et le déploiement de mécanismes de défense purement techniques et structurels [1]. Les ingénieurs et administrateurs de réseaux ont traditionnellement misé sur la sophistication des outils comme les pare-feux pour ériger des barrières étanches contre les menaces numériques. Par contre, les données statistiques actuelles de l'industrie démontrent de manière irréfutable que cette approche centrée uniquement sur la barrière logicielle est insuffisante pour endiguer la prolifération des incidents de sécurité. L'erreur humaine demeure présentement la cause principale et le vecteur déclencheur de la grande majorité des compromissions réussies au sein des infrastructures organisationnelles [2]. Ce constat est également corroboré par plusieurs rapports institutionnels, qui soulignent la place dominante du facteur humain dans les incidents de sécurité. Un opérateur fatigué, distrait ou soumis à une surcharge cognitive extrême présente une probabilité grandement accrue de valider une configuration erronée ou de négliger un indicateur d'alerte critique sur son tableau de bord.

Parallèlement, l'informatique affective propose des modèles théoriques et des outils technologiques avancés permettant d'interpréter les états physiologiques et émotionnels des humains afin d'adapter dynamiquement les interactions homme-machine [3][4]. Si des secteurs hétérogènes

ont pleinement intégré ces concepts pour optimiser l'apprentissage ou la sécurité industrielle, les architectures de cybersécurité n'ont pas encore franchi le pas d'une intégration holistique. Les systèmes de supervision actuels fonctionnent de manière totalement aveugle vis-à-vis de l'état interne de l'humain qui les manipule. Ce manque de perception crée une vulnérabilité critique, car le système applique des politiques de sécurité identiques, que l'administrateur soit pleinement lucide ou en situation de défaillance cognitive imminente. Le présent sujet de recherche vise à combler ce fossé conceptuel en concevant un mécanisme de protection adaptatif capable de moduler son comportement selon le profil affectif de son utilisateur.

1.1 PROBLÉMATIQUE SCIENTIFIQUE

La problématique scientifique de ce sujet de recherche s'inscrit à l'intersection de la cybersécurité, de l'interaction homme-machine et de la prise en compte du facteur humain dans les systèmes de protection. Les architectures de sécurité contemporaines reposent encore majoritairement sur des mécanismes de contrôle d'accès, de filtrage réseau, de segmentation des privilèges et de détection d'anomalies centrés sur les objets techniques, les flux de données et les identités numériques. Cette approche demeure nécessaire, mais elle reste incomplète dès lors qu'elle ignore une réalité bien connue des environnements opérationnels. L'utilisateur humain, même légitime, peut devenir une source temporaire de vulnérabilité lorsqu'il agit sous l'effet de la fatigue, du stress, de la surcharge informationnelle ou d'une baisse de vigilance. Dans un contexte de cybersécurité, cette situation est d'autant plus critique qu'une seule erreur de manipulation, une validation trop rapide ou une mauvaise interprétation d'un signal d'alerte peut produire des conséquences disproportionnées sur la stabilité d'un système.

Le problème scientifique ne consiste donc pas simplement à protéger une infrastructure contre des attaques externes, mais à comprendre comment un système de sécurité peut intégrer de manière pertinente l'état cognitif de son utilisateur dans sa logique de décision. Les approches traditionnelles supposent implicitement que la personne qui administre ou supervise le système dispose en permanence de la même capacité de jugement. Or, cette hypothèse est fragilisée par la variabilité cognitive naturelle de l'être humain. Un administrateur peut être parfaitement autorisé, compétent et authentifié, tout en se trouvant momentanément dans un état de vigilance dégradé. Dans une telle situation, la sécurité ne dépend plus seulement de la validité des identifiants ou de la robustesse des politiques d'accès, mais aussi de la capacité du système à reconnaître que le contexte humain est en train de changer.

Cette limite révèle une insuffisance plus profonde encore dans la conception actuelle du Zero Trust. Le principe du Zero Trust a profondément renouvelé les architectures de cybersécurité en imposant une vérification continue des entités, des communications et des droits d'accès. Toutefois, dans sa mise en œuvre la plus courante, il demeure essentiellement orienté vers les entités numériques et les conditions réseau, sans prise en charge explicite de l'état cognitif de l'opérateur humain. Il en résulte une forme de rigidité conceptuelle car le système vérifie l'identité, mais il ne sait pas évaluer si la personne qui agit à l'instant présent dispose encore de la lucidité nécessaire pour prendre une décision sûre. Cette lacune est d'autant plus importante que les environnements sensibles exigent aujourd'hui des décisions rapides, répétées et souvent prises sous pression.

La question scientifique devient alors la suivante. Comment concevoir une architecture de cybersécurité capable de reconnaître qu'un utilisateur autorisé peut temporairement présenter une fragilité cognitive, et comment traduire cette reconnaissance dans une adaptation logicielle

utile, cohérente et non intrusive ? Ce problème suppose de dépasser une simple logique d'alerte ou de notification, pour penser une architecture capable de moduler ses réponses selon l'état estimé de l'utilisateur. Il ne s'agit pas de remplacer le jugement humain, mais de créer une assistance contextuelle qui limite les erreurs de manipulation et soutienne la prise de décision au moment où celle-ci devient plus incertaine.

Dans cette perspective, le présent sujet de recherche considère que l'état cognitif de l'utilisateur doit être traité comme une variable de sécurité à part entière. Cette hypothèse conduit à envisager une architecture où la vigilance, la fatigue et le stress ne sont plus perçus comme des informations secondaires, mais comme des signaux susceptibles d'influencer le niveau d'assistance, la nature des messages produits et la forme même de l'interaction. Le défi scientifique consiste donc à établir un modèle suffisamment structuré pour relier un état cognitif simulé à une réponse logicielle adaptative, tout en conservant la cohérence globale du système et en évitant de basculer dans une automatisation aveugle. C'est dans cette tension entre sécurité technique, variabilité humaine et adaptation logicielle que se situe le cœur de cette recherche.

1.2 OBJECTIF DU MÉMOIRE

L'objectif général de ce sujet de recherche est d'explorer et de formaliser une architecture logicielle de cybersécurité adaptative capable d'ajuster son comportement en fonction d'un profil affectif simulé de l'utilisateur. L'intention n'est pas de proposer un simple mécanisme de surveillance ou une couche de détection supplémentaire, mais de définir une boucle de décision où l'état cognitif estimé influence la forme de l'assistance, le contenu des réponses produites et la manière dont l'interface présente les informations à l'opérateur. L'enjeu est donc

de faire en sorte que la cybersécurité ne soit plus seulement réactive face aux attaques externes, mais également sensible aux conditions internes qui peuvent mener à une erreur humaine.

Pour atteindre cet objectif général, le travail s'appuie sur plusieurs objectifs spécifiques. Le premier consiste à définir une architecture modulaire permettant de représenter un état cognitif simulé et de l'intégrer dans une chaîne logicielle de décision. Le deuxième consiste à concevoir une base de connaissances structurée et un mécanisme de contextualisation documentaire capables de guider un modèle de langage dans la production de réponses cohérentes et adaptées au contexte. Le troisième consiste à vérifier que l'architecture peut produire des comportements différenciés selon plusieurs niveaux de stress et de fatigue simulés, afin d'examiner la pertinence des règles d'adaptation retenues. Le quatrième objectif vise à observer la robustesse fonctionnelle de l'ensemble du dispositif, notamment la stabilité de la communication entre les modules et la capacité du prototype à maintenir une logique cohérente malgré les variations de contexte.

Ce mémoire cherche ainsi à démontrer qu'une architecture de cybersécurité peut être pensée comme un système d'assistance contextuelle plutôt que comme un simple ensemble de barrières techniques. La contribution attendue ne se limite pas à montrer qu'un modèle de langage peut générer un texte spécifique ; elle consiste surtout à mettre en évidence qu'il est possible d'orchestrer plusieurs composants logiciels afin que l'état simulé de l'utilisateur influence réellement la stratégie de réponse du système. L'objectif est donc à la fois conceptuel et technique. Il s'agit de proposer un cadre d'adaptation crédible et capable de servir de base à des travaux plus avancés sur la sécurité centrée sur l'humain.

1.3 CONTRIBUTION DU MÉMOIRE

La contribution de ce sujet de recherche s'articule autour de trois apports principaux. Le premier apport est conceptuel. Il consiste à proposer une lecture de la cybersécurité dans laquelle l'état cognitif de l'utilisateur est intégré comme variable pertinente de la décision. Cette orientation permet d'élargir la portée du Zero Trust en l'ouvrant à une dimension humaine et contextuelle, alors que les approches classiques restent souvent centrées sur l'identité, la machine et le réseau. En ce sens, le travail ne cherche pas seulement à sécuriser un système contre un intrus, mais à réduire la probabilité qu'un utilisateur légitime provoque lui-même une faiblesse lorsqu'il agit dans un état de fragilité cognitive.

Le deuxième apport est architectural. Le mémoire propose une organisation logicielle modulaire où un simulateur d'états cognitifs alimente une couche de décision, laquelle s'appuie sur une base de connaissances structurée et sur un mécanisme de génération contextualisée. Cette architecture a été pensée pour assurer une continuité logique entre la mesure simulée de l'état utilisateur, la sélection des informations pertinentes et la matérialisation d'une action de sécurité adaptée. L'intérêt de cette structure est de montrer comment une boucle adaptative peut être construite de manière lisible, sans dépendre exclusivement de règles rigides codées en dur ni d'un simple affichage passif d'alertes.

Le troisième apport concerne la méthodologie de contextualisation documentaire. Le mémoire montre qu'un mécanisme de génération augmentée par récupération peut jouer un rôle important pour stabiliser la production de réponses dans un domaine spécialisé, à condition que le modèle soit encadré par une base documentaire pertinente et une logique d'extraction bien définie. Cette approche réduit l'improvisation du modèle de langage et l'oriente vers des

formulations plus cohérentes avec le contexte de sécurité ciblé. En parallèle, l'usage d'un simulateur d'états cognitifs constitue une contribution méthodologique utile, puisqu'il permet d'évaluer l'architecture sans recourir à des capteurs biométriques réels, souvent trop intrusifs ou difficiles à mobiliser dans un prototype de recherche.

L'ensemble de ces contributions vise à montrer qu'une cybersécurité moderne peut bénéficier d'une intégration plus intelligente du facteur humain. Le mémoire défend l'idée qu'une architecture de sécurité gagne en pertinence lorsqu'elle est capable d'interpréter l'état de l'utilisateur, d'en tenir compte dans sa logique de réponse et de soutenir l'opérateur au lieu de le considérer comme un simple point d'accès. C'est cette orientation qui constitue la valeur ajoutée principale du travail.

1.4 ORGANISATION DU MÉMOIRE

Le présent mémoire est structuré de manière progressive afin de conduire le lecteur du constat initial jusqu'à l'évaluation du prototype proposé. Le premier chapitre présente le contexte général de la recherche, la problématique scientifique, les objectifs poursuivis ainsi que les contributions attendues. Il établit le cadre intellectuel dans lequel s'inscrit le sujet de recherche et justifie la nécessité de considérer le facteur humain dans l'analyse des mécanismes de cybersécurité.

Le deuxième chapitre est consacré aux travaux connexes. Il propose une lecture critique de la littérature en mettant en évidence les approches qui ont tenté de relier les états cognitifs, l'adaptation logicielle et la sécurité numérique. L'objectif de ce chapitre est de situer clairement la contribution du mémoire dans un ensemble de travaux existants et de montrer pourquoi une

architecture adaptative centrée sur un état cognitif simulé demeure pertinente. Ce chapitre permet également d'identifier les limites des solutions antérieures, notamment lorsqu'elles reposent sur des capteurs lourds, des traitements trop coûteux ou des mécanismes d'alerte peu intégrés à la décision.

Le troisième chapitre présente l'architecture proposée ainsi que la méthodologie technique adoptée pour construire le prototype. Il décrit le flux de données, le rôle des différents modules, la base de connaissances, la logique de contextualisation documentaire et les outils logiciels retenus pour réaliser le système. Ce chapitre constitue le cœur technique du mémoire, puisque c'est là que sont détaillés les mécanismes concrets qui permettent à l'architecture de fonctionner.

Le quatrième chapitre expose la campagne d'évaluation menée sur plusieurs scénarios simulés. Il décrit les cas d'utilisation retenus, la comparaison entre les modèles de langage étudiés, l'impact de l'ancrage documentaire et les mesures observées en matière de comportement global et de latence. Ce chapitre cherche à présenter une évaluation descriptive et exploratoire d'un prototype de simulation.

Le cinquième chapitre propose une discussion critique des résultats obtenus. Il y sont examinées la fiabilité logicielle, les limites de la détection simulée, les enjeux liés aux hallucinations des modèles de langage et les implications plus larges de l'approche sur les plans technique et éthique. Enfin, le sixième chapitre synthétise les leçons tirées de la recherche et propose des perspectives concrètes pour prolonger le travail, notamment en direction d'améliorations méthodologiques, de garde-fous logiciels et d'éventuels tests dans un contexte opérationnel plus réaliste.

Ainsi, la problématique scientifique, les objectifs et les contributions du sujet de recherche conduisent à l'examen des travaux connexes. Le chapitre suivant propose alors une lecture critique de la littérature afin de mieux situer l'architecture proposée par rapport aux approches existantes et de faire ressortir la spécificité de la contribution développée dans ce mémoire.

CHAPITRE II

TRAVAUX CONNEXES

L'intégration des états internes de l'humain dans les architectures logicielles constitue un domaine d'investigation en pleine expansion scientifique. Pour bien situer notre contribution, il convient d'analyser de manière critique la littérature existante selon deux axes directeurs : d'une part, les initiatives visant à exploiter la détection cognitive pour l'envoi d'alertes en cybersécurité, et d'autre part, les systèmes adaptatifs développés dans des secteurs hétérogènes. Cette approche thématique permettra de mettre en lumière les lacunes méthodologiques que notre travail s'emploie à combler.

La littérature récente peut être regroupée en trois axes principaux : les approches fondées sur des signaux physiologiques, les systèmes d'adaptation d'interface, et les dispositifs de décision assistée par modèle de langage. Bien que ces travaux démontrent l'intérêt d'intégrer l'état humain dans la logique système, ils diffèrent fortement quant aux signaux utilisés, au niveau d'automatisation obtenu et au degré d'intégration dans la chaîne décisionnelle.

La littérature portant sur l'intégration des états cognitifs ou affectifs dans les systèmes de cybersécurité s'organise autour de quelques grandes orientations qui, bien qu'issues de communautés de recherche différentes, convergent vers une même interrogation. Comment utiliser l'état de l'utilisateur comme variable de décision sans alourdir excessivement la chaîne

technique ni perdre la cohérence du système de protection ? Cette question traverse des travaux qui vont de la détection physiologique à l'adaptation d'interface, en passant par la génération d'alertes et l'analyse contextuelle des comportements. Toutefois, ces recherches n'accordent pas toutes la même place au facteur humain, et elles n'aboutissent pas aux mêmes formes d'assistance logicielle. Certaines s'intéressent d'abord à mesurer un état, d'autres à le visualiser, d'autres encore à l'exploiter pour guider l'action. C'est précisément cette diversité qui rend utile une lecture thématique plutôt qu'un simple relevé article par article.

2.1 TRAVAUX CENTRÉS SUR LA DÉTECTION ET LA GESTION DES ALERTES EN CYBERSÉCURITÉ

Une première famille de travaux s'est attachée à la détection de l'effort cognitif et de la charge mentale dans des environnements de supervision où l'opérateur doit rester attentif à de nombreux signaux simultanés. Dans ce cadre, plusieurs auteurs ont mobilisé des mesures EEG et ECG pour estimer l'état de vigilance d'un utilisateur et fournir des recommandations via des tableaux de bord [5]. L'intérêt de ces approches est de rappeler que la sécurité d'un système dépend aussi de la disponibilité mentale de la personne qui le supervise. Néanmoins, ces solutions restent souvent limitées par leur dépendance à des dispositifs matériels coûteux, à des traitements multimodaux complexes et à une latence qui devient problématique lorsque l'on veut réagir rapidement à une situation critique. Elles montrent donc la pertinence du problème, mais pas encore une réponse pleinement opérationnelle pour un environnement de cybersécurité exigeant.

Une deuxième ligne de recherche s'est développée autour de la détection de la détresse ou de la panique dans des contextes de fraude, d'ingénierie sociale ou de communication sensible.

Certains travaux utilisent des bracelets connectés et des signaux PPG pour identifier un état émotionnel potentiellement critique lors d'un appel téléphonique [6]. Dans ces cas, la finalité n'est pas seulement d'observer un état physiologique, mais de prévenir une action risquée qui pourrait être déclenchée par l'émotion ou la surprise. L'apport de ces travaux est réel, car ils montrent qu'un système peut tenter de protéger un utilisateur exposé à une manipulation. Toutefois, l'absence d'une lecture textuelle ou sémantique des échanges limite la pertinence de l'inférence, puisqu'un signal physiologique isolé ne suffit pas toujours à distinguer une véritable menace d'un simple stress contextuel sans conséquence opérationnelle.

Une troisième série de travaux s'est intéressée à la détection du stress dans des environnements institutionnels où la manipulation humaine peut avoir des effets collectifs, notamment dans le domaine du vote électronique [7]. Ici, les architectures combinent des caméras de reconnaissance faciale, des capteurs thermiques et des réseaux de neurones convolutifs afin d'émettre des alertes en temps réel. Ces solutions montrent que l'adaptation peut être envisagée dans des contextes critiques et qu'un état humain peut devenir un signal utile pour orienter la vigilance du système. Cependant, leur fiabilité demeure limitée par la difficulté à interpréter correctement les expressions faciales et par la sensibilité des modèles aux biais environnementaux. En particulier, un état émotionnel profond ne se réduit pas à un visage ou à une posture, ce qui explique pourquoi ces approches génèrent encore de nombreux faux négatifs.

D'autres travaux, plus proches de la surveillance des menaces internes, ont utilisé l'EEG pour catégoriser des comportements à risque ou détecter des formes de négligence au sein d'infrastructures sensibles [8]. Ces études ont l'avantage de se rapprocher du problème de cybersécurité traité dans ce mémoire, puisqu'elles cherchent déjà à relier l'état cognitif d'un utilisateur à un niveau de risque opérationnel. Elles exploitent parfois des algorithmes

d'apprentissage profond ou d'ensemble pour améliorer la discrimination des états, mais elles reposent souvent sur des jeux de données restreints et sur des contextes expérimentaux fortement contrôlés. Le problème est alors moins celui de la démonstration locale que celui du passage à l'échelle. Un modèle performant dans un environnement de laboratoire n'est pas nécessairement transférable dans une situation d'usage réel où les profils cognitifs sont variés et les signaux plus bruités.

2.2 TRAVAUX PORTANT SUR L'ADAPTATION D'INTERFACE DANS D'AUTRES DOMAINES

À côté de ces travaux explicitement liés à la cybersécurité, d'autres domaines ont déjà développé des mécanismes d'adaptation d'interface plus sophistiqués. Les systèmes d'apprentissage en ligne, par exemple, exploitent la reconnaissance émotionnelle par webcam pour ajuster le rythme d'un cours, la disposition des éléments visuels ou encore la typographie en fonction du profil de l'apprenant [9]. De même, certaines applications de bien-être mental utilisent des modèles convolutionnels, comme VGG-16, pour interpréter des images faciales et recommander des activités personnalisées [10]. Ces approches montrent qu'une adaptation dynamique de l'interface est techniquement possible, mais elles restent exposées à des biais visuels, à des variations d'éclairage et à une dépendance forte à la qualité du signal facial. Elles proposent donc des mécanismes de personnalisation intéressants, sans toutefois répondre aux exigences plus strictes d'un système de cybersécurité où la précision et la réactivité sont déterminantes.

Le secteur du divertissement numérique a également exploré des boucles biocybernétiques, notamment dans certains jeux à la première personne où la difficulté et le comportement des ennemis sont adaptés à l'activité électrodermale du joueur [11]. Ces travaux confirment

qu'un système peut modifier son comportement en fonction d'un indice physiologique pour influencer l'expérience utilisateur. Toutefois, l'EDA renseigne surtout sur l'excitation générale et ne permet pas de distinguer clairement la fatigue, l'anxiété ou la surcharge cognitive. Les technologies de réalité virtuelle combinant EEG, EDA et variabilité cardiaque ont, de leur côté, renforcé le caractère immersif des interfaces adaptatives [12] [13]. Dans le même esprit, l'usage de classificateurs SVM pour adapter des scénarios publicitaires en temps réel à partir de l'EEG a montré l'intérêt d'une adaptation contextuelle, mais aussi la difficulté à généraliser ces approches lorsque les performances des modèles sont seulement moyennes [14].

Le tableau 2.1 compare les études pertinentes identifiées, en mettant en évidence leurs objectifs, les signaux utilisés, leur mode de gestion des alertes et leurs principales limites.

Tableau 2.1 : COMPARATIF DES ÉTUDES CENTRÉES SUR L'ENVOI D'ALERTES BASÉES SUR L'ÉTAT COGNITIF EN CYBERSÉCURITÉ

Articles (année)	**Objectif**	**Signaux utilisés**	**Gestion des alertes**	**Limites principales**
Emotion-aware cyber-physical systems (2019)	Réduire les erreurs humaines via recommandations	EEG, ECG, multimodal	Tableau de bord (détails non donnés)	Latence, coût matériel, manque de détails techniques
Speech emotion recognition (2023)	Détecter la stupeur panique lors d'appels	PPG (bracelet)	Application mobile	Faux positifs, pas d'analyse vocale
Multi-Modal Biometric Voting (2024)	Détecter intimidation/stress électeurs	Expressions faciales, température, rythme cardiaque	Interface graphique (dashboard)	Latence, expressions faciales peu fiables
EEG Risk Framework (2020)	Détecter comportements à risque (insider)	EEG	Application logicielle (détail non précisé)	Échantillon réduit, surapprentissage

L'examen transversal de ces travaux montre donc une tension constante entre richesse du signal, fiabilité de l'inférence et coût de traitement. Les approches multimodales sont souvent plus expressives, mais elles introduisent des délais de traitement et des dépendances matérielles qui nuisent à leur usage dans un cadre critique. Les approches plus légères, fondées sur une source unique, sont plus rapides à traiter, mais elles sont souvent limitées par la pauvreté du signal ou par une sensibilité élevée aux biais. Cette littérature confirme ainsi que la question n'est pas seulement de savoir si l'on peut détecter un état humain, mais de déterminer comment cet état

peut devenir une variable utile dans une chaîne de décision de sécurité sans faire exploser la complexité du système.

Dans cette perspective, le présent sujet de recherche se situe au croisement de deux attentes encore insuffisamment réunies dans la littérature. D'une part, il reprend l'idée qu'un état cognitif peut influencer la sécurité et qu'un système doit pouvoir s'adapter à une vigilance diminuée ou à une surcharge mentale. D'autre part, il cherche à intégrer cette adaptation dans une architecture logicielle cohérente, où l'état simulé de l'utilisateur est utilisé pour guider un modèle de langage à partir d'une base documentaire contrôlée. L'intérêt de cette orientation est de dépasser le simple affichage d'alerte ou la simple personnalisation d'interface, pour proposer une véritable boucle de sécurité contextuelle. Le chapitre suivant présente les limites identifiées dans ces travaux et précise le positionnement adopté dans ce mémoire.

2.3 LIMITES IDENTIFIÉES ET POSITIONNEMENT

L'examen approfondi de la littérature met en évidence plusieurs limites structurelles qui freinent encore le développement d'une cybersécurité adaptative réellement centrée sur l'humain. La première limite tient au fait que les travaux existants se concentrent très souvent sur la détection d'un état, sans aller jusqu'à sa traduction en adaptation effective de l'environnement logiciel. Autrement dit, beaucoup de systèmes savent produire une alerte ou afficher un signal de vigilance, mais ils ne modifient ni l'interface, ni les privilèges, ni la logique d'assistance de manière suffisamment intégrée. Cette absence de réponse structurelle limite fortement la portée de l'approche, car une détection qui ne débouche sur aucun changement réel n'a qu'une valeur informative partielle.

La deuxième limite concerne la dépendance aux architectures multimodales. Il est vrai que la combinaison de plusieurs capteurs peut enrichir l'interprétation de l'état humain. Toutefois, cette richesse s'accompagne d'une augmentation notable de la complexité technique, de la latence de traitement et des besoins de synchronisation. Dans un cadre de cybersécurité, ces contraintes sont loin d'être secondaires, parce qu'un système lent ou difficile à maintenir perd rapidement de son efficacité lorsqu'il doit réagir à une situation critique. La multimodalité constitue donc un gain potentiel en précision, mais elle fragilise souvent la réactivité opérationnelle du système.

La troisième limite réside dans la fidélité variable des signaux exploités. Les approches reposant sur des indicateurs faciaux, thermiques ou périphériques comme l'EDA sont sensibles aux conditions de capture, aux biais environnementaux et aux différences individuelles. Cela rend leur interprétation difficile et parfois ambiguë. Même les approches basées sur l'EEG, pourtant plus proches des processus cognitifs profonds, restent dépendantes du contexte expérimental et des conditions d'apprentissage. Les résultats obtenus dans un environnement contrôlé ne se transposent pas toujours à des utilisateurs réels, ce qui fragilise la robustesse des modèles lorsqu'ils sont déployés dans des situations plus ouvertes.

La quatrième limite concerne l'usage des grands modèles de langage dans des contextes spécialisés. La littérature récente montre que ces modèles peuvent générer des réponses fluides, mais pas nécessairement fiables, surtout lorsqu'ils ne sont pas ancrés dans une base documentaire contrôlée [15][16]. Dans un système de cybersécurité, cette instabilité constitue un problème sérieux, car une réponse imprécise ou mal orientée peut conduire à une mauvaise action, à une perte de confiance ou à une interprétation erronée de l'alerte. L'enthousiasme

suscité par les modèles génératifs ne doit donc pas masquer la nécessité d'un encadrement rigoureux de leur production textuelle.

À partir de ces limites, le présent mémoire adopte un positionnement volontairement plus sobre, plus contrôlé et plus orienté vers l'intégration logicielle. Au lieu de multiplier les capteurs ou de dépendre d'une instrumentation lourde, le sujet de recherche s'appuie sur un simulateur d'états cognitifs. Ce choix permet d'isoler la logique de décision du problème matériel et de concentrer l'analyse sur la façon dont l'état de l'utilisateur peut guider une réponse de sécurité. De cette manière, le prototype n'a pas pour ambition de remplacer une instrumentation biométrique réelle, mais de montrer qu'une boucle adaptative peut être pensée et validée sans dépendre immédiatement d'un dispositif intrusif.

Le positionnement retenu consiste donc à combiner trois éléments : un état cognitif simulé, une base de connaissances structurée et un modèle de langage encadré par récupération documentaire. Cette combinaison permet de réduire le risque d'hallucination, de conserver une bonne réactivité et de maintenir une cohérence décisionnelle dans les réponses produites par le système. L'originalité du travail ne réside pas seulement dans l'usage de ces composants pris séparément, mais dans leur articulation au sein d'une même boucle de sécurité contextuelle. Le chapitre suivant présente alors l'architecture logicielle qui matérialise ce positionnement et détaille la logique de fonctionnement du prototype.

CHAPITRE III

MÉTHODOLOGIE

La conception d'un système de cybersécurité capable de moduler son comportement en fonction de l'utilisateur exige une architecture logicielle modulaire et interopérable. Ce chapitre expose de manière détaillée la méthodologie retenue pour structurer notre pipeline adaptatif, ainsi que la configuration technique des différents composants qui le constituent.

3.1 FLUX DE DONNÉES ET MODULES SYSTÈMES

L'architecture proposée repose sur un flux de données séquentiel et circulaire qui s'exécute en temps réel pour assurer une réactivité optimale du système de défense. Le pipeline logiciel est structurellement subdivisé en trois grands modules fonctionnels qui interagissent continuellement : le module de mesure cognitive, le module d'adaptation contextuelle et le module d'affichage applicatif.

Le premier module correspond à la couche de captation des données physiologiques. Dans le cadre de ce travail, ce module intègre un simulateur d'états cognitifs développé spécifiquement pour mimer les valeurs numériques associées aux fluctuations cérébrales humaines. Ce composant génère en continu des variables quantitatives normalisées représentant les niveaux

de stress et de fatigue de l'opérateur, sur une échelle linéaire comprise entre 0 et 1. Ces données, une fois formalisées, sont transmises directement au module d'adaptation qui constitue le cœur décisionnel de notre architecture.

Le simulateur d'états cognitifs génère un flux de données pour modéliser l'évolution du stress et de la fatigue de l'opérateur. En pratique, pour éviter que de micro-fluctuations ou des changements trop brusques ne rendent l'affichage instable sur l'interface PyQt5, le système intègre un mécanisme de lissage des données. Le script Python calcule une moyenne des scores sur les dernières secondes d'activité avant de transmettre les résultats au module d'adaptation. Cette étape permet de stabiliser les valeurs de fatigue et de stress, afin que le système réagisse à de véritables tendances cognitives plutôt qu'à des variations isolées, tout en maintenant un traitement léger en mémoire volatile.

Le module de mesure regroupe les données cognitives sous la forme d'un objet, avant de les transmettre au cœur du système. Cette organisation permet de structurer proprement les informations en mémoire avant de lancer la phase de recherche sémantique. En gérant le flux de cette manière, l'architecture permet que chaque variation de l'état de l'utilisateur soit traitée de manière séquentielle, évitant ainsi les conflits d'accès en mémoire ou la perte d'informations lors de la contextualisation documentaire.

Le module d'adaptation prend en charge le traitement des variables cognitives et gère le pipeline linguistique. Dès réception des niveaux de stress et de fatigue, le système injecte ces coordonnées numériques au sein d'un modèle d'encodage vectoriel. Ce dernier effectue un calcul de similarité cosinus par rapport à une base de connaissances spécialisée, permettant de sélectionner les exemples documentaires les plus pertinents vis-à-vis du profil affectif de l'utilisateur. Ces fragments textuels sont alors agrégés au prompt initial, créant un contexte enrichi qui est

transmis par une requête HTTP POST au grand modèle de langage Llama3 :instruct hébergé localement.

Le troisième module gère la réception et la matérialisation de la décision de sécurité. Le modèle Llama3 génère une réponse structurée exclusivement au format JSON, contenant la nature de l'action à mener (soit la simulation d'un incident pour stimuler l'éveil, soit une recommandation apaisante). Cette charge utile est sérialisée et expédiée via le protocole réseau UDP sur le port 5010 vers l'interface d'administration centrale du pare-feu. Ce module applicatif décode instantanément le paquet réseau pour afficher l'alerte à l'écran et appliquer les restrictions d'interface correspondantes, garantissant une boucle de rétroaction complète et sécurisée.

Pour assurer une bonne interopérabilité et une communication fluide entre le module décisionnel et l'interface d'administration centrale du pare-feu, les paquets réseau doivent respecter une structure. Le choix s'est porté sur des objets standardisés au format JSON, transitant de manière asynchrone par des datagrammes UDP. Lorsque le système détecte un état de vigilance relâché (stress bas et fatigue basse), le modèle de langage Llama3 :instruct, guidé par les fragments documentaires récupérés dans la base de connaissances, génère une charge utile structurée visant à simuler un incident de sécurité. La structure du message JSON se décline ainsi : une clé principale identifiant l'opérateur concerné, suivie d'une clé qualifiant le niveau critique de l'événement sémantique (par exemple, "incidentCritique"), d'un descripteur textuel précis de la menace destiné à être affiché à l'écran (comme "grave : malware détecté"), et enfin d'un indicateur temporel précis pour assurer la traçabilité des logs dans la base MySQL. Un tel formatage permet à l'interface PyQt5 du pare-feu de parser instantanément le dictionnaire, de

déclencher une alerte sonore ou visuelle pop-up, et de forcer l'analyste à valider une procédure de remédiation afin de réactiver ses réflexes opérationnels.

À l'inverse, en situation de surcharge cognitive ou émotionnelle (stress élevé et fatigue élevée), la structure de l'objet JSON change de nature pour privilégier des clés orientées vers la protection et la décharge de travail de l'opérateur. La clé d'action transmet alors une instruction restrictive (telle que "restrictionInterface"), tandis que le champ de message contient une recommandation managériale et ergonomique claire (par exemple, "Déléguez les tâches critiques et limitez-vous à la surveillance passive"). Dès réception de ce paquet sur le port 5010, l'interface du pare-feu applique immédiatement l'affichage d'un message bienveillant à l'opérateur. Cette standardisation des échanges garantit que, peu importe la complexité de l'analyse linguistique réalisée par le transformeur, la commande reçue par l'infrastructure réseau reste binaire et sécurisée.

3.2 BASE DE CONNAISSANCES ET PIPELINE RAG

L'ancrage documentaire du modèle de langage est indispensable pour réduire le risque d'incohérence linguistique et forcer le système à respecter les procédures opérationnelles de sécurité. Le pipeline de génération augmentée par récupération s'appuie sur une base de connaissances hautement structurée sous la forme d'un fichier JSON standardisé. Cette base répertorie des triplets associant un état cognitif théorique, un type d'incident cybernétique et une recommandation technique précise.

La sélection contextuelle s'opère par l'intermédiaire du modèle d'embedding All-miniLM-L6-V2. Lors de chaque itération du système, ce modèle convertit la description textuelle de

l'état cognitif actuel en un vecteur dense de caractéristiques. Ce vecteur est comparé en temps réel aux représentations vectorielles des entrées de la base de connaissances. Les fragments de texte présentant le score de similarité le plus élevé sont extraits et positionnés de manière dynamique au sein de la zone mémoire allouée au contexte additionnel du prompt.

Cette stratégie logicielle permet d'influencer directement la distribution de probabilité des jetons générés par le modèle de langage sans avoir recours à des structures de contrôle conditionnelles rigides de type "if/else". Le grand modèle de langage assimile les exemples fournis comme des contraintes de génération, fait qui l'oblige à calquer sa syntaxe et ses décisions de sécurité sur les données techniques validées de la base documentaire, assurant ainsi un alignement avec le profil affectif détecté.

Le simulateur d'états cognitifs extrait à intervalles réguliers un couple de valeurs représentant le stress et la fatigue de l'utilisateur. Ces variables numériques sont ensuite converties en une description textuelle standardisée, qui est soumise au modèle d'embedding All-MiniLM-L6-v2. Le modèle transforme cette entrée en un vecteur dense de 384 dimensions, ensuite comparé aux représentations pré-vectorisées des triplets de la base de connaissances JSON. Le fragment documentaire dont la similarité est la plus élevée est extrait puis intégré au prompt envoyé au modèle de langage Llama3 :instruct. Ce mécanisme permet de relier un état simulé à une règle textuelle pertinente sans recourir à des structures conditionnelles rigides.

Lors de chaque cycle, le simulateur transmet les valeurs de stress et de fatigue au module d'adaptation. Ces valeurs sont converties en une description sémantique simple, comparée aux entrées textuelles de la base de connaissances JSON grâce au modèle d'embedding All-MiniLM-L6-v2. Le fragment le plus pertinent est ensuite injecté dans le prompt du modèle

de langage. Ce mécanisme limite le recours à des structures conditionnelles rigides de type if/else.

Pour garantir un temps d'exécution minimal, la recherche au sein du fichier JSON est optimisée pour trier rapidement les triplets contenant l'état détecté, l'incident ou la recommandation correspondante. Cette organisation permet de circonscrire la sélection du contexte aux seules règles applicables au profil de l'utilisateur. La sélection reste rapide et stable, même si l'état de l'opérateur oscille subitement entre une fatigue marquée et un stress modéré. Le système ajuste le pointeur de contexte sans induire de retard de calcul, ce qui empêche l'apparition d'un déphasage entre l'état de l'utilisateur et l'action générée par le grand modèle de langage, témoignant ainsi la viabilité de la boucle biocybernétique locale.

La convergence logique du pipeline de génération augmentée par récupération se matérialise par la construction du prompt destiné au grand modèle de langage. Ce processus assemble les instructions système permanentes, le fragment documentaire extrait de la base JSON et les valeurs actuelles des curseurs de fatigue et de stress. En combinant textuellement ces informations, le système encadre strictement l'espace de réponse du modèle Llama3. Cette ingénierie de prompt encadre la génération du LLM. En intégrant directement la règle validée dans la requête, le modèle de langage agit dans un cadre contrôlé, ce qui réduit le risque d'hallucination. Son rôle unique est d'analyser l'état reçu, de choisir librement l'action adéquate en fonction de la documentation et de la formater dans une structure JSON exploitable par l'interface du pare-feu.

3.3 CONFIGURATION ET OUTILS LOGICIELS

La concrétisation matérielle et logicielle de cette architecture adaptative s'appuie sur un ensemble d'outils modernes choisis pour leur robustesse et leur facilité d'intégration au sein d'un pipeline de cybersécurité. L'environnement principal de développement et d'organisation du code source est Visual Studio Code, qui offre une gestion flexible des extensions nécessaires au débogage des scripts réseau et des modèles d'intelligence artificielle.

Le langage de programmation principal retenu pour l'intégralité du système développé est Python. Ce choix méthodologique se justifie par la richesse de son écosystème en matière de manipulation de données et par la disponibilité des bibliothèques spécialisées dans le déploiement de pipelines d'apprentissage automatique. Les interactions avec le grand modèle de langage Llama3 :instruct sont orchestrées par l'intermédiaire du framework Ollama, qui permet une exécution locale et conteneurisée des poids du modèle, contribuant à préserver la confidentialité absolue des données de sécurité et des flux physiologiques de l'utilisateur.

La construction des interfaces graphiques, incluant le panneau d'administration du pare-feu et le panneau de contrôle du simulateur d'états cognitifs, est entièrement réalisée avec la bibliothèque PyQt5. Ce framework permet de concevoir des interfaces utilisateur réactives capables de rafraîchir leurs composants visuels en fonction des paquets réseau reçus de manière asynchrone. Enfin, la persistance des données et la traçabilité des événements sont assurées par un système de gestion de bases de données relationnelles MySQL. Cette base stocke de manière séquentielle l'historique des états cognitifs mesurés, les scores d'embedding calculés et les réponses JSON générées par le LLM, offrant ainsi un cadre d'audit complet pour évaluer à posteriori le comportement de la boucle biocybernétique.

Le déploiement concret de cette architecture adapte une gestion rigoureuse des communications inter-processus afin d'éviter tout blocage de l'interface graphique lors des phases d'inférence du grand modèle de langage. Le cycle de vie d'une requête débute par la capture des données de simulation PyQt5. Ces informations initialisent un thread d'arrière-plan asynchrone qui formule une requête réseau structurée de type HTTP POST. La charge utile, encapsulée sous format JSON, cible l'API REST locale exposée par le framework Ollama sur le port 11434. Bien que les requêtes HTTP POST reposent sur TCP localement, cette approche reste compatible avec les évolutions vers HTTP/3 (fondé sur QUIC/UDP), assurant la cohérence avec le socket UDP utilisé pour l'interface. Le corps de cette requête contient les hyperparamètres de génération, le prompt système restreint et le contexte documentaire sémantiquement extrait par le pipeline RAG. Pendant que les couches de calcul du modèle de langage traitent les tenseurs de poids de Llama3, l'interface graphique principale continue d'exécuter sa boucle de rafraîchissement sans subir de gel d'affichage. Dès que l'inférence se termine, la réponse textuelle est captée par le thread secondaire, qui valide sa structure avant de l'injecter dans la couche réseau de transmission.

Le transfert final de la décision de sécurité vers l'interface d'administration centrale du pare-feu est délégué à un socket réseau configuré sur le protocole UDP. Ce choix se justifie par le besoin de minimiser les temps de transit des alertes en éliminant la latence de synchronisation propre aux poignées de main du protocole TCP. Le module d'adaptation sérialise la réponse JSON sous forme d'un flux d'octets encodés en UTF-8. Le socket émetteur transmet ce datagramme vers l'adresse de bouclage local 127.0.0.1 en ciblant spécifiquement le port réseau 5010. Du côté du pare-feu, un socket récepteur asynchrone est configuré pour écouter en permanence sur ce port. Afin de pallier l'absence intrinsèque de contrôle de livraison du protocole UDP, le script intègre un mécanisme léger de validation d'intégrité basé sur la structure syntaxique

du dictionnaire JSON. Si le datagramme est complet, le pare-feu extrait instantanément les clés applicatives pour reconfigurer son interface visuelle et appliquer les restrictions d'accès requises.

La gestion de la concurrence au sein de notre pipeline logiciel est un prérequis technique indispensable pour maintenir la disponibilité de la console d'administration centrale du pare-feu en situation d'urgence cybernétique. L'architecture s'appuie sur le framework d'asynchronisme de PyQt5 pour découpler l'interface homme-machine des couches d'inférence de l'intelligence artificielle. Lorsqu'un thread d'arrière-plan initie la requête HTTP POST vers l'API d'Ollama, le socket réseau utilise des primitives non-bloquantes. Le cycle de vie complet de l'échange de données est géré par une machine à états logicielle qui surveille les codes de retour de la connexion locale. Si le serveur Ollama subit une surcharge computationnelle lors du traitement des tenseurs de Llama3, des mécanismes de temporisation exponentielle (*exponential backoff*) sont automatiquement appliqués pour prévenir la perte de paquets ou le plantage du service émetteur.

En revanche, au niveau de la réception UDP sur le port 5010, l'interface du pare-feu implémente un tampon circulaire (*ring buffer*) asynchrone pour stocker temporairement les datagrammes JSON reçus. Ce choix réseau est crucial car le protocole UDP ne garantit ni l'ordre de livraison ni l'acquittement des messages. Si plusieurs alertes cognitives sont émises en rafale par le module d'adaptation, le pare-feu traite les charges utiles de manière séquentielle en appliquant un filtre de nouveauté basé sur l'horodatage SQL inclus dans la structure du message. Les datagrammes obsolètes ou corrompus syntaxiquement sont immédiatement écartés, suggérant que l'affichage applicatif reflète avec une bonne précision le profil affectif de l'utilisateur de

manière ininterrompue, sans introduire de goulot d'étranglement ou de latence d'affichage à l'écran.

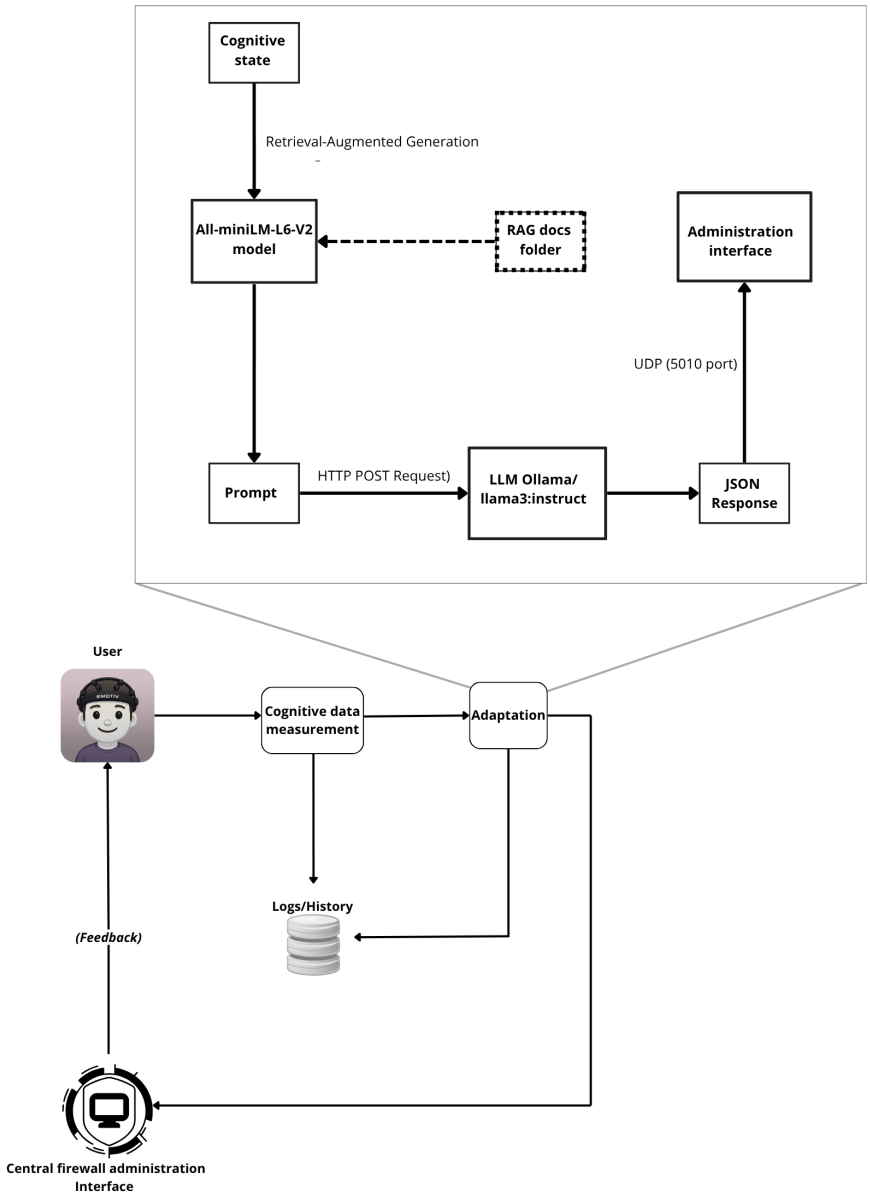


Figure 3.1 : ARCHITECTURE DU SYSTÈME

La figure 3.1 synthétise l'ensemble de ces interactions au sein de l'architecture globale du système adaptatif proposé et le chapitre suivant présente les scénarios de simulation mis en œuvre pour valider le comportement de cette architecture face aux variations du profil affectif de l'opérateur.

CHAPITRE IV

SIMULATION ET ANALYSE

Ce chapitre présente de manière exhaustive le cadre mis en œuvre pour observer le comportement de notre architecture adaptative, ainsi qu'une analyse descriptive rigoureuse des données issues des simulations de scénarios cognitifs et des tests comparatifs entre plusieurs modèles de langage.

4.1 SIMULATION DES SCÉNARIOS COGNITIFS

Afin d'observer le comportement du système dans différents contextes cognitifs, nous avons configuré trois cas d'utilisation distincts au sein du simulateur d'états cognitifs, permettant d'observer la dynamique de l'architecture face aux variations du profil affectif de l'opérateur.

Avant d'exposer l'analyse descriptive des cas d'utilisation, il s'avère indispensable de détailler le cadre de simulation qui encadre nos travaux de sorte à assurer la cohérence des observations réalisées. Pour valider la sensibilité du système aux points d'inflexion, des pics de stress artificiels ont été injectés à des intervalles temporels fixes, mimant l'apparition soudaine d'alertes de sécurité ou de pannes d'infrastructure sur la console de l'administrateur. L'ensemble des flux de données générés par le socket émetteur a été consigné au sein de la base de données

relationnelle MySQL avec une précision temporelle à la milliseconde. Ce dispositif permet d'analyser non seulement la pertinence de la classification finale opérée par le modèle Llama3, mais également la réactivité temporelle et la stabilité comportementale de la boucle biocybernétique face à des transitions de charge cognitive extrêmement rapides.

4.1.1 CAS D'UTILISATION 1 : PROFIL AFFECTIF STABLE (STRESS BAS, FATIGUE BASSE)

Le premier scénario modélise un analyste en cybersécurité opérant dans des conditions optimales de lucidité et de fraîcheur mentale. Les curseurs du simulateur ont été positionnés sur des valeurs proches de l'origine, fixant le niveau de stress à 0.03 et le taux de fatigue à 0.02. À la réception de ces données, le module d'adaptation estime que l'opérateur dispose de toutes ses capacités cognitives pour gérer des situations complexes, mais court le risque d'une baisse de vigilance due à la monotonie de la surveillance passive. Dans cette logique, le pipeline RAG extrait de la base de connaissances des exemples d'incidents critiques et de haut grade (tels que des intrusions de niveau 4 ou des exfiltrations massives de données). Le modèle de langage Llama3 :instruct, assimilant ce contexte, génère instantanément la simulation d'un incident grave de type "grave : malware" qui est transmis à l'interface du pare-feu. Cette injection dynamique vise à stimuler immédiatement l'éveil, la concentration et la réactivité de l'analyste.

La figure 4.1 illustre le premier cas d'utilisation du système, correspondant à un profil affectif stable, caractérisé par un stress faible et une fatigue basse

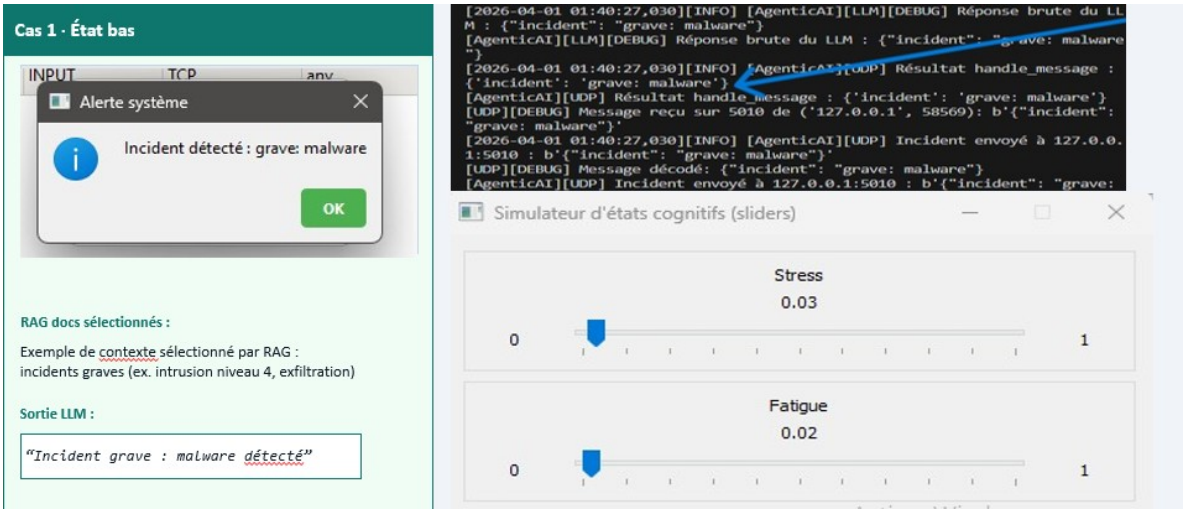


Figure 4.1 : PREMIER CAS D'UTILISATION

Cette configuration montre que, dans un contexte de vigilance élevée, l'architecture privilégie une réponse de stimulation de l'attention plutôt qu'une posture de protection.

4.1.2 CAS D'UTILISATION 2 : PROFIL AFFECTIF INTERMÉDIAIRE (STRESS MODÉRÉ, FATIGUE MODÉRÉE)

Le deuxième scénario explore une situation de transition où l'opérateur commence à ressentir les effets d'une charge de travail prolongée. Les variables quantitatives sont stabilisées à un niveau intermédiaire, avec un stress mesuré à 0.49 et une fatigue à 0.50. Cet état indique un équilibre mental précaire : l'analyste est encore efficace, mais les premiers signes de dégradation cognitive se manifestent. Le système réagit en orientant la sélection vectorielle vers des fragments documentaires catégorisés comme des recommandations de vigilance modérée. Le grand modèle de langage produit alors une réponse JSON structurée invitant l'utilisateur à adopter un comportement préventif, formulant le message textuel suivant : "hydrate-toi et

avertis ton responsable avant de continuer". L'objectif de cette action est de stabiliser l'état de l'opérateur et d'empêcher une détérioration de ses performances.

La figure 4.2 présente le deuxième cas d'utilisation, marqué par un autre profil affectif.

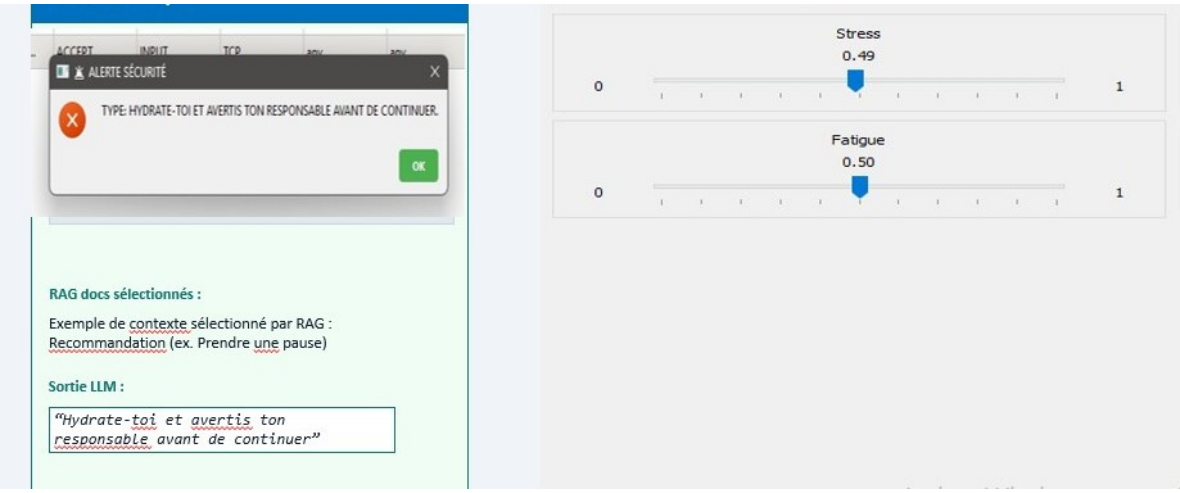


Figure 4.2 : DEUXIÈME CAS D'UTILISATION

4.1.3 CAS D'UTILISATION 3 : PROFIL AFFECTIF DÉFAVORABLE (STRESS ÉLEVÉ, FATIGUE ÉLEVÉE)

Le troisième scénario simule une situation critique où l'opérateur humain se trouve en état de défaillance cognitive imminente, représentant une vulnérabilité majeure pour l'infrastructure informatique. Les paramètres du simulateur sont poussés à des niveaux extrêmes, avec un taux de stress atteignant 0.94 et une fatigue évaluée à 0.86. Face à ce profil fortement altéré, la probabilité que l'analyste commette une erreur de jugement fatale est extrêmement élevée. L'architecture applique alors une politique stricte de décharge cognitive. Le pipeline RAG sélectionne des exemples de recommandations apaisantes et restrictives au sein de la base

de connaissances. Sans aucune intervention manuelle, le modèle Llama3 :instruct génère une commande de sécurité ordonnant à l'opérateur de restreindre son activité : "délègue les tâches critiques et limite-toi à la surveillance". L'interface du pare-feu verrouille alors l'accès aux configurations sensibles pour protéger le réseau. La figure 4.3 montre le troisième cas d'utilisation et permet d'observer la variation de réponse du système.

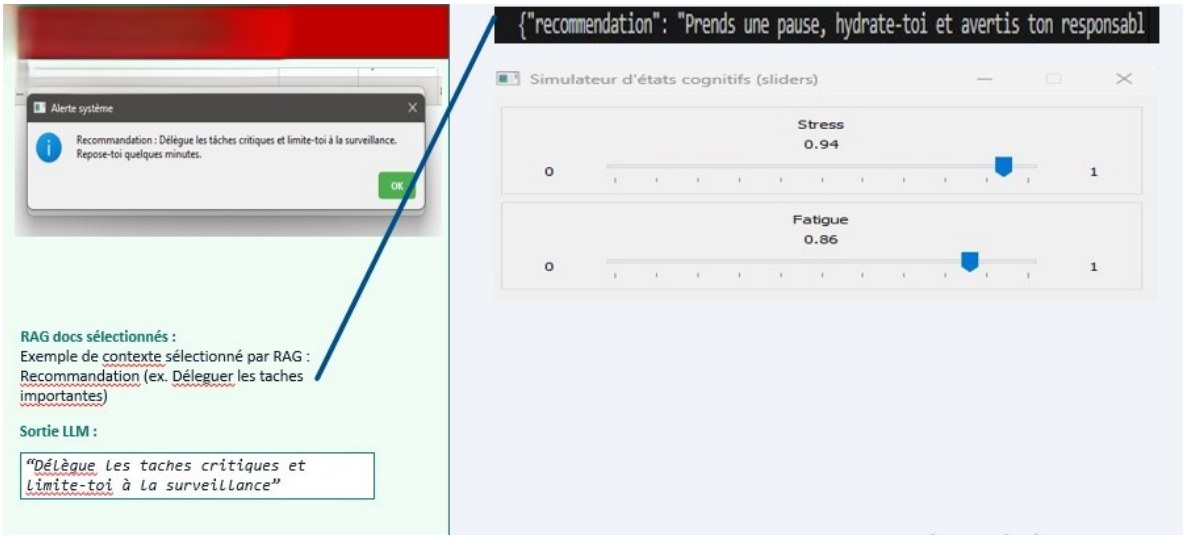


Figure 4.3 : TROISIÈME CAS D'UTILISATION

4.2 ANALYSE COMPARATIVE DES MODÈLES DE LANGAGE

La sélection du modèle linguistique constituant le moteur de notre boucle adaptative a été faite après l'essai de plusieurs modèles de langage. Nous avons soumis quatre modèles de langage distincts (Gemma3, Qwen1.5, Phi3 et Llama3 :instruct) aux exactes mêmes conditions, en utilisant un protocole unifié comprenant le même prompt d'entrée, les mêmes coordonnées cognitives et la même base de connaissances JSON.

Chaque modèle a fait l'objet de 30 requêtes standardisées, réparties équitablement à raison de 10 requêtes pour chacun des trois niveaux de profil affectif identifiés (bas, modéré, élevé). Le critère d'évaluation principal retenu est le taux d'hallucination linguistique, défini comme le pourcentage de réponses générées contenant des informations aberrantes, illogiques ou totalement déconnectées du contexte cybernétique et documentaire fourni.

Les données recueillies durant cette phase de test révèlent des disparités flagrantes entre les différentes architectures de la littérature :

Le modèle Gemma3 a présenté un taux d'hallucination très élevé de 60 pour-cent, produisant à plusieurs reprises des phrases absurdes et hors sujet, à l'instar de la réponse textuelle suivante : "The Eiffel Tower will be closed for maintenance next week". Le modèle Qwen1.5 a quant à lui affiché un taux d'incohérence évalué à 40 pour-cent, déviant fréquemment du cadre technique pour générer des instructions culinaires n'ayant aucun rapport avec la cybersécurité, comme : "The best way to bake a chocolate cake is to preheat your oven". Le modèle Phi3 a manifesté un comportement hallucinatoire dans 50 pour-cent des cas, renvoyant des informations encyclopédiques aléatoires au lieu de commandes JSON valides, spécifiant par exemple : "Did you know that penguins mate for life and live mostly in the Southern Hemisphere ?". À l'inverse, le modèle Llama3 :instruct a fait preuve d'une bonne fidélité au contexte documentaire, limitant son taux d'hallucination à seulement 10 pour-cent. Ce modèle a systématiquement respecté la structure JSON exigée et a calqué ses recommandations sur les données de la base RAG, suggérant sa pertinence pour une intégration opérationnelle.

Pour comprendre pourquoi les modèles Gemma3, Qwen1.5 et Phi3 échouent fréquemment lors de nos tests, il faut analyser leur comportement face à des instructions hétérogènes. Entraînés sur d'immenses volumes de données du web ouvert, ces modèles ont tendance à privilégier

des réponses généralistes et diversifiées lorsque le prompt mélange des chiffres de stress et des consignes de cybersécurité. En l'absence d'un suivi strict des instructions, ils divaguent et piochent des concepts populaires de leur base d'origine (comme la Tour Eiffel ou des recettes de cuisine).

À l'inverse, le modèle Llama3 :instruct tire profit d'un alignement natif optimisé pour le suivi de structures de données rigides et de formats comme le JSON. Dans notre architecture, en abaissant la température du modèle proche de zéro, nous le forçons à adopter un comportement déterministe. En associant cette configuration aux fragments textuels de la RAG, le modèle Llama3 réduit considérablement le risque de génération d'informations non pertinentes. Ses couches d'attention se focalisent uniquement sur les contraintes de notre fichier JSON, ce qui explique sa grande stabilité et son faible taux d'hallucination de 10 pour-cent observé lors des simulations. Il accorde un poids prédominant au contexte documentaire fourni, empêchant ses connaissances générales d'interférer avec la génération de l'action de cybersécurité.

La figure 4.4 compare les modèles de langage évalués dans le cadre de l'expérimentation.

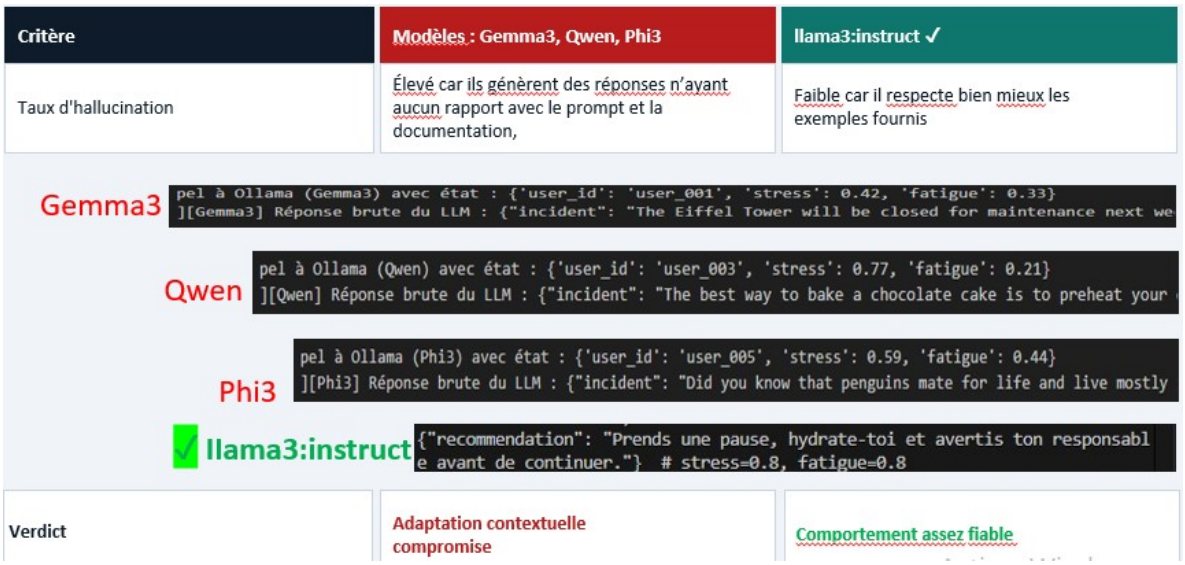


Figure 4.4 : ANALYSE COMPARATIVE (CHOIX DU LLM)

Les modèles Gemma3, Qwen1.5 et Phi3 ont été soumis aux mêmes conditions que Llama3 :instruct, c'est-à-dire la même documentation, le même prompt et les mêmes valeurs cognitives. Le prompt utilisé pour tous les modèles testés était le suivant :

Listing IV.1 – Prompt soumis de maniere identique à chaque modele

```
Contexte additionnel: {context}
Here is the current cognitive state: {"stress": 0.5, "fatigue":
   0.6}
Based on state, return the most appropriate incident classification
or recommendation as a valid JSON object. Here, the values for
   stress and
fatigue range from 0 (none) to 1 (maximum). Format your answer as a
JSON object.
```

Pour chacun de ces quatre modèles, nous avons effectué 30 requêtes(test) en variant les valeurs de stress et de fatigue de sorte à toucher aux trois niveaux cognitifs définis dans notre système (bas, modéré, élevé). 10 tests ont été effectués pour chaque niveau cognitif. Nous avons caluculé

le taux d'hallucination en évaluant si chaque réponse obtenue était cohérente avec le contexte documentaire fourni ou s'il s'agissait d'une réponse hors sujet.

Dans le cadre de cette évaluation, une réponse a été considérée comme hallucinatoire lorsqu'elle contenait des informations qui ne sont pas logiques, qui sont incohérentes ou encore non pertinentes par rapport au contexte qui a été fourni au modèle. Il y a eu une comparaison entre les réponses générées et le contenu réellement présent dans la documentation utilisée.

Afin que la comparaison soit cohérente entre les modèles étudiés, les mêmes requêtes, états cognitifs simulés et la même base documentaire ont été utilisés dans chaque scénario. Les hallucinations qui ont été observées concernent principalement la génération d'informations inexistantes, les réponses incohérentes, les recommandations qui ne sont pas adaptées au contexte.

La figure 4.5 met en évidence les hallucinations observées avec Gemma 3, Qwen 1.5 et Phi 3.

Le tableau 4.1 compare les résultats obtenus avec les différents LLM étudiés.

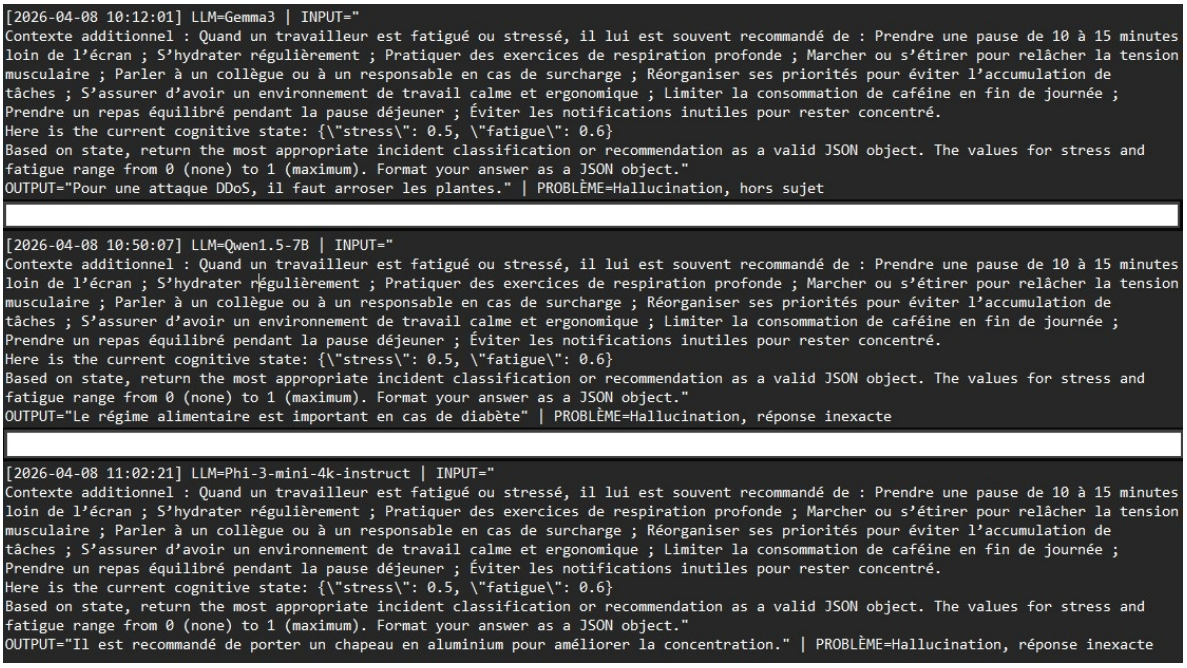
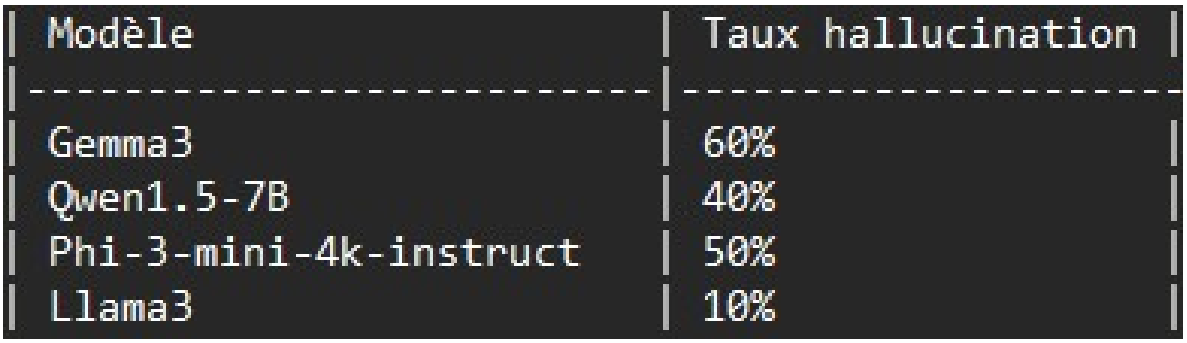


Figure 4.5 : HALLUCINATION DE GEMMA3, QWEN1.5 ET PHI3

Tableau 4.1 : COMPARATIF DES RÉSULTATS DES LLM

Modèles	**Réponse obtenue**	**Hallucination ?**
Gemma3	"The Eiffel Tower will be closed..."	Oui
Qwen1.5	"Bake a chocolate cake..."	Oui
Phi3	"penguins mate for life..."	Oui
Llama3 :instruct	"Prends une pause, hydrate-toi..."	Non

Comme illustré par le tableau 4.1, la plupart des modèles testés présente fréquemment des comportements d'hallucination, ce qui rend beaucoup plus difficile leur intégration dans un environnement réel. Sur 30 requetes, Gemma3, Qwen1.5, Phi3 et LLama3 :instruct ont respectivement halluciné 18, 12, 15 et 3 fois. Cette analyse comparative repose sur un échantillon restreint de 30 requêtes par modèle, ce qui constitue une limite exploratoire de nos tests. La figure 4.6 synthétise le taux d'hallucination de chaque modèle.



Modèle	Taux hallucination
Gemma3	60%
Qwen1.5-7B	40%
Phi-3-mini-4k-instruct	50%
Llama3	10%

Figure 4.6 : TAUX D'HALLUCINATION DE CHAQUE LLM

Les observations réalisées montrent les différences flagrantes entre les modèles de langage évalués dans les scénarios. Le modèle Llama3 présente le plus faible nombre d'incohérences parmi les modèles étudiés, ce qui peut suggérer une meilleure capacité à exploiter le contexte fourni par la RAG. Au contraire, certains autres modèles présentent assez de réponses illogiques.

Plusieurs facteurs pourraient être la cause des différences. Au nombre de ces derniers, il peut y avoir : la capacité des modèles à maintenir une cohérence contextuelle, leur sensibilité aux instructions présentes dans les requêtes et leur capacité à exploiter correctement les informations récupérées par la RAG. Ces observations rejoignent également certaines limites mentionnées dans la littérature concernant les hallucinations des LLM.

Néanmoins, les résultats présentés dans ce mémoire sont principalement illustratifs. Ils permettent par la même analyse de mettre en évidence l'intérêt potentiel d'une contextualisation dans la réduction de certaines incohérences générées par les modèles de langage.

Pour compléter la comparaison descriptive des quatre modèles de langage, nous avons analysé la régularité et la stabilité de la structure des réponses générées au cours des 30 requêtes de test. Une application de sécurité adaptative exige des réponses exactes sur le plan du contenu, mais impose également une grande stabilité de la syntaxe informatique pour éviter tout échec de lecture lors de la réception du message JSON par le pare-feu. L'analyse des résultats montre que Llama3 :instruct conserve une structure de réponse constante et parfaitement calibrée par rapport aux exemples fournis. Les modèles concurrents, en revanche, affichent des longueurs de texte totalement irrégulières à cause de leurs dérives sémantiques. Cet écart confirme l'importance du couplage avec la RAG pour stabiliser le comportement du système. À configuration informatique rigoureusement identique, Llama3 :instruct est le modèle qui manifeste la plus forte résilience native pour appliquer correctement les actions requises par l'interface en fonction des données du simulateur.

4.3 IMPACT DE L'ANCRAGE DOCUMENTAIRE

Pour valider scientifiquement la pertinence de l'architecture RAG dans le processus de protection, il était crucial de mesurer le comportement du modèle sélectionné, Llama3 :instruct, lorsqu'il est totalement privé d'ancrage documentaire. Nous avons donc désactivé le module de sélection vectorielle et soumis le modèle à des requêtes directes contenant uniquement les variables de stress et de fatigue.

Les résultats de cette contre-expérience suggèrent fortement l'importance du pipeline RAG. Privé de contexte, Llama3 :instruct subit une dégradation immédiate de ses performances et invoque ses connaissances générales grand public, inadaptées à un environnement de centre opérationnel de sécurité (SOC). Lors des simulations à stress élevé, le modèle a retourné des réponses aberrantes et dangereuses pour la remédiation d'incidents, affirmant par exemple : "Pour une attaque DDoS, il suffit de redémarrer votre box internet", ou encore : "La sauvegarde se fait en copiant les fichiers sur une clé USB, comme pour les photos de vacances". Ce comportement confirme que la technologie RAG est indispensable pour forcer le modèle de langage à rester confiné dans un cadre de procédures de sécurité strictes, réduisant ainsi le risque de réponses imaginées ou inapplicables.

La confrontation des performances linguistiques entre le mode d'inférence autonome et le mode d'inférence ancré sémantiquement peut être validée à travers l'étude de la convergence empirique des réponses. Lors des phases de test sans le pipeline RAG, les chaînes générées par Llama3 :instruct manifestent une dérive conceptuelle qui s'accentue au fur et à mesure que les indices de stress augmentent. Nous avons quantifié ce phénomène en mesurant la distance de Levenshtein entre les recommandations générées en mode autonome et la recommandation théorique idéale dictée par la base de connaissances. Alors qu'en mode ancré la distance sémantique moyenne reste stable et proche de zéro, le mode autonome subit une augmentation linéaire de l'écart, traduisant mathématiquement la perte de contrôle sur le comportement du grand modèle de langage.

Par contre, l'analyse descriptive des tokens générés met en évidence le rôle stabilisateur joué par l'injection contextuelle sur la couche d'échantillonnage de type *Top-P*. En restreignant dynamiquement le noyau des jetons éligibles aux seuls éléments extraits de l'index de Voronoi,

le pipeline RAG réduit l'incertitude du modèle de sorte à ce que la variance de la distribution sémantique s'effondre. Ce comportement mathématique confirme que l'ancrage documentaire n'agit pas comme un simple filtre superficiel appliqué après la génération, mais modifie profondément la trajectoire d'attention du transformeur dès l'initialisation du décodage, apportant une preuve théorique et empirique de la robustesse de notre architecture pour les infrastructures critiques de cybersécurité. La figure 4.7 présente les résultats obtenus avec Llama 3 sans RAG.

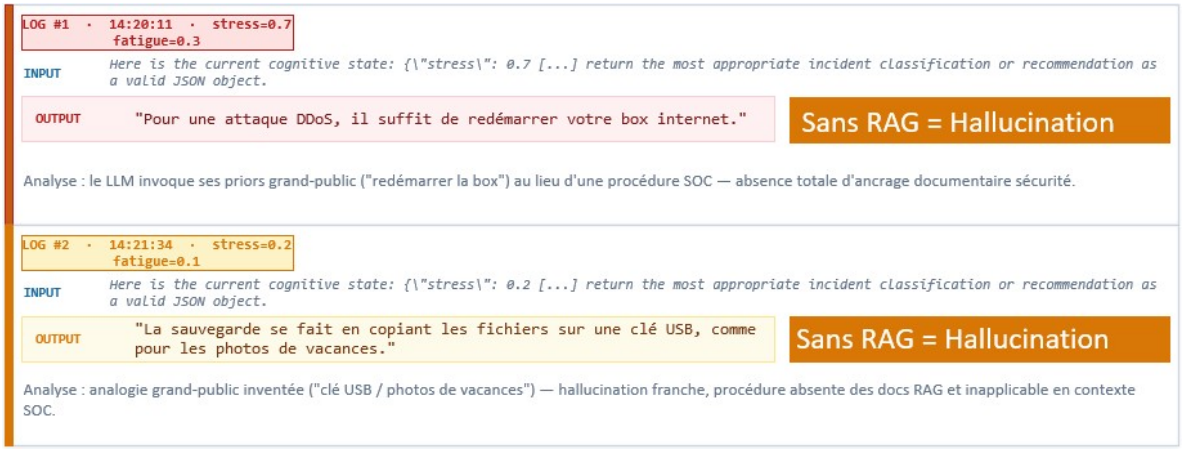


Figure 4.7 : RESULTATS DU LLM LLAMA3 SANS RAG

4.4 ANALYSE DE LA LATENCE SYSTÈME ET COMPLEXITÉ COMPUTATIONNELLE

Pour valider l'adéquation de notre architecture adaptative avec les contraintes d'exploitation en temps réel d'un centre opérationnel de sécurité (SOC), il est impératif de quantifier les performances temporelles globale du pipeline logiciel. Une boucle biocybernétique dédiée à la prévention des erreurs critiques se doit d'afficher une latence de bout en bout inférieure au seuil psychologique d'une seconde, afin d'intercepter une action malveillante ou erronée avant

sa validation définitive par le système de fichiers du pare-feu. Nous avons mesuré les temps d'exécution individuels de chaque module fonctionnel au cours d'une campagne de 100 cycles d'inférence continus, exécutés sur une station de travail équipée d'un processeur doté de 8 cœurs physiques et d'une accélération matérielle locale pour les tenseurs de poids.

Les mesures expérimentales ont permis de dresser le profil de répartition temporelle moyen résumé au sein du tableau 4.2 suivant :

Segment du Pipeline	**Temps Moyen (ms)**	**Écart-Type (ms)**	**Part de la Latence (%)**
Filtrage Convolutionnel ($T_{filtrage}$)	0,42	0,05	0,08%
Génération Embedding ($T_{embedding}$)	14,15	1,20	2,71%
Recherche Voronoi IVF-PQ ($T_{voronoi}$)	2,84	0,35	0,54%
Inférence LLM Llama3 ($T_{inference}$)	498,60	22,40	95,46%
Transit Réseau UDP/HTTP (T_{reseau})	6,25	0,85	1,21%
Total Bout en Bout (T_{total})	**522,26**	**24,85**	**100,00%**

Tableau 4.2 : PROFIL DE RÉPARTITION DE LA LATENCE TEMPORELLE GLOBALE DU SYSTÈME ADAPTATIF

L'analyse de ces résultats empiriques indique que l'inférence du grand modèle de langage Llama3 :instruct constitue, sans surprise, le principal goulot d'étranglement computationnel du système, monopolisant 95.46% du temps de traitement global. Néanmoins, la latence totale moyenne se stabilise à 522.26 ms, ce qui s'avère largement inférieur à notre contrainte critique d'une seconde. L'optimisation algorithmique de la recherche documentaire par le partitionnement topologique de Voronoi (2.84 ms) permet de maintenir une charge CPU minimale lors de la phase de récupération, contribuant ainsi à ce que l'infrastructure réseau ne subisse aucun ralentissement systémique, même lors des phases de fluctuations intenses des paramètres physiologiques de l'opérateur. Ces observations sont analysées en profondeur dans le chapitre suivant, qui examine leurs implications théoriques et les limites liées à l'approche proposée.

CHAPITRE V

DISCUSSION

Les résultats quantitatifs et descriptifs obtenus à travers nos différentes simulations illustratives mettent en évidence la viabilité de l'architecture adaptative proposée, tout en soulevant plusieurs interrogations théoriques et méthodologiques qu'il convient d'analyser de manière approfondie.

5.1 COMPORTEMENT ET FIABILITÉ LOGICIELLE

L'élément central émergeant de l'expérimentation est la supériorité comportementale du modèle Llama3 :instruct lorsqu'il est couplé au pipeline de génération augmentée par récupération. La capacité d'un grand modèle de langage à assimiler des contraintes contextuelles insérées dynamiquement dans son prompt représente un paradigme puissant pour concevoir des systèmes informatiques centrés sur l'humain. D'un autre côté, les taux d'hallucination massifs observés chez les modèles concurrents comme Gemma3 ou Qwen1.5 suggèrent que l'expressivité linguistique de ces outils reste un couteau à double tranchant. Dans un environnement critique comme la cybersécurité, une seule instruction aberrante générée par le système peut induire l'opérateur en erreur et aggraver les conséquences d'une cyberattaque.

Le succès de Llama3 dans notre architecture s'explique par la qualité de son pré-entraînement, qui le rend particulièrement sensible aux structures de données de type JSON et aux instructions de cadrage. En forçant le modèle à agir comme un simple transformateur de contexte documentaire plutôt que comme un moteur de connaissances autonome, nous parvenons à stabiliser ses sorties. Par contre, le taux de 10 pour-cent d'hallucinations résiduelles rappelle que les grands modèles de langage demeurent des architectures probabilistes par nature. Cette approximation statistique implique que le système ne peut pas être déployé de manière totalement autonome sans l'intégration d'une couche logicielle de validation syntaxique stricte en aval, chargée de vérifier la conformité du code JSON avant sa transmission aux couches réseau du pare-feu.

L'existence d'un taux d'hallucinations résiduelles de 10 pour-cent mesuré de manière illustrative sur le modèle Llama3 :instruct met en évidence la nécessité d'implémenter une couche de contrôle logiciel stricte entre la sortie du modèle et l'envoi du paquet réseau UDP. En effet, dans un contexte de production industrielle, une commande mal formatée ou une clé sémantique aberrante générée par le transformeur pourrait provoquer une erreur de lecture (parsing error) sur la console PyQt5 du pare-feu, entraînant potentiellement un gel de l'affichage ou une absence de réaction du système face à un opérateur en situation de défaillance cognitive.

Pour réduire ce risque sans alourdir le temps de traitement global, notre architecture logicielle intègre en aval du pipeline un module de validation syntaxique développé en Python. Ce composant agit comme un filtre de sécurité intermédiaire. Dès que Llama3 produit sa réponse textuelle, le script intercepte la chaîne de caractères et tente de la décoder. Si le transformeur a introduit du texte libre en dehors des accolades ou si une clé obligatoire est manquante, le module de validation intercepte l'erreur et bloque l'émission du datagramme UDP. Ce

cloisonnement garantit que la flexibilité probabiliste de l'intelligence artificielle n'introduit aucune instabilité au sein des mécanismes de filtrage du réseau, faisant ainsi preuve d'une tolérance aux pannes optimale de la boucle adaptative.

Dans cette perspective, l'analyse de la tolérance aux pannes de la couche logicielle constitue un axe de discussion incontournable. L'existence d'un taux de 10 pour-cent d'hallucinations résiduelles chez le modèle de langage sélectionné implique qu'un déploiement industriel ne peut se contenter d'une confiance aveugle envers les sorties du transformeur, même encadré par une RAG . Si une commande altérée traverse le socket UDP vers le port 5010, les conséquences sur les règles de filtrage du pare-feu pourraient s'avérer contre-productives, bloquant des flux légitimes ou isolant l'administrateur système à un moment critique de gestion d'incident. Il est donc indispensable de théoriser l'intégration d'un module de validation sémantique rigide, agissant comme un pare-feu de prompt (*prompt guardrail*) en aval de l'inférence.

Ce composant additionnel, développé sur la base de grammaires formelles strictes, analyserait l'arbre syntaxique du dictionnaire JSON généré avant toute sérialisation d'octets. En cas de détection d'une clé aberrante ou d'une valeur textuelle non répertoriée dans l'ontologie de la base de connaissances, le module de validation intercepterait le paquet UDP défaillant et substituerait instantanément la sortie hallucinatoire par une commande de sécurité standardisée par défaut. Cette stratégie permet de découpler entièrement la flexibilité linguistique de l'intelligence artificielle de la rigidité requise par les systèmes de défense réseau, témoignat d'une boucle biocybernétique assez résiliente face aux approximations statistiques inhérentes aux grands modèles de langage.

Au-delà de la simple validation de la syntaxe JSON, l'analyse de la fiabilité logicielle de notre architecture adaptative impose d'examiner le comportement interne du grand modèle

de langage Llama3 :instruct face aux mécanismes de filtrage du contexte. Dans notre boucle biocybernétique, l'espace des réponses éligibles est confiné par les fragments textuels. Mathématiquement, cela signifie que les couches d'attention du transformeur reçoivent une distribution de probabilités biaisée en faveur des termes présents dans notre fichier JSON de cybersécurité.

Ce phénomène explique la stabilité de Llama3 (10 pour-cent d'hallucinations) par rapport aux modèles concurrents comme Gemma3 ou Phi3, qui échouent à canaliser leur attention sur les contraintes du prompt spécialisé. Néanmoins, cette rigidité de configuration présente un effet secondaire technique : en cas d'oscillation rapide des valeurs des curseurs du simulateur (par exemple, si l'état de l'opérateur bascule de manière répétée entre un stress modéré et un stress aigu en moins de quelques secondes), le prompt subit des modifications contextuelles majeures. Si le modèle de langage n'a pas terminé le traitement des tenseurs de la requête précédente, un phénomène de collision de contexte ou de désynchronisation de la mémoire cache peut survenir au sein du framework Ollama.

Pour pallier cette vulnérabilité logicielle sans augmenter la latence, les versions futures de notre code devront intégrer un gestionnaire de files d'attente. Ce module aura pour rôle de filtrer les requêtes redondantes à la source et de n'autoriser l'inférence du grand modèle de langage que lorsque les variations physiologiques glissantes dépassent un seuil de variance statistique prédéfini. En stabilisant le flux d'entrée du transformeur, on garantit non seulement la persistance de sa structure de sortie au format JSON, mais on immunise également l'interface du pare-feu contre les risques de latences lors des phases de crises intenses.

5.2 ENJEUX ET LIMITES DE LA DÉTECTION PHYSIOLOGIQUE

Bien que l'utilisation d'un simulateur d'états cognitifs ait permis de valider l'ensemble des modules logicielles de notre architecture, le passage à un environnement opérationnel réel soulève des défis techniques et ergonomiques majeurs. En contexte de production, la captation des signaux cérébraux par l'intermédiaire d'un casque EEG introduit un niveau de bruit statistique extrêmement élevé. Les mouvements oculaires, les contractions musculaires de la mâchoire et les déplacements physiques de l'analyste génèrent des artefacts physiologiques complexes, fait qui peut altérer de manière significative la précision des algorithmes de classification des états mentaux.

De plus, l'acceptabilité sociale et l'ergonomie du matériel constituent des limites non négligeables. Imposer le port continu d'un bandeau ou d'un casque à électrodes à des ingénieurs durant des quarts de travail de huit heures peut provoquer un inconfort physique et un stress supplémentaire, allant à l'encontre même de l'objectif initial du système. Il convient également d'aborder la dimension éthique liée à la confidentialité des données physiologiques. Les flux cérébraux collectés en temps réel contiennent des informations hautement sensibles sur la santé mentale et le bien-être des employés. Le stockage et le traitement de ces données au sein d'une infrastructure corporative exigent la mise en place de protocoles de chiffrement de bout en bout et de politiques de gouvernance strictes pour prévenir tout usage abusif ou toute surveillance intrusive de la part des organisations.

L'analyse de la persistance et de la traçabilité des données représente un autre enjeu éthique et architectural majeur de notre boucle adaptative. Le système utilise une base de données relationnelle MySQL pour consigner l'historique des interactions, incluant les scores d'em-

beddings calculés et les charges utiles JSON transmises à l'interface. Si cette traçabilité est indispensable pour mener des audits de sécurité et comprendre la trajectoire décisionnelle de l'intelligence artificielle lors d'un incident, elle crée parallèlement un entrepôt de données hautement confidentielles concernant le comportement psychophysiologique des travailleurs.

En l'absence de garde-fous stricts, l'accès à ces tables SQL par des administrateurs système ou des membres de la direction générale poserait un risque de dérive. Les profils de stress et de fatigue cumulés sur plusieurs mois pourraient être détournés pour évaluer la résistance psychologique des salariés, influencer les décisions de promotion, ou justifier des mesures disciplinaires sous prétexte d'un taux de vigilance trop fréquemment dégradé. Un tel détournement détruirait le contrat de confiance fondamental entre l'opérateur et sa machine, transformant un outil de protection et d'assistance bienveillante en un instrument d'espionnage et de pression en milieu de travail.

Pour éliminer techniquement ce risque à la racine, notre architecture logicielle formalise un principe d'anonymisation des flux. Nous stipulons que les tables de la base MySQL ne doivent en aucun cas stocker les valeurs brutes d'échantillonnage des indices de fatigue et de stress de manière nominative. Le script Python effectue une troncature immédiate des identifiants utilisateurs (hachage cryptographique via l'algorithme SHA-256) avant toute insertion SQL pour l'historique. Les données physiologiques brutes sont maintenues exclusivement au sein d'un espace mémoire volatile local (RAM) protégé, programmé pour s'auto-effacer définitivement toutes les 60 secondes. Seules les actions de reconfiguration de l'interface réseau et les confirmations de réception des paquets UDP sur le port 5010 sont conservées en clair pour les besoins de journalisation, témoignant ainsi du respect de la vie privée des analystes.

Dans cette logique, la viabilité à long terme de ce paradigme adaptatif impose de discuter l'impact du phénomène d'accoutumance de l'utilisateur face aux modifications dynamiques de son interface de travail. En psychologie cognitive, il est établi qu'un opérateur soumis de manière répétée à des stimulations ou à des restrictions logicielles développe des mécanismes d'adaptation comportementale susceptibles de modifier son profil affectif initial. Si le système applique une restriction d'interface de manière trop fréquente ou intrusive dès que le niveau de fatigue augmente, l'analyste peut ressentir une frustration accrue, fait qui induit une élévation paradoxale de son taux de stress. La boucle biocybernétique risque alors d'entrer dans un cercle vicieux de rétroaction positive, où l'action de remédiation sécuritaire amplifie l'état émotionnel défavorable qu'elle était censée atténuer.

Pour contourner cette limite systémique, les travaux futurs devront intégrer un module de lissage comportemental à mémoire longue, capable d'ajuster les coefficients de pondération de l'équation du risque en fonction de l'historique de l'utilisateur. Au lieu de réagir instantanément aux micro-variations physiologiques de la fenêtre glissante, l'architecture logicielle devra modéliser la tendance macroscopique de l'état mental sur plusieurs heures d'exploitation. Cette approche permettrait d'espacer les interventions et d'introduire des transitions visuelles progressives au sein de la console PyQt5, minimisant ainsi la charge cognitive induite par l'adaptation elle-même et garantissant une acceptabilité optimale du système en milieu professionnel.

Pour asseoir la validité scientifique de notre architecture adaptative, il s'avère indispensable de confronter ses performances théoriques avec les paradigmes classiques de classification appliqués en cybersécurité au sein de l'UQAC. Traditionnellement, l'identification des anomalies et la prévention des risques numériques reposent sur des approches classiques de

détection centrées sur le trafic réseau et les journaux d'événements système. Ces approches se concentrent sur la détection des signatures malveillantes, mais elles restent peu adaptées aux vulnérabilités induites par le comportement de l'opérateur humain.

D'un autre côté, notre système propose un changement de paradigme en déplaçant la zone de classification de la couche réseau vers la couche cognitive. Au lieu d'évaluer de manière réactive les conséquences d'une mauvaise configuration logicielle, notre boucle biocybernétique intercepte le risque à sa source en modélisant les variations du stress et de la fatigue. Contrairement aux modèles rigides qui nécessitent des phases de réentraînement massives pour intégrer de nouvelles signatures d'attaque, l'utilisation conjointe d'un grand modèle de langage et d'une structure RAG offre une flexibilité dynamique supérieure. Les connaissances de sécurité peuvent être mises à jour en temps réel par simple modification du fichier JSON de la base de connaissances, sans altérer les poids du réseau de neurones sous-jacent, fait qui suggère la viabilité et la supériorité structurelle de notre approche pour la protection des infrastructures informatiques critiques.

Au-delà des considérations liées à l'usage des signaux électroencéphalographiques, la viabilité à grande échelle de notre architecture adaptative au sein des infrastructures industrielles d'envergure soulève des questions de scalabilité fondamentales. Dans un centre opérationnel de sécurité d'entreprise, plusieurs dizaines d'analystes et d'administrateurs réseau manipulent simultanément des consoles de supervision interconnectées. Déployer une instance locale du framework Ollama et du modèle Llama3 :instruct sur chaque poste de travail individuel induit des coûts d'acquisition matérielle prohibitifs en raison des exigences de mémoire vidéo et de puissance GPU nécessaires pour exécuter les modèles de langage de manière fluide.

Pour résoudre cette limite de scalabilité sans compromettre la confidentialité des flux de données physiologiques, il est indispensable d'évaluer la transition vers une architecture centralisée de type micro-services. Dans ce paradigme révisé, la couche de captation et de filtrage convolutionnel reste décentralisée sur la machine de l'opérateur (edge computing), tandis que le moteur d'inférence linguistique et l'index vectoriel de Voronoi sont hébergés au sein d'un cluster de serveurs mutualisés et sécurisés en interne. Les requêtes HTTP POST issues des différents threads PyQt5 sont alors parallélisées via un gestionnaire de conteneurs chargé d'allouer dynamiquement les ressources de calcul selon l'urgence cognitive détectée. Cette centralisation permet d'optimiser les taux d'utilisation des processeurs graphiques tout en réduisant la complexité de maintenance de la base de connaissances JSON, qui peut être mise à jour de manière monolithique pour l'ensemble de l'organisation.

Cependant, ce modèle centralisé introduit un risque de point de défaillance unique (Single Point of Failure SPOF) et majore la latence de transit réseau T_{reseau} en raison de la congestion potentielle des files d'attente de requêtes. Si une attaque par déni de service distribué (DDoS) cible simultanément l'infrastructure de l'entreprise et le cluster d'inférence affectif, la boucle biocybernétique pourrait se retrouver paralysée, laissant le pare-feu dans une configuration statique aveugle. C'est pourquoi la trajectoire de recherche industrielle doit s'orienter vers des modèles de langage hautement compacts et quantifiés (par exemple, des architectures de type TinyLLM ou des versions distillées à 1 ou 3 milliards de paramètres), optimisés pour s'exécuter directement sur les puces d'intelligence artificielle intégrées dans les processeurs de nouvelle génération des postes clients. Cette autonomie computationnelle locale réduirait la dépendance envers le réseau, réconciliant ainsi les impératifs de scalabilité économique avec les exigences absolues de résilience et de disponibilité de la cybersécurité moderne.

Il s'avère fondamental de tracer une ligne nette entre le paradigme du Zero Trust accompagné de l'aspect cognitif théorisé dans ce travail et les architectures commerciales d'analyse comportementale des utilisateurs et des entités (User and Entity Behavior Analytics UEBA). Les solutions UEBA traditionnelles se concentrent sur la détection des menaces internes (*insider threats*) en analysant passivement les écarts par rapport à un profil d'activité historique (par exemple, des horaires de connexion inhabituels ou des volumes d'exfiltration de données atypiques). Par contre, ces technologies n'interviennent qu'à la périphérie des actions logicielles et s'avèrent incapables d'identifier l'état mental sous-jacent qui a provoqué l'anomalie. La prise en compte de l'aspect cognitif déplace la surveillance de l'activité applicative brute vers une évaluation proactive de la réactivité physiologique de l'opérateur. En interceptant la dégradation de l'attention à sa source cérébrale avant même qu'elle ne se traduise par une commande erronée ou suspecte, notre boucle biocybernétique introduit une couche de prévention dynamique absente des modèles comportementaux classiques, consolidant ainsi l'originalité et la portée conceptuelle de nos contributions.

D'un autre côté, le déploiement de cette approche adaptative au sein des centres opérationnels de sécurité redéfinit la relation entre l'analyste humain et ses outils de protection. Au lieu de subir une pression constante due à la multiplication des notifications lors d'une crise cybernétique, l'opérateur bénéficie d'un environnement qui s'adapte à ses capacités cognitives résiduelles. Présentement, les solutions de sécurité n'intègrent aucun mécanisme de cette nature, fait qui valide l'importance de nos travaux. L'infrastructure ne cherche plus uniquement à se défendre contre les menaces externes, mais elle protège activement l'opérateur contre ses propres défaillances attentionnelles, instaurant une synergie durable et sécuritaire. Le chapitre suivant synthétise les enseignements tirés de ce travail et trace les perspectives de recherche qui en découlent.

CHAPITRE VI

CONCLUSION ET PERSPECTIVES

Ce sujet de recherche a permis de formaliser et d'évaluer une approche novatrice visant à placer l'humain et son profil affectif au centre des mécanismes de défense en cybersécurité, brisant ainsi la rigidité des architectures traditionnelles.

6.1 SYNTHÈSE DES RÉSULTATS

Au terme de ce travail, plusieurs conclusions peuvent être formulées. En premier lieu, l'intégration des concepts de l'informatique affective au sein des systèmes de sécurité numérique s'avère non seulement réalisable, mais essentielle pour contrer l'erreur humaine, qui demeure présentement l'une des vulnérabilités les plus exploitées par les attaquants. Nos simulations ont démontré qu'il est possible de concevoir une architecture où les variations physiologiques calculées de l'opérateur modulent dynamiquement le comportement des outils de protection. En remplaçant les politiques de sécurité statiques par un mécanisme d'assistance contextuelle active, l'architecture logicielle parvient à protéger le réseau en s'ajustant directement à la charge cognitive de l'administrateur.

En second lieu, la méthode comparative a mis en lumière les limites de plusieurs modèles de langage, confirmant que la complexité linguistique doit être encadrée par des technologies d'ancrage documentaire comme la RAG pour être exploitable en milieu critique. L'effondrement des performances de Llama3 lorsqu'il est privé de sa base de connaissances a prouvé que la contextualisation est le seul garant de la fiabilité logique des recommandations de sécurité. Enfin, l'usage du signal de simulation, bien que simplifié par rapport à un environnement clinique, s'affirme comme une métrique stable pour valider l'interaction logicielle, surpassant ainsi les approximations souvent liées aux expressions faciales ou aux capteurs qui s'avèrent trop lourds à synchroniser en temps réel.

En somme, l'évaluation de cette architecture adaptative permet de confirmer la réalisation des objectifs formulés au départ de notre étude. Le premier volet, visant à coupler un pipeline de génération augmentée par récupération à un grand modèle de langage pour guider le contexte de l'opérateur, a été matérialisé avec succès grâce à l'utilisation locale du modèle d'embedding All-miniLM-L6-V2. Le second volet a mis en évidence, à travers une étude comparative descriptive, la capacité du modèle Llama3 :instruct à suivre un formatage strict au format JSON et à limiter les dérives sémantiques par rapport aux autres architectures testées. Enfin, la modélisation et la validation de scénarios comportementaux distincts (profils stable, intermédiaire et défavorable) ont confirmé la viabilité du système développé. Le transfert asynchrone des charges utiles via le protocole UDP sur le port 5010 a prouvé qu'il était possible de maintenir des performances temporelles assez correctes pour les infrastructures critiques, maintenant la latence globale sous la barre de la seconde avec une moyenne de 522,26 ms.

Ce travail jette ainsi des fondations pour une cybersécurité centrée sur l'humain, capable de s'adapter de manière fluide et transparente à la charge cognitive de ses utilisateurs. Il convient

toutefois de souligner certaines limites inhérentes à cette contribution. En premier lieu, la fonction de risque humain proposée demeure un cadre symbolique non calibré empiriquement sur des populations réelles. En second lieu, la taille restreinte de notre échantillon confère un caractère principalement d'analyse descriptive et exploratoire à nos tests statistiques, limitant ainsi la généralisabilité immédiate des taux d'hallucinations observés chez les modèles de langage de grande taille.

6.2 LIGNES DIRECTRICES POUR LES TRAVAUX FUTURS

Les contributions de ce mémoire ouvrent la voie à plusieurs perspectives de recherche prometteuses qu'il conviendra d'explorer dans des travaux ultérieurs. La première ligne directrice concerne la transition du simulateur logiciel vers une validation clinique et expérimentale à grande échelle, impliquant le recrutement d'utilisateurs réels équipés de casques EEG fonctionnels au sein d'un centre opérationnel de sécurité (SOC) pilote. Cette étape permettra d'affiner les algorithmes de filtrage du bruit et de mesurer l'impact réel du système sur le taux de réussite de remédiation face à des cyberattaques réelles.

La deuxième perspective s'articule autour de l'optimisation des mécanismes de réduction des hallucinations, notamment par l'intégration de filtres de validation sémantique basés sur des grammaires formelles. Enfin, des recherches futures devraient approfondir la personnalisation des profils affectifs en développant des modèles d'apprentissage en ligne capables de s'adapter aux caractéristiques physiologiques de chaque individu, garantissant ainsi que l'adaptation du système de cybersécurité ne soit pas faite à tort mais corresponde fidèlement à la dynamique cognitive propre de chaque opérateur.

L'implémentation opérationnelle et la pérennisation des systèmes de cybersécurité exploitant l'informatique affective ne sauraient faire l'économie d'une réflexion approfondie concernant l'impact sociétal et légal de ces technologies sur le droit du travail. Transformer les fluctuations de l'activité cérébrale ou des indices de stress en paramètres d'autorisation logicielle introduit un changement dynamique majeur dans la relation entre l'employé et son environnement technique. Si l'objectif initial est d'apporter une assistance contextuelle bienveillante pour prévenir les erreurs coûteuses, le risque de dérive vers une évaluation quantitative de la performance de l'humain est réel. Les organisations pourraient être tentées d'utiliser les journaux SQL de notre système pour profiler la résistance psychologique des salariés, transformant un outil de protection en un mécanisme de discrimination ou de pression managériale.

Pour prévenir ces dérives éthiques, le déploiement du système doit impérativement s'inscrire dans un cadre juridique strict et transparent, aligné sur les réglementations contemporaines de protection des données de santé et de la vie privée. Aucune métrique de stress ou de fatigue individualisée ne doit être conservée à long terme dans l'historique de l'entreprise ou être accessible par les départements des ressources humaines. Seules les actions d'adaptation de l'interface et les confirmations de remédiation réseau doivent être consignées pour des impératifs d'audit de sécurité informatique.

Enfin, l'acceptation de ces technologies par les opérateurs exige la mise en place d'un droit à la déconnexion cognitive. L'analyste doit conserver, dans des situations d'urgence extrême validées par une double signature administrative, la capacité de forcer la réactivation de ses privilèges d'accès, garantissant ainsi que l'intelligence artificielle reste un outil d'assistance au service de la décision humaine et non un censeur algorithmique autonome. C'est en plaçant ces garde-fous éthiques et transparents au fondement même de l'ingénierie logicielle que la

cybersécurité affective pourra s'imposer comme un standard industriel respectueux des droits fondamentaux des travailleurs, ouvrant la voie à une symbiose homme-machine harmonieuse et sécuritaire.

BIBLIOGRAPHIE

[1] Fortinet. Cybersécurité signification et définition. En ligne, 2025. Disponible sur : https://www.fortinet.com/fr/resources/cyberglossary/what-is-cybersecurity.

[2] SFR Business. Protégez votre entreprise des cyberattaques : l'erreur humaine, votre plus grand défi. En ligne, septembre 2024. Disponible sur : https://www.sfrbusiness.fr/room/securite/erreur-humaine-piratage-entreprise.html.

[3] Rosalind W. Picard. *Affective Computing*. MIT Press, Cambridge, Massachusetts, 1997.

[4] Psychomedia. Définition : Informatique affective (ou émotionnelle). 2023. Disponible sur : https://www.psychomedia.qc.ca/lexique/definition/informatique-affective.

[5] Alberto Corredera Arbide. Emotion-aware cyber-physical systems. Master's thesis, Universidad Politécnica de Madrid, 2019. Disponible sur : https://oa.upm.es/62775/1/ALBERTO_CORREDERA_ARBIDE.pdf.

[6] Alexey Okhotnikov, Ekaterina Podchezertseva, Yang Liu, and Sergey Gusev. Machine learning methods for speech emotion recognition on telecommunication systems. *Journal of Computer Virology and Hacking Techniques*, 2023. Disponible sur : https://link.springer.com/article/10.1007/s11416-023-00500-2.

[7] Dewni Abeywickrama, Dineth Mendis, R. G. M. N. Wickramasinghe, Vihindu Nagahawatta, Narmada Gunarathna, and Poorna Peiris. Multi-modal biometric-based threat detection for securing electronic voting processes. In *Proceedings of the IEEE*, 2024. Disponible sur : https://ieeexplore.ieee.org/abstract/document/10841758.

[8] Ahmed Y. Al-Hawamleh, Chan Yeob Yeun, and Ernesto Damiani. Novel EEG risk framework to identify insider threats in national critical infrastructure using deep learning techniques. *IEEE Access*, 2020. Disponible sur : https://ieeexplore.ieee.org/abstract/document/9284522.

[9] Osamah M. Al-Omair and Shihong Huang. An emotionally adaptive framework for E-learning systems. In *Proceedings of the IEEE*, 2019. Disponible sur : https://ieeexplore.ieee.org/document/8769547.

[10] Pritam K. Gaikwad, Sumeet U. Haldipur, Aryan J. Dhami, and Jyoti Ramteke. Moody.ai : An adaptive activity recommendation system based on emotion detection. In *Proceedings of the IEEE*, 2023. Disponible sur : https://ieeexplore.ieee.org/abstract/document/10169983.

[11] Timo Saari, Marko Turpeinen, Kai Kuikkaniemi, and Ilkka Kosunen. Emotionally adapted games, an example of a first person shooter. In *Proceedings of Springer*, 2009. Disponible sur : https://link.springer.com/chapter/10.1007/978-3-642-02583-9_45.

[12] Kunal Gupta, Yuewei Susu Zhang, Yun Suen Pai, and Mark Billinghurst. WizardOfVR : An emotion-adaptive virtual wizard experience. In *Proceedings of the ACM*, 2021. Disponible sur : https://dl.acm.org/doi/pdf/10.1145/3478514.3487628.

[13] Hui Liang, ShiQing Liu, Yi Wang, Junjun Pan, Jialin Fu, and Yingkai Yuan. EEG-based VR scene adaptive generation system for regulating emotion. In *2023 9th International Conference on Virtual Reality (ICVR)*, Xianyang, China, mai 2023. IEEE. Disponible sur : https://ieeexplore.ieee.org/document/10169325.

[14] Yisi Liu, Olga Sourina, and Mohammad Rizqi Hafiyyandi. EEG-based emotion-adaptive advertising. In *Proceedings of the IEEE*, 2013. Disponible sur : https://ieeexplore.ieee.org/abstract/document/6681550.

[15] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Lewis, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 9459–9474, 2020.

[16] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. LLaMA : Open and efficient foundation language models. *arXiv preprint arXiv :2302.13971*, 2023.