UQAC
Université du Québec
à Chicoutimi

DÉVELOPPEMENT D'UN MODÈLE DE QUALITÉ DES DONNÉES POUR LES ENVIRONNEMENTS DEVSECOPS INTÉGRANT DES SYSTÈMES D'IA : LE CAS DES DONNÉES DE VULNÉRABILITÉS

PAR FATOU BEYE

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

L'utilisation croissante de l'intelligence artificielle (IA) pour automatiser la détection, l'analyse et la priorisation des vulnérabilités dans les environnements DevSecOps a fait de la qualité des données un facteur déterminant de la fiabilité des mécanismes automatisés de cybersécurité. Toutefois, les modèles existants en matière de qualité des données demeurent principalement génériques et ne prennent pas explicitement en compte les exigences propres aux environnements DevSecOps intégrant des systèmes d'IA.

Ce travail propose un modèle consolidé de qualité des données, adapté à ce contexte, fondé sur la consolidation des normes existantes et des travaux de recherche portant sur la qualité des données, l'intelligence artificielle et la cybersécurité. Ce modèle est soutenu par une méthodologie structurée de mesure d'analyse, de nettoyage, et d'évaluation de la qualité des données. Son application à la National Vulnerability Database (NVD) a permis d'identifier plusieurs catégories d'anomalies, notamment des valeurs manquantes, des doublons, des incohérences temporelles et des identifiants Common Platform Enumeration (CPE) invalides.

Les résultats démontrent l'efficacité du modèle proposé pour identifier et corriger les anomalies des données de vulnérabilités dans les jeux de données de la NVD couvrant la période de 2021 à 2025. À titre d'exemple, pour le jeu de données de 2025, le nettoyage des données a permis d'améliorer l'exhaustivité de 89,20 % à 98,01 %, l'exactitude de 85,26 % à 99,99 % et l'unicité de 54,73 % à 91,05 %.

Ces résultats mettent en évidence la capacité du modèle à améliorer la qualité des données et à fournir un cadre systématique pour leur gestion dans les environnements DevSecOps intégrant des systèmes d'IA.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES ABRÉVIATIONS

API	Application Programming Interface
CI/CD	Continuous Integration / Continuous Deployment
CVE	Common Vulnerabilities and Exposures
CVSS	Common Vulnerability Scoring System
CWE	Common Weakness Enumeration
CSRF	Cross-Site Request Forgery
DAST	Dynamic Application Security Testing
DevOps	Development and Operations
DevSecOps	Development, Security and Operations
GPT	Generative Pre-trained Transformer
HITL	Human-In-The-Loop
HTTP	HyperText Transfer Protocol
IA	Intelligence Artificielle
IaC	Infrastructure as Code
ISO	International Organization for Standardization
LLM	Large Language Model
ML	Machine Learning
SAST	Static Application Security Testing
SCA	Software Composition Analysis
SaC	Security as Code
SQL	Structured Query Language
SQuaRE	Software Quality Requirements and Evaluation
XSS	Cross-Site Scripting

DÉDICACE

À la mémoire de ma défunte mère, Mariama Soumah, dont l'amour et le souvenir continuent de m'accompagner chaque jour. Malgré son absence, elle demeure une source permanente de courage, de motivation et de persévérance dans ma vie.

À mon père, Morlaye Beye, pour ses sacrifices, son soutien et les valeurs qu'il m'a transmises tout au long de mon parcours.

À ma tante et homonyme, Fatou Gaye, qui m'a élevée avec amour, patience et bienveillance, et qui a toujours été présente dans les moments les plus importants de ma vie.

À ma grand-mère, Fatoumata N'gady Camara, qui m'a accueillie et entourée d'affection après le décès de ma mère. Sa présence et son soutien ont occupé une place essentielle dans mon parcours personnel et académique.

À ma sœur, Oumou Beye, pour son affection, son soutien et ses encouragements constants.

À mes tantes, Aicha Soumah et Tady Soumah, pour leur présence, leur affection, leurs conseils et leur soutien tout au long de mon parcours.

À mon directeur de recherche, le professeur Jonathan Roy, pour son encadrement, sa disponibilité, ses conseils et son accompagnement tout au long de ce travail.

À toutes les personnes qui m'ont soutenue, encouragée et accompagnée durant ce parcours académique et personnel. Leur présence, leurs conseils et leur confiance ont contribué, chacune à leur manière, à l'aboutissement de ce mémoire.

REMERCIEMENTS

Je rends grâce à Allah (SWT) pour m'avoir donné la force, la santé, le courage et la persévérance nécessaires pour mener ce travail à son terme, malgré les difficultés rencontrées tout au long de ce parcours.

Mes sincères remerciements s'adressent à mon directeur de recherche, le professeur Jonathan Roy, pour son encadrement, sa disponibilité, ses conseils et son accompagnement tout au long de cette recherche. Ses orientations et ses remarques ont grandement contribué à la réalisation de ce mémoire.

Je remercie également l'ensemble des professeurs ainsi que le personnel du Département d'informatique et de mathématique (DIM) de l'Université du Québec à Chicoutimi (UQAC), pour la qualité de leur enseignement, leur soutien et leur disponibilité durant mon parcours académique.

J'adresse aussi mes remerciements aux personnes qui m'ont accueillie au Canada et qui ont contribué, de près ou de loin, à mon intégration et à mon adaptation dans ce nouvel environnement.

J'adresse une pensée particulière à mon amie Léonie Eva Hann, rencontrée ici au Canada, pour son soutien, sa présence et son amitié durant cette aventure académique et personnelle.

Mes remerciements vont également à Ernest Ndecki pour son soutien et ses encouragements.

Enfin, je remercie profondément ma famille pour son amour, ses sacrifices, son soutien moral et ses prières tout au long de mon parcours.

CHAPITRE I
INTRODUCTION

1.1 CONTEXTE

Au cours des dernières années, les environnements de développement logiciel ont connu une transformation profonde. Les organisations doivent désormais concevoir, tester et déployer des applications à un rythme beaucoup plus soutenu afin de répondre aux besoins du marché et à l'évolution constante des technologies numériques. Les applications sont régulièrement modifiées et mises à jour, ce qui rend la gestion de la sécurité de plus en plus complexe [1, 18]. Dans ce contexte, les approches de sécurité appliquées seulement à la fin du développement ne suffisent plus à répondre pleinement aux exigences des systèmes logiciels actuels.

Cette évolution a favorisé l'émergence du DevSecOps, une approche qui cherche à intégrer la sécurité directement dans l'ensemble du cycle de développement logiciel plutôt que de la traiter comme une activité séparée [12, 4]. Le DevSecOps repose principalement sur l'intégration continue de la sécurité dans les différentes étapes du développement logiciel, grâce à l'automatisation de plusieurs contrôles et une collaboration plus étroite entre les équipes impliquées dans le cycle de développement [1, 18]. Les pipelines intègrent aujourd'hui différents outils capables d'analyser automatiquement le code, d'identifier certaines vulnérabilités et d'assurer une surveillance continue des infrastructures afin de limiter les risques de sécurité.

Parallèlement, les cybermenaces deviennent plus nombreuses et plus complexes. Les bases de données de vulnérabilités comme la National Vulnerability Database (NVD) enregistrent chaque année un nombre croissant de failles de sécurité touchant les applications, les infrastructures cloud, les bibliothèques logicielles et les composants open source. Selon FIRST,

49 183 CVE avaient déjà été publiées à deux jours de la fin de l'année 2025 [42]. À titre de comparaison, CVE Details recense environ 40 313 CVE en 2024, 29 066 en 2023 et 18 323 en 2020, ce qui confirme la forte croissance du volume annuel des vulnérabilités recensées [15]. Cette évolution rend la détection rapide des vulnérabilités particulièrement importante dans les environnements DevSecOps. Les mécanismes de détection ont progressivement évolué vers des approches plus automatisées intégrant différentes techniques d'analyse et d'intelligence artificielle [31, 27].

L'intégration croissante de l'intelligence artificielle dans les pipelines DevSecOps transforme également les pratiques de sécurisation des environnements logiciels. Les systèmes intelligents occupent désormais une place importante dans les mécanismes d'analyse, de surveillance et d'assistance à la prise de décision au sein des pipelines logiciels [17, 30]. Ces approches renforcent l'automatisation des mécanismes de sécurité et permettent d'améliorer certaines capacités d'analyse dans des environnements logiciels devenus particulièrement dynamiques.

Toutefois, l'efficacité de ces approches dépend fortement de la qualité des données utilisées pendant les phases d'apprentissage et d'analyse [29, 16]. Lorsque les données sont incomplètes, incohérentes ou peu fiables, les modèles peuvent produire des résultats moins fiables et produire des analyses qui ne reflètent pas correctement la réalité observée. Cette problématique prend une importance particulière dans les environnements DevSecOps, où les mécanismes automatisés reposent continuellement sur de grandes quantités de données issues des pipelines logiciels afin d'analyser, détecter et traiter les vulnérabilités. Dans ce contexte, la qualité des données apparaît comme un élément essentiel pour renforcer la fiabilité des mécanismes d'analyse utilisés en cybersécurité.

1.2 PROBLÉMATIQUE

Malgré les progrès réalisés dans l'automatisation des pipelines DevSecOps, plusieurs difficultés persistent concernant la qualité des données exploitées par les mécanismes d'analyse et de prédiction. Les données utilisées dans les environnements de cybersécurité présentent souvent plusieurs problèmes de qualité pouvant compliquer leur exploitation dans les processus automatisés [14, 9]. Ces problèmes peuvent réduire la fiabilité des prédictions et affecter les performances des modèles intégrés aux pipelines DevSecOps.

Les données de vulnérabilités issues de bases telles que la NVD peuvent notamment contenir des valeurs manquantes, des doublons, des incohérences de représentation, des informations peu exploitables ou des informations temporelles difficiles à interpréter. Ces anomalies peuvent limiter l'efficacité des mécanismes automatisés de détection, de priorisation et d'aide à la décision. Pourtant, la question de la qualité des données appliquée spécifiquement aux environnements DevSecOps demeure relativement insuffisamment étudiée dans la littérature scientifique.

Il apparaît donc nécessaire de disposer d'un modèle de qualité permettant d'évaluer, de caractériser et d'améliorer les données de vulnérabilités utilisées pour l'entraînement et l'utilisation de systèmes d'IA en environnement DevSecOps.

1.3 QUESTIONS DE RECHERCHE

Dans ce contexte, cette recherche vise à répondre aux questions de recherche suivantes :

- **RQ1 :** Quels standards internationaux proposent des modèles ou des caractéristiques de qualité des données applicables aux systèmes d'IA ?

- **RQ2 :** Quels modèles de qualité des données pour les systèmes d'IA ont été proposés dans la littérature scientifique, et quelles caractéristiques sont pertinentes pour les environnements DevSecOps ?
- **RQ3 :** Quelles caractéristiques de qualité des données, identifiées dans la littérature scientifique, ne sont pas couvertes par les standards existants, et comment concevoir un modèle consolidé adapté aux environnements DevSecOps intégrant des systèmes d'IA ?

1.4 OBJECTIFS DE RECHERCHE

L'objectif principal de cette recherche est de proposer un modèle consolidé de qualité des données adapté aux environnements DevSecOps intégrant des systèmes d'IA, puis de l'appliquer à l'évaluation et à l'amélioration des données de vulnérabilités issues de la NVD.

Plus spécifiquement, cette recherche poursuit les objectifs suivants :

- Identifier les standards internationaux proposant des modèles ou caractéristiques de qualité des données applicables aux systèmes d'IA ;
- Analyser les modèles et dimensions de qualité des données proposés dans la littérature scientifique pour l'IA, l'apprentissage automatique et les jeux de données de vulnérabilités ;
- Comparer les caractéristiques issues des standards et de la littérature scientifique afin d'identifier celles qui sont pertinentes pour les environnements DevSecOps ;
- Concevoir un modèle consolidé de qualité des données adapté aux environnements DevSecOps intégrant des systèmes d'intelligence artificielle.

Afin de valider le modèle de qualité des données consolidé, deux objectifs de validation sont poursuivis :

- Identifier et caractériser les problèmes de qualité affectant les données de vulnérabilités issues de la NVD selon les dimensions du modèle de qualité des données consolidé.
- Évaluer l'amélioration de la qualité des données de vulnérabilités issues de la NVD avant et après l'application d'approches de nettoyage, à l'aide des mesures définies par le modèle consolidé.

1.5 MÉTHODOLOGIE GÉNÉRALE

Afin de répondre aux questions de recherche, cette étude adopte une démarche organisée en deux phases complémentaires.

La première phase consiste à analyser les standards internationaux et la littérature scientifique afin d'identifier les caractéristiques de qualité des données pertinentes pour les systèmes d'IA et les environnements DevSecOps. Cette analyse permet de mettre en évidence les convergences et les écarts entre les approches normatives et académiques, puis de proposer un modèle consolidé de qualité des données.

La seconde phase vise à valider l'applicabilité du modèle proposé à travers une étude de cas portant sur les données de vulnérabilités issues de la NVD. Le modèle consolidé est d'abord traduit en règles de mesure permettant d'identifier et de caractériser les problèmes de qualité affectant ces données selon différentes dimensions, notamment la complétude, l'unicité, la cohérence, l'exactitude, la crédibilité et l'actualité. Des mécanismes de nettoyage et de validation sont ensuite appliqués afin de corriger les anomalies identifiées. Enfin, l'évolution de la qualité des données est évaluée à l'aide des mesures définies par le modèle consolidé.

1.6 STRUCTURE DU MÉMOIRE

Ce mémoire est structuré en huit chapitres.

- Le premier chapitre présente l'introduction générale, le contexte de recherche, la problématique, les questions de recherche, les objectifs ainsi que la méthodologie générale de cette étude.
- Le deuxième chapitre expose le contexte général du DevSecOps, les vulnérabilités logicielles ainsi que l'automatisation intelligente dans les environnements de cybersécurité.
- Le troisième chapitre présente la revue de la littérature. Il aborde les concepts fondamentaux liés à la qualité des données, les standards ISO/IEC ainsi que les travaux scientifiques portant sur la qualité des données, l'intelligence artificielle, l'apprentissage automatique et la cybersécurité.
- Le quatrième chapitre présente le modèle consolidé de qualité des données proposé dans cette recherche. Il décrit les caractéristiques retenues, leur origine dans les standards et la littérature, ainsi que leur organisation dans un modèle adapté aux environnements DevSecOps intégrant des systèmes d'IA.
- Le cinquième chapitre décrit la méthodologie de mesure et d'amélioration de la qualité des données. Il présente le cas d'étude fondé sur les données de vulnérabilités de la NVD, les règles de mesure, les indicateurs de conformité ainsi que les deux phases de nettoyage appliquées aux données.
- Le sixième chapitre présente et analyse les résultats obtenus avant et après les différentes phases de nettoyage. Il compare les données brutes, les données après le premier nettoyage et les données finales après le second nettoyage.
- Le septième chapitre propose une discussion générale des résultats. Il positionne l'étude par rapport aux travaux existants, présente les limites de la recherche et ouvre des perspectives futures.
- Le huitième chapitre présente la conclusion générale du mémoire.

CHAPITRE II

CONTEXTE

2.1 DEVSECOPS

2.1.1 CONTEXTE ET ÉVOLUTION DU DEVSECOPS

L'évolution rapide des technologies numériques a profondément transformé les pratiques de développement logiciel. Les équipes sont désormais amenées à concevoir, tester et déployer des applications dans des délais toujours plus courts, tout en s'adaptant aux besoins changeants des utilisateurs et aux avancées technologiques. L'adoption des approches Agile, DevOps et des pipelines CI/CD a permis d'accélérer considérablement les cycles de développement et de déploiement des logiciels [1, 18, 34]. Cependant, cette accélération a également mis en évidence plusieurs limites des approches traditionnelles de sécurité, dans lesquelles les mécanismes de protection étaient souvent appliqués uniquement à la fin du cycle de développement.

Dans les environnements modernes, les applications évoluent en permanence à travers des cycles fréquents de modification, de test et de déploiement. Les équipes de développement doivent gérer un nombre important de composants logiciels, de bibliothèques open source, de services cloud et d'infrastructures automatisées. Dans ce contexte, les contrôles de sécurité réalisés manuellement deviennent difficiles à maintenir et risquent de ralentir fortement les processus de livraison [35]. Les vulnérabilités peuvent alors être détectées trop tardivement, parfois après la mise en production des applications, ce qui augmente les risques de compromission et les coûts de correction [37].

Pour répondre à ces défis, le DevSecOps s'est progressivement imposé comme une évolution du DevOps visant à intégrer la sécurité à l'ensemble du cycle de développement logiciel [1]. Contrairement aux approches classiques où la sécurité était traitée comme une étape séparée, le DevSecOps propose une intégration continue des mécanismes de sécurité depuis la phase de développement jusqu'au déploiement et à la supervision des applications [4]. Il repose principalement sur l'automatisation des contrôles de sécurité, la collaboration entre les équipes ainsi que l'adoption d'une culture de sécurité partagée [1].

Le DevSecOps ne cherche donc pas uniquement à sécuriser les applications, mais aussi à maintenir l'équilibre entre rapidité de livraison, qualité logicielle et gestion des risques [33]. Dans les infrastructures cloud et distribuées, l'intégration continue de la sécurité devient essentielle pour détecter plus rapidement les vulnérabilités et réduire l'exposition aux cybermenaces [6]. La littérature récente souligne également que l'automatisation occupe désormais une place centrale dans cette approche, notamment à travers l'intégration d'outils d'analyse de sécurité, de mécanismes de surveillance continue ainsi que de techniques plus récentes fondées sur l'intelligence artificielle [37, 35].

Afin de mieux comprendre cette approche, les sections suivantes présenteront les principaux concepts et mécanismes du DevSecOps, ainsi que les pratiques permettant d'intégrer la sécurité de façon continue dans les environnements de développement logiciel.

2.1.2 LES PILIERS DE DEVSECOPS

Le DevSecOps repose sur plusieurs principes permettant d'intégrer la sécurité dans les pipelines logiciels sans ralentir les cycles de développement. La littérature présente généralement cette approche autour de trois piliers principaux : l'automatisation de la sécurité, la collaboration entre les équipes et le développement d'une culture de sécurité continue [1, 3].

Ces piliers sont étroitement liés et fonctionnent de manière complémentaire. L'automatisation permet par exemple d'exécuter les contrôles de sécurité à grande vitesse, mais son efficacité dépend aussi de la coordination entre les équipes et de l'intégration effective des pratiques de sécurité dans les habitudes de développement.

2.1.2.1 AUTOMATISATION DE LA SÉCURITÉ

L'automatisation constitue l'un des fondements les plus importants du DevSecOps. Avec l'adoption des pipelines CI/CD, les cycles de développement logiciel sont devenus beaucoup plus rapides et fréquents. Les applications évoluent en permanence grâce à des mises à jour et des intégrations continues, ce qui rend les approches de sécurité traditionnelles, principalement fondées sur des vérifications manuelles, de plus en plus difficiles à appliquer efficacement. Dans ce contexte, l'automatisation est devenue indispensable pour préserver à la fois la rapidité des pipelines et l'intégration des contrôles de sécurité [27, 18].

L'objectif principal de cette automatisation est d'intégrer les mécanismes de sécurité directement dans les différentes étapes du cycle de développement logiciel afin de détecter les vulnérabilités le plus tôt possible. Cette approche favorise l'intégration de la sécurité tout au long du développement, plutôt que de la concentrer à une étape réalisée uniquement avant la mise en production [12, 13]. Cette logique rejoint directement le principe du *Shift-Left Security*, qui consiste à déplacer les contrôles de sécurité vers les premières phases du développement afin de réduire les coûts, les délais et les risques associés aux corrections tardives.

Dans les pipelines DevSecOps, l'automatisation intervient à plusieurs niveaux. Elle concerne d'abord l'intégration continue et les mécanismes de tests automatiques. Les analyses de sécurité peuvent être exécutées automatiquement à chaque modification du code source grâce à des outils d'analyse statique (SAST) et dynamique (DAST). Les outils SAST per-

mettent d'examiner le code source afin d'identifier certaines vulnérabilités avant l'exécution des applications, tandis que les outils DAST analysent le comportement des applications en cours d'exécution afin de détecter des failles plus difficiles à repérer directement dans le code [27]. L'intégration automatique de ces outils dans les pipelines CI/CD permet ainsi d'accélérer la détection des vulnérabilités et de réduire les interventions manuelles.

La littérature souligne également que les pipelines peuvent intégrer un grand nombre d'activités de sécurité simultanément. Il existe également des recherches qui recensent jusqu'à 16 activités de tests réparties à différentes étapes du pipeline logiciel [18]. Cette multiplication des mécanismes de contrôle illustre le rôle structurant de l'automatisation dans les environnements DevSecOps. Sans automatisation, la gestion continue de ces différents contrôles deviendrait rapidement difficile dans des infrastructures distribuées et des environnements cloud complexes.

L'automatisation concerne également la surveillance continue des infrastructures et des applications après le déploiement. Les mécanismes de supervision s'appuient aujourd'hui sur des outils capables d'analyser en continu les journaux d'événements, les activités système, les métriques de performance ainsi que les comportements inhabituels observés dans les infrastructures [22]. Dans certains environnements DevOps, des mécanismes d'apprentissage profond sont même utilisés afin d'identifier automatiquement des anomalies à partir de données de surveillance multidimensionnelles. Cette surveillance continue permet de détecter plus rapidement certains incidents de sécurité ou comportements inhabituels dans les systèmes en production.

Aujourd'hui, l'automatisation dépasse largement le simple cadre des tests de sécurité. Les pipelines intègrent désormais des mécanismes de déploiement automatisé, des contrôles de conformité, des systèmes d'alertes en temps réel, des mécanismes de gestion automatisée

des configurations ainsi que des politiques de sécurité définies directement sous forme de code [13]. Cette approche, souvent appelée *Security as Code*, permet de versionner, tester et appliquer automatiquement certaines politiques de sécurité dans les pipelines CI/CD.

Malgré ses avantages, la littérature montre cependant que l'automatisation présente aussi plusieurs limites. L'intégration des outils automatisés peut devenir complexe dans les infrastructures hétérogènes ou dans les environnements contenant de nombreux outils différents [25]. Certains travaux soulignent également que les mécanismes de scan automatisés peuvent parfois ralentir les pipelines lorsqu'ils deviennent trop nombreux ou insuffisamment optimisés [27]. D'autres travaux rappellent enfin que l'automatisation seule ne suffit pas à garantir la sécurité des systèmes et qu'elle doit être accompagnée d'une bonne gouvernance, d'une coordination efficace entre les équipes et d'une véritable culture de sécurité.

2.1.2.2 LA COLLABORATION ET L'INTÉGRATION DES ÉQUIPES

Le deuxième pilier important du DevSecOps concerne la collaboration entre les différentes équipes impliquées dans le cycle de vie logiciel. Contrairement aux approches traditionnelles où les équipes de développement, d'exploitation et de sécurité travaillent souvent de manière séparée, le DevSecOps cherche à rapprocher ces différents acteurs afin d'intégrer la sécurité directement dans les activités quotidiennes du développement logiciel.

La littérature souligne que cette collaboration favorise la circulation de l'information, la coordination des activités et l'intégration des contrôles de sécurité dans les pipelines logiciels [18, 33]. Plusieurs recherches rappellent également que la sécurité ne doit plus être considérée comme une responsabilité réservée uniquement aux spécialistes en cybersécurité, mais comme une responsabilité partagée entre tous les acteurs du pipeline logiciel [13].

Cette collaboration devient particulièrement importante dans les environnements CI/CD, où les déploiements sont rapides et les infrastructures souvent très distribuées. Les équipes doivent alors coordonner simultanément les activités de développement, de déploiement, de supervision et de gestion des risques sans ralentir les pipelines. Les organisations rencontrent fréquemment des difficultés liées aux silos organisationnels, aux différences de méthodes de travail ou encore à l'utilisation d'outils non compatibles entre les équipes [25].

Il ressort également que les aspects humains et organisationnels représentent une part importante des difficultés rencontrées lors de l'adoption du DevSecOps. Le manque de communication, la résistance au changement ou encore l'absence de responsabilités clairement partagées peuvent limiter l'efficacité des mécanismes de sécurité automatisés [32, 4]. Certains auteurs considèrent même que les problèmes culturels et organisationnels sont parfois plus difficiles à résoudre que les défis purement techniques.

2.1.2.3 LA CULTURE DE SÉCURITÉ

Le troisième pilier du DevSecOps concerne la culture de sécurité. Ce pilier vise à intégrer les préoccupations de cybersécurité dans l'ensemble des activités du développement logiciel afin que la sécurité ne soit plus perçue comme une étape supplémentaire ajoutée uniquement à la fin du projet. Le DevSecOps cherche plutôt à faire de la sécurité une composante naturelle et permanente du processus de développement logiciel [1, 20].

Dans la pratique, cette culture de sécurité se traduit par plusieurs mécanismes organisationnels et techniques. Les travaux analysés mentionnent notamment la formation continue au codage sécurisé, les revues de code orientées sécurité, la modélisation des menaces, la sensibilisation des développeurs aux vulnérabilités courantes ainsi que la surveillance conti-

nue des applications après leur déploiement [12, 13]. Ces mécanismes permettent d'intégrer progressivement les bonnes pratiques de sécurité dans les habitudes de travail des équipes.

Ce changement culturel reste toutefois difficile à mettre en place dans certaines organisations. De nombreux développeurs continuent parfois de considérer la sécurité comme une responsabilité réservée aux équipes spécialisées en cybersécurité [18]. Tant que cette séparation persiste, l'intégration complète du DevSecOps reste limitée. La réussite du DevSecOps dépend ainsi autant des transformations humaines et organisationnelles que des outils techniques utilisés dans les pipelines.

2.1.3 PHASES D'UN MODÈLE DEVSECOPS AVEC SÉCURITÉ EMBARQUÉE

Le DevSecOps repose sur une approche dans laquelle la sécurité accompagne en permanence les processus de conception, d'intégration et de déploiement des applications. Les contrôles de sécurité ne sont donc plus isolés dans une phase unique, mais répartis tout au long du pipeline CI/CD afin de détecter les vulnérabilités le plus tôt possible et de maintenir une surveillance continue des applications [37, 13].

Les architectures DevSecOps peuvent varier selon les organisations et les outils adoptés. Certains modèles présentent des pipelines centrés sur les principales activités CI/CD, alors que d'autres proposent des approches plus détaillées intégrant plusieurs niveaux de validation et de contrôle de sécurité [39]. Malgré ces différences, certaines phases apparaissent de manière récurrente dans la littérature : la planification, le développement et le codage, la compilation et l'intégration continue, les tests de sécurité, la publication et le déploiement, puis l'exploitation et la supervision continue [27, 12].

2.1.3.1 PHASE DE PLANIFICATION

La phase de planification représente le point de départ du pipeline DevSecOps. Cette étape consiste à identifier les besoins fonctionnels, les objectifs du projet ainsi que les exigences de sécurité avant même le début du développement [12]. Contrairement aux approches classiques où la sécurité était souvent ajoutée après le développement, le DevSecOps cherche à intégrer les préoccupations de sécurité dès les premières décisions de conception.

Le principe de *Shift-Left Security* occupe une place centrale dans les approches DevSecOps : il consiste à déplacer les contrôles de sécurité vers les premières étapes du cycle de développement afin de réduire les coûts de correction et de limiter la propagation des vulnérabilités dans les phases suivantes [13, 37]. Les équipes doivent ainsi identifier les risques, définir les politiques de sécurité et préparer les mécanismes automatisés qui seront intégrés dans le pipeline CI/CD.

Yelkoti souligne également que les approches de type *Security as Code* permettent de transformer certaines politiques de sécurité en configurations automatisées intégrées directement dans les pipelines de développement [39]. Cette approche facilite l'application continue des règles de sécurité tout au long du cycle de vie logiciel.

2.1.3.2 PHASE DE DÉVELOPPEMENT ET DE CODAGE

La phase de développement correspond à l'implémentation des fonctionnalités logicielles par les développeurs. Dans les environnements DevSecOps, cette étape ne se limite plus uniquement à l'écriture du code, mais intègre également plusieurs mécanismes de sécurité directement pendant le développement [12].

Les développeurs soumettent régulièrement leurs modifications dans des dépôts de code partagés afin de permettre l'intégration continue et les validations automatisées [26]. Les pratiques DevSecOps encouragent fortement l'utilisation des revues de code, du contrôle de version et des politiques d'accès afin de limiter les erreurs humaines et les modifications non autorisées [37].

Plusieurs auteurs mettent également en avant l'importance des outils d'analyse automatisée durant cette phase. Les mécanismes SAST permettent par exemple d'analyser automatiquement le code source afin d'identifier certaines vulnérabilités avant l'exécution des applications [37, 27]. Cette intégration précoce des contrôles contribue à réduire les vulnérabilités qui pourraient atteindre les phases de test ou de production.

2.1.3.3 LA COMPILATION ET L'INTÉGRATION CONTINUE

Après le développement, les modifications du code sont intégrées dans des pipelines d'intégration continue où plusieurs mécanismes automatisés sont exécutés afin de vérifier la stabilité et la sécurité des applications [26]. Cette phase joue un rôle essentiel dans le DevSecOps puisqu'elle permet de détecter rapidement les erreurs d'intégration, les dépendances vulnérables ou encore certaines mauvaises configurations avant le déploiement.

Thopalle souligne que les pipelines utilisent différents outils automatisés afin d'analyser les dépendances logicielles, les bibliothèques open source ainsi que les composants tiers susceptibles d'introduire des vulnérabilités dans les applications [37]. Les outils de *Software Composition Analysis* (SCA) permettent notamment d'identifier les composants contenant des vulnérabilités connues.

Les architectures DevSecOps intègrent également des mécanismes de validation de l'infrastructure définie sous forme de code (*Infrastructure as Code*) afin de détecter les configurations non sécurisées avant leur déploiement [39]. Cette automatisation contribue à renforcer la cohérence et la sécurité des environnements déployés.

2.1.3.4 LES TESTS DE SÉCURITÉ

La phase de test constitue une étape importante du pipeline DevSecOps puisqu'elle permet de valider le comportement des applications et d'identifier différentes vulnérabilités avant la mise en production [12]. Les environnements DevSecOps utilisent généralement plusieurs niveaux de tests automatisés afin de renforcer la couverture de sécurité.

Les travaux étudiés distinguent principalement deux grandes approches : l'analyse statique (*SAST*) et l'analyse dynamique (*DAST*) [27, 37]. Les analyses SAST se concentrent directement sur le code de l'application, alors que les approches DAST examinent les applications pendant leur exécution afin d'identifier certaines failles visibles uniquement dans un contexte d'utilisation réel.

Les analyses de sécurité appliquées aux conteneurs, aux dépendances logicielles et aux infrastructures cloud occupent désormais une place importante dans les environnements DevSecOps [39]. Les pipelines peuvent ainsi intégrer des mécanismes de balayage des images Docker, des validations d'infrastructure ainsi que des contrôles automatisés de conformité avant le déploiement final.

2.1.3.5 LA PUBLICATION ET LE DÉPLOIEMENT

Une fois les validations terminées, les applications peuvent être publiées puis déployées dans les environnements de production. Dans les architectures DevSecOps, cette phase reste fortement automatisée afin d'assurer des déploiements rapides tout en maintenant les mécanismes de sécurité [37].

Les pipelines intègrent également des mécanismes de validation d'intégrité permettant de vérifier que les artefacts logiciels n'ont pas été modifiés de manière non autorisée avant leur déploiement [37]. Les politiques de sécurité peuvent également être appliquées automatiquement à travers des outils de validation d'infrastructure et des mécanismes de contrôle d'accès.

Certaines architectures mettent aussi en place des *security gates*, c'est-à-dire des points de contrôle automatisés capables de bloquer un déploiement lorsque certaines vulnérabilités critiques sont détectées [39]. Cette approche permet d'empêcher la mise en production de configurations jugées trop risquées.

2.1.3.6 L'EXPLOITATION ET LA SUPERVISION CONTINUE

Pendant longtemps, les mécanismes de sécurité étaient surtout appliqués avant la mise en production des applications. Le DevSecOps adopte toutefois une approche plus continue dans laquelle la sécurité reste présente même après le déploiement des applications [13]. Les phases d'exploitation et de supervision permettent ainsi d'assurer une surveillance continue des environnements de production.

Le *Continuous Monitoring* joue également un rôle important pour détecter rapidement les comportements anormaux, les nouvelles vulnérabilités ainsi que les incidents de sécurité [13, 39]. Les mécanismes de surveillance continue permettent également de produire des alertes automatiques et d'améliorer la capacité de réaction des équipes de sécurité.

Les architectures DevSecOps récentes intègrent aussi des mécanismes de protection à l'exécution, des systèmes de journalisation ainsi que des outils d'analyse intelligente capables d'adapter certaines politiques de sécurité en fonction du contexte et du niveau de risque [37]. La supervision continue contribue à maintenir la sécurité des applications tout au long de leur cycle de vie.

2.2 VULNÉRABILITÉS ET MÉTHODES DE DÉTECTION

2.2.1 INTRODUCTION

La multiplication des cyberattaques et l'évolution rapide des environnements logiciels ont profondément transformé les besoins en matière de sécurité informatique. Les approches Agile, DevOps et les pipelines CI/CD ont considérablement accéléré le rythme de développement des applications. Cependant, cette accélération du développement s'accompagne également d'une augmentation importante des vulnérabilités logicielles et des surfaces d'attaque. Les vulnérabilités sont aujourd'hui plus nombreuses, plus diversifiées et plus difficiles à maîtriser dans les infrastructures numériques [21, 5].

Les données issues des bases CVE et NVD illustrent clairement cette évolution. Guo et al. rapportent que plus de 29 065 vulnérabilités ont été recensées durant l'année 2023 et que plus de 2 000 nouvelles vulnérabilités avaient déjà été publiées dès le début de l'année 2024 [21]. La littérature rapporte également une augmentation importante des vulnérabilités

exploitables à distance ainsi qu'une diversification croissante des types d'attaques ciblant les systèmes [31, 5].

Cette situation rend la détection des vulnérabilités particulièrement importante dans les environnements DevSecOps. Il devient désormais essentiel de pouvoir identifier rapidement les failles de sécurité afin de limiter les risques d'exploitation avant la mise en production des applications. Cependant, les approches de sécurité reposant principalement sur des contrôles réalisés manuellement s'adaptent de plus en plus difficilement au rythme des déploiements continus, à l'utilisation croissante de composants tiers et à la complexité des architectures logicielles [27, 34].

Afin de répondre à ces contraintes, les mécanismes de détection des vulnérabilités ont fortement évolué au cours des dernières années. Les premières approches reposaient principalement sur des règles définies manuellement, des signatures connues et l'expertise des analystes de sécurité. Progressivement, ces mécanismes ont été complétés par des outils d'analyse statique et dynamique intégrés directement dans les pipelines CI/CD. Les travaux les plus récents exploitent désormais des techniques d'apprentissage automatique, d'apprentissage profond et des modèles pré-entraînés capables d'automatiser davantage la détection des vulnérabilités dans de grands volumes de code source [31, 21].

Les recherches récentes indiquent également que les mécanismes de détection ne reposent plus sur une seule approche isolée. Les pipelines DevSecOps actuels combinent généralement plusieurs techniques complémentaires afin d'améliorer la couverture, la rapidité et la précision des analyses de sécurité. Les approches intelligentes fondées sur l'IA permettent aussi d'identifier progressivement des motifs complexes difficiles à repérer avec les seules méthodes traditionnelles [31].

Ainsi, la détection des vulnérabilités apparaît aujourd'hui comme un processus continu, automatisé et de plus en plus intelligent au sein des pipelines DevSecOps. Cette évolution vise non seulement à accélérer l'identification des vulnérabilités, mais également à améliorer la capacité des organisations à sécuriser des infrastructures logicielles devenues particulièrement dynamiques et complexes.

2.2.2 VULNÉRABILITÉS LOGICIELLES

Les vulnérabilités logicielles représentent aujourd'hui l'un des principaux défis des environnements numériques. Avec l'évolution rapide des technologies et l'adoption croissante de nouvelles architectures et environnements logiciels, les systèmes deviennent de plus en plus complexes et difficiles à sécuriser [4, 12]. Cette complexité augmente progressivement les risques d'erreurs, de mauvaises configurations et de failles de sécurité pouvant être exploitées par des attaquants. Des données montrent une forte augmentation des vulnérabilités exploitables par réseau, passées d'environ 1180 en 2013 à plus de 19 231 en 2023 [31]. Cette progression illustre l'élargissement continu de la surface d'attaque des environnements logiciels.

La littérature montre également que les vulnérabilités peuvent apparaître à différents niveaux des systèmes informatiques. Certaines failles proviennent directement du code source des applications, tandis que d'autres sont liées aux dépendances logicielles, aux bibliothèques open source, aux conteneurs ou encore aux mauvaises configurations des infrastructures cloud [12, 4]. Dans les environnements DevSecOps, les vulnérabilités doivent ainsi être gérées sur l'ensemble du pipeline logiciel, depuis les phases de développement jusqu'aux infrastructures de production.

La diversité des *types de vulnérabilités* présents dans les environnements de cybersécurité est également largement mise en évidence dans la littérature. Les injections SQL figurent parmi les vulnérabilités les plus connues et permettent à un attaquant de manipuler les requêtes envoyées aux bases de données afin d'accéder à des informations sensibles ou de modifier certaines données [5]. Les attaques XSS (*Cross-Site Scripting*) représentent également une catégorie importante de vulnérabilités web et permettent l'injection de scripts malveillants dans les applications afin de compromettre les utilisateurs ou de voler certaines informations [5].

D'autres vulnérabilités concernent directement la gestion de la mémoire et des ressources système. Les *buffer overflows* permettent par exemple d'écrire des données en dehors des zones mémoire prévues, ce qui peut entraîner des comportements inattendus ou l'exécution de code malveillant [5]. Les études mentionnent également des vulnérabilités liées aux mécanismes d'authentification, aux élévations de privilèges ainsi qu'aux attaques réseau pouvant affecter les infrastructures distribuées et les environnements connectés [31, 5].

Certaines vulnérabilités deviennent particulièrement critiques lorsqu'elles touchent des composants largement utilisés dans les infrastructures numériques. Qi et al. rappellent notamment le cas de la vulnérabilité Heartbleed, qui permettait l'extraction d'informations sensibles directement depuis la mémoire des serveurs affectés [31]. Les auteurs évoquent également la fuite massive de données d'Equifax associée à une vulnérabilité d'Apache Struts, ayant conduit à l'exposition des données personnelles d'environ 148 millions de personnes [31]. Ces exemples montrent que certaines vulnérabilités peuvent avoir des impacts techniques, financiers et sociétaux importants.

Les menaces associées aux vulnérabilités évoluent également au rythme des technologies utilisées dans les environnements DevSecOps. Les déploiements continus, l'intégration

massive de composants tiers ainsi que l'utilisation d'infrastructures cloud rendent les systèmes beaucoup plus dynamiques et augmentent la difficulté de surveillance et de correction des failles de sécurité [4, 32]. Certaines familles de vulnérabilités sont également fortement représentées dans les jeux de données utilisés pour entraîner les mécanismes intelligents de détection, tandis que d'autres restent beaucoup moins présentes [5]. Cette distribution déséquilibrée peut limiter la capacité des modèles intelligents à détecter correctement certaines vulnérabilités rares ou plus complexes [21].

Ainsi, les vulnérabilités logicielles ne constituent pas uniquement un problème technique isolé. Elles représentent un phénomène complexe influencé par l'évolution des architectures logicielles, la diversité des technologies utilisées ainsi que par la rapidité des pipelines de développement et de déploiement. Cette complexité explique pourquoi la détection rapide et efficace des vulnérabilités est devenue un enjeu central dans les environnements DevSecOps.

2.2.3 PROCESSUS DE DÉTECTION DES VULNÉRABILITÉS

La détection des vulnérabilités occupe aujourd'hui une place essentielle dans les environnements DevSecOps. L'augmentation constante du nombre de vulnérabilités oblige les organisations à mettre en place des mécanismes de détection capables de fonctionner de manière rapide, continue et automatisée afin de suivre le rythme des environnements logiciels [21, 4, 27]. Les pipelines logiciels intègrent désormais de nombreux composants et services externes, ce qui élargit considérablement les surfaces d'attaque et complique le maintien des mécanismes de contrôle manuel [4, 12, 27].

Face à cette complexité, plusieurs approches ont progressivement été développées dans les environnements DevSecOps. Elles regroupent notamment les mécanismes fondés sur des règles et des signatures, les analyses statiques et dynamiques intégrées aux pipelines CI/CD

ainsi que les méthodes reposant sur l'apprentissage automatique et l'apprentissage profond [21, 27, 31]. L'objectif commun de ces approches est d'améliorer la rapidité, la précision et le niveau d'automatisation des analyses de sécurité dans des environnements logiciels toujours plus dynamiques.

2.2.3.1 DÉTECTION TRADITIONNELLE

Les premières approches de détection des vulnérabilités reposaient principalement sur des règles définies manuellement, des signatures connues ainsi que sur l'expertise des analystes de sécurité [21]. Ces mécanismes cherchaient à identifier des motifs spécifiques associés à certaines familles de vulnérabilités, comme les injections SQL, les attaques XSS, les débordements mémoire ou encore certains problèmes d'authentification [5]. Pendant longtemps, ces approches ont constitué les principaux moyens de détection utilisés dans les environnements logiciels.

Les mécanismes traditionnels visaient principalement à repérer des comportements ou des motifs déjà associés à des vulnérabilités connues afin de signaler la présence potentielle de failles de sécurité dans les applications. Lorsqu'un motif correspondant était détecté, le système pouvait signaler la présence potentielle d'une faille de sécurité. Ces approches étaient particulièrement efficaces pour identifier des vulnérabilités déjà documentées dans les bases de connaissances de sécurité [21].

Cependant, les approches traditionnelles présentent également certaines limites importantes dans les environnements DevSecOps. Guo et al. expliquent notamment que les mécanismes basés uniquement sur des règles deviennent difficiles à maintenir face à l'évolution rapide des cybermenaces et à l'apparition constante de nouvelles vulnérabilités [21]. Qi et al. soulignent également que ces approches nécessitent une forte intervention humaine afin de

définir, mettre à jour et maintenir les règles de détection [31]. Cette dépendance à l'expertise humaine réduit leur capacité à fonctionner efficacement dans des environnements caractérisés par des déploiements continus et des volumes importants de données de sécurité.

Malgré ces limites, les approches traditionnelles continuent de jouer un rôle important dans les mécanismes de détection des vulnérabilités. Elles restent utiles pour détecter rapidement certaines vulnérabilités connues et servent encore de base à plusieurs outils de sécurité intégrés dans les pipelines DevSecOps actuels [27].

2.2.3.2 DÉTECTION PAR ANALYSE STATIQUE ET DYNAMIQUE

Avec l'évolution des environnements DevSecOps et l'accélération des cycles de développement logiciel, les mécanismes de détection des vulnérabilités ont progressivement évolué vers des approches davantage intégrées aux pipelines CI/CD. Parmi les méthodes les plus utilisées figurent aujourd'hui l'analyse statique (SAST) et l'analyse dynamique (DAST) [27]. Ces approches permettent d'automatiser une partie importante des contrôles de sécurité et de repérer plus rapidement les vulnérabilités au sein des applications.

L'analyse statique consiste à examiner le code source, les fichiers de configuration ou certaines dépendances logicielles sans exécuter directement l'application [27]. Cette approche permet d'identifier certaines vulnérabilités dès les premières étapes du développement, avant même le déploiement des applications. Les outils SAST sont notamment utilisés pour détecter des problèmes comme les injections SQL, certaines erreurs d'authentification, les débordements mémoire ou encore certaines mauvaises pratiques de programmation [5]. Dans les environnements DevSecOps, ces mécanismes sont souvent intégrés directement dans les pipelines CI/CD afin d'automatiser les vérifications de sécurité pendant le développement logiciel.

L'analyse dynamique adopte quant à elle une approche différente puisqu'elle évalue le comportement des applications lorsqu'elles sont en cours d'exécution [34]. Les outils DAST cherchent ainsi à identifier des vulnérabilités qui peuvent être difficiles à détecter uniquement à partir du code source, notamment certaines failles liées aux interactions réseau, aux configurations d'exécution ou aux comportements réels des applications. Cette approche permet également de simuler certains scénarios d'attaque afin d'évaluer la résistance des applications face à différentes menaces de sécurité.

Les approches SAST et DAST sont souvent complémentaires dans les environnements DevSecOps. Marandi et al. soulignent notamment l'intérêt d'intégrer simultanément les analyses statiques et dynamiques dans les pipelines CI/CD afin d'améliorer la couverture des mécanismes de détection [27]. Les pipelines combinent ainsi plusieurs niveaux de vérification afin d'identifier les vulnérabilités dès les premières phases du cycle logiciel tout en surveillant le comportement des applications lors des étapes de test et de déploiement.

Malgré leurs avantages, ces approches présentent également certaines limites. Les outils SAST peuvent générer un nombre important de faux positifs et rencontrent parfois des difficultés à détecter certaines vulnérabilités liées au contexte d'exécution réel des applications [19]. Les mécanismes DAST nécessitent quant à eux des environnements d'exécution fonctionnels et peuvent être plus coûteux en temps d'analyse dans les pipelines de déploiement continu [34]. Les organisations doivent donc souvent combiner plusieurs approches de sécurité afin d'obtenir des mécanismes de détection plus complets et plus fiables.

Les mécanismes SAST et DAST sont progressivement renforcés par des techniques d'automatisation avancées, d'apprentissage automatique et d'analyse intelligente capables de traiter des volumes de données beaucoup plus importants dans les environnements DevSecOps [31, 21]. Cette évolution vise à améliorer la rapidité de détection, la précision des analyses

ainsi que la capacité des systèmes de sécurité à s'adapter à des menaces de plus en plus complexes.

2.2.3.3 DÉTECTION PAR APPRENTISSAGE PROFOND ET MODÈLES PRÉ-ENTRAÎNÉS

L'augmentation massive du volume de code à analyser et la complexité croissante des environnements logiciels ont progressivement conduit à l'utilisation de techniques d'apprentissage automatique et d'apprentissage profond dans la détection des vulnérabilités. Contrairement aux approches traditionnelles, ces méthodes apprennent automatiquement des représentations du code source et des comportements associés aux vulnérabilités à partir des informations disponibles.

L'apprentissage profond repose sur des réseaux de neurones capables d'extraire automatiquement des représentations complexes à partir de grands volumes d'information. Contrairement aux méthodes nécessitant souvent une extraction manuelle des caractéristiques, les modèles profonds peuvent apprendre directement des motifs présents dans le code source, les journaux système, les traces réseau ou les bases de vulnérabilités [40]. Cette capacité est particulièrement importante dans les environnements DevSecOps où les données de sécurité sont massives, hétérogènes et continuellement mises à jour.

Les modèles pré-entraînés occupent désormais une place importante dans les mécanismes modernes de détection des vulnérabilités. Ces modèles sont généralement entraînés sur de très grands ensembles de données avant d'être adaptés à des tâches spécifiques de cybersécurité. Cette approche permet de réutiliser des connaissances déjà apprises afin d'améliorer les performances sur des tâches de détection spécialisées, même lorsque les informations disponibles demeurent limitées [21, 31]. Ils peuvent aussi améliorer la compréhension du

contexte du code source ainsi que la capacité à identifier des comportements anormaux ou des vulnérabilités difficiles à détecter avec des approches classiques.

Dans les systèmes de détection, les mécanismes d'apprentissage profond sont souvent associés à des architectures collaboratives intégrant supervision humaine, mécanismes de rétro-action et modèles explicables. Nuthalapati explique notamment que les systèmes d'intelligence artificielle ne doivent plus être conçus comme des boîtes noires complètement autonomes, mais comme des systèmes collaboratifs dans lesquels l'humain conserve un rôle important dans la validation, l'interprétation et le contrôle des décisions produites par l'IA [36].

Certaines méthodes intègrent ainsi des mécanismes de *Human-in-the-Loop* (HITL), dans lesquels les modèles d'intelligence artificielle réalisent une première analyse automatique tandis que les cas plus complexes ou plus sensibles sont examinés par des experts humains [36]. Cette collaboration permet d'améliorer progressivement les performances des modèles grâce aux mécanismes de rétroaction continue et d'apprentissage adaptatif. Les architectures explicables deviennent également importantes afin de permettre aux équipes de sécurité de comprendre les décisions produites par les modèles d'apprentissage profond et d'identifier les facteurs ayant conduit à la détection d'une vulnérabilité.

Malgré leurs performances prometteuses, les approches fondées sur l'apprentissage profond présentent encore plusieurs limites. Les modèles nécessitent souvent d'importants volumes d'information fiables pour être correctement entraînés. Les biais présents dans les ensembles d'apprentissage peuvent également affecter les performances des systèmes et produire des résultats erronés ou difficiles à interpréter [40]. Cette dépendance à la qualité de l'information a favorisé l'émergence des approches Data-Centric AI, qui accordent une attention particulière à l'amélioration et à la gestion des ressources informationnelles utilisées par les systèmes intelligents. De plus, certains modèles profonds demeurent coûteux en

ressources computationnelles et parfois difficiles à expliquer, ce qui peut compliquer leur intégration dans des environnements DevSecOps nécessitant des analyses rapides et continues.

2.3 AUTOMATISATION INTELLIGENTE DES PIPELINES DEVSECOPS

L'évolution des environnements DevSecOps a progressivement conduit à l'apparition de formes d'automatisation plus avancées, capables non seulement d'exécuter des tâches répétitives, mais également d'analyser l'information disponible, d'identifier des comportements inhabituels et d'assister la prise de décision dans les pipelines logiciels. Cette évolution repose principalement sur l'intégration de l'intelligence artificielle, de l'apprentissage automatique et des modèles de langage dans les mécanismes d'intégration continue et de livraison continue [30, 17, 31]. Les pipelines DevSecOps deviennent ainsi progressivement des environnements capables d'adapter certains traitements de sécurité en fonction des données observées, des événements détectés ou encore des vulnérabilités identifiées.

Cette automatisation intelligente s'intègre aujourd'hui à différentes étapes du pipeline logiciel. Pattanayak et al. expliquent notamment que les mécanismes fondés sur l'intelligence artificielle sont de plus en plus utilisés dans différentes phases des pipelines DevSecOps [30]. Les auteurs décrivent des pipelines capables d'automatiser différentes tâches d'analyse et de contrôle à partir des informations produites tout au long du cycle de développement. Dans ces environnements, les outils intelligents ne se limitent plus à appliquer des règles fixes, mais exploitent également les observations issues des systèmes afin d'améliorer progressivement les analyses réalisées.

Qi et al. montrent par exemple que les modèles de langage pré-entraînés sont désormais utilisés pour automatiser certaines tâches de détection des vulnérabilités dans le code source [31]. Ces approches exploitent des modèles capables d'apprendre automatiquement des repré-

sentations du code à partir de très grands volumes d'informations issues du développement logiciel. Les auteurs expliquent que des modèles comme CodeBERT, GraphCodeBERT ou UnixCoder peuvent être entraînés afin d'identifier automatiquement des schémas de vulnérabilités dans des projets logiciels complexes [31]. Cette automatisation permet ainsi d'intégrer des mécanismes de détection avancés directement dans les pipelines CI/CD sans dépendre uniquement de signatures statiques ou de règles manuelles.

Les recherches récentes montrent aussi une utilisation croissante des grands modèles de langage (*LLM*) dans les pipelines de sécurité automatisés. Deng et al. utilisent notamment GPT-4 pour générer automatiquement de nouveaux exemples de vulnérabilités afin d'améliorer les mécanismes de détection des vulnérabilités rares [17]. Leur approche repose sur une génération automatique de code vulnérable suivie d'un processus de filtrage automatisé permettant de sélectionner les échantillons les plus pertinents avant leur intégration dans les modèles de détection [17]. Les auteurs décrivent ainsi une approche dans laquelle les modèles d'IA participent directement à la génération, à l'évaluation et à l'amélioration des ressources informationnelles exploitées par les mécanismes de cybersécurité. Cependant, cette automatisation intelligente ne concerne pas uniquement la détection des vulnérabilités, mais aussi la gestion continue des pipelines et des infrastructures. Pattanayak et al. décrivent plusieurs mécanismes capables d'assister automatiquement certaines opérations de gestion des branches, de fusion du code, de prédiction des échecs de compilation ou encore d'optimisation des configurations des conteneurs [30]. Les auteurs expliquent notamment que certains outils basés sur l'IA peuvent analyser les historiques des modifications de code afin d'anticiper des conflits de fusion ou détecter des comportements inhabituels dans les pipelines CI/CD [30]. D'autres mécanismes intelligents sont également utilisés pour générer automatiquement des tests logiciels ou pour améliorer les analyses statiques de sécurité directement pendant les phases de développement [30].

Cette évolution vers des pipelines DevSecOps plus intelligents s'accompagne également d'un changement dans la manière de gérer l'information exploitée par les systèmes automatisés. Les approches récentes de *Data-Centric AI* considèrent désormais la qualité de l'information comme un élément central du fonctionnement des systèmes intelligents [28]. Contrairement aux approches traditionnellement centrées sur les modèles, cette vision cherche à améliorer en continu la qualité, la cohérence et la représentativité des ressources utilisées par les mécanismes d'intelligence artificielle. Dans ce contexte, les environnements DevSecOps évoluent progressivement vers des pipelines capables d'intégrer des processus continus de validation, d'analyse et d'amélioration de l'information exploitée par les outils de sécurité automatisés.

L'automatisation intelligente tend à évoluer vers des architectures plus collaboratives entre les humains et les systèmes intelligents. Nuthalapati explique notamment que les architectures d'IA ne considèrent plus les modèles intelligents comme des composants isolés, mais comme des éléments intégrés dans des systèmes sociotechniques plus larges [36]. Cette vision favorise des pipelines dans lesquels les outils intelligents assistent les équipes de développement et de sécurité dans l'analyse des vulnérabilités, la supervision des événements ou la gestion des opérations de sécurité, tout en maintenant une interaction continue entre les mécanismes automatisés et les experts humains [36].

2.3.1 APPORT DE L'INTELLIGENCE ARTIFICIELLE DANS L'AUTOMATISATION

Les pipelines DevSecOps intelligents ne se limitent plus à l'automatisation de tâches répétitives. Les mécanismes fondés sur l'intelligence artificielle sont désormais capables d'automatiser diverses tâches d'analyse et de détection dans les environnements CI/CD tout en perfectionnant progressivement leurs capacités d'apprentissage.

L'un des principaux apports observés dans la littérature concerne l'amélioration des capacités de détection des vulnérabilités. Deng et al. montrent par exemple que l'utilisation de grands modèles de langage pour générer automatiquement des exemples de vulnérabilités rares permet d'améliorer significativement les performances des systèmes de détection [17]. Les auteurs rapportent notamment une amélioration moyenne du score F1 de 26,5 % pour les vulnérabilités rares dans les modèles d'apprentissage profond, ainsi qu'une amélioration moyenne de 18,3 % pour les modèles fondés sur les grands modèles de langage [17]. Les résultats montrent également que les modèles entraînés à partir des données générées automatiquement parviennent à détecter davantage de vulnérabilités réelles dans des projets inconnus auparavant [17]. Ces résultats illustrent le potentiel des mécanismes intelligents pour renforcer la détection de vulnérabilités peu représentées dans les jeux de données traditionnels.

Les techniques intelligentes d'augmentation de données peuvent également améliorer les performances des systèmes de cybersécurité automatisés. Qi et al. proposent une approche fondée sur des transformations préservant la sémantique des vulnérabilités afin d'enrichir automatiquement les jeux de données utilisés pour entraîner les modèles de détection [31]. Grâce à cette approche, les auteurs ont généré plus de 3 233 635 nouveaux échantillons de vulnérabilités et augmenté la taille du jeu de données initial d'un facteur 17 [31]. Les expérimentations montrent ensuite une amélioration des performances pouvant atteindre 23,6 % pour le score F1 ainsi qu'une augmentation de précision allant jusqu'à 10,1 % selon les modèles utilisés [31]. Ces résultats montrent que ces approches ne se limitent pas à accélérer les traitements ; elles contribuent aussi à améliorer les capacités de détection des modèles de cybersécurité entraînés dans les environnements DevSecOps.

L'automatisation intelligente permet également d'améliorer l'efficacité opérationnelle des pipelines logiciels. Pattanayak et al. expliquent que les mécanismes fondés sur l'intelligence artificielle peuvent automatiser plusieurs tâches historiquement longues et répétitives

dans les environnements DevOps et DevSecOps [30]. Les auteurs décrivent notamment des systèmes capables d'assister automatiquement les développeurs dans la gestion des branches, la détection précoce des conflits de fusion, l'analyse statique du code, la génération automatique de tests ou encore l'optimisation des configurations d'infrastructure [30]. Cette automatisation réduit fortement certaines opérations manuelles qui pouvaient ralentir les cycles de développement et augmente ainsi la rapidité des déploiements logiciels.

Les pipelines intelligents permettent également d'améliorer l'analyse continue des environnements logiciels. Marandi et al. montrent que l'intégration automatisée des outils SAST et DAST dans les pipelines CI/CD favorise une détection plus rapide des vulnérabilités tout au long du cycle de développement logiciel [27]. Les auteurs soulignent également que l'automatisation des analyses de sécurité permet de réduire considérablement les délais associés aux processus de vérification et de déploiement. En s'appuyant sur des travaux antérieurs intégrant DevSecOps dans des environnements CI/CD, ils rappellent notamment que certaines opérations auparavant réalisées en plusieurs heures peuvent être exécutées en seulement 3 à 4 minutes grâce à l'automatisation [27]. L'étude met également en évidence l'importance de la surveillance continue des vulnérabilités dans les pipelines DevSecOps afin de détecter rapidement les problèmes de sécurité avant le déploiement en production [27].

Les approches intelligentes permettent progressivement de rendre les pipelines plus adaptatifs. Les systèmes ne se contentent plus d'exécuter des tâches prédéfinies, mais exploitent désormais les informations produites par les environnements logiciels afin d'ajuster certains comportements et d'améliorer progressivement leurs capacités d'analyse [28]. Les approches de *Data-Centric AI* insistent notamment sur l'amélioration continue des données et des modèles tout au long du cycle de vie des systèmes intelligents [28]. Cette logique favorise des pipelines capables d'intégrer des mécanismes continus d'analyse, de validation et d'amélioration des données utilisées dans les outils de sécurité automatisés.

Enfin, l'automatisation intelligente favorise également une meilleure collaboration entre les systèmes automatisés et les experts humains. Nuthalapati explique que les architectures d'intelligence artificielle tendent de plus en plus vers des modèles collaboratifs où les systèmes intelligents assistent les équipes humaines dans les tâches d'analyse, de supervision et de prise de décision [36]. Cette collaboration permet d'exploiter simultanément les capacités analytiques des systèmes intelligents et l'expertise contextuelle des professionnels de la cybersécurité, ce qui contribue à améliorer la supervision globale des environnements DevSecOps [36].

2.3.2 LIMITES DE L'AUTOMATISATION INTELLIGENTE

Malgré les avancées apportées par l'automatisation intelligente dans les environnements DevSecOps, ces approches présentent encore plusieurs limites, notamment en ce qui concerne les informations exploitées par les systèmes intelligents. Les mécanismes fondés sur l'intelligence artificielle, l'apprentissage profond ou les grands modèles de langage dépendent fortement de la qualité, de la diversité et de la représentativité des données utilisées pendant les phases d'entraînement et d'analyse [31, 17, 28]. Lorsque la qualité des données est insuffisante, les performances des systèmes automatisés peuvent diminuer de manière importante.

Une première limite concerne le manque de jeux de données de vulnérabilités suffisamment riches et correctement annotés. Qi et al. expliquent notamment que les ressources publiques disponibles pour l'entraînement des modèles de détection contiennent souvent un nombre limité d'échantillons représentatifs des vulnérabilités réelles [31]. Les auteurs indiquent également que certaines méthodes classiques d'augmentation peuvent produire des résultats contre-productifs lorsqu'elles ne préservent pas correctement la sémantique des vulnérabilités [31]. Cette situation réduit la capacité des modèles à généraliser correctement les comportements observés dans les environnements logiciels réels.

Les déséquilibres présents dans les jeux de données constituent également une limite importante pour les pipelines DevSecOps intelligents. Deng et al. montrent que les ensembles de données de vulnérabilités présentent souvent des distributions très déséquilibrées entre les différents types de vulnérabilités [17]. Certaines catégories de vulnérabilités sont fortement représentées alors que d'autres apparaissent beaucoup plus rarement dans les ensembles utilisés pour l'entraînement. Les auteurs montrent notamment que, dans le jeu de données DiverseVul, les performances des modèles varient fortement selon les types de vulnérabilités détectés [17]. Les scores F1 observés varient par exemple de 0,03 à 0,57 selon les catégories de vulnérabilités pour certains modèles évalués [17]. Cette forte variation montre que les systèmes intelligents peuvent devenir très performants pour certaines vulnérabilités fréquentes tout en restant beaucoup moins efficaces pour les vulnérabilités rares ou peu représentées dans les données d'entraînement.

La littérature montre également que les grands modèles de langage introduisent certaines limites liées à la qualité des contenus générés automatiquement. Deng et al. expliquent que les échantillons produits par les LLM peuvent contenir des erreurs, des incohérences ou des éléments non conformes aux vulnérabilités recherchées, ce qui nécessite des mécanismes de validation supplémentaires [17]. Les auteurs ont ainsi intégré une phase complète d'autoévaluation et de filtrage automatique afin d'éliminer les contenus de faible qualité avant leur intégration dans les modèles de détection [17]. Ces résultats montrent que l'automatisation intelligente peut également produire des informations imparfaites lorsqu'aucun mécanisme de contrôle qualité n'est appliqué.

Les approches récentes de *Data-Centric AI* soulignent que les performances des systèmes intelligents ne dépendent pas uniquement de la sophistication des modèles, mais également de la qualité de l'information exploitée tout au long de leur cycle de vie [28]. Les auteurs expliquent que les approches centrées exclusivement sur les modèles ont longtemps relégué

la gestion de l'information à une simple étape de prétraitement, sans véritable démarche d'amélioration continue. Cette situation peut conduire au problème bien connu du « garbage in, garbage out », où des informations de faible qualité produisent directement des résultats peu fiables. La littérature souligne également que les erreurs présentes dans les ressources exploitées peuvent provoquer des effets en cascade et affecter durablement les performances ainsi que la fiabilité des mécanismes de décision automatisés.

D'autres limites concernent la difficulté de maintenir des informations cohérentes et adaptées à l'évolution rapide des environnements numériques et des cybermenaces. Les environnements DevSecOps produisent un volume considérable d'éléments provenant de multiples sources, notamment les journaux d'exécution, les rapports de sécurité, les résultats d'analyses SAST/DAST, les événements réseau ou encore les infrastructures cloud. Selon les travaux sur la *Data-Centric AI*, ces ressources doivent être surveillées, validées et maintenues en continu afin de préserver leur fiabilité dans le temps [40]. En l'absence de mécanismes efficaces de contrôle et d'amélioration continue, les systèmes automatisés risquent progressivement d'exploiter des informations obsolètes, incohérentes ou biaisées, ce qui réduit la fiabilité globale des pipelines DevSecOps.

Enfin, plusieurs recherches montrent que les systèmes intelligents deviennent parfois difficiles à interpréter lorsque les modèles utilisés gagnent en complexité. Nuthalapati explique notamment que certaines architectures fondées sur l'intelligence artificielle fonctionnent comme des systèmes de type « boîte noire », dans lesquels les mécanismes internes de décision restent peu visibles pour les équipes humaines [36]. Ce manque de transparence peut compliquer la compréhension des décisions produites automatiquement, mais également rendre plus difficile l'analyse des erreurs ou la validation des résultats de sécurité générés dans les pipelines DevSecOps [36]. Les auteurs soulignent ainsi l'importance de mettre en place

des mécanismes d'explicabilité et de supervision humaine afin d'améliorer la confiance et la fiabilité des systèmes intelligents [36].

2.3.3 CONCLUSION

L'analyse réalisée dans ce chapitre met en évidence l'évolution des environnements DevSecOps vers des processus de développement où la sécurité est intégrée de manière continue tout au long du cycle de vie des applications. Contrairement aux approches traditionnelles reposant sur des contrôles effectués en fin de développement, le DevSecOps favorise une prise en compte précoce des exigences de sécurité grâce à l'automatisation, à la collaboration entre les équipes et à l'adoption d'une culture de sécurité partagée [1, 37]. L'intégration de mécanismes tels que l'analyse du code, les tests de sécurité, la validation des dépendances et la surveillance continue contribue ainsi à améliorer la détection des risques et à réduire les vulnérabilités avant et après le déploiement des applications [27, 39].

Malgré ces avancées, les vulnérabilités logicielles demeurent un enjeu important dans les environnements DevSecOps. La généralisation des composants open source, des infrastructures cloud, des conteneurs et des architectures distribuées accroît la complexité des systèmes ainsi que leur surface d'attaque [37]. Pour répondre à ces défis, les approches de détection ont progressivement évolué, passant des méthodes fondées sur des règles et des signatures vers des mécanismes intégrant l'analyse statique, l'analyse dynamique ainsi que des techniques d'apprentissage automatique, d'apprentissage profond et des modèles pré-entraînés [21, 31]. Ces avancées permettent d'automatiser davantage les analyses de sécurité et d'améliorer l'identification de comportements complexes ou difficilement détectables par les approches classiques [30, 17].

Toutefois, les travaux étudiés montrent également que les performances des mécanismes intelligents dépendent fortement de la qualité des données exploitées. Le manque de données correctement annotées, les déséquilibres entre catégories de vulnérabilités, la présence de bruit ou encore certaines incohérences peuvent limiter l'efficacité des modèles développés [31, 17]. Les approches de *Data-Centric AI* soulignent d'ailleurs que l'amélioration des systèmes intelligents repose autant sur la qualité, la cohérence et la représentativité des données que sur la sophistication des algorithmes utilisés [28]. Ainsi, la qualité des données apparaît aujourd'hui comme un facteur essentiel pour garantir la fiabilité des analyses de sécurité et soutenir efficacement les mécanismes de priorisation des vulnérabilités dans les environnements DevSecOps.

CHAPITRE III
REVUE DE LA LITTÉRATURE

3.1 QUALITÉ DES DONNÉES

3.1.1 FONDEMENTS ET DÉFINITIONS DE LA QUALITÉ DES DONNÉES

La notion de qualité des données peut paraître simple à première vue, mais elle reste difficile à définir de manière universelle. Dans la littérature, la qualité des données est généralement associée à la capacité des données à répondre correctement aux besoins liés à leur utilisation. Autrement dit, des données considérées comme fiables dans un contexte donné peuvent devenir inadaptées dans un autre selon les objectifs du système, les traitements réalisés ou les décisions attendues.

Pendant longtemps, les données étaient principalement vues comme un support permettant de stocker et de transmettre l'information. Toutefois, avec le développement des systèmes d'intelligence artificielle et des environnements fortement automatisés, leur rôle a progressivement évolué. Les données occupent désormais une place centrale dans le fonctionnement des systèmes intelligents, notamment dans les tâches d'analyse, de prédiction ou de détection automatisée [40].

Cette évolution a conduit plusieurs chercheurs à accorder davantage d'attention à la manière dont les données sont collectées, préparées, transformées et maintenues dans les systèmes numériques. Les approches récentes de *Data-Centric AI* montrent notamment que les performances des systèmes intelligents ne dépendent pas uniquement des modèles utilisés, mais également de la qualité des données exploitées pendant les phases d'apprentissage et d'analyse. Selon Zha et al., l'amélioration continue des données devient aujourd'hui un

élément important dans le développement de systèmes d'intelligence artificielle plus fiables et mieux adaptés aux conditions réelles d'utilisation [40].

La qualité des données ne peut désormais plus être réduite à une simple phase de nettoyage réalisée avant leur utilisation. Elle s'inscrit plutôt dans un processus continu comprenant différentes étapes telles que la collecte, l'intégration, la transformation et l'amélioration des données. Cette vision est particulièrement importante dans les environnements où les systèmes exploitent des informations provenant de sources multiples et parfois très hétérogènes.

Dans les environnements DevSecOps et les systèmes de cybersécurité, cette problématique prend une dimension particulière puisque plusieurs mécanismes automatisés reposent directement sur les données utilisées lors des phases d'analyse et de détection. Les jeux de données de vulnérabilités peuvent contenir diverses anomalies, notamment des incohérences, des informations manquantes ou des problèmes de structuration, susceptibles d'affecter les résultats produits par les systèmes intelligents [21].

Ainsi, les fondements de la qualité des données reposent aujourd'hui sur une vision qui considère celles-ci comme un élément essentiel au bon fonctionnement des systèmes intelligents. Leur gestion, leur préparation et leur amélioration continue constituent désormais des enjeux majeurs dans les domaines de l'intelligence artificielle, de la cybersécurité et de l'automatisation intelligente.

3.1.2 PROBLÈMES LIÉS À LA QUALITÉ DES DONNÉES

Les approches d'automatisation intelligente ont mis en lumière le rôle important des données dans les systèmes exploités par l'intelligence artificielle. Les performances des modèles d'apprentissage automatique dépendent fortement de la qualité des données utilisées pour leur fonctionnement [16, 21]. Cette vision est notamment présente dans le *Data-Centric*

AI, où l'amélioration des données occupe une place centrale dans le développement des systèmes intelligents [16]. Zha et al. rappellent par exemple que GPT-1 utilisait environ 4,8 Go de données, GPT-2 près de 40 Go de données filtrées et GPT-3 environ 570 Go de données sélectionnées à partir de 45 To de données brutes [16]. Cela montre que les systèmes intelligents reposent sur des volumes massifs de données nécessitant différentes opérations de préparation, de filtrage et de validation avant leur exploitation.

Malgré ces avancées, les ensembles de données utilisés dans les systèmes intelligents présentent encore de nombreux problèmes pouvant affecter la fiabilité des résultats obtenus [21]. Jarrahi et al. expliquent notamment que certains défauts présents dans les données peuvent se propager dans différentes composantes d'un système intelligent et produire des effets en cascade lors du déploiement [28]. Cette problématique devient particulièrement critique dans les environnements automatisés où les mécanismes d'IA exploitent continuellement des données provenant de multiples sources afin de produire des analyses et soutenir la prise de décision.

En cybersécurité, les performances des modèles d'apprentissage automatique peuvent être fortement influencées par les caractéristiques des données utilisées pendant l'entraînement et l'évaluation [14, 21]. Chen et al. expliquent notamment que certaines performances très élevées rapportées dans la littérature étaient en partie liées à des défauts présents dans les données ou dans les méthodologies d'évaluation utilisées [11]. Une telle situation peut conduire à une surestimation des capacités réelles des modèles et donner une vision biaisée de leur efficacité dans des environnements réels.

Parmi les problèmes les plus fréquemment observés figurent notamment les erreurs d'étiquetage, les doublons, les données manquantes ainsi que les incohérences de représentation [14, 29]. Dans les systèmes de détection de vulnérabilités, certaines annotations utilisées pour

identifier les vulnérabilités peuvent être incorrectes, incomplètes ou contradictoires [14]. Les modèles risquent alors d'apprendre des comportements erronés et de produire des résultats peu fiables. Les doublons représentent également une difficulté majeure puisque certaines données similaires peuvent apparaître à la fois dans les ensembles d'entraînement et de test, créant ainsi des fuites d'information et une impression artificielle de bonnes performances [14]. Mohammed et al. montrent d'ailleurs que ce phénomène influence fortement plusieurs métriques d'évaluation utilisées dans les datasets de cybersécurité [29].

Les données manquantes et les incohérences de représentation constituent également des défis récurrents pour les traitements automatisés. L'absence de certaines informations peut réduire les capacités d'analyse et limiter les performances des modèles d'apprentissage automatique [29]. Dans les bases de vulnérabilités, certaines données peuvent être représentées sous différents formats ou comporter des variations dans les noms, les versions logicielles ou les identifiants utilisés [14]. Ces différences compliquent la fusion et l'exploitation des données dans les systèmes automatisés et réduisent l'efficacité des mécanismes d'analyse et de corrélation [29].

D'autres difficultés concernent également le déséquilibre et la faible diversité des ensembles de données utilisés en cybersécurité. Deng et al. montrent notamment que certaines catégories de vulnérabilités sont largement représentées alors que d'autres apparaissent très rarement dans les jeux de données [17]. Les auteurs observent ainsi des variations importantes des scores F1, compris entre 0.03 et 0.57 selon les catégories de vulnérabilités étudiées [17]. Guo et al. soulignent également que plusieurs jeux de données utilisés pour la détection des vulnérabilités couvrent seulement un nombre limité de catégories de vulnérabilités et de langages de programmation [21]. Les auteurs montrent notamment que 18 datasets étudiés couvrent uniquement 28 types de vulnérabilités répartis dans 10 grandes familles, tandis que certaines vulnérabilités importantes ne sont représentées dans aucun dataset [21]. Cette

couverture limitée réduit les capacités de généralisation des systèmes intelligents dans des environnements réels beaucoup plus variés.

Les problèmes liés à l'actualité et à la pertinence des données représentent également un défi majeur dans les environnements DevSecOps et de cybersécurité. Les vulnérabilités évoluent rapidement et de nouvelles menaces apparaissent continuellement. L'utilisation de données obsolètes ou peu représentatives peut alors limiter l'efficacité des modèles d'intelligence artificielle et réduire leur capacité à détecter les nouvelles formes d'attaques [29]. Dans les pipelines DevSecOps, cette situation devient particulièrement critique puisque les mécanismes automatisés s'appuient continuellement sur des données issues des outils de développement, des scanners de sécurité et des plateformes de surveillance afin d'automatiser différentes tâches liées à la cybersécurité [27].

Les difficultés associées à la qualité des données ne concernent pas uniquement les aspects techniques. Guerra-García et al. expliquent également que les exigences de qualité des données restent encore insuffisamment prises en compte dans plusieurs processus de développement logiciel [10]. Les auteurs soulignent notamment l'absence de méthodologies universelles permettant de gérer efficacement les exigences de qualité dans les systèmes d'information [10]. Cette situation complique l'évaluation, le contrôle et l'amélioration des données utilisées dans les systèmes intelligents.

Ainsi, les problèmes liés à la qualité des données influencent directement la fiabilité des systèmes intelligents et des mécanismes automatisés utilisés dans les environnements DevSecOps. Les erreurs d'étiquetage, les doublons, les données manquantes, les incohérences de représentation, le déséquilibre des jeux de données ou encore l'obsolescence des informations peuvent fortement affecter les performances des modèles ainsi que l'efficacité des mécanismes d'analyse et de priorisation des vulnérabilités. Dans ce contexte, l'évaluation et

l'amélioration de la qualité des données apparaissent comme des éléments essentiels pour renforcer la robustesse et la fiabilité des systèmes intelligents appliqués à la cybersécurité.

3.1.3 STANDARDS ET NORMES ISO LIÉS À LA QUALITÉ DES DONNÉES

L'augmentation importante du volume de données exploitées dans les systèmes logiciels, les environnements automatisés et les applications d'intelligence artificielle a progressivement mis en évidence la nécessité d'encadrer la gestion et l'évaluation des données. La diversité des sources, les problèmes d'incohérence, les données manquantes ou encore les difficultés liées à leur exploitation ont ainsi conduit au développement de plusieurs standards et modèles internationaux consacrés à la qualité des données. Ces standards fournissent des cadres méthodologiques permettant de définir, d'évaluer et d'améliorer les données utilisées dans les systèmes logiciels et les systèmes intelligents [23, 24]. Avec l'essor de l'intelligence artificielle, du Big Data et des environnements fortement automatisés, ces référentiels occupent aujourd'hui une place de plus en plus importante puisqu'ils contribuent à structurer les mécanismes de mesure, de validation et de gouvernance des données [23].

Les travaux d'Oviedo et al. montrent notamment que les organismes de normalisation ont développé, au cours des dernières années, plusieurs normes destinées à améliorer la gouvernance, la fiabilité et l'évaluation des systèmes d'intelligence artificielle [23]. Les auteurs rappellent également que les meilleures pratiques en ingénierie de l'IA restent encore insuffisamment établies et que les systèmes intelligents nécessitent désormais des approches capables d'assurer simultanément la qualité des logiciels, des modèles et des données [23].

Dans les environnements DevSecOps, ces standards présentent un intérêt particulier puisqu'ils permettent de transformer la qualité des données en caractéristiques mesurables et comparables. Ils servent également de base pour concevoir des modèles de qualité adaptés

aux pipelines automatisés, aux systèmes d'analyse de vulnérabilités ainsi qu'aux mécanismes d'automatisation intelligente intégrés dans les pipelines CI/CD.

3.1.3.1 ISO/IEC 25012

La norme ISO/IEC 25012 constitue l'un des principaux modèles de référence consacrés à la qualité des données. Cette norme appartient à la famille SQuaRE et propose un modèle structuré permettant d'évaluer les données indépendamment du domaine d'utilisation [23].

La norme ISO/IEC 25012 organise les caractéristiques de qualité des données en deux grandes catégories : les caractéristiques directement liées aux données elles-mêmes et celles associées aux systèmes qui stockent, manipulent ou exploitent ces données [23, 10]. Les caractéristiques directement liées aux données comprennent notamment l'exactitude, l'exhaustivité, la cohérence, la crédibilité, l'actualité, la conformité ainsi que la précision [23, 10]. Ces caractéristiques permettent principalement d'évaluer la fiabilité et la validité des données elles-mêmes. La norme définit également plusieurs caractéristiques davantage associées aux systèmes exploitant les données, parmi lesquelles figurent l'accessibilité , la confidentialité, la disponibilité, la traçabilité, la portabilité, la récupérabilité, l'efficacité ainsi que la compréhensibilité [23]. Ces caractéristiques concernent principalement la manière dont les données sont accessibles, protégées, comprises et gérées au sein des systèmes numériques [23].

Les travaux de Ramirez-Torres et al. rappellent également que la qualité des données représente un concept multidimensionnel fortement dépendant du contexte d'utilisation des données [10]. Les auteurs expliquent notamment que les organisations doivent définir leurs exigences de qualité selon les besoins des systèmes développés et les objectifs visés [10].

Bautista Villalpando et al. montrent également que les caractéristiques proposées dans les normes ISO peuvent être intégrées dans les mécanismes d'évaluation des systèmes Big Data et Cloud Computing [8]. Ils expliquent que ces caractéristiques peuvent être associées à des indicateurs techniques permettant d'évaluer différents aspects de qualité au sein des systèmes numériques [8].

3.1.3.2 ISO/IEC 5259

La norme ISO/IEC 5259 s'inscrit parmi les standards les plus récents consacrés à la qualité des données pour l'intelligence artificielle et l'apprentissage automatique. Contrairement aux approches plus générales, cette famille de normes s'intéresse directement aux données utilisées dans les tâches analytiques et les systèmes d'apprentissage automatique [23].

Oviedo et al. expliquent que la famille ISO/IEC 5259 comprend plusieurs parties consacrées au modèle de qualité des données, aux mesures de qualité, aux exigences de gestion, aux processus d'évaluation ainsi qu'à la gouvernance des données utilisées dans les systèmes d'intelligence artificielle [23].

La norme ISO/IEC 5259-2 adapte également plusieurs caractéristiques déjà présentes dans la norme ISO/IEC 25012 au contexte spécifique de l'analyse de données et de l'apprentissage automatique [23]. Elle introduit notamment des caractéristiques supplémentaires comme la pertinence, la représentativité, la similarité, la diversité, l'équilibre ainsi que l'efficacité des étiquettes de données [23].

Dans les pipelines DevSecOps, cette norme apparaît particulièrement pertinente puisqu'elle permet de structurer les mécanismes d'évaluation et d'amélioration des ensembles de

données de vulnérabilités utilisés dans les tâches automatisées de détection, d'analyse et de priorisation des risques.

3.1.3.3 ISO/IEC 25000 ET ISO/IEC 25010

Les normes ISO/IEC 25000 et ISO/IEC 25010 appartiennent également à la famille SQuaRE dédiée à l'évaluation de la qualité des systèmes logiciels. La norme ISO/IEC 25010 propose un modèle structuré permettant d'évaluer différentes caractéristiques de qualité logicielle [23].

Parmi les principales caractéristiques définies dans cette norme figurent notamment l'efficacité des performances , la compatibilité , l'utilisabilité , la fiabilité, la sécurité , la maintenabilité ainsi que la portabilité [23]. Dans les environnements DevSecOps, ces caractéristiques sont particulièrement importantes puisque les pipelines doivent assurer simultanément rapidité, sécurité, robustesse et continuité des services.

Bautista Villalpando et al. expliquent également que la norme ISO/IEC 25010 distingue la qualité du produit logiciel et la qualité d'utilisation [8]. Les auteurs montrent que plusieurs caractéristiques peuvent être décomposées en sous-caractéristiques mesurables comme le comportement temporel, l'utilisation des ressources ou encore la tolérance aux fautes [8].

3.1.3.4 ISO/IEC 25059

Avec l'évolution récente des systèmes d'IA, de nouvelles extensions de la famille SQuaRE ont été développées afin d'adapter les modèles de qualité aux spécificités des systèmes intelligents. Oviedo et al. indiquent notamment que la norme ISO/IEC 25059 :2023 propose un modèle de qualité spécialement conçu pour les systèmes d'intelligence artificielle [23].

Cette norme introduit plusieurs caractéristiques importantes associées aux systèmes d'intelligence artificielle comme la transparence, la robustesse, la fiabilité, la contrôlabilité, l'intervenabilité ainsi que l'adaptabilité fonctionnelle [23].

Oviedo et al. montrent également que certaines caractéristiques déjà présentes dans la norme ISO/IEC 25010 ont été adaptées afin de mieux prendre en compte les spécificités des systèmes d'intelligence artificielle [23]. Cette évolution illustre l'adaptation progressive des standards ISO/IEC aux nouveaux besoins des systèmes intelligents et des environnements fortement orientés données [23].

3.1.4 DIMENSIONS DE QUALITÉ DES DONNÉES ISSUES DE LA LITTÉRATURE

Les dimensions de qualité des données proposées dans la littérature ne se limitent pas uniquement aux modèles définis par les standards ISO. De nombreux travaux issus de l'intelligence artificielle, de l'apprentissage automatique et de la cybersécurité introduisent des caractéristiques directement liées aux difficultés rencontrées lors de l'utilisation concrète des données dans les systèmes intelligents. Ces dimensions répondent aux besoins d'environnements où les données proviennent de sources multiples, évoluent rapidement et circulent dans des processus automatisés.

La qualité des annotations, la fiabilité des labels, la diversité, la représentativité, l'hétérogénéité des sources et la traçabilité des transformations apparaissent comme des préoccupations récurrentes dans la littérature [9, 14, 21, 17]. Dans les contextes de cybersécurité et de DevSecOps, ces caractéristiques sont particulièrement importantes, car les données de vulnérabilités sont utilisées pour entraîner, évaluer et alimenter des mécanismes automatisés de détection, de priorisation et d'aide à la décision.

La littérature scientifique apporte ainsi un complément aux standards ISO/IEC en introduisant des dimensions plus directement liées aux usages de l'IA, de l'apprentissage automatique et de la cybersécurité. La section suivante propose une synthèse structurée de ces dimensions et précise leur pertinence pour les environnements DevSecOps.

3.1.4.1 SYNTHÈSE DES MODÈLES DE QUALITÉ DES DONNÉES POUR L'IA ET LES ENVIRONNEMENTS DEVSECOPS

Dans cette perspective, cette section synthétise les principaux modèles et cadres de qualité des données proposés dans la littérature scientifique, en mettant l'accent sur les travaux liés à l'apprentissage automatique, à l'analyse logicielle et aux datasets de vulnérabilités logicielles.

Plusieurs travaux montrent que la qualité des données constitue un facteur déterminant pour les performances des systèmes d'intelligence artificielle et d'apprentissage automatique. Dans une perspective centrée sur les données, Mohammed et al. étudient l'effet de six dimensions de qualité sur les performances de différents algorithmes d'apprentissage automatique : la cohérence de représentation, la complétude, l'exactitude des caractéristiques, l'exactitude de la cible, l'unicité et l'équilibre des classes [29]. Ces dimensions montrent que la qualité des données ne concerne pas seulement l'absence de valeurs manquantes, mais aussi la fiabilité des attributs utilisés par les modèles, la qualité des labels et la distribution des classes.

Dans le domaine du génie logiciel, Bhatia et al. proposent une taxonomie des anti-modèles de qualité des données pour l'analyse logicielle. Ces anti-modèles incluent notamment les valeurs aberrantes, le chevauchement de classes, les distributions asymétriques, la redondance et les problèmes de bruit dans les données [9]. Ces éléments sont particulièrement importants pour les modèles d'apprentissage automatique utilisés dans les pipelines logiciels,

car ils peuvent affecter non seulement les performances prédictives, mais aussi l'interprétation des résultats produits par les modèles.

Les travaux portant spécifiquement sur les datasets de vulnérabilités logicielles confirment cette importance. Croft et al. analysent plusieurs ensembles de données de vulnérabilités et identifient cinq attributs majeurs de qualité : l'exactitude, l'unicité, la cohérence, l'exhaustivité et l'actualité [14]. Leurs résultats montrent que les datasets de vulnérabilités peuvent contenir des étiquettes inexactes, des doublons, des incohérences et des informations incomplètes, ce qui peut fausser l'entraînement et l'évaluation des modèles de prédiction des vulnérabilités.

D'autres travaux insistent sur la représentativité et l'équilibre des données. Deng et al. montrent que les datasets de vulnérabilités souffrent souvent d'un problème de distribution longue traîne, où certaines catégories de vulnérabilités sont fortement représentées tandis que d'autres sont rares [17]. Cette situation limite la capacité des modèles à détecter efficacement les vulnérabilités peu fréquentes. Dans le même sens, Guo et al. soulignent que les modèles d'IA appliqués à la détection de vulnérabilités nécessitent des ensembles de données complets, équilibrés, correctement étiquetés et représentatifs des situations réelles.

Ainsi, les modèles et cadres proposés dans la littérature scientifique définissent plusieurs caractéristiques pertinentes pour les environnements DevSecOps : la complétude, l'exactitude, la cohérence, l'unicité, l'actualité, la qualité des labels, l'équilibre des classes, la diversité, la représentativité et la réduction du bruit. Ces caractéristiques sont essentielles dans les environnements DevSecOps, car les données de vulnérabilités alimentent des mécanismes automatisés de détection, de priorisation et d'aide à la décision. Une faible qualité des données peut donc entraîner des prédictions erronées, une mauvaise priorisation des vulnérabilités ou une surestimation des performances des modèles.

3.2 BASES DE DONNÉES DE VULNÉRABILITÉS LOGICIELLES

Les travaux existants montrent que la qualité des données constitue un enjeu important pour les systèmes d'intelligence artificielle, l'apprentissage automatique et la cybersécurité [16, 40, 21]. Plusieurs études ont montré que les valeurs manquantes, les doublons, les incohérences, les erreurs d'annotation, les distributions déséquilibrées ou encore le bruit dans les données peuvent affecter les performances des modèles et la fiabilité des résultats produits. Ces contributions permettent de mieux comprendre les effets des problèmes de qualité sur les systèmes intelligents.

Dans le domaine des vulnérabilités logicielles, certains travaux se sont intéressés plus directement à la qualité des jeux de données utilisés pour l'analyse et la détection des vulnérabilités. Ces études ont mis en évidence des problèmes liés aux labels, aux doublons, à l'incomplétude, aux incohérences et à la représentativité des données [21, 11]. Elles montrent que les données de vulnérabilités ne sont pas toujours directement exploitables pour entraîner ou évaluer des modèles d'intelligence artificielle.

Les travaux d'Anwar et al. occupent une place importante dans ce contexte, car ils portent directement sur l'évaluation et l'amélioration de la qualité des données de la NVD [7]. Leur étude part du constat que la NVD est largement utilisée par les chercheurs, les analystes de sécurité et les outils automatisés, alors que ses données contiennent plusieurs anomalies susceptibles d'influencer les analyses de cybersécurité. Les auteurs examinent notamment les informations relatives aux dates de publication, aux fournisseurs, aux produits, aux scores CVSS, aux métriques CVSS et aux catégories CWE.

Leur contribution consiste d'abord à réaliser une analyse systématique des problèmes présents dans la NVD. Ils mettent en évidence des incohérences dans les dates de publication, des variations dans les noms de fournisseurs et de produits, des valeurs manquantes ou

incorrectes dans les scores de sévérité, ainsi que des problèmes liés à l'attribution des catégories CWE. Ces anomalies peuvent compliquer l'exploitation des données, fausser certaines analyses statistiques et réduire la fiabilité des outils qui utilisent la NVD comme source d'information.

Les auteurs proposent ensuite plusieurs méthodes de correction afin d'améliorer la qualité de ces données. Pour les noms de fournisseurs et de produits, ils utilisent des techniques de normalisation permettant de réduire les variations d'écriture et les doublons sémantiques. Pour les scores CVSS et les informations de sévérité, ils vérifient la cohérence entre les différentes composantes du score et les valeurs associées. Concernant les catégories CWE, ils cherchent à améliorer la précision des classifications en corrigeant certaines affectations inexactes ou trop générales. Leur approche combine ainsi des règles heuristiques, des traitements automatiques et des vérifications destinées à rendre les données plus cohérentes et plus exploitables.

Un aspect important de leur travail est qu'ils ne se limitent pas à signaler les erreurs présentes dans la NVD. Ils montrent également que ces erreurs peuvent avoir des conséquences sur les analyses de cybersécurité. Par exemple, des informations incorrectes sur les fournisseurs, les produits ou les scores de sévérité peuvent modifier les résultats d'une analyse de tendances, influencer la priorisation des vulnérabilités ou conduire à une interprétation erronée de l'évolution des risques. Leur étude montre donc que la qualité des données NVD a un effet direct sur les décisions et les analyses qui en dépendent.

Le présent mémoire s'inscrit dans la continuité de ces travaux, tout en adoptant une perspective différente. Alors que plusieurs recherches se concentrent sur l'identification de problèmes spécifiques ou sur la correction de certaines anomalies, cette recherche propose un modèle consolidé de qualité des données construit à partir des standards ISO/IEC et de la littérature scientifique. Ce modèle permet d'organiser les caractéristiques de qualité selon plusieurs catégories adaptées aux environnements DevSecOps intégrant les systèmes d'IA.

L'originalité de cette recherche réside également dans l'articulation entre trois niveaux d'analyse. Le premier concerne la construction d'un modèle de qualité des données adapté aux environnements DevSecOps intégrant les systèmes d'IA. Le deuxième porte sur l'application de ce modèle aux données NVD à travers des règles explicites de mesure et de nettoyage.

Ainsi, ce mémoire ne se limite pas à constater la présence d'anomalies dans les données de vulnérabilités. Il propose une démarche complète allant de l'identification des caractéristiques de qualité jusqu'à l'évaluation de leur impact sur la prédiction. Cette approche permet de positionner la qualité des données comme un élément central dans la fiabilité des mécanismes d'automatisation intelligente utilisés dans les pipelines DevSecOps.

CHAPITRE IV
MODÈLE CONSOLIDÉ DE QUALITÉ DES DONNÉES

Les standards ISO/IEC ainsi que les dimensions présentées dans la littérature permettent de mieux comprendre les différentes caractéristiques associées à la qualité des données. Toutefois, les environnements DevSecOps et les systèmes de cybersécurité possèdent certaines particularités liées à l'automatisation des tâches, à la diversité des sources de données, à l'évolution rapide des vulnérabilités et à l'utilisation croissante de mécanismes d'intelligence artificielle. Dans ce contexte, il devient nécessaire de disposer d'un modèle de qualité des données capable de répondre aux besoins des systèmes intelligents utilisés dans les pipelines DevSecOps.

Le modèle proposé dans cette recherche repose à la fois sur les caractéristiques issues des standards ISO/IEC et sur plusieurs dimensions identifiées dans les domaines de l'intelligence artificielle, de l'apprentissage automatique, de l'analyse logicielle et de la cybersécurité. Les caractéristiques retenues ont été sélectionnées en fonction de leur importance pour l'évaluation des ensembles de données de vulnérabilités exploités par les mécanismes automatisés d'analyse, de détection et de priorisation des vulnérabilités.

La construction du modèle repose sur une comparaison entre les caractéristiques proposées par les normes et standards, notamment ISO/IEC 25012, ISO/IEC 5259 et ISO/IEC 25059, et celles identifiées dans la littérature scientifique. Le Tableau 4.1 présente l'origine de chaque caractéristique ainsi que son statut dans le modèle proposé.

TABLEAU 4.1 : Origine des caractéristiques de qualité des données du modèle consolidé

Caractéristique	Normes / standards	Littérature scientifique	Statut
Complétude / exhaustivité	ISO/IEC 25012 ; ISO/IEC 5259	[14, 29, 11]	Retenue et mesurée
Exactitude	ISO/IEC 25012 ; ISO/IEC 5259	[14, 29]	Retenue et mesurée
Cohérence	ISO/IEC 25012	[14, 29, 9]	Retenue et mesurée
Actualité	ISO/IEC 25012	[14]	Retenue et mesurée
Unicité	ISO/IEC 25012 indirectement ; ISO/IEC 25024	[14, 29, 9]	Retenue et mesurée
Crédibilité / fiabilité	ISO/IEC 25012 ; ISO/IEC 25059	Littérature sur la qualité des données et l'IA [41, 11, 14, 29, 17]	Retenue et mesurée
Validité	ISO/IEC 5259 ; ISO/IEC 25012 partiellement	[9] ; littérature sur l'apprentissage automatique [41, 29, 23]	Retenue partiellement
Intégrité	ISO/IEC 25012 ; ISO/IEC 25059	Littérature sur la qualité des données [38, 23]	Non mesurée directement
Traçabilité	ISO/IEC 25012 ; ISO/IEC 5259	Littérature DevSecOps et apprentissage automatique [41, 14, 29, 21]	Intégrée au modèle, non mesurée comme dimension
Exploitabilité	ISO/IEC 25012 indirectement ; ISO/IEC 25059	Littérature DevSecOps, cybersécurité et apprentissage automatique [14, 17]	Intégrée au modèle, évaluée indirectement
Disponibilité	ISO/IEC 25012	Peu présente dans les travaux étudiés [38, 10, 23]	Intégrée au modèle, non mesurée
Conformité CVE / CWE / CPE / CVSS	Standards CVE, CWE, CPE et CVSS	Littérature sur les données de vulnérabilités et DevSecOps [14, 17, 21]	Intégrée au modèle, évaluée indirectement
Confidentialité	ISO/IEC 25012 ; ISO/IEC 27001	Peu présente dans les travaux étudiés [38, 10, 23, 14]	Non retenue dans l'expérimentation
Récupérabilité	ISO/IEC 25012	Peu présente dans les travaux étudiés [38, 17, 8]	Non retenue dans l'expérimentation
Qualité des labels	ISO/IEC 5259 indirectement	[14, 29, 11, 17, 41]	Retenue partiellement
Équilibre des classes	ISO/IEC 5259	[17, 29]	Non mesurée directement
Représentativité	ISO/IEC 5259	[17, 21]	Non mesurée directement
Diversité	ISO/IEC 5259	[17, 21]	Non mesurée directement
Réduction du bruit	ISO/IEC 5259 indirectement	[9, 11]	Retenue indirectement
Gestion des valeurs aberrantes	Non explicitement définie dans ISO/IEC 25012	[9] ; littérature sur l'apprentissage automatique	Retenue partiellement
Hétérogénéité	Non centrale dans ISO/IEC 25012	[21] ; littérature sur les jeux de données de vulnérabilités	Retenue partiellement

Ce tableau montre que les caractéristiques du modèle consolidé proviennent à la fois des standards ISO/IEC, des standards techniques utilisés dans les bases de vulnérabilités, tels que CVE, CWE, CPE et CVSS, et de la littérature scientifique.

À partir des caractéristiques recensées dans les standards et dans la littérature scientifique, le modèle consolidé proposé dans cette recherche est organisé en cinq catégories principales, comme le montre la Figure 4.1. Ces catégories sont la qualité intrinsèque, la qualité structurelle, la qualité temporelle, la qualité orientée IA/ML et la qualité opérationnelle DevSecOps. Cette organisation permet de représenter la qualité des données de manière hiérarchique, en tenant compte à la fois des dimensions générales de qualité, des besoins des systèmes d'intelligence artificielle et des exigences propres aux environnements DevSecOps.

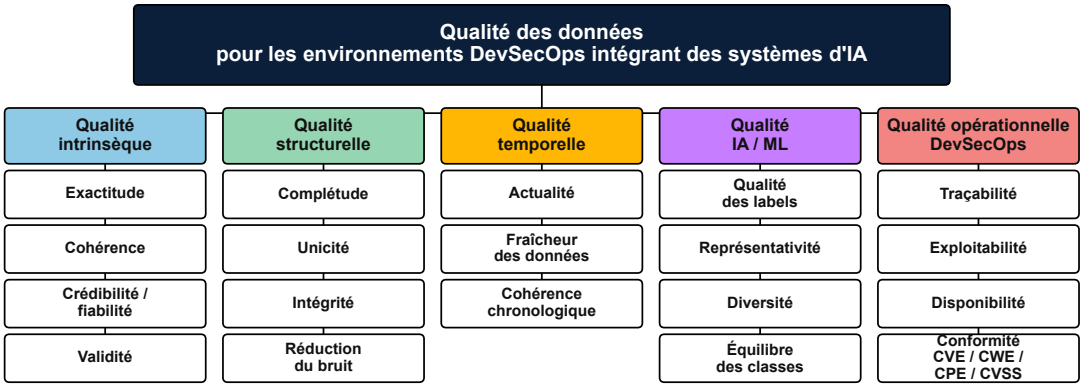

FIGURE 4.1 : Modèle consolidé de qualité des données pour les environnements DevSecOps intégrant les systèmes d'IA

La première catégorie, appelée qualité intrinsèque, regroupe les caractéristiques liées à la fiabilité interne des données. Elle comprend l'exactitude, la cohérence, la crédibilité ou fiabilité, ainsi que la validité. Ces caractéristiques permettent d'évaluer si les données décrivent correctement les vulnérabilités, si elles respectent une logique interne et si elles peuvent être considérées comme fiables pour les analyses automatisées.

La deuxième catégorie correspond à la qualité structurelle. Elle concerne l'organisation des données et leur capacité à être correctement exploitées dans des traitements automatisés. Cette catégorie regroupe la complétude, l'unicité, l'intégrité et la réduction du bruit. Dans le contexte des données NVD, elle permet notamment d'évaluer la présence des informations attendues, l'absence de doublons ou de répétitions inutiles, ainsi que la qualité générale de la structure des enregistrements.

La troisième catégorie est la qualité temporelle. Elle regroupe l'actualité, la fraîcheur des données et la cohérence chronologique. Cette catégorie est importante dans les environnements DevSecOps, car les données de vulnérabilités évoluent rapidement. Une vulnérabilité peut être publiée, modifiée, enrichie ou corrigée au fil du temps. Il est donc nécessaire de vérifier que les informations temporelles sont disponibles, cohérentes et suffisamment à jour pour soutenir les mécanismes de détection et de priorisation.

La quatrième catégorie est la qualité orientée IA/ML. Elle regroupe les caractéristiques particulièrement importantes pour les systèmes d'apprentissage automatique, notamment la qualité des labels, la représentativité, la diversité et l'équilibre des classes. Ces caractéristiques permettent d'évaluer si les données utilisées pour entraîner, tester ou évaluer des modèles intelligents sont suffisamment fiables, variées et adaptées aux tâches d'analyse ou de prédiction.

La cinquième catégorie correspond à la qualité opérationnelle DevSecOps. Elle regroupe la traçabilité, l'exploitabilité, la disponibilité et la conformité aux standards CVE, CWE, CPE et CVSS. Cette catégorie permet de prendre en compte les besoins pratiques des pipelines automatisés de cybersécurité. La traçabilité concerne la capacité à suivre l'origine des données et les transformations appliquées. L'exploitabilité renvoie à la possibilité d'utiliser les données dans des traitements automatisés. La disponibilité concerne l'accès aux données dans les

environnements opérationnels. Enfin, la conformité permet de vérifier que les informations respectent les formats et conventions attendus dans les bases de données vulnérabilités.

CHAPITRE V
MESURES DE QUALITÉ DES DONNÉES

5.1 CHOIX DU CAS D'ÉTUDE : DONNÉES DE VULNÉRABILITÉS ET NATIONAL VULNERABILITY DATABASE

Afin de valider l'applicabilité du modèle consolidé de qualité des données proposé dans cette recherche, le choix s'est porté sur les données de vulnérabilités issues de la National Vulnerability Database (NVD). La NVD a été retenue comme source principale, car elle représente l'une des bases de référence les plus utilisées pour la diffusion publique d'informations sur les vulnérabilités logicielles. Elle est publique, documentée et accessible dans un format exploitable automatiquement, ce qui favorise la reproductibilité de l'expérimentation.

Ce choix est également motivé par la place centrale occupée par les données de vulnérabilités dans les environnements de cybersécurité et les pipelines DevSecOps. Les informations relatives aux vulnérabilités sont utilisées pour l'identification des risques, la priorisation des correctifs, l'analyse de l'exposition des systèmes, l'automatisation des contrôles de sécurité et l'alimentation de mécanismes d'intelligence artificielle ou d'apprentissage automatique. La qualité de ces données influence donc directement la fiabilité des analyses produites et la pertinence des décisions prises à partir de ces analyses.

La NVD offre un accès structuré à des données normalisées autour de référentiels largement utilisés, tels que CVE pour l'identification des vulnérabilités, CWE pour la classification des faiblesses, CPE pour la description des produits affectés et CVSS pour l'évaluation de la sévérité. La présence de ces standards facilite la définition de règles de mesure explicites, reproductibles et automatisables, conformément à la démarche de métrologie logicielle mobilisée dans cette recherche.

Les données de vulnérabilités constituent ainsi un cas d'étude particulièrement pertinent pour l'évaluation de la qualité des données, car elles regroupent plusieurs types d'informations hétérogènes. Un enregistrement de vulnérabilité peut contenir un identifiant CVE, une description textuelle, des dates de publication et de modification, des scores CVSS, des niveaux de sévérité, des faiblesses CWE, ainsi que des critères CPE permettant d'identifier les produits ou plateformes affectés. Cette diversité permet d'observer plusieurs dimensions de qualité des données, notamment la complétude, l'unicité, la cohérence, l'exactitude, la crédibilité et l'actualité. Les données de vulnérabilités offrent ainsi un terrain expérimental riche pour transformer les caractéristiques théoriques du modèle en règles de mesure observables.

Le choix de la NVD se justifie également par la disponibilité de données historiques couvrant plusieurs années. Cette dimension temporelle permet d'étudier non seulement l'état des données à un moment donné, mais aussi certaines propriétés liées à leur évolution, comme l'actualité, la cohérence entre dates de publication et de modification, ou encore la stabilité des attributs associés aux vulnérabilités. Dans le cadre de cette recherche, les fichiers NVD des années 2021 à 2025 ont été retenus afin de disposer d'un corpus suffisamment large pour appliquer les règles de mesure et comparer les résultats sur plusieurs périodes.

Ce cas d'étude présente aussi un intérêt méthodologique. Les données NVD permettent de construire un pipeline de mesure dans lequel les entités observées, les attributs mesurés, les règles de mesure et les indicateurs calculés peuvent être définis de manière explicite. Cette configuration correspond aux exigences méthodologiques associées à la mesure logicielle, selon lesquelles l'évaluation doit reposer sur des critères observables, interprétables et reproductibles.

Enfin, l'utilisation des données de vulnérabilités permet de valider le modèle dans un contexte directement lié aux objectifs du mémoire. Les pipelines DevSecOps reposent de plus

en plus sur des traitements automatisés pour détecter, analyser et prioriser les vulnérabilités. Dans ce contexte, une donnée incomplète, incohérente, obsolète ou peu crédible peut avoir des effets importants sur la qualité des résultats produits. L'expérimentation menée sur les données NVD permet donc d'évaluer la capacité du modèle consolidé à rendre mesurables les problèmes de qualité susceptibles d'affecter les systèmes intelligents utilisés en cybersécurité.

Ainsi, les données de vulnérabilités de la NVD constituent un cas d'étude approprié pour l'expérimentation et la validation du modèle proposé. Elles combinent une forte pertinence métier, une structure fondée sur des standards reconnus, une disponibilité publique, une richesse d'attributs et une importance directe pour les mécanismes d'automatisation en DevSecOps.

5.2 MÉTROLOGIE LOGICIELLE

Les travaux d'Alain Abran en métrologie logicielle constituent le cadre méthodologique retenu dans cette recherche pour structurer la mesure de la qualité des données [2]. Ce choix se justifie par le besoin de passer d'un modèle conceptuel de qualité à une démarche de mesure opérationnelle, fondée sur des règles explicites, des résultats interprétables et une logique de vérification et de validation.

Selon Abran, une mesure ne se limite pas à un simple calcul numérique ni à l'application d'une formule mathématique. Elle correspond à un processus structuré permettant d'attribuer une valeur à une propriété observée selon des règles clairement définies [2]. Cette définition est pertinente dans le cadre de cette recherche, car elle rappelle qu'une mesure doit être construite à partir d'éléments clairement identifiés et interprétée selon une logique méthodologique rigoureuse.

La méthodologie d'Abran repose sur plusieurs principes fondamentaux. Elle exige d'abord de préciser l'objet sur lequel porte la mesure et la propriété qui doit être évaluée. Elle nécessite ensuite la définition de règles de mesure explicites, stables et applicables de manière reproductible. Enfin, elle suppose que les résultats obtenus soient interprétés à partir de critères définis, afin que les valeurs produites puissent réellement soutenir l'analyse.

Dans cette perspective, la mesure établit une relation entre un phénomène observé et sa représentation quantitative. Elle ne consiste donc pas seulement à produire un nombre, mais à construire une correspondance entre un monde empirique et un monde formel. Cette correspondance permet de représenter certaines propriétés sous une forme exploitable pour l'analyse et la prise de décision [2].

Abran distingue également plusieurs notions souvent confondues en métrologie logicielle, notamment les données, les mesures, les métriques et les indicateurs [2]. Les données correspondent aux observations de départ. Les mesures servent à quantifier une propriété observée à partir d'une règle définie. Les métriques correspondent aux mécanismes de calcul ou de transformation construits à partir des mesures. Les indicateurs permettent enfin d'interpréter les résultats obtenus afin de soutenir l'analyse.

La vérification et la validation constituent deux autres principes importants de cette méthodologie. La vérification consiste à s'assurer que les règles de mesure sont correctement appliquées et que les calculs produits sont exacts. La validation consiste à déterminer si la mesure représente réellement la propriété qu'elle prétend évaluer. Cette distinction renforce la rigueur de la démarche, car une valeur numérique ne constitue une mesure pertinente que si elle est correctement produite et correctement interprétée.

Ainsi, la méthodologie d'Abran a été retenue parce qu'elle offre un cadre rigoureux pour organiser la mesure de la qualité des données. Elle permet de définir les conditions générales

de mesure, de distinguer les données, les mesures, les métriques et les indicateurs, puis de vérifier et de valider les résultats obtenus. La section suivante présente l'application de ce cadre à la méthodologie de mesure développée dans ce mémoire.

5.3 MÉTHODOLOGIE DE MESURE DE LA QUALITÉ DES DONNÉES

Cette section présente l'application opérationnelle du cadre de métrologie logicielle défini précédemment. L'objectif est de décrire la manière dont les caractéristiques de qualité retenues dans le modèle consolidé ont été traduites en règles de mesure applicables aux données de vulnérabilités de la NVD. La Figure 5.1 présente la méthodologie de mesure adoptée dans cette recherche.

L'évaluation a été réalisée à travers un processus de mesure de la qualité des données de vulnérabilités extraites de la NVD pour les années 2021 à 2025. Les données, initialement fournies au format JSON, ont été transformées sous forme tabulaire afin de permettre l'identification des champs disponibles, l'application des règles de mesure et le calcul des indicateurs de qualité. Le Tableau 5.1 présente un exemple simplifié du format tabulaire obtenu après cette transformation.

Dans ce processus, les entités observées correspondent ici aux données de vulnérabilités issues des fichiers NVD. Les attributs à mesurer correspondent aux différentes dimensions de qualité des données retenues dans le modèle consolidé.

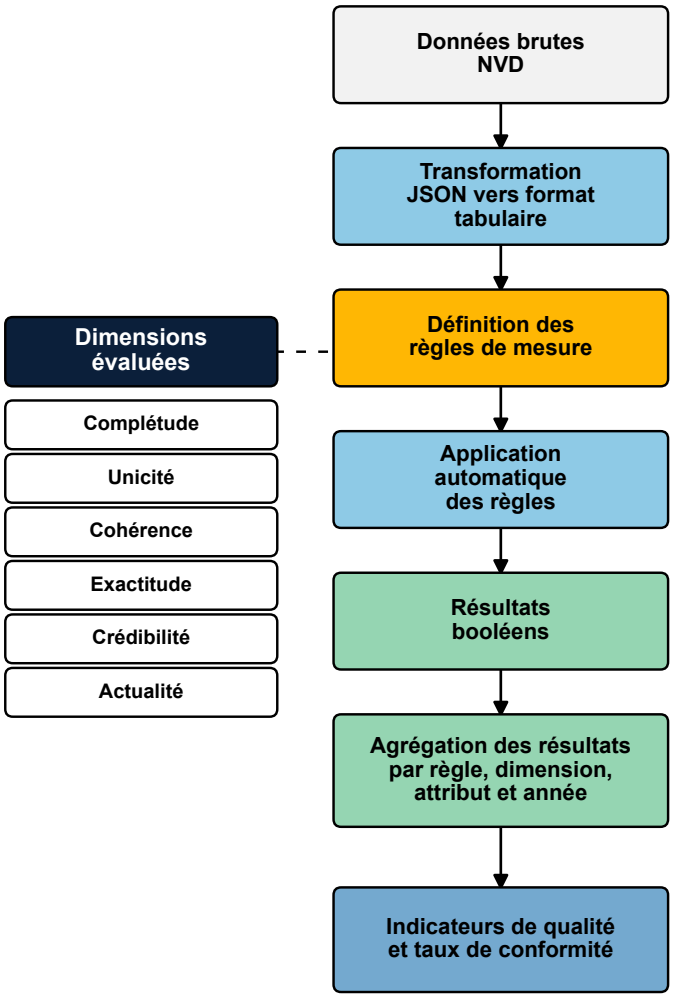

FIGURE 5.1 : Méthodologie de mesure de la qualité des données

TABLEAU 5.1 : Exemple de données NVD après transformation tabulaire

CVE_ID	PublishedDate	LastModifiedDate	Status	CVSS	Severity	CWE	Nombre_CPE
CVE-2021-3002	2021-01-01	2024-11-21	Modified	6.1	MEDIUM	CWE-79	1
CVE-2021-3004	2021-01-03	2024-11-21	Modified	7.5	HIGH	CWE-682	1
CVE-2021-3005	2021-01-03	2024-11-21	Modified	4.3	MEDIUM	NVD-CWE-noinfo	1
CVE-2025-0168	2025-01-01	2025-02-25	Analyzed	6.3	MEDIUM	CWE-74	1
CVE-2025-22214	2025-01-02	2026-04-15	Deferred	4.3	MEDIUM	CWE-89	0

Le modèle consolidé présenté à la Figure 4.1 du chapitre 4 regroupe plusieurs caractéristiques issues des standards ISO/IEC, des référentiels propres aux données de vulnérabilités et de la littérature scientifique sur la qualité des données, l'apprentissage automatique et les environnements DevSecOps. Selon Abran, une mesure ne correspond pas uniquement à un calcul numérique, mais à un processus structuré permettant d'associer une valeur à un attribut clairement défini d'une entité observée. Une caractéristique ne peut donc être mesurée directement que si elle peut être reliée à une entité observable, à un attribut mesurable, à une règle de mesure explicite et à une méthode d'interprétation reproductible.

TABLEAU 5.2 : Conditions associées aux statuts selon l'approche d'Abran

Famille de statut	Statuts concernés	Conditions satisfaites	Justification
Mesure directe	Retenue et mesurée	Entité observable ; attribut mesurable ; règle explicite ; interprétation reproductible	La caractéristique peut être mesurée directement avec les attributs disponibles dans les fichiers NVD.
Mesure partielle	Retenue partiellement	Entité observable ; attribut partiellement mesurable ; règle partielle ; interprétation reproductible	La caractéristique est importante, mais seulement une partie peut être mesurée avec les données disponibles.
Évaluation indirecte	Retenue indirectement ; intégrée au modèle, évaluée indirectement	Entité observable ; attribut indirect ; règle liée à une autre dimension ; interprétation reproductible	La caractéristique n'est pas mesurée seule. Elle est évaluée à travers d'autres règles.
Intégration conceptuelle sans mesure autonome	Intégrée au modèle, non mesurée ; intégrée au modèle, non mesurée comme dimension ; non mesurée directement	Entité partiellement observable ; attribut non directement mesurable ; absence de règle autonome	La caractéristique reste dans le modèle, mais elle ne peut pas être mesurée directement avec les fichiers NVD.
Exclusion expérimentale	Non retenue dans l'expérimentation	Conditions de mesure non satisfaites dans cette étude	La caractéristique ne correspond pas au contenu des enregistrements NVD ou dépasse le cadre de l'expérimentation.

Par conséquent, comme le montre le Tableau 5.2, cinq statuts ont été définis afin de passer du modèle conceptuel au dispositif expérimental. Ces mêmes statuts sont également présentés, pour chacune des caractéristiques identifiées, dans le Tableau 4.1 du chapitre 4. Le statut *retenu et mesuré* désigne une caractéristique intégrée au noyau opérationnel de mesure

et évaluée directement à l'aide de règles automatisables appliquées aux attributs disponibles dans les fichiers NVD. Le statut *retenu partiellement* désigne une caractéristique conservée dans le modèle, mais seulement mesurée sur certains aspects observables ou à travers un sous-ensemble de règles. Le statut *évalué indirectement* désigne une caractéristique qui n'est pas traitée comme une dimension autonome, mais dont certains effets sont captés par des règles rattachées à d'autres dimensions. Le statut *non mesuré directement* désigne une caractéristique conceptuellement pertinente, mais qui ne peut pas être évaluée de façon autonome dans cette expérimentation, soit parce qu'elle dépend d'éléments externes aux enregistrements NVD, soit parce que les attributs disponibles ne permettent pas de définir une règle de mesure suffisamment objective et reproductible. Enfin, le statut *non retenu dans l'expérimentation* désigne une caractéristique dont les conditions de mesure ne sont pas satisfaites dans le cadre de cette étude, car elle ne correspond pas au contenu des enregistrements NVD ou dépasse le cadre de l'expérimentation.

Le noyau opérationnel appliqué dans cette recherche est composé de six dimensions : la complétude, l'unicité, la cohérence, l'exactitude, la crédibilité et l'actualité. Ces dimensions ont le statut *retenue et mesurée*, car elles correspondent à des anomalies directement observables dans les fichiers NVD, telles que les valeurs manquantes, les doublons, les incohérences de format, les valeurs peu exploitables, les contradictions entre attributs et les anomalies temporelles. Elles peuvent également être traduites en règles de mesure automatisables à partir des attributs disponibles, notamment les identifiants CVE, les faiblesses CWE, les critères CPE, les scores CVSS, les niveaux de sévérité et les dates de publication ou de modification.

La validité, la qualité des labels, la gestion des valeurs aberrantes et l'hétérogénéité ont le statut *retenue partiellement*. Ces caractéristiques sont pertinentes pour les jeux de données utilisés dans les systèmes intelligents, mais elles ne sont pas toutes directement mesurables à partir des seuls champs disponibles dans les fichiers NVD. La validité est prise en compte

à travers les règles de format et de domaine de valeurs, par exemple pour les identifiants CVE, CWE, CPE et les scores CVSS. La qualité des labels est abordée indirectement par les contrôles portant sur les niveaux de sévérité, les scores CVSS et les catégories CWE, mais elle ne fait pas l'objet d'une évaluation supervisée complète, car l'étude ne dispose pas d'un jeu de labels de référence externe. La gestion des valeurs aberrantes est prise en compte à travers des règles de plausibilité, notamment sur les scores, les dates et les nombres de CPE. L'hétérogénéité est considérée dans la transformation des données JSON vers une structure tabulaire, mais elle n'est pas mesurée comme une dimension indépendante.

La traçabilité, l'exploitabilité, la conformité CVE/CWE/CPE/CVSS et la réduction du bruit ont le statut *évaluée indirectement*. La traçabilité est prise en compte à travers le suivi des transformations, les journaux de nettoyage et les règles temporelles permettant d'observer l'évolution des enregistrements. L'exploitabilité est évaluée à travers la présence, la validité et le caractère informatif des attributs nécessaires aux traitements automatisés. La conformité aux standards CVE, CWE, CPE et CVSS est mesurée par plusieurs règles de cohérence et d'exactitude, sans être isolée comme une dimension séparée. La réduction du bruit est également évaluée à travers les règles qui identifient les valeurs génériques, inconnues, vides ou peu informatives.

D'autres caractéristiques sont conservées dans le modèle consolidé, mais ne sont pas mesurées directement dans cette expérimentation. L'intégrité, la disponibilité, l'équilibre des classes, la représentativité et la diversité relèvent de ce statut. L'intégrité nécessiterait une vérification plus large des relations entre plusieurs sources ou référentiels externes. La disponibilité dépend davantage de l'accès aux sources, de la continuité de publication et de l'infrastructure de diffusion de la NVD que du contenu interne des enregistrements analysés. L'équilibre des classes, la représentativité et la diversité nécessitent un cadre d'analyse spécifique portant sur

la constitution d'échantillons, les distributions de classes ou les populations de référence, ce qui dépasse l'objectif principal de la présente démarche méthodologique.

Enfin, la confidentialité et la récupérabilité ne sont pas retenues dans l'expérimentation. La confidentialité concerne principalement la protection des données sensibles, alors que les données NVD utilisées dans cette recherche sont des données publiques de vulnérabilités. La récupérabilité relève davantage de la capacité d'un système ou d'une infrastructure à restaurer des données après incident que de la qualité intrinsèque des enregistrements contenus dans les fichiers NVD. Conformément à la méthodologie d'Abran, ces caractéristiques ne sont donc pas opérationnalisées, car elles ne peuvent pas être rattachées de manière pertinente à des attributs mesurables du jeu de données étudié.

Conformément au cadre de métrologie logicielle retenu, chaque dimension de qualité du noyau opérationnel a été associée à un ensemble explicite de règles de mesure. Chaque règle définit précisément les conditions qu'une donnée doit respecter afin d'être considérée comme conforme. Le processus de mesure appliqué aux données de vulnérabilités de la NVD n'effectue donc pas directement des opérations de nettoyage ou de correction ; il mesure uniquement le niveau de conformité des données par rapport aux règles établies. Cette distinction est importante dans la perspective d'Abran, puisque la mesure doit rester indépendante des opérations de transformation des données afin de garantir une évaluation objective et reproductible [2].

5.3.1 PROCESSUS DE DÉFINITION DES RÈGLES

La définition des règles de mesure a été réalisée à partir des caractéristiques de qualité présentées dans le Chapitre 4 et ayant le statut *retenue et mesurée*. Ces caractéristiques constituent le noyau opérationnel de mesure retenu dans cette recherche. L'objectif est de

transformer ces dimensions théoriques de qualité des données en critères mesurables pouvant être évalués automatiquement sur les données de vulnérabilités de la NVD.

Les règles ont été associées aux attributs du jeu de données sur lesquels elles pouvaient être évaluées. Cette mise en correspondance a été réalisée en fonction de la nature des informations contenues dans chaque colonne. Par exemple, les règles portant sur les identifiants ont été appliquées à la colonne *CVE_ID*, les règles relatives aux scores de vulnérabilité aux colonnes *CVSS* et *Severity*, tandis que les règles temporelles ont été appliquées aux colonnes PublishedDate et LastModifiedDate.

Dans un quatrième temps, des règles de validation explicites ont été définies pour chaque combinaison dimension-attribut jugée pertinente. Chaque règle traduit une condition précise permettant d'évaluer un aspect particulier de la qualité des données. La formulation des règles s'est appuyée sur les définitions théoriques des dimensions de qualité, sur les spécifications des standards CVE, CWE, CPE et CVSS ainsi que sur certaines caractéristiques observées dans les données analysées. La démarche a conduit à la construction d'un ensemble de règles couvrant les dimensions de complétude, d'unicité, de cohérence, d'exactitude, de crédibilité et d'actualité.

Les règles de mesure définies pour les caractéristiques ayant le statut *retenue et mesurée* sont présentées dans les sections suivantes selon la dimension de qualité qu'elles permettent d'évaluer. Pour chaque dimension, la démarche ayant conduit à la définition des règles est d'abord décrite. L'ensemble des règles retenues pour les différentes dimensions de qualité est synthétisé dans les Tableaux 5.3 à 5.8.

5.3.1.1 COMPLÉTUDE

Les règles de complétude ont été définies à partir de la définition classique de la complétude dans la littérature sur la qualité des données. Selon cette définition, une donnée est considérée comme complète lorsque l'information attendue est présente et disponible pour l'utilisation [11, 14]. Cette définition a conduit à la formulation de deux règles complémentaires. La première vérifie la présence effective de l'information en contrôlant l'absence de valeurs nulles. La seconde vérifie que les valeurs présentes sont réellement exploitables et ne correspondent pas à des contenus non informatifs tels que null, unknown, n/a, NVD-CVE-noinfo, ou certaines chaînes vides etc. La distinction a été introduite afin d'éviter de considérer comme complètes des données présentes mais dépourvues de valeur informative. Les règles retenues pour évaluer la dimension de complétude sont présentées dans le Tableau 5.3.

TABLEAU 5.3 : Règles associées à la dimension de complétude

ID	Règle de mesure	Justification de la règle
COMP_001	La valeur ne doit pas être nulle	Une information absente ne peut pas être considérée comme complète puisqu'elle ne peut pas être utilisée dans les analyses ou les traitements automatisés.
COMP_002	La valeur ne doit pas être vide ou correspondre à une valeur non exploitable	Une information présente mais non informative (*null*, *unknown*, *n/a*, *NVD-CVE-noinfo*, chaîne vide) ne contribue pas à la complétude effective des données.

5.3.1.2 UNICITÉ

Les règles d'unicité ont été définies à partir du principe selon lequel une même entité ne doit pas être représentée plusieurs fois dans les données. Toutefois, cette dimension n'a

pas été appliquée uniformément à l'ensemble des colonnes du jeu de données. Certaines colonnes peuvent en effet contenir naturellement des répétitions sans pour autant constituer des anomalies de qualité. Par exemple, plusieurs vulnérabilités peuvent partager le même niveau de sévérité, le même fournisseur ou la même catégorie de faiblesse. Le tableau 5.1 présente un exemple de données illustrant cette situation.

L'analyse s'est donc concentrée sur les attributs pour lesquels l'unicité constitue une propriété attendue. Dans le cas des identifiants CVE, chaque identifiant est conçu pour représenter une vulnérabilité unique. Une règle a ainsi été définie afin de vérifier qu'un même CVE_ID n'apparaît qu'une seule fois dans le jeu de données. Des règles complémentaires ont également été définies afin de détecter les répétitions inutiles pouvant apparaître à l'intérieur de certaines cellules contenant plusieurs valeurs, notamment les listes de CPE, de fournisseurs ou de produits. Ces répétitions n'apportent aucune information supplémentaire et peuvent introduire des biais dans les analyses réalisées sur les données. Les règles associées à la dimension d'unicité sont présentées dans le Tableau 5.4.

TABLEAU 5.4 : Règles associées à la dimension d'unicité

ID	Règle de mesure	Justification de la règle
UNI_001	Chaque identifiant CVE_ID doit apparaître une seule fois dans le jeu de données.	Un identifiant CVE est conçu pour représenter une vulnérabilité unique. La présence de doublons peut entraîner une surreprésentation artificielle de certaines vulnérabilités dans les analyses.
UNI_002	Une même vulnérabilité ne doit pas contenir des CPE répétés.	La répétition d'un même CPE dans une cellule multi-valeurs n'apporte aucune information supplémentaire et peut fausser certains indicateurs calculés à partir des produits affectés.
UNI_003	Une même vulnérabilité ne doit pas contenir des fournisseurs répétés.	La présence répétée d'un même fournisseur dans une même vulnérabilité constitue une redondance informationnelle susceptible de dégrader la qualité des données.
UNI_004	Une même vulnérabilité ne doit pas contenir des produits répétés.	La répétition d'un même produit dans une cellule multi-valeurs représente une duplication inutile pouvant introduire des biais dans les analyses statistiques.

5.3.1.3 COHÉRENCE

Les règles de cohérence ont été définies à partir des contraintes structurelles imposées par les standards utilisés dans la NVD ainsi que des relations logiques existant entre différents attributs du jeu de données. L'objectif de cette dimension est de vérifier que les informations respectent les formats attendus, appartiennent aux domaines de valeurs autorisés et ne présentent pas de contradictions internes.

La définition de ces règles s'est appuyée principalement sur les spécifications officielles des standards CVE, CWE, CPE et CVSS. Ces standards définissent des conventions précises concernant la structure des identifiants, les formats de représentation ainsi que certaines contraintes de cohérence entre les informations décrivant une vulnérabilité. Des règles complé-

mentaires ont également été définies afin de vérifier la cohérence entre plusieurs colonnes du jeu de données, notamment les informations temporelles et les attributs décrivant les produits affectés. Le tableau 5.1 présente un exemple de données illustrant cette situation.

La démarche a conduit à la définition de règles portant sur les formats des identifiants, la validité des dates, les relations chronologiques entre les événements ainsi que la cohérence entre certaines informations présentes dans différentes colonnes du jeu de données. Les règles utilisées pour évaluer la cohérence des données sont présentées dans le Tableau 5.5.

TABLEAU 5.5 : Règles associées à la dimension de cohérence

ID	Règle de mesure	Justification de la règle
COH_001	Le CVE_ID doit respecter le format CVE-YYYY-NNNN+.	Cette règle est définie à partir de la nomenclature officielle du standard CVE afin de garantir la validité syntaxique des identifiants de vulnérabilités.
COH_002	L'année présente dans le CVE_ID doit correspondre à la colonne Year.	Les deux attributs représentent la même information temporelle et doivent donc être cohérents.
COH_003	La date de publication doit avoir un format valide.	La date de publication doit respecter un format de date valide. Cette règle est définie afin de garantir la validité syntaxique des informations temporelles et de permettre leur exploitation fiable dans les analyses chronologiques.
COH_004	La date de modification doit avoir un format valide.	Cette règle garantit l'utilisation correcte des informations relatives aux mises à jour de la vulnérabilité.
COH_005	La date de modification doit être postérieure ou égale à la date de publication.	Une vulnérabilité ne peut pas être modifiée avant d'avoir été publiée.
COH_006	Le score CVSS doit être une valeur numérique.	Le standard CVSS définit le score comme une valeur numérique utilisée pour évaluer la sévérité d'une vulnérabilité.
COH_007	La sévérité doit appartenir aux catégories autorisées.	Les niveaux de sévérité doivent respecter les catégories définies par le standard CVSS.
COH_008	Le CWE doit respecter le format CWE-XXXX ou CWE-UNKNOWN.	Cette règle est définie à partir de la nomenclature officielle du standard CWE.
COH_009	Chaque CPE doit respecter une structure valide CPE 2.3.	Les identifiants CPE doivent suivre la structure normalisée définie par le standard CPE.
COH_010	Le nombre déclaré de CPE doit correspondre au nombre réel de CPE présents.	Les informations contenues dans les colonnes descriptives et numériques doivent être cohérentes entre elles.

5.3.1.4 EXACTITUDE

Les règles d'exactitude ont été définies à partir des contraintes métier associées à la description des vulnérabilités logicielles ainsi que des spécifications du standard CVSS utilisé

par la NVD. L'objectif de cette dimension est de vérifier que les valeurs observées sont plausibles, cohérentes avec leur signification métier et suffisamment représentatives de la réalité décrite.

Certaines règles ont été directement dérivées du standard CVSS. C'est notamment le cas des règles portant sur l'intervalle de valeurs autorisé pour les scores CVSS ainsi que sur la correspondance entre les scores et les niveaux de sévérité associés. D'autres règles ont été définies afin d'évaluer la qualité des informations textuelles présentes dans le jeu de données. Ces règles visent à s'assurer que les descriptions, les fournisseurs et les produits contiennent des informations exploitables et ne se limitent pas à des contenus insuffisants ou peu représentatifs.

Des contrôles complémentaires ont également été introduits afin de vérifier la plausibilité de certains attributs numériques et la cohérence de certaines informations avec le contexte de l'étude. Ce choix permet d'évaluer dans quelle mesure les données reflètent correctement les caractéristiques des vulnérabilités décrites. Les règles associées à la dimension d'exactitude sont présentées dans le Tableau 5.6.

TABLEAU 5.6 : Règles associées à la dimension d'exactitude

ID	Règle de mesure	Justification de la règle
EXA_001	Le score CVSS doit être compris entre 0 et 10.	Cette règle est définie à partir des spécifications du standard CVSS, qui établit une échelle de notation comprise entre 0 et 10 pour évaluer la sévérité des vulnérabilités. Toute valeur située en dehors de cet intervalle est considérée comme invalide.
EXA_002	Le niveau de sévérité doit correspondre au score CVSS observé.	Cette règle vise à garantir la cohérence entre le score CVSS et le niveau de sévérité associé, les catégories de sévérité étant déterminées à partir des intervalles définis par le standard CVSS.
EXA_003	La description ne doit pas être vide et doit contenir un texte descriptif.	Cette règle vise à garantir la présence d'informations descriptives permettant d'identifier et de comprendre la vulnérabilité.
EXA_004	Le fournisseur doit être renseigné sous forme de texte et ne pas être constitué uniquement de caractères numériques.	Cette règle vise à garantir l'identification correcte du fournisseur associé à la vulnérabilité. Un nom composé uniquement de chiffres ne permet généralement pas d'identifier de manière fiable l'entité concernée.
EXA_005	Le produit doit être renseigné sous forme de texte et ne pas être constitué uniquement de caractères numériques.	Le produit doit être textuel et ne pas être constitué uniquement de chiffres. Cette règle vise à garantir l'identification correcte du produit associé à la vulnérabilité.
EXA_006	Le nombre de CPE doit être supérieur ou égal à 0.	Le nombre de produits affectés représente une quantité qui ne peut pas être négative.
EXA_007	La colonne Year doit correspondre à l'année du fichier analysé.	Cette règle est définie à partir de l'organisation des données de la NVD, chaque fichier annuel étant destiné à regrouper les vulnérabilités associées à une année donnée.
EXA_008	La longueur de la description doit être supérieure ou égale à 10 caractères.	Seuil défini à partir de l'analyse exploratoire de la distribution des longueurs des descriptions, afin d'écarter les descriptions extrêmement courtes et peu informatives.

5.3.1.5 CRÉDIBILITÉ

Les règles de crédibilité ont été définies à partir de l'analyse exploratoire des valeurs textuelles et catégorielles présentes dans les données de la NVD, notamment les champs relatifs aux fournisseurs, aux produits et aux faiblesses CWE. Cette analyse a consisté à repérer les valeurs génériques, inconnues ou peu informatives utilisées comme substituts lorsqu'une information n'est pas disponible ou n'a pas pu être déterminée avec précision. Bien que ces valeurs ne soient pas nécessairement erronées, elles apportent peu d'information utile pour l'analyse des vulnérabilités et peuvent réduire la confiance accordée aux données.

Dans cette étude, la crédibilité a donc été associée au caractère informatif des données. Les règles définies visent à identifier les valeurs génériques ou artificielles susceptibles de limiter l'exploitation réelle des informations. Une donnée est considérée comme plus crédible lorsqu'elle fournit une information précise et exploitable plutôt qu'une valeur de remplacement indiquant une absence d'information. Les règles utilisées pour évaluer la crédibilité des données sont présentées dans le Tableau 5.7.

TABLEAU 5.7 : Règles associées à la dimension de crédibilité

ID	Règle de mesure	Justification de la règle
CRE_001	La valeur du fournisseur ne doit pas être *unknown_vendor*.	Cette valeur indique que le fournisseur n'a pas pu être identifié avec précision. Sa présence réduit la crédibilité et la valeur informative des données relatives à la vulnérabilité.
CRE_002	La valeur du produit ne doit pas être *unknown_product*.	Cette valeur correspond à une information générique qui ne permet pas d'identifier clairement le produit concerné par la vulnérabilité.
CRE_003	La valeur du CWE ne doit pas être *CWE-UNKNOWN* .	Cette règle vise à garantir l'identification précise de la faiblesse à l'origine de la vulnérabilité. La valeur CWE-UNKNOWN traduit une absence d'information spécifique, ce qui limite l'exploitation et l'interprétation des données.

5.3.1.6 ACTUALITÉ

Les règles d'actualité ont été définies à partir de la dimension temporelle des données de vulnérabilités. Cette dimension vise à évaluer dans quelle mesure les informations temporelles associées aux vulnérabilités sont disponibles, cohérentes et exploitables pour l'analyse. Dans le contexte de cette recherche, l'actualité ne se limite pas à la récence des données, mais renvoie également à leur capacité à situer correctement une vulnérabilité dans le temps et à assurer la traçabilité de son évolution.

La définition des règles s'est appuyée à la fois sur les principes de qualité temporelle des données et sur l'organisation annuelle des fichiers publiés par la NVD. Certaines règles ont ainsi été définies afin de vérifier la cohérence entre l'année de publication d'une vulnérabilité et l'année du fichier analysé. D'autres règles visent à garantir la disponibilité des informations temporelles nécessaires au suivi des vulnérabilités, notamment les dates de publication et de modification. Enfin, une règle complémentaire a été introduite afin de vérifier que le délai

entre la publication et la dernière modification peut être calculé, ce qui permet d'évaluer la traçabilité temporelle des informations. Les règles associées à la dimension d'actualité sont présentées dans le Tableau 5.8.

TABLEAU 5.8 : Règles associées à la dimension d'actualité

ID	Règle de mesure	Justification de la règle
ACT_001	L'année de publication doit correspondre à l'année du fichier analysé.	Chaque fichier annuel de la NVD est associé à une année donnée. Cette règle vise à garantir la cohérence temporelle entre les informations de publication de la vulnérabilité et le fichier source auquel elle appartient.
ACT_002	La date de modification doit être présente.	Cette règle vise à garantir la disponibilité des informations relatives aux mises à jour des vulnérabilités. L'absence de date de modification limite la traçabilité et le suivi de leur évolution dans le temps.
ACT_003	Le délai entre la publication et la dernière modification doit être calculable.	Cette règle est définie afin de permettre l'évaluation du suivi temporel des vulnérabilités. Le calcul du délai entre la publication et la dernière modification nécessite la présence de dates valides et cohérentes.

Les règles de mesure définies précédemment constituent le référentiel d'évaluation de la qualité des données retenu dans le cadre de cette recherche. Elles sont ensuite appliquées automatiquement à l'ensemble du jeu de données étudié afin de mesurer le niveau de conformité des enregistrements et de générer les indicateurs présentés dans la section suivante.

5.3.2 APPLICATION DES RÈGLES DE MESURE ET CALCUL DES INDICATEURS

Une fois les règles de mesure définies, elles sont appliquées automatiquement à l'ensemble des données analysées. Les résultats obtenus permettent ensuite de calculer différents indicateurs quantitatifs destinés à évaluer le niveau de conformité des données par rapport au référentiel de qualité retenu.

Les mesures réalisées dans cette recherche n'ont pas pour objectif de fournir une évaluation absolue de la qualité des données. Elles visent plutôt à déterminer dans quelle mesure les données respectent les règles de qualité définies dans le cadre de l'étude. Pour chaque règle évaluée, le processus automatisé de mesure appliqué aux données de vulnérabilités de la NVD vérifie si les valeurs analysées satisfont ou non la contrainte correspondante. Les résultats prennent alors la forme de valeurs booléennes : *True* lorsqu'une donnée est conforme à la règle considérée et *False* lorsqu'elle ne l'est pas.

Le principe d'application d'une règle de mesure est illustré dans le Tableau 5.9. Dans cet exemple, la règle COMP_001 vérifie la présence d'un identifiant CVE non nul. Chaque enregistrement est évalué individuellement et se voit attribuer une valeur booléenne indiquant sa conformité ou sa non-conformité à la règle appliquée.

TABLEAU 5.9 : Exemple de contrôle de conformité de la colonne CVE_ID

Ligne	CVE_ID	Règle appliquée	Résultat booléen	Interprétation
1	CVE-2024-0001	CVE_ID non nul	True	Conforme
2	CVE-2024-0002	CVE_ID non nul	True	Conforme
3	NONE	CVE_ID non nul	False	Non conforme
4	CVE-2024-0004	CVE_ID non nul	True	Conforme
5	NONE	CVE_ID non nul	False	Non conforme

À partir de ces contrôles booléens, les résultats individuels sont agrégés afin de produire différents indicateurs quantitatifs. Le nombre total de valeurs vérifiées correspond au nombre d'éléments analysés pour une règle donnée. Le nombre de valeurs conformes correspond au total des observations pour lesquelles le résultat du contrôle est *True*, tandis que le nombre de valeurs non conformes correspond au total des observations pour lesquelles le résultat est *False*.

Les indicateurs obtenus sont ensuite utilisés pour calculer le taux de conformité, lequel permet d'évaluer le niveau de respect d'une règle de qualité au sein du jeu de données. Ce taux est calculé à l'aide de la formule suivante :

$$\text{Taux de conformité} = \frac{\text{Nombre de valeurs conformes}}{\text{Nombre total de valeurs vérifiées}} \times 100$$

Cette formule exprime, sous forme de pourcentage, la proportion de données conformes à une règle donnée. Par exemple, si une règle vérifie la présence d'un identifiant CVE valide sur 10 000 enregistrements et que 9 500 d'entre eux respectent cette contrainte, le taux de conformité obtenu est de 95%.

Les résultats booléens obtenus lors de l'application de la règle, illustrés dans le Tableau 5.9, sont ensuite agrégés afin de calculer les différents indicateurs de conformité. Le Tableau 5.10 présente la synthèse obtenue à partir de cet exemple, en regroupant le nombre total de valeurs vérifiées, le nombre de valeurs conformes, le nombre de valeurs non conformes ainsi que le taux global de conformité associé à la règle évaluée.

TABLEAU 5.10 : Synthèse des mesures calculées

Indicateur	**Valeur**
Total vérifié	5
Valeurs conformes	3
Valeurs non conformes	2
Taux de conformité (%)	60

Dans le processus automatisé de mesure appliqué aux données de vulnérabilités de la NVD, les résultats calculés pour chaque règle sont ensuite regroupés afin de produire des indicateurs globaux par dimension de qualité, par attribut et par année étudiée. Cette agrégation

consiste à compter le nombre de valeurs conformes et non conformes, puis à calculer des taux de conformité pour chaque dimension de qualité ainsi que pour chaque attribut analysé. Cette méthode permet de quantifier de manière structurée le niveau de qualité des données et d'identifier les principales sources de non-conformité observées dans le jeu de données.

Cette méthodologie permet ainsi d'obtenir une évaluation structurée, quantitative et reproductible de la qualité des données utilisées dans les pipelines DevSecOps. Les mesures obtenues servent de base à l'analyse de la qualité des données et permettent d'évaluer l'impact de cette qualité sur les mécanismes d'analyse et d'automatisation étudiés dans ce mémoire.

L'évaluation de la qualité des données a permis d'identifier les différentes non-conformités présentes dans le jeu de données étudié. Toutefois, la détection de ces problèmes constitue uniquement une première étape. Afin d'améliorer la qualité des données et de réduire l'impact des anomalies observées sur les analyses ultérieures, un processus de nettoyage a été mis en œuvre.

La section suivante présente la méthodologie de nettoyage des données adoptée dans cette recherche. Cette méthodologie s'appuie sur les résultats de l'évaluation de la qualité et vise à corriger, supprimer ou normaliser les données non conformes identifiées par les règles de mesure précédemment définies.

5.4 MÉTHODOLOGIE DE NETTOYAGE DES DONNÉES

La démarche méthodologique adoptée dans ce chapitre ne se limite pas à la mesure initiale de la qualité des données. Elle comprend également des étapes successives de nettoyage et de réévaluation permettant d'améliorer progressivement la conformité du jeu de données aux règles de qualité définies. Le processus général comprend d'abord la transformation des données brutes de la NVD en format tabulaire, puis l'application des règles de mesure afin

d'établir un état initial de la qualité. Un premier nettoyage conservateur est ensuite appliqué, suivi d'une réévaluation des données. Enfin, un second nettoyage plus strict est réalisé sur les anomalies résiduelles, avant une réévaluation finale du jeu de données obtenu. La Figure 5.2 présente les principales étapes du processus de nettoyage et de réévaluation mis en œuvre dans cette recherche.

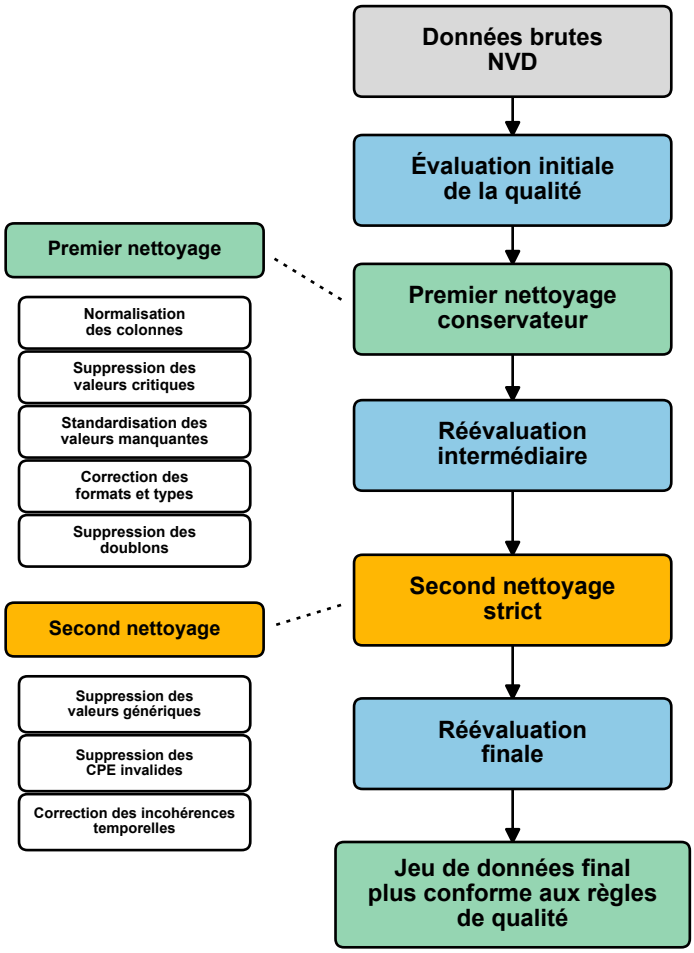

FIGURE 5.2 : Méthodologie de nettoyage des données NVD

5.4.1 PREMIER NETTOYAGE

À la suite de la phase d'évaluation de la qualité des données brutes, un premier pipeline de nettoyage a été mis en œuvre afin de corriger progressivement les principales anomalies détectées dans les jeux de données NVD. Cette étape ne visait pas à supprimer systématiquement les données jugées non conformes, mais plutôt à améliorer leur qualité tout en préservant un maximum d'informations exploitables pour les analyses DevSecOps.

Les opérations de nettoyage appliquées ont été définies à partir des problèmes identifiés lors de l'évaluation de la qualité des données. Elles concernent notamment le traitement des valeurs manquantes, la correction des incohérences de format, l'élimination des doublons, la résolution de certaines anomalies temporelles ainsi que la gestion de différentes formes de valeurs peu informatives ou non exploitables présentes dans le jeu de données.

Le nettoyage a été réalisé séparément pour chaque année étudiée afin de préserver la cohérence temporelle des jeux de données annuels. Le pipeline débute par une phase de normalisation générale des colonnes. Les noms des attributs sont d'abord harmonisés afin d'éviter les problèmes liés aux différences d'écriture, aux variations de casse ou à la présence d'espaces superflus.

Un nettoyage textuel global est ensuite appliqué aux principales colonnes textuelles du jeu de données. Cette opération permet notamment de supprimer les espaces inutiles, les retours à la ligne, les espaces multiples ainsi que différents caractères parasites susceptibles d'altérer l'analyse des données. Elle vise également à identifier plusieurs formes de valeurs non exploitables telles que *null*, *none*, *n/a*, *unknown*, les chaînes vides ou certaines structures incomplètes. Ces valeurs sont ensuite converties en valeurs manquantes standardisées afin de faciliter leur traitement lors des étapes ultérieures du pipeline de nettoyage.

Une seconde étape concerne le traitement des attributs considérés comme essentiels pour les analyses ultérieures. Les enregistrements ne contenant pas d'identifiant *CVE_ID* valide sont supprimés, cet identifiant constituant la référence principale permettant d'identifier de manière unique chaque vulnérabilité dans la base NVD. En l'absence de cet attribut, il devient impossible d'assurer la traçabilité et l'identification correcte de la vulnérabilité concernée.

Les enregistrements ne contenant pas de description exploitable sont également supprimés. La description constitue en effet une source d'information essentielle pour la compréhension des vulnérabilités et pour les traitements automatisés réalisés dans cette recherche. Les enregistrements dépourvus d'information descriptive pertinente ne peuvent être exploités de manière fiable dans les analyses textuelles ni dans les mécanismes d'analyse fondés sur le contenu des vulnérabilités.

Contrairement aux attributs critiques, certaines colonnes importantes, notamment le *CWE*, le fournisseur et le produit, ne sont pas supprimées lorsqu'elles contiennent des valeurs manquantes. En effet, bien que ces informations soient utiles pour l'analyse des vulnérabilités, leur absence ne rend pas l'enregistrement inutilisable. Afin de préserver autant que possible les informations exploitables, les valeurs absentes sont remplacées par des valeurs de substitution telles que *CWE-UNKNOWN*, *unknown_vendor* ou *unknown_product*.

Ce choix permet de limiter la perte de données tout en conservant la structure générale des jeux de données. Ces valeurs de substitution ne constituent toutefois pas des valeurs de qualité optimale. Elles ont principalement pour objectif de rendre explicites les informations manquantes tout en préservant les enregistrements concernés. Leur présence continue d'être prise en compte lors de l'évaluation des règles de qualité associées à la crédibilité des données.

Afin d'assurer la traçabilité de ces remplacements, des colonnes indicatrices supplémentaires ont également été générées pour identifier les valeurs initialement manquantes ou remplacées au cours du processus de nettoyage.

Le Tableau 5.11 présente plusieurs exemples issus du jeu de données NVD 2024 pour lesquels la valeur du champ CWE était absente avant nettoyage. Afin de préserver les enregistrements tout en rendant explicite l'absence d'information, les valeurs manquantes ont été remplacées par la valeur *CWE-UNKNOWN*.

TABLEAU 5.11 : Extrait du jeu de données NVD 2024 après standardisation des valeurs manquantes

CVE_ID	SourceIdentifier	Vendor_-Brut	Product_Brut	CWE
CVE-2024-22362	vultures@jpcert.or.jp	drupal	drupal	CWE-UNKNOWN
CVE-2024-23347	cve-assign@fb.com	facebook	meta_spark_studio	CWE-UNKNOWN
CVE-2024-22216	cve@mitre.org	microchip	maxview_storage_-manager	CWE-UNKNOWN
CVE-2024-0584	secalert@redhat.com	unknown_-vendor	unknown_product	CWE-UNKNOWN
CVE-2024-0395	patrick@puiterwijk.org	unknown_-vendor	unknown_product	CWE-UNKNOWN
CVE-2024-0333	chrome-cve-admin@google.com	google, fedo-raproject	chrome, fedora	CWE-UNKNOWN

Le pipeline applique ensuite plusieurs opérations de validation et de correction relatives aux formats et aux types de données. Les identifiants CVE sont conservés uniquement lorsqu'ils respectent la structure normalisée définie par le standard CVE (*CVE-YYYY-NNNN*+). Cette vérification permet d'éliminer les identifiants incomplets ou incorrects susceptibles d'introduire des erreurs dans les analyses ultérieures.

Les informations temporelles font également l'objet d'un traitement spécifique. Les dates sont d'abord converties dans un format temporel uniforme afin de garantir leur compatibilité avec les analyses chronologiques. Les enregistrements contenant une date de publication invalide sont supprimés, cette information étant indispensable pour l'analyse temporelle des vulnérabilités.

Lorsque la date de dernière modification est absente, celle-ci est remplacée par la date de publication. Cette stratégie permet de conserver les enregistrements concernés tout en garantissant la disponibilité d'une information temporelle cohérente pour les traitements ultérieurs. Elle ne vise toutefois pas à reconstituer la date réelle de modification, mais à fournir une valeur de substitution compatible avec les analyses réalisées dans cette recherche.

Certaines incohérences temporelles ont également été corrigées lorsque la date de dernière modification était antérieure à la date de publication. Dans ces situations, la date de modification est automatiquement réalignée sur la date de publication afin de rétablir une cohérence chronologique minimale conforme aux règles de qualité définies précédemment.

Les scores CVSS utilisés pour évaluer la sévérité des vulnérabilités sont ensuite convertis en valeurs numériques afin de permettre leur exploitation dans les traitements statistiques. Les valeurs non numériques ainsi que les scores situés en dehors de l'intervalle officiel défini par le standard CVSS, compris entre 0 et 10, sont supprimés. Une fois les scores validés, la catégorie de sévérité est reconstruite automatiquement à partir du score CVSS selon les seuils officiels associés aux niveaux *NONE*, *LOW*, *MEDIUM*, *HIGH* et *CRITICAL*. L'opération permet de garantir la cohérence entre les scores numériques et les catégories de sévérité utilisées dans le jeu de données.

Une attention particulière est également portée aux colonnes contenant plusieurs valeurs au sein d'une même cellule, notamment les fournisseurs, les produits et les critères CPE. Ces

colonnes sont d'abord séparées, puis nettoyées afin de supprimer les répétitions internes et les valeurs non exploitables. Les différentes valeurs sont ensuite normalisées, notamment par conversion en minuscules, afin de réduire les incohérences de représentation. Concernant les critères CPE, seuls les éléments conformes à une structure compatible avec le standard *CPE 2.3* sont conservés. Le nombre réel de critères CPE associés à chaque vulnérabilité est ensuite recalculé afin de garantir la cohérence entre les informations descriptives et les informations quantitatives du jeu de données.

Le pipeline applique également une contrainte d'actualité afin de préserver la cohérence temporelle des jeux de données annuels. Seules les vulnérabilités dont l'année de publication correspond à l'année du fichier analysé sont conservées. Cette étape permet d'éviter l'introduction d'enregistrements incohérents provenant d'autres périodes temporelles et de maintenir la cohérence des analyses réalisées sur chaque année étudiée.

Après les principales opérations de nettoyage, plusieurs attributs dérivés sont reconstruits automatiquement afin de faciliter les analyses ultérieures. Parmi ces attributs figurent notamment l'année de publication, le nombre réel de critères CPE associés à chaque vulnérabilité ainsi que la longueur textuelle des descriptions. Ces attributs permettent de soutenir certaines vérifications de qualité et de fournir des informations complémentaires utiles aux traitements automatisés.

Les descriptions dont la longueur est inférieure au seuil minimal défini précédemment sont supprimées afin d'éviter la conservation d'informations insuffisantes ou peu représentatives pour les analyses. Cette opération vise à garantir la présence d'un contenu descriptif minimalement exploitable dans les données conservées.

Une phase d'unicité est ensuite appliquée afin d'éliminer les doublons présents dans le jeu de données. Les doublons de *CVE_ID* sont supprimés afin de garantir qu'une même

vulnérabilité ne soit représentée qu'une seule fois dans les analyses. Les lignes entièrement dupliquées sont également éliminées. Une fois l'ensemble des opérations terminé, l'index du jeu de données est réinitialisé afin d'obtenir une structure cohérente et directement exploitable.

Afin de garantir la traçabilité complète du processus de nettoyage, un journal de traitement est généré automatiquement pendant l'exécution du pipeline. Pour chaque opération appliquée, ce journal enregistre notamment le nombre de lignes avant traitement, le nombre de lignes restantes après traitement ainsi que le nombre de lignes supprimées ou corrigées. Cette traçabilité permet de mesurer précisément l'impact des différentes opérations et de justifier les transformations réalisées tout au long du processus de nettoyage. Un extrait de ce journal est présenté dans le Tableau 5.12, lequel synthétise les principales opérations de nettoyage effectuées sur les données NVD 2024 et leurs effets sur le volume et la qualité des données.

TABLEAU 5.12 : Exemples de règles appliquées lors du nettoyage des données NVD

Année	Étape	Colonne	Règle appliquée	Avant	Après	Suppr.	Corr.
2021	CVSS manquants	*CVSS*	Suppression des scores CVSS non exploitables	23358	22490	868	0
2021	Fournisseurs normalisés	*Vendor_Brut*	Minuscules, espaces nettoyés et doublons supprimés	22490	22490	0	11778
2021	Produits normalisés	*Product_Brut*	Minuscules, espaces nettoyés et doublons supprimés	22490	22490	0	8255
2022	CWE invalides	*CWE*	Remplacement par `CWE-UNKNOWN`	26098	26098	0	4226
2022	Filtrage par année	*PublishedDate*	Conservation des publications de 2022	26098	19793	6305	0
2023	Critères CPE normalisés	*CPE_Criteria_-Brut*	CPE valides, minuscules et doublons supprimés	30029	30029	0	4120
2024	Fournisseurs manquants	*Vendor_Brut*	Remplacement par `unknown_vendor`	39065	39065	0	9241
2024	CVSS manquants	*CVSS*	Suppression des scores CVSS non exploitables	39065	37853	1212	0
2024	Produits normalisés	*Product_Brut*	Minuscules, espaces nettoyés et doublons supprimés	37853	37853	0	9475
2024	Filtrage par année	*PublishedDate*	Conservation des publications de 2024	37853	32361	5492	0

5.4.2 SECOND NETTOYAGE

Après l'application du premier pipeline de nettoyage, une nouvelle mesure de la qualité des données a été réalisée afin d'identifier les non-conformités restantes. Cette évaluation a montré que certaines anomalies persistaient, notamment la présence de valeurs de substitution telles que *CWE-UNKNOWN*, *unknown_vendor* et *unknown_product*, l'absence de critères CPE valides ainsi que certaines incohérences temporelles résiduelles.

Contrairement au premier nettoyage, qui visait à préserver un maximum d'enregistrements exploitables en standardisant certaines valeurs manquantes, ce second passage adopte une stratégie plus stricte. Il ne remet pas en cause la première phase de nettoyage, mais intervient après celle-ci afin de produire une version finale du jeu de données davantage conforme aux règles de qualité définies précédemment.

Les valeurs de substitution introduites lors du premier nettoyage ont permis de conserver les enregistrements tout en rendant explicite l'absence d'information. Toutefois, ces valeurs demeurent non optimales du point de vue de la crédibilité des données. Dans le second passage, les enregistrements contenant encore des valeurs telles que *CWE-UNKNOWN*, *unknown_vendor* ou *unknown_product* sont donc exclus lorsque les analyses nécessitent des informations pleinement spécifiques et exploitables.

De la même manière, les enregistrements ne contenant aucun critère CPE valide sont supprimés dans cette phase stricte. Les critères CPE étant utilisés pour identifier les plateformes, produits ou composants affectés, leur absence limite fortement l'exploitation des données dans les analyses portant sur les produits vulnérables. Cette suppression est donc cohérente avec les règles de complétude, de cohérence et d'unicité appliquées aux informations CPE.

Les incohérences temporelles restantes sont traitées selon les règles établies lors du premier nettoyage. Les dates de modification absentes sont remplacées par les dates de publication lorsque cela est possible, tandis que les dates de modification antérieures aux dates de publication sont réalignées sur ces dernières. Les enregistrements dont la date de publication demeure invalide ou ne correspond pas à l'année du fichier analysé sont supprimés, conformément aux règles d'actualité et de cohérence temporelle.

Ce second passage ne vise donc pas à maximiser la conservation des données, mais à construire un jeu de données final plus strict, destiné aux analyses nécessitant un niveau élevé de conformité aux règles de qualité. La distinction entre le premier nettoyage conservateur et le second nettoyage strict permet ainsi de concilier la préservation initiale des données avec l'exigence de fiabilité nécessaire aux analyses finales.

5.5 CONCLUSION

Ce chapitre a présenté la méthodologie adoptée pour mesurer et améliorer la qualité des données de vulnérabilités issues de la NVD. Dans un premier temps, les principes de métrologie logicielle ont permis de définir un cadre rigoureux pour transformer les dimensions théoriques de qualité des données en règles de mesure observables et applicables automatiquement. Les dimensions retenues dans cette recherche, à savoir la complétude, l'unicité, la cohérence, l'exactitude, la crédibilité et l'actualité, ont ainsi été associées à des règles explicites fondées sur la littérature, les standards CVE, CWE, CPE et CVSS, ainsi que sur les caractéristiques observées dans les données analysées.

Dans un deuxième temps, ces règles ont été appliquées aux données brutes afin de produire des indicateurs quantitatifs de conformité. Cette étape a permis d'identifier les principales anomalies présentes dans les données et de guider les opérations de nettoyage. Le premier

nettoyage a été conçu comme une approche conservatrice visant à corriger les incohérences les plus importantes tout en préservant un maximum d'informations exploitables. Une nouvelle évaluation de la qualité a ensuite permis d'identifier les non-conformités restantes.

Dans un troisième temps, un second nettoyage plus strict a été appliqué afin de produire un jeu de données final davantage conforme aux règles de qualité définies. Ce processus progressif permet de comparer plusieurs états du même jeu de données : les données brutes, les données après premier nettoyage et les données finales après nettoyage strict.

La qualité des données est abordée comme un processus itératif combinant la mesure, le nettoyage et la réévaluation. Cette méthodologie sert de base à l'analyse présentée dans le chapitre suivant, qui examine les résultats obtenus et discute de l'amélioration de la qualité des données de vulnérabilités exploitées dans un contexte DevSecOps.

CHAPITRE VI
PRÉSENTATION ET ANALYSE DES RÉSULTATS

Ce chapitre présente les résultats obtenus à partir du processus méthodologique de mesure, de nettoyage et de réévaluation de la qualité des données de vulnérabilités issues de la NVD, décrit dans le chapitre précédent. L'objectif est d'analyser l'évolution de la qualité des données en comparant trois états successifs du jeu de données : les données brutes, les données obtenues après le premier nettoyage, puis les données finales produites après le nettoyage strict.

Dans un premier temps, les résultats relatifs aux dimensions de qualité retenues sont présentés, à savoir la complétude, l'unicité, la cohérence, l'exactitude, la crédibilité et l'actualité. Cette analyse permet d'identifier les principales anomalies présentes dans les données initiales, d'en évaluer la fréquence et d'observer leur évolution au fil des années étudiées.

Dans un second temps, les résultats du processus de nettoyage sont analysés afin de mesurer l'impact des différentes étapes sur le volume, la structure et l'exploitabilité des données. Une attention particulière est portée aux enregistrements supprimés, corrigés ou standardisés, ainsi qu'aux variations observées entre les données brutes, les données nettoyées et les données finales.

Enfin, la réévaluation de la qualité après nettoyage permet d'apprécier l'effet des règles appliquées sur chaque dimension étudiée. Cette comparaison met en évidence les améliorations obtenues, mais aussi les limites persistantes liées à certaines colonnes, notamment les fournisseurs, les produits, les CWE et les critères CPE.

L'ensemble de ces résultats permet ainsi de vérifier dans quelle mesure les règles de qualité définies précédemment contribuent à améliorer la fiabilité, la cohérence et l'exploitabilité des données utilisées dans un contexte DevSecOps.

6.1 ANALYSE DES DONNÉES BRUTES

L'analyse commence par les résultats obtenus lors de l'évaluation initiale de la qualité des données brutes issues de la NVD. À ce stade, les données n'ont pas encore fait l'objet d'un nettoyage ou d'une correction. L'objectif est donc d'établir un état initial de la qualité des jeux de données afin d'identifier les principales anomalies présentes avant toute transformation.

L'analyse des données brutes permet de mesurer le niveau de conformité initial des données par rapport aux règles de qualité définies dans la méthodologie. Elle porte notamment sur les dimensions de complétude, d'unicité, de cohérence, d'exactitude, de crédibilité et d'actualité. Cette première évaluation constitue une base de comparaison essentielle pour mesurer ensuite l'effet des opérations de nettoyage et du passage strict sur la qualité globale des données.

Les résultats présentés dans cette section permettent ainsi de mettre en évidence les limites des données initiales, telles que la présence de valeurs manquantes, de valeurs non exploitables, d'incohérences de format, de doublons ou encore d'informations temporelles non conformes. Ces constats servent de point de départ à l'analyse des améliorations observées après l'application du processus de nettoyage des données de vulnérabilités de la NVD.

6.1.1 ANALYSE DES ANOMALIES OBSERVÉES DANS LES DONNÉES BRUTES

Le Tableau 6.1 présente les principales anomalies détectées dans les données brutes. Les résultats montrent que la dimension d'unicité constitue la source majeure de non-conformité.

Les anomalies les plus importantes concernent les colonnes *Vendor_Brut*, *Product_Brut* et *CPE_Criteria_Brut*, qui contiennent de nombreuses répétitions internes au sein d'une même cellule.

TABLEAU 6.1 : Principales anomalies observées dans les données brutes

Année	Dimension	Colonne	Règle	Description	Non conformes	Total	Conformité (%)
2025	Cohérence	*CPE_Criteria_-Brut*	COH_009	Structure CPE invalide	18593	44148	57.88
2025	Exactitude	*Vendor_Brut*	EXA_004	Fournisseur absent ou non exploitable	18584	44148	57.91
2025	Exactitude	*Product_Brut*	EXA_005	Produit absent ou non exploitable	18572	44148	57.93
2024	Unicité	*Vendor_Brut*	UNI_003	Répétitions internes de fournisseurs	21921	39065	43.89
2024	Unicité	*Product_Brut*	UNI_004	Répétitions internes de produits	18662	39065	52.23
2023	Unicité	*Vendor_Brut*	UNI_003	Répétitions internes de fournisseurs	15016	31170	51.83
2024	Unicité	*CPE_Criteria_-Brut*	UNI_002	Répétitions internes de critères CPE	14670	39065	62.45
2022	Unicité	*Vendor_Brut*	UNI_003	Répétitions internes de fournisseurs	14504	27470	47.20
2021	Unicité	*Vendor_Brut*	UNI_003	Répétitions internes de fournisseurs	12825	23358	45.09
2023	Unicité	*Product_Brut*	UNI_004	Répétitions internes de produits	10850	31170	65.19
2022	Unicité	*Product_Brut*	UNI_004	Répétitions internes de produits	10805	27470	60.67
2021	Unicité	*Product_Brut*	UNI_004	Répétitions internes de produits	9283	23358	60.26
2024	Cohérence	*CPE_Criteria_-Brut*	COH_009	Structure CPE invalide	9271	39065	76.27
2024	Exactitude	*Product_Brut*	EXA_005	Produit absent ou non exploitable	9242	39065	76.34
2024	Exactitude	*Vendor_Brut*	EXA_004	Fournisseur absent ou non exploitable	9242	39065	76.34
2024	Complétude	*Vendor_Brut*	COMP_002	Valeur vide ou non exploitable	9241	39065	76.34
2024	Complétude	*Vendor_Brut*	COMP_001	Valeur nulle	9241	39065	76.34

Entre 2021 et 2024, les anomalies d'unicité touchent fortement les colonnes *Vendor_Brut* et *Product_Brut*. Pour les fournisseurs, le nombre de valeurs non conformes passe de 12 825 en 2021 à 14 504 en 2022, puis à 15 016 en 2023, avant d'atteindre 21 921 en 2024. Cette progression montre que les répétitions internes de fournisseurs deviennent plus importantes

avec l'augmentation du volume de données. Les produits suivent une tendance similaire, avec 9 283 valeurs non conformes en 2021, 10 805 en 2022, 10 850 en 2023 et 18 662 en 2024.

L'année 2024 se distingue donc par une forte concentration d'anomalies d'unicité. La colonne *Vendor_Brut* présente un taux de conformité de seulement 43,89 %, tandis que *Product_Brut* atteint 52,23 %. Ces doublons internes peuvent fausser les analyses statistiques, car un même fournisseur ou un même produit peut être comptabilisé plusieurs fois pour une seule vulnérabilité.

En 2025, les anomalies les plus visibles se déplacent davantage vers la cohérence et l'exactitude des données. La colonne *CPE_Criteria_Brut* présente 18 593 valeurs non conformes pour la règle COH_009, soit un taux de conformité de 57,88 %. Les colonnes *Vendor_Brut* et *Product_Brut* présentent également un nombre élevé d'anomalies d'exactitude, avec respectivement 18 584 et 18 572 valeurs non conformes. Cela indique qu'une part importante des fournisseurs, produits ou critères CPE reste absente, générique, mal structurée ou insuffisamment exploitable.

Les critères CPE constituent donc une autre source importante de non-conformité. En 2024, 14 670 valeurs non conformes sont observées pour l'unicité des critères CPE, tandis que 9 271 valeurs ne respectent pas la structure attendue. En 2025, ce problème de structure devient encore plus marqué, avec 18 593 valeurs non conformes. Ces anomalies réduisent la capacité des données brutes à identifier correctement les plateformes, produits ou composants affectés.

Ainsi, sur l'ensemble des années étudiées, les anomalies les plus significatives ne concernent pas uniquement l'absence d'information. Elles touchent aussi la structure interne des valeurs, les répétitions au sein des cellules et la qualité descriptive des fournisseurs, produits et critères CPE. Ces résultats montrent que les données brutes ne sont pas directement

exploitables et qu'elles nécessitent une étape préalable de préparation et de contrôle de la qualité avant toute analyse.

Après l'identification des anomalies les plus fréquentes, l'analyse suivante examine les taux globaux de conformité afin d'obtenir une vision synthétique de la qualité des données brutes selon les années et les dimensions évaluées.

6.1.2 ANALYSE GLOBALE DES TAUX DE CONFORMITÉ

Le Tableau 6.2 présente les taux globaux de conformité obtenus pour chaque dimension de qualité entre 2021 et 2025. Ces résultats montrent que certaines dimensions conservent des niveaux de conformité élevés, notamment la crédibilité, l'actualité et, dans une moindre mesure, la cohérence. Toutefois, ces taux doivent être interprétés avec prudence, car ils peuvent coexister avec un volume important de valeurs non conformes, particulièrement lorsque le nombre total de vulnérabilités augmente.

La crédibilité atteint 100 % pour toutes les années dans les données brutes, ce qui signifie qu'aucune valeur générique ciblée par les règles de crédibilité n'a été détectée à ce stade. L'actualité suit également une progression régulière, passant de 90,13 % en 2021 à 96,38 % en 2025. Cette évolution indique une meilleure correspondance entre l'année de publication des vulnérabilités et l'année du fichier analysé, même si certaines anomalies temporelles persistent.

La cohérence demeure globalement élevée entre 2021 et 2024, avec des taux compris entre 96,17 % et 97,13 %. Toutefois, une baisse plus marquée apparaît en 2025, où le taux descend à 93,06 %. Cette diminution s'explique notamment par l'augmentation des valeurs non conformes liées à la structure des critères CPE, comme observé précédemment dans la colonne *CPE_Criteria_Brut*.

La complétude et l'exactitude présentent une dégradation progressive plus visible à partir de 2024. La complétude passe de 98,24 % en 2021 à 94,32 % en 2024, puis à 89,20 % en 2025. De même, l'exactitude diminue de 97,66 % en 2021 à 92,35 % en 2024, puis à 85,26 % en 2025. Cette évolution confirme la présence croissante de valeurs absentes, non exploitables ou incorrectes dans les données brutes, notamment pour les colonnes liées aux fournisseurs, aux produits et aux critères CPE.

L'unicité demeure la dimension la plus problématique sur l'ensemble de la période. Son taux de conformité est de 71,29 % en 2021 et 2022, puis de 74,24 % en 2023, avant de diminuer à 64,64 % en 2024 et à 54,73 % en 2025. Cette baisse confirme les résultats précédents : les répétitions internes dans les colonnes *Vendor_Brut*, *Product_Brut* et *CPE_Criteria_Brut* constituent une faiblesse importante des données brutes.

Ainsi, l'analyse globale par année montre que les données brutes présentent une qualité initiale acceptable pour certaines dimensions, mais que cette qualité se dégrade lorsque le volume de données augmente. L'année 2025 met particulièrement en évidence les limites des données brutes, avec une baisse notable de la complétude, de l'exactitude, de la cohérence et surtout de l'unicité. Ces résultats confirment la nécessité d'un processus de nettoyage afin d'améliorer la fiabilité, la structure et l'exploitabilité des données avant leur utilisation dans un contexte DevSecOps.

TABLEAU 6.2 : Taux globaux de conformité par année et par caractéristique dans les données brutes

Année	Caractéristique	Valeurs conformes	Valeurs non conformes	Total vérifié	Taux global (%)
2021	Actualité	63155	6919	70074	90.13
2021	Cohérence	226552	7028	233580	96.99
2021	Complétude	642524	11500	654024	98.24
2021	Crédibilité	70074	0	70074	100.00
2021	Exactitude	159672	3834	163506	97.66
2021	Unicité	66611	26821	93432	71.29
2022	Actualité	75076	7334	82410	91.10
2022	Cohérence	264807	9893	274700	96.40
2022	Complétude	751206	17954	769160	97.67
2022	Crédibilité	82410	0	82410	100.00
2022	Exactitude	186521	5769	192290	97.00
2022	Unicité	78330	31550	109880	71.29
2023	Actualité	86026	7484	93510	92.00
2023	Cohérence	302764	8936	311700	97.13
2023	Complétude	852176	20584	872760	97.64
2023	Crédibilité	93510	0	93510	100.00
2023	Exactitude	211601	6589	218190	96.98
2023	Unicité	92558	32122	124680	74.24
2024	Actualité	111249	5946	117195	94.93
2024	Cohérence	375701	14949	390650	96.17
2024	Complétude	1031696	62124	1093820	94.32
2024	Crédibilité	117195	0	117195	100.00
2024	Exactitude	252547	20908	273455	92.35
2024	Unicité	101007	55253	156260	64.64
2025	Actualité	127651	4793	132444	96.38
2025	Cohérence	410821	30659	441480	93.06
2025	Complétude	1102678	133466	1236144	89.20
2025	Crédibilité	132444	0	132444	100.00
2025	Exactitude	263476	45560	309036	85.26
2025	Unicité	96650	79942	176592	54.73

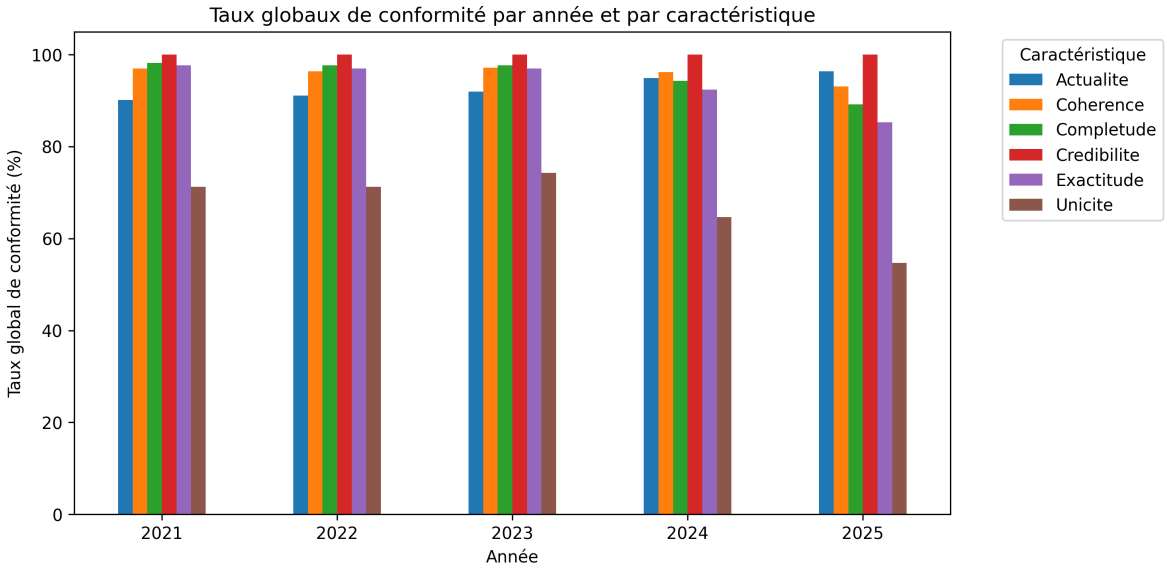

FIGURE 6.1 : Taux globaux de conformité par année et par caractéristique dans les données brutes

La Figure 6.1 confirme visuellement les tendances observées dans le tableau. Les dimensions de crédibilité, de cohérence, de complétude et d'exactitude présentent globalement des taux élevés, tandis que l'unicité reste nettement inférieure aux autres dimensions.

En conclusion, l'analyse des taux globaux de conformité met en évidence une qualité variable des données brutes selon les dimensions étudiées. Certaines dimensions, comme la crédibilité et l'actualité, présentent des résultats globalement élevés sur l'ensemble de la période 2021–2025. Cependant, d'autres dimensions montrent des limites importantes, notamment l'unicité, la complétude et l'exactitude.

L'année 2025 confirme cette tendance avec une baisse marquée des taux de conformité pour plusieurs dimensions. Les résultats montrent que l'augmentation du volume de données s'accompagne d'un accroissement des valeurs non conformes, en particulier dans les colonnes liées aux fournisseurs, aux produits et aux critères CPE. Ces constats justifient la nécessité

d'appliquer des étapes de nettoyage et de réévaluation afin d'améliorer la fiabilité, la cohérence et l'exploitabilité des données NVD.

6.2 PREMIER NETTOYAGE

Après l'évaluation initiale des données brutes, une première étape du processus de nettoyage a été appliquée afin de corriger ou de standardiser les principales anomalies observées. Cette étape adopte une approche conservatrice : elle vise à améliorer la qualité des données tout en conservant un maximum d'enregistrements exploitables. Certaines valeurs manquantes ou non informatives sont donc remplacées par des valeurs standardisées, tandis que les formats, les types de données, les incohérences temporelles et les doublons internes sont traités lorsque cela est possible.

L'objectif de cette section est d'analyser les résultats obtenus après ce premier nettoyage et de les comparer à l'état initial des données brutes. L'analyse porte sur l'évolution des valeurs conformes, des valeurs non conformes et des taux globaux de conformité pour les différentes dimensions de qualité retenues.

6.2.1 ANALYSE DES RÉSULTATS DU PREMIER NETTOYAGE

Après ce premier nettoyage, une nouvelle évaluation globale de la qualité des données a été réalisée. Les résultats montrent une amélioration notable de la majorité des dimensions de qualité par rapport aux données brutes. La complétude, l'exactitude, l'actualité, la cohérence et l'unicité atteignent des taux de conformité élevés, en particulier pour les années 2021, 2022 et 2023. Cette amélioration s'explique par les opérations appliquées durant le nettoyage, notamment le traitement des valeurs manquantes, la validation des formats, la normalisation

des dates, la correction des scores CVSS et la suppression des répétitions internes dans les listes de fournisseurs, de produits et de critères CPE.

Les résultats doivent toutefois être interprétés en tenant compte du fait que les données brutes et les données nettoyées ne correspondent pas exactement au même état du jeu de données. Après nettoyage, certaines lignes ont été supprimées, certaines valeurs ont été remplacées ou standardisées, et certains attributs ont été reconstruits. Les valeurs conformes, non conformes et les totaux vérifiés peuvent donc varier entre les deux évaluations.

La crédibilité présente une évolution particulière après le premier nettoyage. Dans les données brutes, cette dimension atteignait 100 %. Après le premier nettoyage, son taux diminue pour atteindre 94,44 % en 2021, 94,85 % en 2022, 96,11 % en 2023, 84,25 % en 2024 et 75,14 % en 2025. Cette baisse s'explique par la logique conservatrice du premier nettoyage. Au lieu de supprimer les enregistrements dont certaines informations n'étaient pas disponibles ou n'étaient pas identifiées avec précision, ces informations ont été remplacées par des valeurs standardisées, notamment *unknown_vendor*, *unknown_product*, *CWE-UNKNOWN* .

Ces valeurs standardisées rendent explicite l'absence d'information et permettent de conserver les enregistrements concernés dans le jeu de données. Cependant, elles restent peu informatives pour l'analyse des vulnérabilités. Elles sont donc considérées comme non conformes par les règles de crédibilité CRE_001, CRE_002 et CRE_003, qui visent respectivement les valeurs génériques associées aux fournisseurs, aux produits et aux faiblesses CWE. Il s'agit donc d'un compromis entre la complétude et la crédibilité : les enregistrements sont conservés afin de ne pas réduire la couverture du jeu de données, mais les valeurs trop génériques sont signalées comme moins fiables pour l'analyse.

TABLEAU 6.3 : Taux globaux de conformité après le premier nettoyage des données

Année	Caractéristique	Valeurs conformes	Valeurs non conformes	Total vérifié	Taux global (%)
2021	Actualité	48413	196	48609	99.60
2021	Cohérence	161819	211	162030	99.87
2021	Complétude	583304	4	583308	100.00
2021	Crédibilité	45904	2705	48609	94.44
2021	Exactitude	129621	3	129624	100.00
2021	Unicité	64810	2	64812	100.00
2022	Actualité	59093	286	59379	99.52
2022	Cohérence	197602	328	197930	99.83
2022	Complétude	712544	4	712548	100.00
2022	Crédibilité	56322	3057	59379	94.85
2022	Exactitude	158334	10	158344	99.99
2022	Unicité	79171	1	79172	100.00
2023	Actualité	69677	328	70005	99.53
2023	Cohérence	232986	364	233350	99.84
2023	Complétude	840032	28	840060	100.00
2023	Crédibilité	67285	2720	70005	96.11
2023	Exactitude	186662	18	186680	99.99
2023	Unicité	93328	12	93340	99.99
2024	Actualité	96601	482	97083	99.50
2024	Cohérence	316605	7005	323610	97.84
2024	Complétude	1151987	13009	1164996	98.88
2024	Crédibilité	81797	15286	97083	84.25
2024	Exactitude	258875	13	258888	99.99
2024	Unicité	122943	6501	129444	94.98
2025	Actualité	108196	464	108660	99.57
2025	Cohérence	348762	13438	362200	96.29
2025	Complétude	1277993	25927	1303920	98.01
2025	Crédibilité	81646	27014	108660	75.14
2025	Exactitude	289733	27	289760	99.99
2025	Unicité	131917	12963	144880	91.05

Le Tableau 6.3 confirme l'amélioration importante obtenue après le premier nettoyage. La complétude atteint 100 % pour les années 2021, 2022 et 2023, puis 98,88 % en 2024 et 98,01 % en 2025. L'exactitude est presque totalement conforme sur l'ensemble de la période,

avec 100 % en 2021 et 99,99 % de 2022 à 2025. L'actualité reste également supérieure à 99 % pour toutes les années, avec 99,57 % en 2025.

L'unicité, qui constituait la principale faiblesse des données brutes, progresse nettement après nettoyage. Elle atteint 100 % en 2021 et 2022, 99,99 % en 2023, puis diminue à 94,98 % en 2024 et 91,05 % en 2025. Cette amélioration montre l'efficacité des opérations appliquées pour réduire les répétitions internes dans les listes de fournisseurs, de produits et de critères CPE. Toutefois, les années 2024 et 2025 conservent encore des valeurs non conformes pour cette dimension, respectivement 6 501 et 12 963, ce qui indique que certaines répétitions ou structures complexes persistent malgré le premier nettoyage.

La cohérence présente également des taux élevés, mais elle diminue pour les années les plus récentes. Elle atteint 99,87 % en 2021, 99,83 % en 2022 et 99,84 % en 2023, puis descend à 97,84 % en 2024 et 96,29 % en 2025. Cette baisse semble principalement associée à la complexité croissante des champs CPE et à la persistance de certaines incohérences dans les informations structurées. Enfin, la crédibilité reste la dimension la plus limitée après nettoyage, avec une baisse marquée en 2024 et surtout en 2025. Cette évolution montre que le premier nettoyage améliore fortement la structure des données, mais fait aussi apparaître explicitement des valeurs génériques qui restent peu exploitables pour l'analyse.

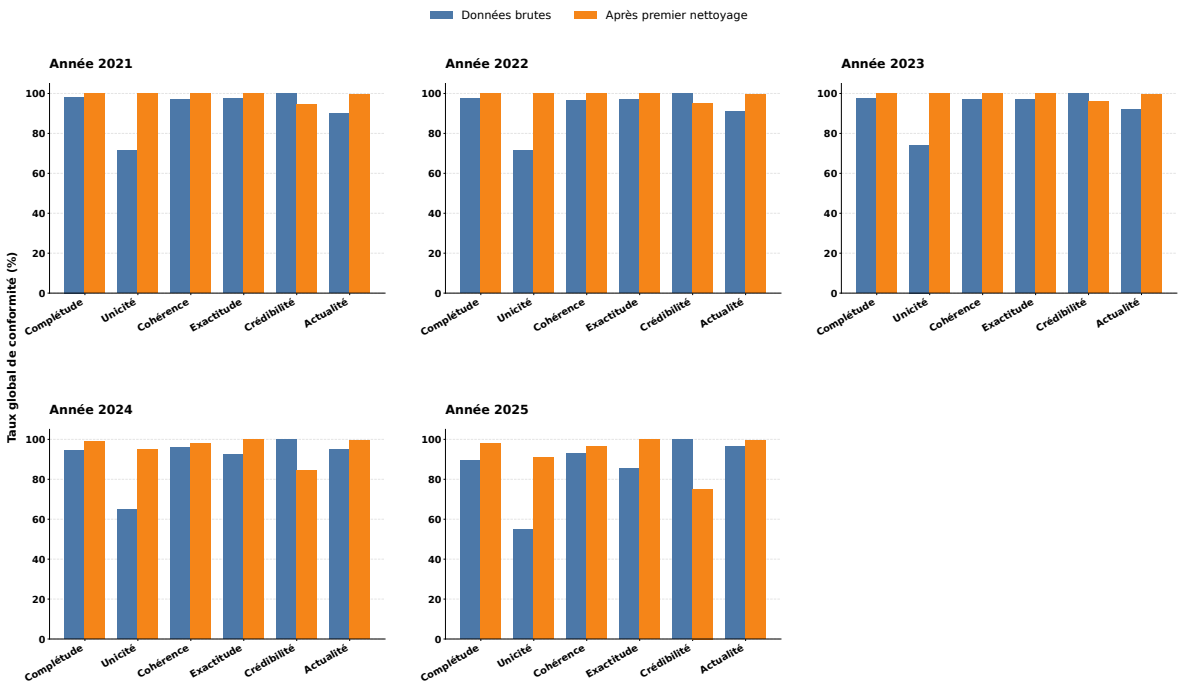

FIGURE 6.2 : Comparaison annuelle des taux globaux de conformité avant et après le premier nettoyage

La Figure 6.2 illustre ces évolutions en comparant, pour chaque année, les taux de conformité avant et après le premier nettoyage. Elle met en évidence l'amélioration globale de plusieurs dimensions, notamment la complétude, la cohérence, l'exactitude et l'unicité, tout en confirmant la baisse de la crédibilité, particulièrement marquée en 2024 et en 2025. Cette baisse s'explique par la standardisation de certaines valeurs manquantes sous forme de valeurs génériques peu informatives, ce qui justifie l'application d'un second nettoyage plus strict.

6.2.2 ANALYSE DES ANOMALIES RÉSIDUELLES AVANT LE SECOND NETTOYAGE : EXEMPLE ANNÉE 2024

Avant l'application du second nettoyage ciblé, une analyse des anomalies résiduelles a été réalisée sur les données nettoyées de l'année 2024. Cette année a été retenue comme

exemple, car elle présente encore plusieurs non-conformités après le premier nettoyage, malgré l'amélioration globale observée précédemment.

Les résultats indiquent que les anomalies restantes sont principalement concentrées dans la colonne *CPE_Criteria_Brut*. Cette colonne présente 6 523 valeurs non conformes pour la règle de cohérence COH_009, qui vérifie que chaque CPE respecte une structure valide. Elle présente également 6 501 valeurs non conformes pour les règles de complétude COMP_001 et COMP_002, ainsi que pour la règle d'unicité UNI_002. Ces résultats indiquent que certains enregistrements ne disposent toujours pas de critères CPE valides, ou contiennent des valeurs CPE absentes, vides ou répétées.

Les anomalies de crédibilité restent également importantes. Les colonnes *Vendor_Brut* et *Product_Brut* présentent chacune 6 500 valeurs non conformes, avec un taux de conformité de 79,91 %. Ces non-conformités correspondent à la présence de valeurs standardisées telles que *unknown_vendor* et *unknown_product*. Ces valeurs ont été introduites ou conservées lors du premier nettoyage afin de préserver les enregistrements concernés, mais elles restent peu informatives pour les analyses nécessitant l'identification précise des fournisseurs et des produits affectés.

La colonne *CWE* présente également 2 286 valeurs non conformes à la règle de crédibilité CRE_003, avec un taux de conformité de 92,94 %. Ces anomalies correspondent à la présence de la valeur *CWE-UNKNOWN*, qui indique que la faiblesse associée à la vulnérabilité n'a pas pu être identifiée précisément. Même si cette valeur permet de rendre explicite l'absence d'information, elle limite l'exploitation des données pour les analyses portant sur les catégories de faiblesses.

Des anomalies temporelles persistent également, mais dans des proportions beaucoup plus faibles. Les règles portant sur la relation entre *PublishedDate* et *LastModifiedDate*

présentent 241 valeurs non conformes, tandis que les règles relatives au format ou à la présence des dates présentent entre 112 et 129 valeurs non conformes. Ces résultats montrent que la majorité des incohérences temporelles ont été corrigées par le premier nettoyage, mais que quelques cas résiduels subsistent.

Enfin, les anomalies d'exactitude et de complétude restantes sont très limitées. La colonne *Product_Brut* présente 12 valeurs non conformes pour l'exactitude, la colonne *Severity* présente 7 valeurs non conformes pour la complétude, et la colonne *Vendor_Brut* ne présente qu'une seule valeur non conforme pour l'exactitude. Ces faibles volumes indiquent que le premier nettoyage a été efficace pour traiter la majorité des anomalies simples.

Ces résultats indiquent que le premier nettoyage a permis de corriger une grande partie des anomalies présentes dans les données brutes, mais que certaines incohérences critiques demeuraient encore dans les jeux de données. Les anomalies restantes concernent principalement des informations essentielles pour les analyses DevSecOps, notamment les critères CPE, les classifications CWE, les fournisseurs, les produits ou certaines descriptions insuffisamment exploitables. Cette situation a donc justifié la mise en place d'un second nettoyage appliquant des règles beaucoup plus strictes afin de renforcer la conformité finale des données.

Les anomalies résiduelles observées après le premier nettoyage pour l'année 2024 sont synthétisées dans le Tableau 6.4.

TABLEAU 6.4 : Anomalies résiduelles observées après le premier nettoyage pour l'année 2024

Année	Dimension	Colonne	Règle	Description	Non conformes	Total_Verifié	Conformité (%)
2024	Cohérence	*CPE_Criteria_Brut*	COH_009	Structure CPE invalide	6523	32361	79.84
2024	Complétude	*CPE_Criteria_Brut*	COMP_001	Valeur CPE nulle	6501	32361	79.91
2024	Unicité	*CPE_Criteria_Brut*	UNI_002	Répétitions internes de critères CPE	6501	32361	79.91
2024	Complétude	*CPE_Criteria_Brut*	COMP_002	Valeur CPE vide ou non exploitable	6501	32361	79.91
2024	Crédibilité	*Vendor_Brut*	CRE_001	Valeur *unknown_vendor*	6500	32361	79.91
2024	Crédibilité	*Product_Brut*	CRE_002	Valeur *unknown_product*	6500	32361	79.91
2024	Crédibilité	*CWE*	CRE_003	Valeur *CWE-UNKNOWN*	2286	32361	92.94
2024	Cohérence	*PublishedDate / LastModifiedDate*	COH_005	Date de modification antérieure à la publication	241	32361	99.26
2024	Actualité	*PublishedDate / LastModifiedDate*	ACT_003	Délai temporel non conforme	241	32361	99.26
2024	Actualité	*PublishedDate*	ACT_001	Année de publication non conforme	129	32361	99.60
2024	Cohérence	*PublishedDate*	COH_003	Format de publication non valide	129	32361	99.60
2024	Cohérence	*LastModifiedDate*	COH_004	Format de modification non valide	112	32361	99.65
2024	Actualité	*LastModifiedDate*	ACT_002	Date de modification absente	112	32361	99.65
2024	Exactitude	*Product_Brut*	EXA_005	Produit non exploitable	12	32361	99.96
2024	Complétude	*Severity*	COMP_002	Sévérité vide ou non exploitable	7	32361	99.98
2024	Exactitude	*Vendor_Brut*	EXA_004	Fournisseur non exploitable	1	32361	100.00

6.3 SECOND NETTOYAGE

6.3.1 ANALYSE GLOBALE DES RÉSULTATS APRÈS LE SECOND NETTOYAGE

Après l'application du second nettoyage, une nouvelle mesure de la qualité des données a été réalisée sur les jeux de données finaux. Cette mesure a été effectuée à partir des mêmes règles de qualité que celles utilisées pour l'évaluation des données brutes et des données issues du premier nettoyage. L'objectif est de vérifier dans quelle mesure cette seconde phase a permis de réduire les anomalies résiduelles observées après le premier nettoyage.

Contrairement au premier nettoyage, qui adoptait une approche conservatrice en remplaçant certaines valeurs manquantes par des valeurs standardisées, le second nettoyage ciblé applique une stratégie plus stricte. Les enregistrements contenant des valeurs peu informatives telles que *unknown_vendor*, *unknown_product* ou *CWE-UNKNOWN*, ainsi que les cas

présentant des critères CPE absents ou invalides, ont été supprimés. Cette étape vise ainsi à produire un jeu de données final plus fiable, plus spécifique et mieux adapté aux analyses de vulnérabilités et aux mécanismes de priorisation.

Les résultats obtenus après cette nouvelle mesure indiquent une amélioration très importante de la qualité globale des données pour l'ensemble des années étudiées, de 2021 à 2025. Toutes les dimensions atteignent des taux globaux de conformité très proches de 100 %, plusieurs d'entre elles étant affichées à 100 % après arrondi. Cela montre que les principales anomalies détectées dans les données brutes et après le premier nettoyage ont été éliminées ou réduites à un niveau marginal.

La complétude, l'actualité, la cohérence, la crédibilité, l'exactitude et l'unicité atteignent désormais un niveau de conformité maximal dans les jeux de données finaux. Cela indique que les valeurs manquantes, les valeurs génériques, les anomalies temporelles majeures, les incohérences de format et les répétitions internes ont été efficacement traitées par le second nettoyage ciblé.

Il convient toutefois de noter que certains taux affichés à 100 % résultent d'un arrondi. Par exemple, quelques valeurs non conformes subsistent encore pour la complétude en 2022, 2023 et 2024, avec respectivement 2, 4 et 5 valeurs non conformes. Ces écarts restent néanmoins négligeables au regard du volume total de valeurs vérifiées, qui atteint plusieurs centaines de milliers d'observations selon les années.

Les résultats de 2025 confirment également l'efficacité du second nettoyage. Après cette étape, les six dimensions évaluées atteignent un taux de conformité de 100 %, sans aucune valeur non conforme recensée dans le tableau final. Cette évolution est particulièrement importante, car l'année 2025 présentait auparavant des non-conformités élevées en crédibilité, en complétude, en exactitude et en unicité après le premier nettoyage.

Ainsi, le second nettoyage ciblé permet de produire un jeu de données final nettement plus fiable, plus cohérent et plus exploitable. Il réduit fortement les limites observées après le premier nettoyage, notamment la présence de valeurs génériques dans les colonnes relatives aux fournisseurs, aux produits et aux faiblesses CWE, ainsi que les problèmes liés aux critères CPE. Les résultats détaillés de cette mesure après le second nettoyage ciblé sont présentés dans le Tableau 6.5.

TABLEAU 6.5 : Taux globaux de conformité après le second nettoyage ciblé

Année	Caractéristique	Valeurs conformes	Valeurs non conformes	Total vérifié	Taux global (%)
2021	Actualité	40371	0	40371	100
2021	Cohérence	134570	0	134570	100
2021	Complétude	484452	0	484452	100
2021	Crédibilité	40371	0	40371	100
2021	Exactitude	107656	0	107656	100
2021	Unicité	53828	0	53828	100
2022	Actualité	49959	0	49959	100
2022	Cohérence	166530	0	166530	100
2022	Complétude	599506	2	599508	100
2022	Crédibilité	49959	0	49959	100
2022	Exactitude	133224	0	133224	100
2022	Unicité	66612	0	66612	100
2023	Actualité	61512	0	61512	100
2023	Cohérence	205040	0	205040	100
2023	Complétude	738140	4	738144	100
2023	Crédibilité	61512	0	61512	100
2023	Exactitude	164032	0	164032	100
2023	Unicité	82016	0	82016	100
2024	Actualité	70659	0	70659	100
2024	Cohérence	235530	0	235530	100
2024	Complétude	847903	5	847908	100
2024	Crédibilité	70659	0	70659	100
2024	Exactitude	188424	0	188424	100
2024	Unicité	94212	0	94212	100
2025	Actualité	66504	0	66504	100
2025	Cohérence	221680	0	221680	100
2025	Complétude	798048	0	798048	100
2025	Crédibilité	66504	0	66504	100
2025	Exactitude	177344	0	177344	100
2025	Unicité	88672	0	88672	100

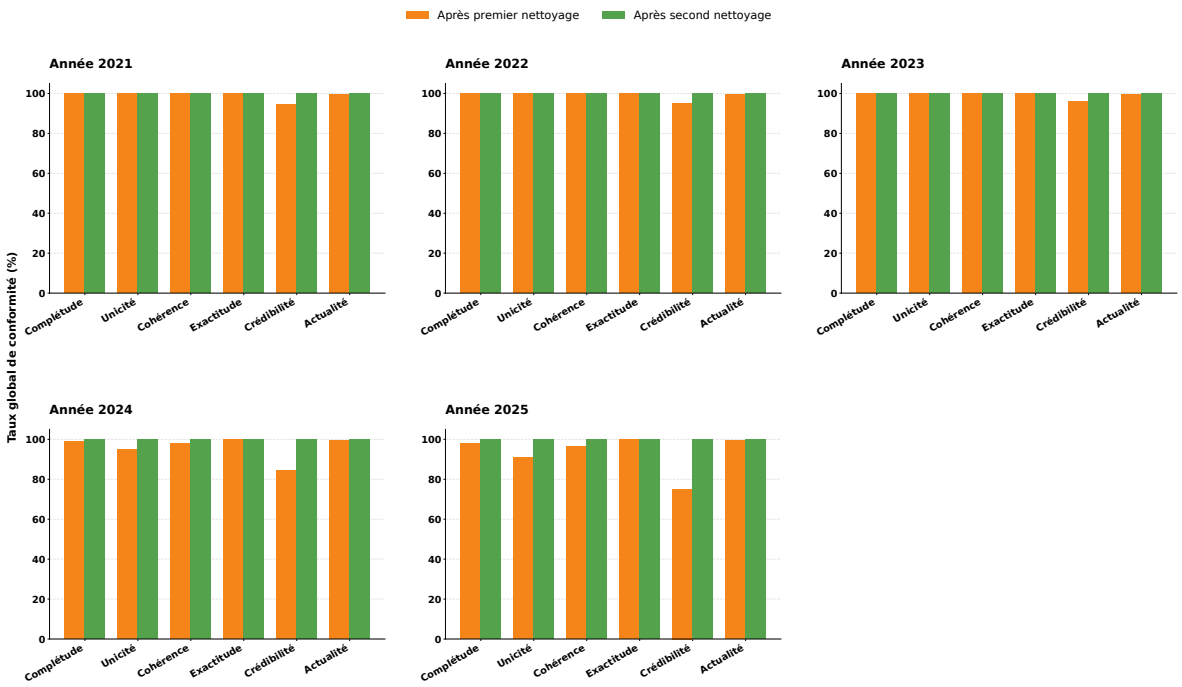

FIGURE 6.3 : Comparaison annuelle des taux globaux de conformité après le premier et le second nettoyage

La Figure 6.3 illustre visuellement les résultats présentés dans le tableau. Les taux de conformité sont quasiment tous égaux à 100 % pour l'ensemble des années et des caractéristiques étudiées. Les faibles écarts observés pour la cohérence et l'exactitude restent peu visibles graphiquement, car les taux demeurent très proches de la conformité totale.

Cette représentation met donc en évidence l'effet du second nettoyage : les anomalies résiduelles liées aux valeurs génériques, aux critères CPE invalides et aux répétitions internes ont été nettement réduites. Les jeux de données finaux obtenus après cette étape présentent ainsi un niveau de qualité nettement plus homogène que les données brutes et que les données issues du premier nettoyage.

6.4 CONCLUSION

Ce chapitre a présenté et analysé les résultats obtenus à partir de l'application du pipeline de mesure, de nettoyage et d'évaluation de la qualité des données de vulnérabilités issues de la NVD. L'analyse a d'abord permis d'établir un état initial des données brutes, puis d'observer l'effet progressif des deux phases de nettoyage sur les différentes dimensions de qualité retenues : la complétude, l'unicité, la cohérence, l'exactitude, la crédibilité et l'actualité.

L'évaluation des données brutes a montré que celles-ci présentaient une qualité hétérogène. Certaines dimensions affichaient des taux de conformité relativement élevés, mais plusieurs anomalies importantes ont été identifiées, notamment les répétitions internes dans les fournisseurs, les produits et les critères CPE, ainsi que la présence de valeurs manquantes, non exploitables ou temporellement incohérentes. Ces résultats ont confirmé que les données brutes ne pouvaient pas être utilisées directement sans traitement préalable.

Le premier nettoyage a permis d'améliorer fortement la qualité structurelle des données. Les taux de conformité ont augmenté pour la plupart des dimensions, en particulier pour la complétude, l'unicité, l'exactitude et l'actualité. Toutefois, cette première phase, volontairement conservatrice, a maintenu certaines valeurs de substitution telles que *unknown_vendor*, *unknown_product* et *CWE-UNKNOWN*. Ces valeurs ont amélioré la traçabilité des absences d'information, mais ont aussi mis en évidence des limites persistantes au niveau de la crédibilité des données.

L'analyse des anomalies résiduelles a ensuite justifié l'application d'un second nettoyage plus strict. Les résultats obtenus après cette seconde phase indiquent une amélioration nette de la qualité finale des jeux de données. Les dimensions de complétude, d'actualité, de crédibilité et d'unicité atteignent des niveaux de conformité proches ou égaux à 100 %, tandis que les rares anomalies restantes en cohérence et en exactitude demeurent marginales.

CHAPITRE VII
DISCUSSION GÉNÉRALE

7.1 LIMITES DE L'ÉTUDE

Cette étude présente certaines limites. Premièrement, l'expérimentation repose principalement sur les données de vulnérabilités issues de la NVD entre 2021 et 2025. Même si cette base est largement utilisée dans les outils et les recherches en cybersécurité, les résultats obtenus ne peuvent pas être généralisés automatiquement à toutes les bases de vulnérabilités ni à tous les contextes DevSecOps.

Toutes les caractéristiques du modèle consolidé n'ont pas été mesurées directement. Le noyau opérationnel appliqué dans cette recherche se concentre sur la complétude, l'unicité, la cohérence, l'exactitude, la crédibilité et l'actualité. D'autres dimensions comme la représentativité, la diversité, l'équilibre des classes, la traçabilité complète, l'exploitabilité, la disponibilité ou la conformité opérationnelle ont été intégrées au modèle conceptuel, mais n'ont pas toutes fait l'objet d'une mesure indépendante.

Le nettoyage strict améliore les taux de conformité, mais il réduit aussi le volume final de données. Cette situation met en évidence un compromis entre la conservation maximale des enregistrements et l'obtention d'un jeu de données plus conforme aux règles de qualité. Dans certains cas, la suppression d'enregistrements peut réduire la couverture de certaines vulnérabilités rares ou modifier la distribution des données utilisées dans les analyses prédictives.

7.2 IMPLICATIONS POUR LES ENVIRONNEMENTS DEVSECOPS

Les résultats de cette recherche ont plusieurs implications pour les environnements DevSecOps. Dans ces environnements, les mécanismes automatisés reposent sur des données continuellement produites, collectées et transformées par différents outils. Si ces données contiennent des erreurs, des valeurs manquantes, des doublons ou des incohérences, les analyses automatisées peuvent produire des résultats incomplets ou trompeurs.

L'intégration de mécanismes de mesure de la qualité des données dans les pipelines DevSecOps permettrait de détecter plus tôt les anomalies présentes dans les données utilisées pour l'analyse des vulnérabilités. Les règles de qualité peuvent être exécutées automatiquement avant l'utilisation des données dans des mécanismes de détection, de priorisation ou de prédiction. Cela permettrait de réduire les risques associés à l'utilisation de données incorrectes ou insuffisamment exploitables.

Les résultats montrent également que le nettoyage des données ne doit pas être considéré comme une opération unique, mais comme un processus progressif. Un premier nettoyage conservateur peut préserver un maximum d'informations tout en corrigeant les anomalies les plus visibles. Un second nettoyage plus strict peut ensuite être appliqué lorsque les analyses exigent un niveau plus élevé de conformité. Cette logique est particulièrement adaptée aux environnements DevSecOps, où les exigences peuvent varier selon les usages : surveillance continue, analyse exploratoire, priorisation ou prédiction.

7.3 PERSPECTIVES FUTURES

Ce travail ouvre plusieurs perspectives de recherche. Le modèle consolidé proposé pourrait d'abord être appliqué à d'autres types de données utilisées dans les environnements DevSecOps, au-delà des données de vulnérabilités issues de la NVD. Les rapports produits

par les scanners de sécurité, les journaux d'exécution, les dépôts de code source ou encore les systèmes de suivi des incidents pourraient permettre de tester la capacité du modèle à s'adapter à des données et pratiques opérationnelles plus diverses.

Une autre perspective concerne l'intégration progressive des règles de mesure et de nettoyage dans les pipelines DevSecOps. Cette intégration permettrait de suivre la qualité des données de manière continue, dès leur collecte et tout au long de leur utilisation par les mécanismes automatisés. Elle contribuerait ainsi à limiter la propagation des anomalies dans les processus d'analyse, de détection et de priorisation des vulnérabilités.

Il serait également pertinent d'élargir l'évaluation expérimentale à d'autres caractéristiques du modèle consolidé. Certaines dimensions, comme la représentativité, la diversité, l'équilibre des classes, la traçabilité, l'exploitabilité, la disponibilité et la conformité, n'ont pas été mesurées directement dans cette recherche.

CHAPITRE VIII
CONCLUSION GÉNÉRALE

Ce mémoire a proposé une démarche structurée pour évaluer et améliorer la qualité des données dans un environnement DevSecOps intégrant des systèmes d'IA, en s'appuyant sur l'inventaire des standards internationaux, la revue de la littérature scientifique sur les modèles de qualité des données pour l'IA, ainsi que sur la comparaison de ces deux sources ayant conduit à la conception d'un modèle consolidé.

La principale contribution de cette recherche réside dans la construction de ce modèle consolidé de qualité des données et dans la démonstration de son applicabilité. Son application aux données de vulnérabilités de la base NVD, à travers des règles de mesure et un processus de nettoyage progressif, a permis d'identifier plusieurs problèmes de qualité affectant les données brutes, puis d'améliorer effectivement la qualité de ces données et de renforcer leur exploitabilité dans un environnement DevSecOps. Ce cas d'application illustre ainsi que le modèle consolidé constitue un outil opérationnel, capable à la fois de diagnostiquer et de corriger des problèmes réels de qualité des données de vulnérabilités.

Cette recherche comporte certaines limites, notamment la mesure partielle des caractéristiques du modèle consolidé selon les champs disponibles, ainsi qu'un compromis observé entre l'amélioration de la qualité mesurée et la conservation du volume de données lors du nettoyage strict. Les perspectives futures pourraient porter sur la mesure de caractéristiques non évaluées directement dans cette recherche, ainsi que sur l'intégration de cette démarche dans un processus continu de suivi de la qualité des données en cybersécurité. En définitive, ce modèle consolidé apporte une contribution à l'amélioration de la qualité des données au sein des environnements DevSecOps intégrant des systèmes d'IA.

BIBLIOGRAPHIE

[1] Abiona, O., Oladapo, O., Modupe, O., Oyeniran, O., Adewusi, A. & Komolafe, A. (2024). The emergence and importance of DevSecOps : Integrating and reviewing security practices within the DevOps pipeline. *World Journal of Advanced Engineering Technology and Sciences*, pp. 127–133. doi: 10.30574/wjaets.2024.11.2.0093

[2] Abran, A. (2010). *Software Metrics and Software Metrology*. John Wiley & Sons. Google-Books-ID : rvF1nsgwa54C.

[3] Ahmed, Z. & Francis, S. C. (2019). Integrating Security with DevSecOps : Techniques and Challenges. doi: 10.1109/icd47981.2019.9105789

[4] Akbar, M. A., Smolander, K., Mahmood, S. & Alsanad, A. (2022). Toward Successful DevSecOps in Software Development Organizations : A Decision-Making Framework. *Information & Software Technology*. doi: 10.1016/j.infsof.2022.106894

[5] Aktuğ, S. S., Ozkan-Okay, M., Yilmaz, A. A. & Akin, E. (2023). A comprehensive review of cyber security vulnerabilities, threats, attacks, and solutions. *Electronics*, *12*(6), 1333.

[6] Alghawli, A. S. & Radivilova, T. (2024). Resilient Cloud cluster with DevSecOps security model, automates a data analysis, vulnerability search and risk calculation. *Alexandria Engineering Journal*. doi: 10.1016/j.aej.2024.07.036

[7] Anwar, A., Abusnaina, A., Chen, S., Li, F. & Mohaisen, D. (2022, 6). Cleaning the NVD : Comprehensive Quality Assessment, Improvements, and Analyses. *IEEE Transactions on Dependable and Secure Computing*, pp. 4255–4269. doi: 10.1109/TDSC.2021.3125270

[8] Bautista Villalpando, L. E., April, A. & Abran, A. (2014, 1). Performance analysis model for big data applications in cloud computing. *Journal of Cloud Computing*, p. 19. doi: 10.1186/s13677-014-0019-z

[9] Bhatia, A., Lin, D., Rajbahadur, G. K., Adams, B. & Hassan, A. E. (2024, août). *Data Quality Antipatterns for Software Analytics*. arXiv :2408.12560 [cs], doi: 10.48550/arXiv.2408.12560

[10] C. Guerra-García, Anastasija Nikiforova, S. Jiménez, H. G. Pérez-González, M. Ramírez-Torres & Luis Ontañon-García (2023). ISO/IEC 25012-based methodology for managing data quality requirements in the development of information systems : Towards Data Quality by Design. *Data & Knowledge Engineering*. S2ID : 14f08eacca7c72f575dd24b923b90e8dbf6e0a42.

[11] Chen, H., Chen, J. & Ding, J. (2021, 2). Data Evaluation and Enhancement for Quality Improvement of Machine Learning. *IEEE Transactions on Reliability*, pp. 831–847. doi: 10.1109/TR.2021.3070863

[12] Chen, T. & Suo, H. (2022). Design and Practice of Security Architecture via DevSecOps Technology. *2022 IEEE 13th International Conference on Software Engineering and Service Science (ICSESS)*. doi: 10.1109/icsess54813.2022.9930212

[13] Chittala, S. (2024). Securing DevOps Pipelines : Automating Security in DevSecOps Frameworks. *JOURNAL OF RECENT TRENDS IN COMPUTER SCIENCE AND ENGINEERING*, pp. 31–44. doi: 10.70589/JRTCSE.2024.5.5

[14] Croft, R., Babar, M. A. & Kholoosi, M. M. (2023, mai). Data Quality for Software Vulnerability Datasets. Dans *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*, pp. 121–133. ISSN : 1558-1225, doi: 10.1109/ICSE48619.2023.00022

[15] CVE DETAIL (2026, mai). *Browse CVE vulnerabilities by date*. Repéré le 2026-06-20, à https://www.cvedetails.com/browse-by-date.php

[16] D. Zha, Zaid Pervaiz Bhat, Kwei-Herng Lai, Fan Yang, Zhimeng Jiang, Shaochen Zhong & Xia Hu (2023). Data-centric Artificial Intelligence : A Survey. *ACM Computing Surveys*. ARXIV_ID : 2303.10158 S2ID : 12c6be503e4e5b7c9cb1810152d4364f26628a8d, doi: 10.1145/3711118

[17] Deng, X., Duan, F., Xie, R., Ye, W. & Zhang, S.-B. (2024). Improving Long-Tail Vulnerability Detection Through Data Augmentation Based on Large Language Models. *IEEE International Conference on Software Maintenance and Evolution*. doi: 10.1109/icsme58944.2024.00033

[18] Feio, C., Santos, N., Escravana, N. & Pacheco, B. (2024). An Empirical Study of DevSecOps Focused on Continuous Security Testing. *2024 IEEE European Symposium on*

Security and Privacy Workshops (EuroS&PW). doi: 10.1109/eurospw61312.2024.00074

[19] Gauthier, F., Keynes, N., Allen, N., Corney, D. & Krishnan, P. (2018). Scalable Static Analysis to Detect Security Vulnerabilities : Challenges and Solutions. doi: 10.1109/sec-dev.2018.00030

[20] Grigorieva, N. M., Petrenko, A. S. & Petrenko, S. A. (2024). Development of Secure Software Based on the New Devsecops Technology. *2024 Conference of Young Researchers in Electrical and Electronic Engineering (ElCon)*. doi: 10.1109/elcon61730.2024.10468425

[21] Guo, Y., Bettaieb, S. & Casino, F. (2024, 5). A comprehensive analysis on software vulnerability detection datasets : trends, challenges, and road ahead. *International Journal of Information Security*, pp. 3311–3327. Company : Springer Distributor : Springer Institution : Springer Label : Springer Number : 5, doi: 10.1007/s10207-024-00888-y

[22] Hrusto, A., Engström, E. & Runeson, P. (2023). Towards optimization of anomaly detection in DevOps. *Information and Software Technology*, p. 107241. doi: 10.1016/j.infsof.2023.107241

[23] J. Oviedo, Moisés Rodríguez, Andrea Trenta, Dino Cannas, Domenico Natale & Mario Piattini (2024). ISO/IEC quality standards for AI engineering. *Computer Science Review*. S2ID : a6613ec72969203f6861d119451b096f104f8651, doi: 10.1016/j.cosrev.2024.100681

[24] Javier Verdugo & Moisés Rodríguez (2019). Assessing Data Cybersecurity Using ISO/IEC 25012. *Software quality journal*, pp. 33–46. MAG ID : 2971207148 S2ID : 3dd976714816ba1a2cf44fee1ac675693c505011, doi: 10.1007/978-3-030-29238-6_3

[25] Leite, L., Rocha, C., Kon, F., Milojicic, D. & Meirelles, P. (2020, 6). A Survey of DevOps Concepts and Challenges. *ACM Computing Surveys*, pp. 1–35. doi: 10.1145/3359981

[26] Lombardi, F. & Fanton, A. (2023). From DevOps to DevSecOps is not enough. CyberDevOps : an extreme shifting-left architecture to bring cybersecurity within software security lifecycle pipeline. *Software quality journal*. doi: 10.1007/s11219-023-09619-3

[27] Marandi, M., Bertia, A. & Silas, S. (2023). Implementing and Automating Security Scanning to a DevSecOps CI/CD Pipeline. *null*. doi: 10.1109/wconf58270.2023.10235015

[28] Mohammad Hossein Jarrahi, Ali Memariani & Min Kyung Lee (2023, 8). The Principles of Data-Centric AI. *Communications of the ACM*, pp. 84–92. ARXIV_ID : 2211.14611 MAG ID : 4385248297 S2ID : e109d1b64b69e4800cd4e0ff14103bbd63fb164c, doi: 10.1145/3571724

[29] Mohammed, S., Budach, L., Feuerpfeil, M., Ihde, N., Nathansen, A., Noack, N., Patzlaff, H., Naumann, F. & Harmouch, H. (2025). The effects of data quality on machine learning performance on tabular data. *Information Systems*, p. 102549. doi: 10.1016/j.is.2025.102549

[30] Pattanayak, S., Murthy, P. & Mehra, A. (2024). Integrating AI into DevOps pipelines : Continuous integration, continuous delivery, and automation in infrastructural management : Projections for future. *International Journal of Science and Research Archive*. doi: 10.30574/ijsra.2024.13.1.1838

[31] Qi, W., Cao, J., Poddar, D., Li, S. & Wang, X. (2024). Enhancing Pre-Trained Language Models for Vulnerability Detection via Semantic-Preserving Data Augmentation. *arXiv.org*. doi: 10.48550/arxiv.2410.00249

[32] Rajapakse, R. N., Zahedi, M., Babar, M. A. & Shen, H. (2022). Challenges and solutions when adopting DevSecOps : A systematic review. *Information and software technology*, *141*, 106700. Repéré le 2025-03-29, à https://www.sciencedirect.com/science/article/pii/S0950584921001543?casa_token=WOQ0aTiMq4sAAAAA:yzpskPEIFqpKa1FEFRY9Eoxbb8GNufK0lUr1tL9ORbH84ww44XQSTfjXVj-0Zw2-vnlWf9Ztmbo

[33] Ramaj, X., Sánchez-Gordón, M.-L., Gkioulos, V. & Palacios, R. C. (2024). On DevSecOps and Risk Management in Critical Infrastructures : Practitioners' Insights on Needs and Goals. *2024 IEEE/ACM 4th International Workshop on Engineering and Cybersecurity of Critical Systems and 2024 IEEE/ACM Second International Workshop on Software Vulnerability (EnCyCriS/SVM)*. doi: 10.1145/3643662.3643954

[34] Rangnau, T., Buijtenen, R. v., Fransen, F. & Turkmen, F. (2020). Continuous Security Testing : A Case Study on Integrating Dynamic Security Testing Tools in CI/CD Pipelines. doi: 10.1109/edoc49727.2020.00026

[35] Reddy, A. & Kumar, S. (2023). AI-Driven Security Automation in DevSecOps Pipelines.

[36] Tejasvi Nuthalapati (2025). Designing with AI, Not Around It – Human-Centric Archi-

tecture in the Age of Intelligence. *Journal of Computer Science and Technology Studies*. S2ID : b9276199e3a9c39fa6f73081c213a02080dcc7f0, doi: 10.32996/jcsts.2025.7.4.19

[37] Thopalle, P. K. (2024, 3). DevSecOps : Integrating Security Into the DevOps Lifecycle with AI and Automation. *INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN ENGINEERING AND TECHNOLOGY (IJARET)*, pp. 452–466. Repéré le 2026-05-03, à https://openlibindex.com/index.php/IJARET/article/view/IJARET_15_03_038

[38] Verdugo, J. & Rodríguez, M. (2020, 3). Assessing data cybersecurity using ISO/IEC 25012. *Software Quality Journal*, pp. 965–985. doi: 10.1007/s11219-019-09494-x

[39] Yelkoti, N. K. K. R. (2025, 6). Security as Code : An Architectural Framework for Automated Risk Mitigation in DevSecOps Pipelines. *Journal of Computer Science and Technology Studies*, pp. 235–244. doi: 10.32996/jcsts.2025.7.6.26

[40] Zha, D., Bhat, Z. P., Lai, K.-H., Yang, F. & Hu, X. (2023). Data-centric AI : Perspectives and Challenges. Dans *Proceedings of the 2023 SIAM International Conference on Data Mining (SDM)*, Proceedings pp. 945–948. Society for Industrial and Applied Mathematics

[41] Zhou, Y., Tu, F., Sha, K., Ding, J. & Chen, H. (2024, juillet). A Survey on Data Quality Dimensions and Tools for Machine Learning Invited Paper. Dans *2024 IEEE International Conference on Artificial Intelligence Testing (AITest)*, pp. 120–131. ISSN : 2835-3560, doi: 10.1109/AITest62860.2024.00023

[42] Éireann Leverett (2025, décembre). *2025 Vulnerability Forecast : Year-End Review*. Repéré le 2026-06-16, à https://www.first.org/blog/20251229-Vulnerability-Forecast-Review