

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
École Nationale Polytechnique



المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique



Département : Génie Industriel
Entreprise : SLB NAF Ex-Schlumberger

Mémoire de Projet de Fin d'Études
En vue de l'obtention du diplôme d'Ingénieur d'État en Génie Industriel
Option : Data Science et Intelligence Artificielle

Développement d'une application web
permettant l'extraction de renseignements sur le forage
Et prévision des risques de forage.

Application : SLB NAF Ex-Schlumberger

Ahmed Abdelnour TERFANI

Adel Sofiane IZEROUEL

Sous la direction de M. Oussama ARKI

Présenté et soutenu publiquement le (29 / 06 / 2025)

Composition du jury :

Président	M. Hakim FOURAR-LAIDI	MCA	ENP
Examineur	Mme. Imene BARECHE	MAB	ENP
Promoteur	M. Oussama ARKI	MCB	ENP
Invité	Mme. Mounia CHEKAI	AI Technical Sales	SLB

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
École Nationale Polytechnique



المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique



Département : Génie Industriel
Entreprise : SLB NAF Ex-Schlumberger

Mémoire de Projet de Fin d'Études
En vue de l'obtention du diplôme d'Ingénieur d'État en Génie Industriel
Option : Data Science et Intelligence Artificielle

Développement d'une application web
permettant l'extraction de renseignements sur le forage
Et prévision des risques de forage.

Application : SLB NAF Ex-Schlumberger

Ahmed Abdelnour TERFANI

Adel Sofiane IZEROUEL

Sous la direction de M. Oussama ARKI

Présenté et soutenu publiquement le (29 / 06 / 2025)

Composition du jury :

Président	M. Hakim FOURAR-LAIDI	MCA	ENP
Examineur	Mme. Imene BARECHE	MAB	ENP
Promoteur	M. Oussama ARKI	MCB	ENP
Invité	Mme. Mounia CHEKAI	AI Technical Sales	SLB

Dédicaces

Je dédie ce travail :

Je tiens à rendre hommage à ma mère, qui a consenti d'innombrables efforts depuis mon enfance pour me permettre d'accéder à la position que j'occupe à présent.

À mon père, qui a toujours fait preuve d'une abnégation sans faille au service de sa profession et de sa famille,

Je tiens à exprimer ma profonde gratitude à l'endroit de mes oncles et tantes. Leur présence constante et leur soutien inestimable ont été des sources d'inspiration et de réconfort tout au long de mon parcours. Leur soutien indéfectible et leur écoute attentive ont été des piliers essentiels dans ma vie,

Dans l'hypothèse où les grands-parents seraient encore en vie, il conviendrait de formuler des prières à Allah, dans l'espoir qu'ils ressentent de la fierté à mon égard,

À l'attention de mes soeurs et cousins, pour lesquels j'éprouve une profonde affection, même si je ne le manifeste pas de manière suffisante,

À mes amis et proches (vous vous reconnaîtrez), vous qui avez forgé des souvenirs inoubliables dans ma mémoire,

Aux DAÏTAS ! Ces trois dernières années sont certainement les meilleures que j'ai pu passer durant mon parcours académique, et ce grâce à vous, une classe légendaire. Il n'y aura certainement pas de deuxième comme vous !

Et à toute personne que j'aime et estime.

Abdelnour.

Dédicaces

Je dédie ce travail :

Ma chère mère, merci pour tout. Ta présence et ton soutien ont toujours été essentiels pour moi. Que Dieu te protège et t'accorde une longue vie remplie de bonheur et de santé.

Mon cher père, merci pour ton soutien, tes conseils et ton exemple. Grâce à toi, j'ai appris l'importance du travail et des responsabilités. Qu'Allah te garde auprès de nous.

Ma chère tante, merci pour ton soutien et ta générosité. Tu as toujours été là pour moi. Que Dieu te protège et t'apporte du bonheur.

À mes frères et sœur, merci d'être toujours là. Vous comptez beaucoup pour moi. Beaucoup de réussite à vous, inchallah.

À mes professeurs du cycle préparatoire et de la spécialité, merci pour votre enseignement et votre soutien. Je vous dédie ce rapport avec respect et gratitude.

À mes amis du cycle préparatoire et de la promo DSIA, merci d'avoir été là tout au long de cette aventure. Vous avez compté pour moi, et je ne vous oublierai pas.

Pour finir je dédie aussi ce travail à mon grand-père, Allah yarahmou. Il m'a toujours encouragé à bien travailler.

Adel.

Remerciements

Tout d'abord, ce travail n'a été réalisé que grâce à Allah Tout Puissant.

Nous tenons à exprimer notre profonde gratitude à notre promoteur, Monsieur Oussama ARKI, pour son soutien indéfectible tout au long de ce projet.

Nous tenons également à exprimer notre gratitude envers Monsieur Mohamed Yacine SAHAR et Madame Mounia CHEKAI, nos tuteurs dans le milieu professionnel, pour leurs conseils judicieux et leur soutien indéfectible.

Nos plus vifs remerciements s'adressent à Monsieur Ahmed Moncef BOUSSELAT, dont la contribution s'est avérée inestimable et dont les conseils avisés ont été une source d'enrichissement.

Nous tenons à exprimer notre profonde gratitude aux membres du jury qui ont consacré un effort considérable à l'évaluation de notre travail.

Nous tenons à exprimer notre gratitude envers nos enseignants, dont la contribution a été déterminante pour notre réussite.

Nous sommes fortement reconnaissants envers toute personne ayant contribué de près ou de loin à la réalisation de ce travail.

ملخص

يهدف هذا العمل إلى تطوير تطبيق ويب بشكل كامل من الصفر، مخصص لاستخلاص المعلومات الهامة والتنبؤ بمخاطر عمليات الحفر استنادًا إلى تحليل تقارير الحفر اليومية. يسعى التطبيق إلى تزويد مهندسي الحفر والمديرين برؤى قابلة للتنفيذ لتحسين كفاءة العمليات وتقليل المخاطر المحتملة. تعتمد المنهجية، التي تم بناؤها وتصميمها بالكامل داخليًا، على معالجة البيانات من تقارير الحفر، وتطبيق تقنيات علوم البيانات لاستخراج الخصائص الهامة، واستخدام نماذج التعلم الآلي والذكاء الاصطناعي المتقدمة لتوليد الرؤى والتنبؤ الدقيق بالمخاطر.

يتم تقديم كافة هذه الإمكانيات عبر واجهة ويب تفاعلية على هيئة لوحة تحكم Dashboard سهلة الاستخدام، تمكن مستخدمي شركة SLB من تحليل البيانات بفعالية وعرض المخاطر المتوقعة بوضوح.

الكلمات المفتاحية : تقارير الحفر اليومية، استخلاص البيانات ، Visualization ، التنبؤ بالمخاطر ، التعلم الآلي ، الذكاء الاصطناعي ، تطبيق ويب ، صناعة النفط والغاز

Abstract

This work focuses on the development of a web application entirely from scratch for extracting insights and predicting risks from daily drilling reports (DDR) within the oil and gas industry. The primary aim is to provide drilling engineers and managers with actionable intelligence to optimize operational efficiency and mitigate potential hazards. The solution, conceived and built in-house, involves processing DDR data, applying data science techniques for pattern recognition, and utilizing advanced Machine Learning and AI models for robust insight generation and risk prediction.

All these functionalities are delivered through a user-friendly dashboard web interface, specifically designed for effective data analysis and clear visualization of predicted risks.

Keywords : Daily Drilling Reports (DDR), Insight Extraction, Visualization, Risk Prediction, Machine Learning, AI Models, Oil Gas Drilling, Web Application.

Résumé

Ce travail porte sur le développement intégral, à partir de zéro, d'une application web pour l'extraction d'informations et la prédiction des risques à partir des rapports journaliers de forage (DDR) dans l'industrie pétrolière et gazière. L'objectif principal est de fournir aux ingénieurs et responsables de forage des informations exploitables pour optimiser l'efficacité opérationnelle et atténuer les dangers potentiels. La solution, conçue et réalisée entièrement en interne, comprend le traitement des données des DDR, l'application de techniques de science des données pour la reconnaissance de formes, et l'utilisation de modèles d'apprentissage automatique et d'intelligence artificielle (IA) avancés pour la génération d'informations pertinentes et la prédiction fiable des risques.

Toutes ces fonctionnalités sont accessibles via une interface web de type tableau de bord (dashboard) conviviale, spécifiquement conçue pour une analyse efficace des données et une visualisation claire des risques prédits.

Mots-clés : Rapports Journaliers de Forage (DDR), Extraction d'Informations, Visualisation, Prédiction des Risques, Apprentissage Automatique, Modèles d'IA, Forage Pétrolier et Gazier, Application Web.

TABLE DES MATIÈRES

Liste des Tableaux

Liste des Figures

Liste des Abréviations

Introduction Générale	16
État des lieux	18
1 État des lieux	18
1.1 Introduction	19
1.2 Présentation du marché pétrolier dans le monde	19
1.3 Présentation de SLB	20
1.3.1 Divisions Et Business Lines	21
1.3.2 Organisation de SLB	22
1.3.3 Implication de SLB dans l'innovation et la gestion des données de forage	23
1.3.4 SLB North Africa (NAF)	23
1.4 Description des Données : Les Rapports Journaliers de Forage (DDR) . .	26
1.4.1 Format et Contenu Typique des DDR	26
1.4.2 Défis Liés à l'Exploitation Manuelle des DDR	28
1.5 Identification des Informations Clés à Extraire et des Risques à Prédire . .	30
1.6 Positionnement, Problématique et Objectifs du Projet	31
1.6.1 Analyse du Paysage Logiciel Existant et Identification de la Lacune	31
1.6.2 Problématique et Objectifs	31
1.7 Conclusion	32
État de l'art	34

2	État de l'art	34
2.1	Introduction	35
2.2	Le forage pétrolier et gazier : Principes et défis	35
2.2.1	Principes Fondamentaux et Composants Clés	35
2.2.2	Les Défis Inhérents aux Opérations de Forage	38
2.2.3	Processus de forage et opérations clés	39
2.2.4	Rapports journaliers de forage (DDR) : Structure et importance	44
2.3	Extraction d'informations et analyse de données textuelles	48
2.3.1	Techniques de Traitement Automatique du Langage (TAL/NLP)	48
2.3.2	Application du TAL aux rapports techniques : Le Défi du Langage de Domaine	52
2.4	Prédiction des risques en forage par apprentissage automatique	56
2.4.1	Types de risques en forage et leur impact	56
2.4.2	Approches de modélisation prédictive pour les risques de forage	61
2.4.3	Métriques d'évaluation et Systèmes d'Alerte : De la Prédiction à l'Action	65
	Conception et implémentation de la solution	70
3	Conception et implémentation de la solution	70
3.1	Introduction	71
3.2	Architecture générale de l'application web	71
3.2.1	Composant front-end	73
3.2.2	Composant back-end	76
3.3	Construction du dataset utilisé	78
3.3.1	Identification et collecte des rapports techniques	78
3.3.2	Nettoyage et normalisation des rapports	79
3.3.3	Extraction des contenus des rapports	80
3.3.4	Extraction d'informations pertinentes	80
3.3.5	Compréhension des données	82
3.3.6	Préparation et finalisation des données pour l'analyse et la modélisation	83
3.4	Conception et Implémentation des Modules d'Intelligence Artificielle	85
3.4.1	Modélisation Prédictive des Risques par Apprentissage Automatique	85
3.5	Analyse Hybride des Risques de Forage : Combinaison de Modèles de Machine Learning Classiques et d'un Modèle de Langage Large (LLM) via une Architecture RAG et MCP	92
3.5.1	Principes Fondamentaux de l'Approche Hybride	93
3.5.2	Composants Clés de l'Architecture Hybride	93
3.5.3	Processus d'Analyse d'un DDR (Workflow Hybride RAG+MCP)	95
3.5.4	Exemple de Sortie LLM et Interprétation	95
3.5.5	Évaluation du Système Hybride	97
3.5.6	Discussion et Avantages de l'Approche Hybride	98
3.5.7	Limitations et Perspectives	98
3.6	Développement d'un Assistant Conversationnel Technique pour le Domaine du Forage	98
3.6.1	Objectif et Cas d'Usage	99
3.6.2	Architecture Technique Distribuée	99

3.6.3	Détail du Flux de Traitement d'une Requête	100
3.6.4	Évaluation du Chatbot Technique	102
3.6.5	Avantages et Limites de l'Approche Implémentée	104
3.7	Conclusion	106
Conclusion Générale		107
Bibliographie		109
Annexes		113
	Annexe A : Prétraitement et Structuration des Données de Forag	114
	Annexe B : Architecture et Entraînement des Modèles LSTM	117
	Annexe C : Architecture du Système d'Analyse Contextuelle (RAG)	120
	Annexe D : Guide d'Utilisation de l'Application Web	122
	Annexe E : Déploiement d'Applications Web avec Docker	124
	Annexe F : Architecture Technique et Technologies de l'Application Web	128
	Annexe H : Extraits de Code Significatifs	131

LISTE DES TABLEAUX

1.1	Fiche d'identité de la multinationale SLB (documents internes)	21
1.2	Répartition des GeoUnits de SLB. Ltd à travers le monde(document interne)	22
1.3	Répartitions des bases de SLB dans la région NAF	24
3.1	Performances comparées de différents modèles de classification obtenues avec LazyPredict.	86
3.2	Rapport de classification pour un modèle LSTM, illustrant les meilleures performances obtenues sur une tâche binaire.	87

TABLE DES FIGURES

1.1	Acteurs du Marché Parapétrolier dans le monde[53, p.18]	20
1.2	Acteurs du marché parapétrolier en Algérie[53, p.19]	20
1.3	Répartition des GUs dans les 5 bassins de l'entreprise (document interne)	23
1.4	SLB North Africa GeoUnit (document interne)	24
1.5	Siège social et bases de SLB Algérie (document interne)	25
1.6	Exemple Anonymisé d'une Section d'un Rapport Journalier de Forage (DDR)	26
1.7	Format et Contenu Typique des DDR	27
1.8	Défis Liés à l'Exploitation Manuelle des DDR	29
2.1	Schéma Détaillé d'une Installation de Forage et de ses Composants Principaux.	36
2.2	Illustration du processus de forage rotatif en fond de trou.	40
2.3	Opération de manœuvre : les tiges de forage sont assemblées ou désassemblées et stockées verticalement dans le derrick.	41
2.4	Étapes de la cimentation d'un tubage : (a) Descente du tubage, (b) Pompage du ciment dans l'annulaire.	42
2.5	Principe de la diagraphie au câble (Wireline Logging).	43
2.6	Schéma du circuit de circulation de la boue en surface (Système de Contrôle des Solides).	43
2.7	Configuration typique d'un équipement de test de puits en surface(flowhead-max).	44
2.8	Configuration typique d'un autre équipement de test de puits en surface(ssv-combo).	44
2.9	Structure Mixte d'un Rapport Journalier de Forage (DDR) Typique.	45
2.10	Flux d'Information et Rôle Central du DDR comme Pivot entre les Parties Prenantes des Opérations de Forage.	46
2.11	Exemple de pipeline de traitement pour l'extraction d'informations à partir de texte non structuré, illustrant la progression de la tokenisation à l'extraction de relations [46].	49
2.12	l'approche par affinage (Fine-Tuning)	54
2.13	l'approche par Génération Augmentée par Récupération (RAG)	55

2.14	Illustration des principaux mécanismes de coincement de la garniture de forage.	57
2.15	Schématisation des différents types d'instabilité des parois du puits. . . .	58
2.16	Mécanismes des pertes de circulation dans des formations fracturées ou à haute perméabilité.	59
2.17	Séquence d'un incident de contrôle de puits : de la détection d'un <i>kick</i> à la potentielle escalade vers une éruption (<i>blowout</i>).	60
2.18	Architecture Détaillée du Modèle LSTM Utilisé pour la Prédiction des Risques à partir des Données Séquentielles Structurées.	63
2.19	Architecture Détaillée du Système de Génération Augmentée par Récupération (RAG) pour l'Analyse Contextuelle des DDR.	64
2.20	Matrices de Confusion pour chaque modèle de prédiction de risque sur l'ensemble de test.	66
2.21	Courbes ROC et Aires Sous la Courbe (AUC) pour chaque modèle de prédiction de risque.	67
3.1	Architecture générale de l'application web	71
3.2	Représentation conceptuelle de la solution	73
3.3	Vue du tableau de bord principal de l'application, affichant les KPIs globaux et des graphiques sur la flotte de puits.	74
3.4	Vues détaillées de l'application pour un puits spécifique.	75
3.5	Interface de navigation permettant d'accéder aux rapports journaliers de forage (DDR) originaux au format HTML.	75
3.6	Schéma des Principaux Endpoint	77
3.7	Structure de répertoires illustrant l'organisation hiérarchique des fichiers DDR, où chaque sous-dossier représente un puits de forage.	79
3.8	image qui represent une partie du fichier csv final.	82
3.9	Matrice de confusion du modèle LSTM pour une tâche binaire.	87
3.10	confusion matrix target confirmed equipment failure next day	88
3.11	roc curve target confirmed equipment failure next day	89
3.12	confusion matrix target confirmed wellbore instability next day	89
3.13	roc curve target confirmed wellbore instability next day	90
3.14	confusion matrix target confirmed stuck pipe next day	90
3.15	roc curve target confirmed stuck pipe next day	91
3.16	confusion matrix target confirmed stuck pipe next day	91
3.17	roc curve target confirmed stuck pipe next day	92
3.18	Exploiter le Protocole MCP pour Connecter les LLM aux Modèles LSTM et aux Données Personnelles	93
3.19	Mise en place du tunnel 'ngrok' dans l'environnement Kaggle pour exposer l'API du chatbot.	100
3.20	Journal d'initialisation des composants d'IA dans le notebook Kaggle. . .	101
3.21	Trace d'exécution d'une requête dans le terminal du serveur, montrant les étapes de récupération et de génération.	102
3.22	Exemple d'interaction avec le chatbot technique au sein de l'application web.	105
E.1	Architecture des composants de Docker, illustrant la relation entre le client, le démon Docker, les images et les conteneurs [17].	125

E.1	Architecture Détaillée de l'Application Web Développée, montrant la séparation entre le client (Frontend), le serveur principal (Backend Flask), et le service d'IA externe (LLM sur Kaggle) accessible via ngrok.	129
E.1	Extraction des donner tabulaires.	131
E.1	Extraction des donner textuel libre.	132

LISTE DES ABRÉVIATIONS

AFE : Authorization for Expenditure

API : Application Programming Interface

AUC : Area Under the Curve (Aire Sous la Courbe)

BDD : Base de Données

BHA : Bottom Hole Assembly (Assemblage de Fond de Puits)

BOP : Blowout Preventer (Bloc d'Obturbateurs de Puits)

COPS : Compagnie des Opérations Pétrolières SLB

CRUD : Create, Read, Update, Delete

CSV : Comma-Separated Values

DDR : Daily Drilling Report (Rapport Journalier de Forage)

DNS : Domain Name System

DOM : Document Object Model

EDA : Exploratory Data Analysis (Analyse Exploratoire des Données)

ENP : École Nationale Polytechnique

FAISS : Facebook AI Similarity Search

FN : Faux Négatif (False Negative)

FP : Faux Positif (False Positive)

GU : GeoUnit

HSE : Health, Safety, and Environment (Hygiène, Sécurité et Environnement)

HTML : HyperText Markup Language

HTTP : Hypertext Transfer Protocol

IA : Intelligence Artificielle

JS : JavaScript

JSON : JavaScript Object Notation

KPI : Key Performance Indicator (Indicateur Clé de Performance)

LLM : Large Language Model (Grand Modèle de Langage)

LSTM : Long Short-Term Memory

LWD : Logging While Drilling

MCP : Model Context Protocol

ML : Machine Learning (Apprentissage Automatique)

MWD : Measurement While Drilling

NAF : North Africa (Afrique du Nord)

NER : Named Entity Recognition (Reconnaissance d'Entités Nommées)

NLP : Natural Language Processing

NPT : Non-Productive Time (Temps Non Productif)

PDC : Polycrystalline Diamond Compact

RAG : Retrieval-Augmented Generation (Génération Augmentée par Récupération)

ROC : Receiver Operating Characteristic

ROP : Rate of Penetration (Vitesse de Pénétration)

RPM : Revolutions Per Minute (Tours par Minute)

SLB : Schlumberger

SPS : Services Pétroliers SLB

SQL : Structured Query Language

TAL : Traitement Automatique du Langage

UI : User Interface (Interface Utilisateur)

URL : Uniform Resource Locator

UX : User Experience (Expérience Utilisateur)

VM : Virtual Machine (Machine Virtuelle)

VN : Vrai Négatif (True Negative)

VP : Vrai Positif (True Positive)

WC : Well Control (Contrôle de Puits)

WOB : Weight On Bit (Poids sur l'Outil)

INTRODUCTION GÉNÉRALE

L'industrie pétrolière et gazière, un pilier fondamental de l'économie mondiale, s'appuie sur des opérations de forage qui sont à la fois des prouesses techniques et des entreprises à haut risque. La réussite de ces opérations critiques, qui impliquent des investissements considérables, dépend d'une gestion rigoureuse des données, d'une prise de décision éclairée et d'une anticipation proactive des problèmes potentiels.

Dans ce contexte, une quantité massive de données est générée quotidiennement, dont une part significative est consignée dans les rapports journaliers de forage (DDR – *Daily Drilling Reports*). Ces rapports constituent une mine d'informations précieuses, documentant en détail les activités, les paramètres, les événements et les observations des dernières 24 heures d'opération. Cependant, l'exploitation de cette richesse informationnelle se heurte à des défis de taille. L'analyse manuelle des DDR est un processus laborieux, chronophage et sujet à l'erreur humaine. Plus important encore, cette approche traditionnelle limite la capacité à identifier des schémas complexes et des corrélations cachées qui sont pourtant cruciaux pour l'optimisation des performances et la prévention des incidents.

Face à ces limitations, les avancées en intelligence artificielle (IA), combinées à la flexibilité des applications web modernes, offrent des perspectives transformatrices. Ces technologies permettent d'automatiser le traitement de vastes volumes de données, d'identifier des modèles significatifs et de développer des systèmes prédictifs capables d'anticiper les risques avec une précision accrue.

Le présent mémoire a pour objectif de concevoir et développer une **application web intelligente** capable d'extraire automatiquement des informations pertinentes des DDR et de fournir une analyse prédictive des risques de forage. Notre solution intègre trois modules technologiques complémentaires au sein d'une plateforme unifiée :

1. **Extraction et Structuration des Données par Traitement Automatique du Langage (TAL).** Le premier pilier de notre solution est un module robuste d'extraction automatisée. Utilisant des techniques de TAL [15], ce module analyse le contenu dense et varié des DDR pour identifier, extraire et structurer les informations les plus pertinentes : phases d'opérations, temps non productifs (NPT), paramètres de forage critiques (ROP, WOB), et observations textuelles. L'objectif est de convertir la masse de données brutes en une base de connaissances propre et sémantiquement

riche, prête à être exploitée par les autres composants de l'application.

2. **Visualisation Interactive via un Tableau de Bord Web.** Une fois les informations clés extraites, il est impératif de les présenter de manière claire et exploitable. Notre deuxième axe est donc le développement d'une interface web sophistiquée, conçue comme un tableau de bord interactif. Ce portail centralisé offre aux ingénieurs une vue d'ensemble des opérations et une gamme d'outils de visualisation dynamique. L'interactivité est au cœur de l'interface, permettant aux utilisateurs de filtrer les données, d'explorer les tendances et d'identifier rapidement les anomalies, supportant ainsi une prise de décision plus rapide et mieux informée.
3. **Analyse et Prédiction des Risques par un Système d'IA Hybride.** Le troisième pilier vise à dépasser l'analyse descriptive pour s'engager dans une démarche prédictive. Nous avons développé un système d'IA innovant qui combine deux approches :
 - Des modèles d'apprentissage automatique supervisé, notamment des réseaux de neurones **LSTM**, sont entraînés pour prédire la probabilité de survenue de risques opérationnels spécifiques (coincements de garniture, pannes, etc.) à partir des données séquentielles.
 - Un Grand Modèle de Langage (**LLM**) [34], intégré via une architecture **MCP** et **RAG** [32], analyse les alertes du LSTM et le texte des rapports pour fournir une **explication contextuelle** du risque, en justifiant ses conclusions et en proposant des recommandations.

Les prédictions et analyses sont ensuite intégrées à l'interface, offrant aux utilisateurs des alertes précoces et des informations prospectives pour une gestion proactive des risques.

Ce travail s'articulera autour de trois chapitres principaux. Le **premier chapitre, "État des lieux"**, établira le contexte industriel et la problématique en présentant le marché pétrolier, l'entreprise SLB, et la nature des données au cœur de notre étude, les DDR. Le **deuxième chapitre, "État de l'art"**, posera les fondations théoriques en explorant en détail les principes du forage, les techniques d'analyse de données, et les approches de modélisation prédictive. Enfin, le **troisième chapitre, "Conception et implémentation de la solution"**, constituera le cœur de notre contribution, en détaillant l'architecture logicielle, la construction du dataset, et le développement de nos modules d'intelligence artificielle. Une conclusion générale synthétisera nos travaux, leurs limites, et les perspectives d'évolution.

CHAPITRE

1

ÉTAT DES LIEUX

1.1 Introduction

Ce premier chapitre se consacre à la mise en contexte de notre projet. Pour bien cerner les enjeux et la pertinence de notre démarche, il est essentiel de comprendre l'environnement dans lequel elle s'inscrit. Nous commencerons par présenter l'entreprise Schlumberger (SLB), acteur majeur de l'industrie des services pétroliers, afin de situer le cadre industriel potentiel de notre application. Ensuite, nous décrirons en détail les données au cœur de notre étude, les rapports journaliers de forage (DDR), en précisant leur format et les défis inhérents à leur exploitation. Nous poursuivrons par l'identification des informations clés que nous visons à extraire et des risques que nous cherchons à prédire. Enfin, ce chapitre aboutira à l'affinage de la problématique et à la définition claire des objectifs de notre projet.

1.2 Présentation du marché pétrolier dans le monde

Le marché pétrolier mondial occupe une place centrale dans l'économie globale, constituant une source d'énergie indispensable pour une multitude de pays et de secteurs industriels. De ce fait, la demande mondiale de pétrole est particulièrement élevée, principalement alimentée par les besoins en carburant des secteurs des transports, de la fabrication industrielle et de la production d'électricité. Les États-Unis, la Chine, l'Inde, l'Union européenne et le Japon figurent parmi les principaux consommateurs de pétrole à l'échelle mondiale.

Le segment des services pétroliers englobe l'intégralité des activités et des entreprises qui proposent des services et des produits spécialisés aux compagnies opérant dans le secteur pétrolier et gazier. Ces services sont fondamentaux pour appuyer les différentes phases d'exploration, de production, de transformation et de distribution des hydrocarbures.

En 2020, la valeur du marché des services pétroliers était estimée à 96,65 milliards de dollars américains. D'après les projections, cette valeur est susceptible d'atteindre 135 milliards de dollars américains d'ici l'année 2027 (Intelligence, 2022). Cette expansion est majoritairement portée par une demande croissante en hydrocarbures, ainsi que par la découverte de nouveaux gisements en mer (offshore). L'offre de services sur ce marché est variée, toutefois, les services liés à la production et au forage y tiennent une place prépondérante. Il est à noter que le taux de croissance du marché des services pétroliers varie considérablement en fonction de la région géographique concernée [21].

Plusieurs acteurs majeurs dominent le marché des services pétroliers, parmi lesquels on compte SLB, Baker Hughes, Halliburton, General Electric, et DMC Global, entre autres[47, 5, 25, 23, 16].

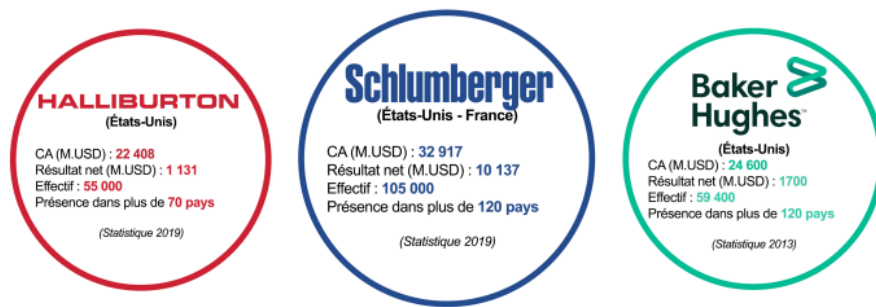


FIGURE 1.1 – Acteurs du Marché Parapétrolier dans le monde[53, p.18]

L'état de santé du marché des services pétroliers est intimement corrélé à l'activité globale de l'industrie pétrolière et gazière. En effet, les fluctuations des prix du pétrole, les niveaux d'investissement consentis dans l'exploration et la production, ainsi que les décisions stratégiques des entreprises pétrolières concernant leurs dépenses d'exploitation, peuvent tous exercer une influence notable sur la demande de services pétroliers.

L'industrie des services pétroliers est caractérisée par une évolution constante, avec une demande soutenue pour des technologies et des solutions innovantes. Les entreprises du secteur s'efforcent de développer des équipements toujours plus performants, des techniques de forage plus efficaces, des technologies de récupération améliorées, et des solutions numériques destinées à optimiser les opérations pétrolières.

Concernant l'Algérie, en 2021, le pays se positionnait au quatrième rang des producteurs de pétrole sur le continent africain, affichant une production moyenne de 969 000 barils par jour. En 2022, l'Algérie s'est distinguée en prenant la tête du classement en termes de nombre de découvertes pétrolières, avec trois nouvelles découvertes recensées, selon un rapport publié par l'Organisation des pays exportateurs de pétrole (OPEP) en 2022. Cette dynamique positive en matière de découvertes pétrolières devrait logiquement se traduire par une demande accrue de services pétroliers en Algérie.



FIGURE 1.2 – Acteurs du marché parapétrolier en Algérie[53, p.19]

1.3 Présentation de SLB

SLB est une entreprise multinationale franco-américaine, fondée en 1926, qui s'est imposée comme le leader mondial dans le domaine des services et équipements pétroliers. Sa mission principale consiste à fournir des technologies de pointe, des services de gestion

de projets intégrés, ainsi que des solutions d'information à l'industrie de l'exploration et de l'exploitation pétrolière et gazière à l'échelle internationale. Active dans plus de 120 pays et forte d'équipes composées de 170 nationalités différentes, SLB intervient sur l'ensemble de la chaîne de valeur des processus d'exploration et d'exploitation, depuis la phase de découverte et de prospection des gisements pétroliers et gaziers jusqu'à la fin de vie des puits. L'entreprise se distingue par un engagement constant en matière de recherche et développement, une valorisation affirmée de sa ressource humaine et de sa technologie, et un réinvestissement systématique de ses profits dans ces domaines afin de maintenir son avantage concurrentiel sur le marché[47].

La fiche suivante synthétise les informations clés concernant l'entreprise :

TABLE 1.1 – Fiche d'identité de la multinationale SLB (documents internes)

Nom de l'entreprise	
Date de création	1926
Fondateurs	Les frères Conrad et Marcel SLB
Forme Juridique	Société anonyme avec appel public à l'épargne
Siège sociale	Bureaux principaux à Houston (USA), Paris (France), et la Haye (Pays-Bas).
Direction	CEO : Olivier Le Peuch EVP et CFO : Stephane Biguet
Secteur d'activité	Service pétrolier
Effectif	92000 de 160 nationalités (2021)
Capital sociale	57 000 millions USD (2021)
Chiffre d'affaire	22 930 millions USD (2021)
Résultat net	188 millions USD (2021)

1.3.1 Divisions Et Business Lines

Dans le but d'améliorer son portefeuille d'offres et d'aligner les flux de travail de ses clients avec les capacités internes de l'entreprise, SLB s'est organisée en quatre divisions distinctes, chacune regroupant plusieurs "Business Lines" (lignes de métier)[49, 26, 31, 53, 11].

- **Digital Integration** Cette division regroupe les technologies et les processus liés à la digitalisation et à l'intégration des données, visant à améliorer les performances globales de l'entreprise. Elle présente un potentiel d'évolution très élevé, compte tenu de la transformation digitale que traverse actuellement l'industrie.
- **Reservoir Performance** Cette division se concentre sur le suivi de la performance des réservoirs. Son objectif est de permettre aux clients d'obtenir des estimations de réserves les plus précises possible et de maximiser la récupération des hydrocarbures.
- **Production Systems** L'objectif de cette division est de maintenir la production d'un puits et de la prolonger autant que possible, en mettant en œuvre des technologies

planifiées et exécutées avec soin. C’est à ce niveau qu’interviennent l’expertise, les systèmes et les outils de production de SLB.

- **Well Construction** Cette division englobe un ensemble d’activités, de processus et de technologies de forage intelligent. Ces derniers visent à maximiser la précision, l’efficacité et la valeur des opérations de forage, tout en minimisant les risques. Elle se focalise sur l’implémentation de changements dynamiques pour améliorer continuellement les opérations de forage, en s’appuyant sur des approches innovantes et des technologies de pointe.

1.3.2 Organisation de SLB

En 2020, SLB Ltd a initié une restructuration majeure de son organisation, adoptant une structure simplifiée et plus réactive, tout en conservant une orientation prioritaire vers ses clients. Antérieurement, l’entreprise était structurée en deux grands hémisphères : l’Hémisphère Est (Eastern Hemisphere) et l’Hémisphère Ouest (Western Hemisphere). La nouvelle organisation est désormais articulée autour de cinq bassins et de 30 unités géographiques (GeoUnits), comme l’illustre le tableau suivant[47, 26, 53, 11].

TABLE 1.2 – Répartition des GeoUnits de SLB. Ltd à travers le monde(document interne)

Bassins	Americas Land	Offshore Atlantic	Russia and Central Asia	Asia	Middle East and North Africa
Capitale	Houston	London	Moscow	Kuala Lumpur	Dubai
Geo Units	Canada	Land Angola, Central And East Africa	Russia Land	East Asia	North Africa
	US Land	Brazil	Arctic And South Off-shore	India	Egypt, Sudan and East Mediterranean
	Ecuador, Colombia And Peru	Europe	Azerbaijan And Turkménistan	Australia, New Zealand And Papua New Guinea	Iraq
	Argentina, Bolivia And Chile	Guyana, Trinidad And Caribbean	Kazakhstan	Indonesia	Kuwait
	Mexico And Central	America	Sakhalin	China	Saudi And Bahrain
		America Off-shore			Qatar
		Nigeria And West Africa			Emirates
		Scandinavia			Oman, Yèmen et pakistan



FIGURE 1.3 – Répartition des GUs dans les 5 bassins de l'entreprise (document interne)

1.3.3 Implication de SLB dans l'innovation et la gestion des données de forage

SLB a une longue tradition d'innovation technologique et investit continuellement dans la recherche et le développement pour relever les défis de l'industrie énergétique. Dans le domaine du forage, cela se traduit par le développement de solutions numériques visant à améliorer la sécurité, l'efficacité et la durabilité des opérations. La gestion et l'analyse des données de forage, y compris les DDR, sont essentielles pour optimiser les processus, réduire les temps non productifs (NPT) et anticiper les risques. Un outil capable d'extraire intelligemment des informations des DDR et de fournir des analyses prédictives s'inscrirait donc parfaitement dans la stratégie de digitalisation et d'amélioration continue de la performance prônée par SLB. Bien que notre projet soit développé dans un cadre académique, le contexte d'une entreprise comme SLB illustre la pertinence et l'applicabilité industrielle de notre démarche.

1.3.4 SLB North Africa (NAF)

1.3.4.a SLB North Africa (NAF)

SLB Ltd assure sa présence en Afrique du Nord par le biais de sa division géographique, SLB NAF (North Africa GeoMarket). Cette entité regroupe cinq pays : l'Algérie, le Maroc, la Tunisie, la Libye et le Tchad. Cette région géographique se caractérise par une richesse en ressources pétrolières et gazières, réparties à travers l'étendue du Sahara. Pour l'année 2020, l'Algérie a enregistré une production d'environ 898 700 barils de pétrole par jour, bien que ce volume ait subi un impact à la baisse en raison de la pandémie de COVID-19 (OPEP, 2021). SLB est implantée sur le continent africain depuis 1950 et opère dans la région de l'Afrique du Nord sous la dénomination de "North Africa Geo Unit". Cette division dispose de bases opérationnelles dans chacun des pays concernés, à savoir l'Algérie, le Maroc, la Tunisie, la Libye et le Tchad, et joue un rôle significatif dans cette région stratégique pour l'industrie pétrolière et gazière[49, 26, 31, 53, 47].



FIGURE 1.4 – SLB North Africa GeoUnit (document interne)

TABLE 1.3 – Répartitions des bases de SLB dans la région NAF

Pays	Base
Algérie	Hassi Messaoud (MD1, MD2, MD3, MD5, MI Base Cameron) - Ain Amenas - Hassi Berkine - Ain Salah
Maroc	Ain Hamra
Tunisie	Golf de Gabès
Libye	Benghazi- Tripoli
Tchad	Komé

1.3.4.b SLB Algérie

SLB a initié sa présence en Algérie dès 1955. L'entreprise y dispose d'un siège social situé dans la micro-zone d'activité Hydra – Alger, au 27 Rue Doudou Mokhtar. Ce siège héberge également les bureaux de la direction du "North Africa GeoMarket". En outre, SLB possède 11 bases opérationnelles qui sont réparties dans les principales zones d'activités pétrolières du pays, notamment à Hassi Messaoud, Ain Amenas, Hassi Berkine et Ain Salah. En complément de ces bases opérationnelles, SLB dispose également en Algérie de bases logistiques, de bunkers destinés au stockage d'explosifs, ainsi que de maisons d'hôtes.

Sur le territoire algérien, SLB mène ses opérations par l'intermédiaire de deux entités légales distinctes : COPS (Compagnie des Opérations Pétrolières SLB) et SPS (Services Pétroliers SLB). Ses principaux clients en Algérie incluent la compagnie pétrolière nationale SONATRACH, ainsi que d'autres sociétés internationales telles que Total, Anadarko, British Petroleum (BP), AGIP, et également le ministère algérien de l'Énergie et des Mines. Les services proposés par SLB en Algérie couvrent un large spectre, comprenant l'installation de bases opérationnelles, la réalisation d'études géologiques et sismiques, la construction et le test de puits, ainsi que l'importation des équipements nécessaires à ces activités.

L'Algérie occupe une position stratégique et centrale pour SLB dans la région nord-africaine, contribuant à plus de 60% de son chiffre d'affaires régional. Par ailleurs, SLB détient une part de marché considérable dans le secteur des services pétroliers en Algérie, celle-ci représentant plus de 70 % du marché national.

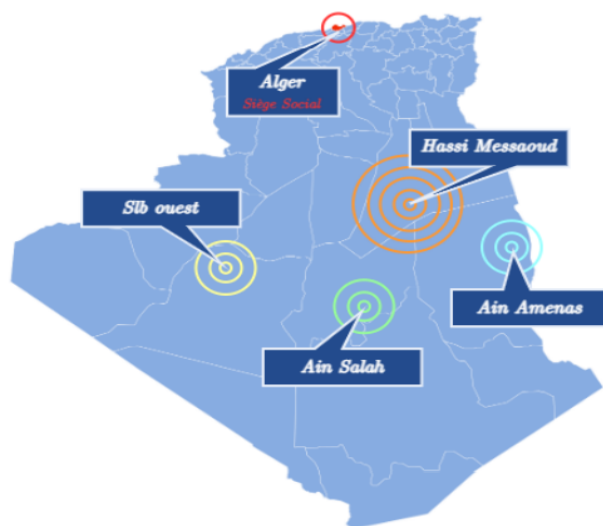


FIGURE 1.5 – Siège social et bases de SLB Algérie (document interne)

1.4 Description des Données : Les Rapports Journaliers de Forage (DDR)

Les rapports journaliers de forage (DDR) constituent la principale source de données pour notre étude. Ce sont des documents essentiels dans l'industrie pétrolière et gazière, produits quotidiennement pour chaque opération de forage en cours. Ils servent de compte-rendu détaillé des activités, des progrès, des problèmes rencontrés, des équipements utilisés et des observations faites sur le site de forage au cours des dernières 24 heures[47].

Summary of planned activities (24 Hours)						
CONTINUE WAITING DUE TO OFS STRIKE.						
Operations						
Start time	End time	End Depth mMD	Main - Sub Activity	State	Remark	
00:00	06:00	0	interruption -- other	ok	OPERATIONS SUSPENDED DUE TO OFS STRIKE.	
06:00	00:00	0	interruption -- other	ok	OPERATIONS SUSPENDED DUE TO OFS STRIKE.	

FIGURE 1.6 – Exemple Anonymisé d'une Section d'un Rapport Journalier de Forage (DDR)

1.4.1 Format et Contenu Typique des DDR

Bien que le format exact puisse varier légèrement d'une entreprise à l'autre ou d'une région à l'autre, un DDR typique est un document textuel (souvent au format PDF, fichier html, Word ou Excel) qui comprend plusieurs sections standardisées[47].

On y retrouve généralement :

- **Informations générales** : Nom du puits, date, numéro du rapport, localisation, opérateur, contracteur de forage, etc.
- **Profondeur et progression** : Profondeur atteinte, métrage foré dans les dernières 24h, profondeur du sabot précédent.
- **Résumé des opérations** : Description chronologique des activités principales menées (forage, manœuvre, cimentation, diagraphies, etc.).
- **Paramètres de forage** : Valeurs enregistrées pour des paramètres clés tels que le poids sur l'outil (WOB), la vitesse de rotation (RPM), le débit de la boue, la pression, le couple.
- **Garniture de forage (BHA)** : Description détaillée des composants de l'assemblage de fond de trou.

- **Boue de forage** : Propriétés de la boue (densité, viscosité, pH, etc.) et volumes utilisés.
- **Cimentation** : Détails des opérations de cimentation, le cas échéant.
- **Temps non productifs (NPT)** : Description des arrêts d'opérations, leurs causes et leur durée.
- **Consommables et inventaire** : Utilisation des matériaux, état des stocks.
- **Personnel et sécurité (HSE)** : Informations sur l'équipe, réunions de sécurité, incidents ou observations HSE.
- **Commentaires et observations** : Section souvent narrative où les ingénieurs ou superviseurs consignent des remarques importantes, des anomalies, ou des recommandations.
- **Prévisions pour les prochaines 24h** : La richesse et la diversité de ces informations font des DDR une source inestimable pour comprendre la dynamique des opérations et identifier les facteurs d'influence.

Summary report

Wellbore: 15/9-19 A

Period: 1997-09-04 00:00 - 1997-09-05 00:00

Status: normal	Dist Drilled (m): -999.99	Depth at Kick Off mMD: 2178
Report creation time: 2018-05-03 13:53	Penetration rate (m/h): -999.99	Depth at Kick Off mTVD:
Report number: 43	Hole Dia ():	Depth mMd: 4131
Days Ahead/Behind (+/-):	Pressure Test Type:	Depth mTVD: 3319
Operator: Statoil	Formation strength ():	Plug Back Depth mMD: 4019
Rig Name: BYFORD DOLPHIN	Dia Last Casing ():	Depth at formation strength mMD:
Drilling contractor:		Depth At Formation Strength mTVD:
Sput Date: 1997-07-25 00:00		Depth At Last Casing mMD: 4643
Wellbore type:		Depth At Last Casing mTVD:
Elevation RKB-MSL (m): 25		
Water depth MSL (m): 84		
Tight well: Y		
HPHT: Y		
Temperature ():		
Pressure ():		
Date Well Complete: 1997-08-30		

Summary of activities (24 Hours)

RIG OPERATIONS SUSPENDED DUE TO OFS STRIKE.

Summary of planned activities (24 Hours)

CONTINUE WAITING DUE TO OFS STRIKE.

Operations

Start time	End time	End Depth mMD	Main - Sub Activity	State	Remark
00:00	06:00	0	interruption -- other	ok	OPERATIONS SUSPENDED DUE TO OFS STRIKE.
06:00	00:00	0	interruption -- other	ok	OPERATIONS SUSPENDED DUE TO OFS STRIKE.

FIGURE 1.7 – Format et Contenu Typique des DDR

1.4.2 Défis Liés à l'Exploitation Manuelle des DDR

Actuellement, l'exploitation des DDR est un processus largement manuel et humain. Typiquement, un ingénieur de forage ou un superviseur lit le rapport chaque matin pour se faire une idée de l'avancement et des problèmes de la veille. Si une anomalie est détectée (par exemple, une augmentation du couple), l'ingénieur doit utiliser son expérience pour en évaluer la criticité. Un plan d'action est ensuite élaboré de manière collaborative, souvent lors de réunions quotidiennes ou par des échanges d'e-mails et d'appels téléphoniques avec l'équipe sur site. Ce processus, bien que basé sur l'expertise, est sujet à la subjectivité de l'analyste, dépend de sa disponibilité, et manque de systématisme pour l'analyse des tendances à long terme, ce qui justifie l'automatisation proposée par notre projet.

Malgré leur importance, l'exploitation des DDR présente plusieurs défis majeurs lorsqu'elle est effectuée manuellement[47] :

- **Informations générales :** Nom du puits, date, numéro du rapport, localisation, opérateur, contracteur de forage, etc.
- **Volume de données :** Une campagne de forage peut générer des centaines de DDR, rendant leur lecture et leur analyse exhaustives extrêmement chronophages.
- **Hétérogénéité des formats :** Même avec des sections standard, la manière de consigner l'information, l'utilisation d'abréviations, et la présence de texte libre introduisent une variabilité qui complique l'agrégation et la comparaison des données.
- **Extraction d'informations spécifiques :** Localiser et extraire systématiquement des informations précises (par exemple, toutes les occurrences d'un type de NPT spécifique sur plusieurs puits) est une tâche ardue et sujette à l'erreur ou à l'omission.
- **Identification de tendances et de schémas :** L'analyse manuelle limite la capacité à détecter des tendances subtiles, des corrélations entre différents paramètres ou événements, ou des schémas récurrents qui pourraient indiquer des risques émergents ou des opportunités d'optimisation.
- **Subjectivité et manque de standardisation dans les sections narratives :** L'interprétation des commentaires et des descriptions narratives peut varier d'un lecteur à l'autre.

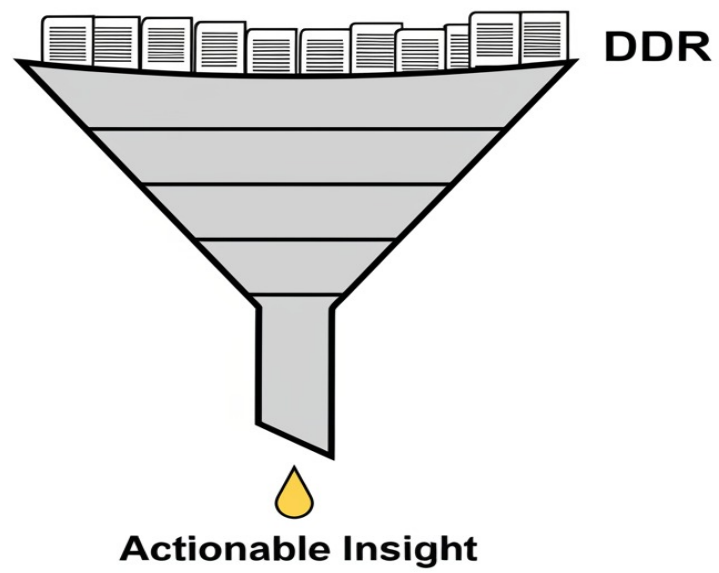


FIGURE 1.8 – Défis Liés à l'Exploitation Manuelle des DDR

1.5 Identification des Informations Clés à Extraire et des Risques à Prédire

Compte tenu de la richesse des DDR et des besoins de l'industrie, notre projet se concentrera sur l'extraction d'informations spécifiques et la prédiction de risques pertinents pour l'optimisation des opérations de forage [29, 42].

Informations clés à extraire

- Phases d'opérations précises (ex : forage rotatif, manœuvre de remontée, attente ciment).
- Événements de Temps Non Productifs (NPT) : Identification des types de NPT, leur durée, et les causes associées.
- Paramètres de forage critiques associés à des événements ou des phases spécifiques.
- Problèmes techniques mentionnés : Coincement de garniture, pertes de circulation, problèmes d'équipement [19, 45, 54].
- Observations géologiques ou liées à la formation (ex : présence de gaz, instabilité du trou).
- Utilisation et performance des outils de forage (BHA – Bottom Hole Assembly).

Risques à prédire

- Probabilité d'occurrence de NPT majeurs (ex : NPT lié au coincement, NPT lié à l'équipement).
- Risque de coincement de la garniture de forage (Stuck Pipe).
- Risque de pertes de circulation.
- Indicateurs prédictifs de problèmes d'instabilité du puits.

L'identification précise de ces éléments guidera le développement de nos modules d'extraction et de modélisation.

1.6 Positionnement, Problématique et Objectifs du Projet

L'industrie pétrolière et gazière a largement investi dans la digitalisation, donnant naissance à un écosystème logiciel riche mais souvent fragmenté. Pour définir précisément la problématique de notre projet et établir nos objectifs, il est essentiel d'analyser d'abord le paysage des outils existants afin d'identifier les lacunes technologiques et fonctionnelles, en particulier en ce qui concerne l'analyse des données de forage et des DDR.

1.6.1 Analyse du Paysage Logiciel Existant et Identification de la Lacune

Le marché des solutions pour le forage peut être segmenté en plusieurs catégories distinctes, chacune répondant à un besoin spécifique mais souvent de manière cloisonnée :

Systèmes de monitoring en temps réel (RTOC) : Ces plateformes excellent dans la surveillance des données de capteurs en direct (MWD/LWD). Leurs limitations résident dans leur faible capacité à analyser le contenu textuel et contextuel des DDR, se concentrant sur le "quoi" (une déviation de paramètre) plutôt que sur le "pourquoi" décrit dans les narrations [2].

Logiciels de planification de puits : Ces outils (e.g., Petrel de SLB) sont des références pour la conception en amont des opérations. Cependant, leur fonction n'est pas le suivi ou l'analyse des opérations quotidiennes en cours.

Plateformes de Business Intelligence (BI) : Des outils comme Power BI ou Tableau sont puissants pour visualiser des KPIs à partir de données structurées. Leur limitation majeure est leur incapacité à traiter nativement le texte non structuré, qui doit être extrait et codifié manuellement au préalable, un processus fastidieux et subjectif [35].

Initiatives d'IA spécialisées : Plusieurs solutions d'IA propriétaires excellent dans des tâches très ciblées (e.g., prédire l'usure d'un trépan). Leur limitation est leur nature de "boîte noire" et leur focus sur un seul type de risque, sans offrir une vision holistique ni exploiter le contexte narratif des DDR.

L'analyse de cet écosystème révèle une lacune claire : **il manque une plateforme intégrée qui fasse le pont entre l'analyse quantitative des données structurées et l'interprétation sémantique approfondie des données textuelles non structurées des DDR.** Les outils actuels traitent soit l'un, soit l'autre, mais rarement les deux de manière synergique.

1.6.2 Problématique et Objectifs

Cette lacune identifiée constitue le fondement de notre problématique. La question centrale de ce projet découle directement des défis inhérents à l'exploitation manuelle des DDR et du potentiel offert par les technologies modernes pour y remédier.

Problématique Générale : Comment transformer la masse d'informations contenues dans les DDR, souvent hétérogènes et textuelles, en connaissances actionnables permettant une meilleure compréhension des opérations, une identification proactive des risques et, in fine, une amélioration de l'efficacité et de la sécurité du forage ?

Problématique Spécifique : Plus concrètement, comment concevoir et développer une solution applicative web intégrée, capable d’automatiser l’extraction d’informations pertinentes à partir des DDR, de visualiser ces informations de manière interactive, et d’utiliser des modèles d’apprentissage automatique pour prédire les risques opérationnels en forage ?

Pour répondre à cette problématique, notre projet se positionne comme une **plateforme de synthèse et d’intelligence opérationnelle**. Il ne vise pas à remplacer les outils existants, mais à les compléter en créant une couche d’analyse post-opérationnelle (journalière) qui exploite la source de données la plus riche mais la plus sous-utilisée : le DDR dans son intégralité. De cette problématique et de ce positionnement découlent les objectifs principaux de notre projet de fin d’études :

1. **Analyser et structurer les données des DDR** pour identifier les formats, les contenus clés et les défis spécifiques à leur traitement.
2. **Développer un module d’extraction d’informations (IE) robuste** utilisant des techniques de TAL/NLP pour identifier et extraire automatiquement les entités, événements et paramètres pertinents.
3. **Construire et évaluer des modèles d’apprentissage automatique (ML)** pour la prédiction des risques de forage identifiés (e.g., NPT, pannes, coincement), en utilisant la synergie des données structurées et textuelles.
4. **Concevoir une interface web de type tableau de bord (dashboard)** permettant la visualisation interactive des données extraites, des tendances, des KPIs et des résultats des modèles prédictifs.
5. **Intégrer les modules d’extraction, de visualisation et de modélisation** au sein d’une application web fonctionnelle et conviviale.
6. **Valider l’approche proposée** en évaluant la performance globale de la solution développée, démontrant ainsi sa valeur ajoutée par rapport à l’existant.

En atteignant ces objectifs, notre projet fournit une réponse concrète à la problématique posée, en proposant une plateforme holistique qui transforme les DDR d’un simple outil d’archivage en une source dynamique de connaissances pour l’optimisation des performances et la gestion prédictive des risques.

1.7 Conclusion

Ce chapitre a permis d’établir le contexte fondamental de notre projet. Nous avons d’abord exploré la dynamique du marché pétrolier mondial et le rôle central d’acteurs majeurs comme SLB, soulignant l’importance stratégique de l’innovation technologique pour maintenir un avantage concurrentiel. Par la suite, nous avons disséqué la source de données au cœur de notre étude, les Rapports Journaliers de Forage (DDR), en détaillant leur structure hétérogène et les défis inhérents à leur exploitation manuelle. Cette analyse a mis en lumière la nécessité d’une approche automatisée pour transformer cette masse d’informations brutes en connaissances exploitables. En définissant précisément les informations clés à extraire et les risques opérationnels à prédire, nous avons pu affiner notre problématique et formaliser les objectifs de notre projet. Forts de cette mise en contexte, nous allons maintenant, dans le chapitre suivant, présenter l’état de l’art des technologies et des méthodologies qui constituent le socle de notre solution, en abordant les principes

du forage, les techniques d'analyse de données et les approches de modélisation prédictive.

CHAPITRE

2

ÉTAT DE L'ART

2.1 Introduction

La réussite des opérations de forage dans l'industrie pétrolière et gazière, secteur critique à haut risque, dépend de plus en plus de l'exploitation intelligente de vastes quantités de données. Au cœur de ce défi se trouvent les rapports journaliers de forage (DDR), des documents riches mais notoirement hétérogènes, mêlant données structurées et texte libre au jargon technique dense. Leur analyse manuelle, lente et subjective, limite considérablement la capacité à anticiper les risques et à optimiser les performances. Pour répondre à cette problématique, ce projet vise à concevoir et développer une solution logicielle innovante capable de transformer cette masse de données brutes en une source de connaissance proactive. L'originalité de notre approche réside dans la conception d'un système hybride qui combine deux méthodologies d'intelligence artificielle de pointe : d'une part, un modèle d'apprentissage profond basé sur les réseaux de neurones récurrents (LSTM) pour analyser les données séquentielles et prédire les risques à partir des tendances quantifiables ; d'autre part, un système d'IA générative exploitant un grand modèle de langage (LLM) au sein d'une architecture de Génération Augmentée par Récupération (RAG) pour interpréter le contexte textuel et fournir une analyse de risque justifiée. La finalité de ce travail est une application web complète, construite avec Flask et Plotly.js, qui sert de tableau de bord interactif, intègre les prédictions de nos modèles, et alerte les ingénieurs en temps réel. Ce rapport détaillera l'état de l'art sous-jacent, la conception et l'implémentation de notre solution, avant de présenter et d'analyser les résultats obtenus et de discuter des perspectives futures.

2.2 Le forage pétrolier et gazier : Principes et défis

Le forage pétrolier et gazier constitue l'ensemble des opérations techniques complexes et à forte intensité capitalistique visant à créer un puits, ou un trou, traversant les formations géologiques du sous-sol jusqu'à un réservoir d'hydrocarbures. Cette phase est l'une des plus critiques et des plus coûteuses dans le cycle de vie d'un champ pétrolier, que ce soit en phase d'exploration, pour confirmer la présence et évaluer le potentiel d'un gisement, ou en phase de développement, pour mettre en production les réserves découvertes. Les puits peuvent être forés verticalement, déviés, ou s'étendre horizontalement sur plusieurs kilomètres pour maximiser le contact avec la roche réservoir. L'objectif principal est d'atteindre la cible géologique de manière sûre, efficace et économique, tout en préservant l'intégrité du puits et en minimisant l'impact environnemental.

2.2.1 Principes Fondamentaux et Composants Clés

Pour atteindre ces objectifs, les opérations de forage mobilisent des équipements sophistiqués et des technologies de pointe, orchestrés sur une plateforme de forage (*rig*), qu'elle soit terrestre (*onshore*) ou maritime (*offshore*). La figure 2.1 illustre les composants majeurs d'une telle installation.

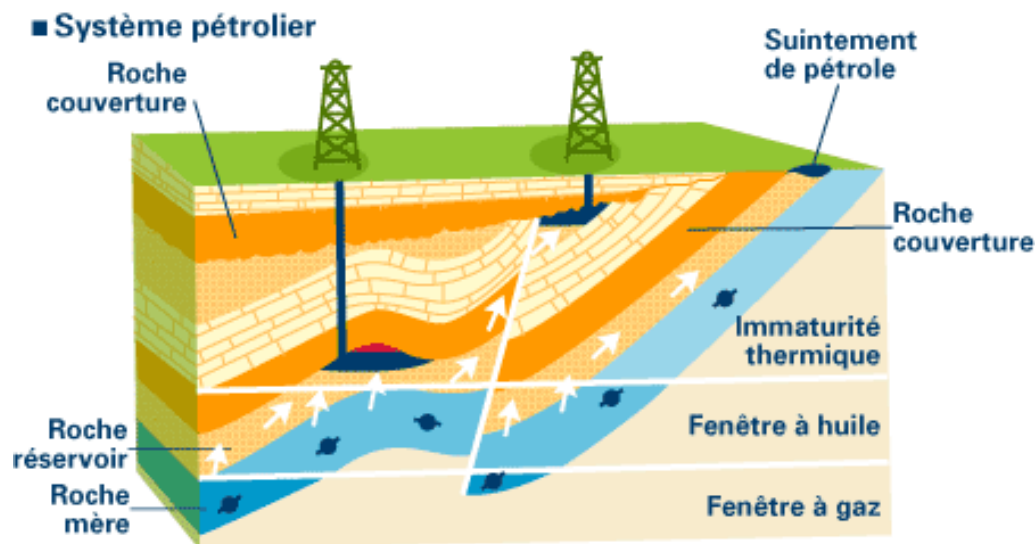


FIGURE 2.1 – Schéma Détaillé d'une Installation de Forage et de ses Composants Principaux.

Les composants essentiels de cet écosystème sont :

- **Le système de levage (Hoisting System)** : Composé du derrick (la tour de forage métallique), du treuil (*drawworks*) et d'un système de poulies (moufle fixe et mobile), il permet de lever et de descendre la garniture de forage, qui peut peser plusieurs centaines de tonnes.
- **Le système de rotation (Rotating System)** : Il fournit l'énergie rotative au trépan. Historiquement, une table de rotation sur le plancher de la plateforme entraînait la garniture. Aujourd'hui, la plupart des plateformes modernes utilisent un **top drive** [8], un moteur puissant suspendu dans le derrick qui fait tourner la garniture directement par le haut, offrant une meilleure efficacité et un meilleur contrôle.
- **La garniture de forage (Drill String)** : C'est l'ensemble des tiges de forage (*drill pipes*), des masses-tiges (*drill collars*) et des outils spécialisés (regroupés sous le nom de BHA - *Bottom Hole Assembly*) qui relie la surface au trépan. Elle transmet la rotation et le poids au trépan et sert de conduit pour la boue de forage.
- **Le trépan (Drill Bit)** : Situé à l'extrémité de la garniture, c'est l'outil qui désagrège la roche. Il existe de nombreux types de trépan (PDC, tricônes, etc.), choisis en fonction de la dureté et de l'abrasivité des formations à forer.
- **Le système de circulation (Circulating System)** : Il est responsable de la gestion de la boue de forage. Il comprend les pompes à boue, les bassins de stockage (*mud pits*), et les équipements de traitement des déblais (tamis vibrants, hydrocyclones) qui nettoient la boue avant sa réinjection dans le puits.
- **Le système de contrôle du puits (Well Control System)** : C'est un ensemble de barrières de sécurité, dont le composant le plus visible est le bloc d'obturateurs de puits (*BOP - Blowout Preventer*) [9]. Le BOP est un ensemble de vannes à haute pression capable de fermer le puits en cas d'afflux de fluides non contrôlé, prévenant ainsi une éruption.

2.2.1.a Le Rôle Multifonctionnel de la Boue de Forage

Au cœur du processus de forage, la boue de forage (*drilling mud*) est un fluide technique dont le rôle est loin de se limiter à la simple lubrification. Sa composition (à base

d'eau, d'huile ou de produits synthétiques, avec de nombreux additifs) est constamment ajustée pour remplir plusieurs fonctions critiques simultanément :

- **Nettoyage du trou** : La boue, injectée dans la garniture et sortant par les buses du trépan, remonte les déblais de roche (*cuttings*) vers la surface.
- **Refroidissement et lubrification** : Elle dissipe la chaleur intense générée par la friction du trépan contre la roche et lubrifie la garniture de forage.
- **Contrôle de la pression** : La colonne de boue exerce une pression hydrostatique sur les parois du puits. Cette pression doit être soigneusement gérée pour être supérieure à la pression des fluides contenus dans les formations (pression de pore) afin d'éviter leur intrusion dans le puits, mais inférieure à la pression de fracturation de la roche pour ne pas créer de pertes de fluide.
- **Maintien de la stabilité des parois** : Le dépôt d'une fine couche imperméable (*filter cake*) par la boue sur les parois du trou aide à prévenir leur effondrement.
- **Transmission d'informations** : Les outils de mesure en fond de trou (MWD/LWD) utilisent souvent la boue comme milieu pour transmettre des données à la surface par impulsions de pression (*mud-pulse telemetry*).

La gestion des propriétés de la boue est donc une science en soi et un élément central du succès des opérations [10].

2.2.2 Les Défis Inhérents aux Opérations de Forage

Le forage est une entreprise semée de défis, où l'incertitude est la norme. Ces défis peuvent être regroupés en plusieurs catégories interdépendantes.

2.2.2.a Défis Géologiques

Le sous-sol est un environnement complexe et imprévisible. Les principaux défis géologiques incluent :

- **Pressions de formation anormales** : Les formations peuvent être en surpression (*overpressured*), où la pression des fluides est supérieure à la normale hydrostatique, créant un risque élevé de *kick* [27]. Inversement, des zones de sous-pression ou des fractures naturelles peuvent entraîner des pertes de circulation massives. La gestion de la densité de la boue pour naviguer dans cette "fenêtre de pression" étroite est un défi constant.
- **Lithologies complexes** : La succession de couches de roches dures (calcaire, grès cimenté) et tendres (argiles) rend difficile le choix du trépan et des paramètres de forage optimaux. Les formations abrasives usent prématurément les équipements, tandis que les argiles réactives (*reactive shales*) peuvent gonfler au contact de la boue à base d'eau, provoquant un resserrement du trou et un risque de coincement.
- **Instabilités structurales** : La présence de failles non cartographiées, de dômes de sel ou de formations peu consolidées peut compromettre la stabilité mécanique du puits et créer des trajectoires de forage complexes.

2.2.2.b Défis Techniques et Opérationnels

Même dans des conditions géologiques connues, les défis techniques sont nombreux :

- **Intégrité et stabilité du puits** : Maintenir le trou "ouvert" et calibré est essentiel. L'effondrement des parois (*breakouts*, *washouts*) peut entraîner le coincement de la garniture.
- **Forage directionnel et horizontal** : Guider le trépan avec précision sur plusieurs kilomètres pour atteindre une cible de quelques mètres de large requiert des technologies de mesure et de guidage sophistiquées (MWD/LWD, moteurs de fond) et une gestion fine de la trajectoire pour éviter une tortuosité excessive qui augmente les frictions.
- **Gestion des vibrations de la garniture** : Les vibrations axiales, latérales ou de torsion peuvent endommager le trépan et les outils de fond, réduire l'efficacité du forage et mener à des défaillances par fatigue.
- **Fiabilité des équipements** : Les composants sont soumis à des contraintes extrêmes (pression, température, chocs, corrosion). La défaillance d'un seul élément, de la pompe de surface au moteur de fond, peut entraîner des heures, voire des jours, de temps non productif (NPT) [7].

2.2.2.c Défis Économiques et Environnementaux

Enfin, les opérations sont encadrées par des contraintes économiques et réglementaires fortes :

- **Maîtrise des coûts** : Le coût journalier d'une plateforme de forage (*spread rate*) peut atteindre plusieurs centaines de milliers de dollars. Chaque heure de NPT a un impact financier direct et significatif. L'optimisation des performances et la minimisation des temps morts sont donc des objectifs économiques primordiaux.
- **Sécurité du personnel** : Le forage reste une activité à haut risque. La sécurité des équipes est la priorité absolue et dicte de nombreuses procédures opérationnelles.
- **Protection de l'environnement** : La gestion des rejets (déblais de forage, fluides usagés) est strictement réglementée. La prévention des fuites et des déversements, en particulier en milieu marin, est une contrainte majeure qui influence la conception des puits et les procédures de contrôle.

La maîtrise de l'ensemble de ces défis passe par une planification rigoureuse, une exécution soignée, et surtout, une surveillance constante et une interprétation intelligente des données disponibles, au premier rang desquelles figurent les rapports journaliers de forage.

2.2.3 Processus de forage et opérations clés

Le forage d'un puits est un processus séquentiel et itératif, méticuleusement planifié pour naviguer en toute sécurité à travers des milliers de mètres de formations géologiques. Après la préparation du site, le forage débute par une section de surface de grand diamètre (par exemple, 36 pouces). Une fois cette section forée, elle est sécurisée par la pose et la cimentation d'un premier tubage en acier, appelé le conducteur (*conductor casing*). Ce tubage sert de fondation stable pour les équipements de surface, notamment le BOP, et prévient l'érosion des formations superficielles, souvent peu consolidées et contenant des aquifères d'eau douce qu'il faut protéger.

Le forage se poursuit ensuite par sections successives, de diamètres décroissants (e.g., 26", 17"1/2, 12"1/4, 8"1/2). Chaque section est forée jusqu'à une profondeur déterminée, choisie en fonction des contraintes géologiques (pression, stabilité). Un nouveau tubage, de diamètre inférieur au précédent, est alors descendu et cimenté en place. Cette architecture télescopique du puits est fondamentale pour :

- Isoler les zones à problèmes (formations instables, zones de pertes).
- Permettre l'utilisation de densités de boue de forage progressivement plus élevées pour contrôler des pressions de formation croissantes avec la profondeur.
- Assurer l'intégrité structurale du puits sur le long terme.

Ce processus itératif est composé de plusieurs opérations clés, chacune étant une source potentielle de données et de risques, et donc un objet d'analyse pour notre système.

2.2.3.a Le forage rotatif (Rotary Drilling)

Le forage rotatif est l'opération "productive" par excellence, durant laquelle le puits est approfondi. Le trépan, à l'extrémité de la garniture de forage, est mis en rotation par le top drive tandis qu'un poids contrôlé (Poids sur l'Outil - WOB, Weight On Bit) est appliqué pour permettre aux éléments coupants du trépan de pénétrer et de désagréger la roche. Simultanément, la boue de forage est pompée à haute pression à travers la garniture, s'échappe par des buses sur le trépan, et remonte les déblais de forage (*cuttings*) vers la surface via l'espace annulaire. Le succès de cette opération dépend d'un équilibre délicat

entre de nombreux paramètres (WOB, vitesse de rotation, débit de la boue) pour maximiser la vitesse de pénétration (ROP - Rate of Penetration) tout en minimisant l'usure de l'outil et en maintenant la stabilité du puits. Les données issues de cette phase (paramètres mécaniques, hydrauliques, et observations sur les cuttings) sont parmi les plus riches pour l'évaluation des conditions de forage.



FIGURE 2.2 – Illustration du processus de forage rotatif en fond de trou.

2.2.3.b La manœuvre de la garniture de forage (Tripping)

La manœuvre est l'action de remonter (*trip out*) ou de descendre (*trip in*) la garniture de forage. Bien qu'elle ne contribue pas directement à l'avancement du puits, c'est une opération essentielle, fréquente et à haut risque. Les raisons d'une manœuvre sont multiples : changer un trépan usé, remplacer un outil défaillant dans le BHA, ou pour réaliser des opérations spécifiques comme les diagraphies ou la pose de tubage.

Durant la manœuvre, le contrôle de la pression dans le puits est un défi majeur. Lors de la remontée, le volume de l'acier retiré doit être remplacé par un volume équivalent de boue pour maintenir la pression hydrostatique constante. De plus, un mouvement trop

rapide de la garniture peut créer un effet de piston (*swab effect*), aspirant les fluides de la formation dans le puits (un *kick*). Inversement, une descente trop rapide peut créer une surpression (*surge effect*) qui peut fracturer la formation et causer des pertes de circulation. Le suivi attentif des volumes de boue et des pressions pendant les manœuvres est donc critique pour la sécurité.

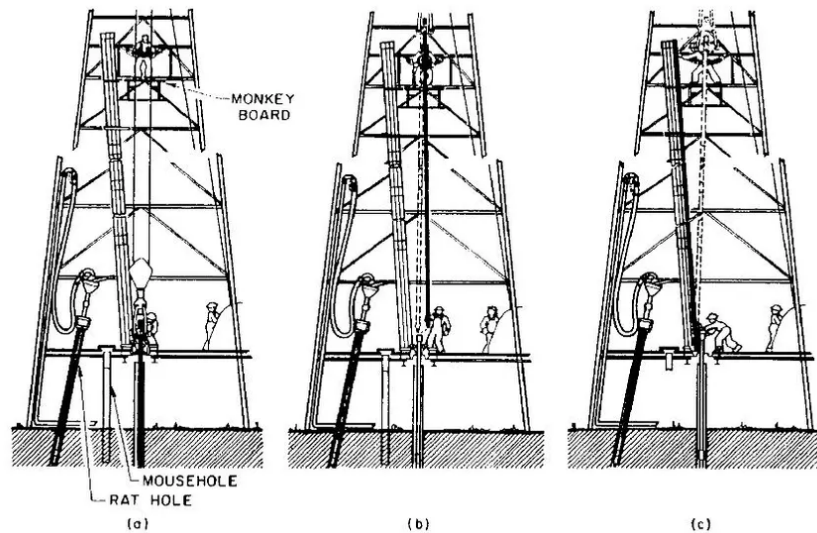


FIGURE 2.3 – Opération de manœuvre : les tiges de forage sont assemblées ou désassemblées et stockées verticalement dans le derrick.

2.2.3.c La pose et la cimentation des tubages (Casing and Cementing)

Une fois qu'une section de puits est forée, la pose et la cimentation d'un tubage sont nécessaires pour sécuriser cette section avant de forer plus profondément. L'opération se déroule en plusieurs étapes : la garniture de forage est entièrement remontée, puis le tubage en acier (une série de tubes vissés) est descendu dans le trou. Ensuite, un laitier de ciment est pompé à l'intérieur du tubage, poussé vers le bas par un fluide de déplacement (souvent de la boue). Arrivé en bas, le ciment remonte dans l'espace annulaire entre le tubage et la formation, déplaçant la boue qui s'y trouvait. Une fois le ciment en place, on attend sa prise (WOC - Waiting On Cement).

La qualité de cette opération est fondamentale pour l'intégrité du puits sur toute sa durée de vie. Un bon travail de cimentation doit assurer une isolation hydraulique parfaite pour empêcher les fluides (gaz, pétrole, eau) de migrer d'une formation à une autre, et fournir un support structurel solide au tubage. Un défaut de cimentation peut avoir des conséquences graves et coûteuses, nécessitant des réparations complexes.

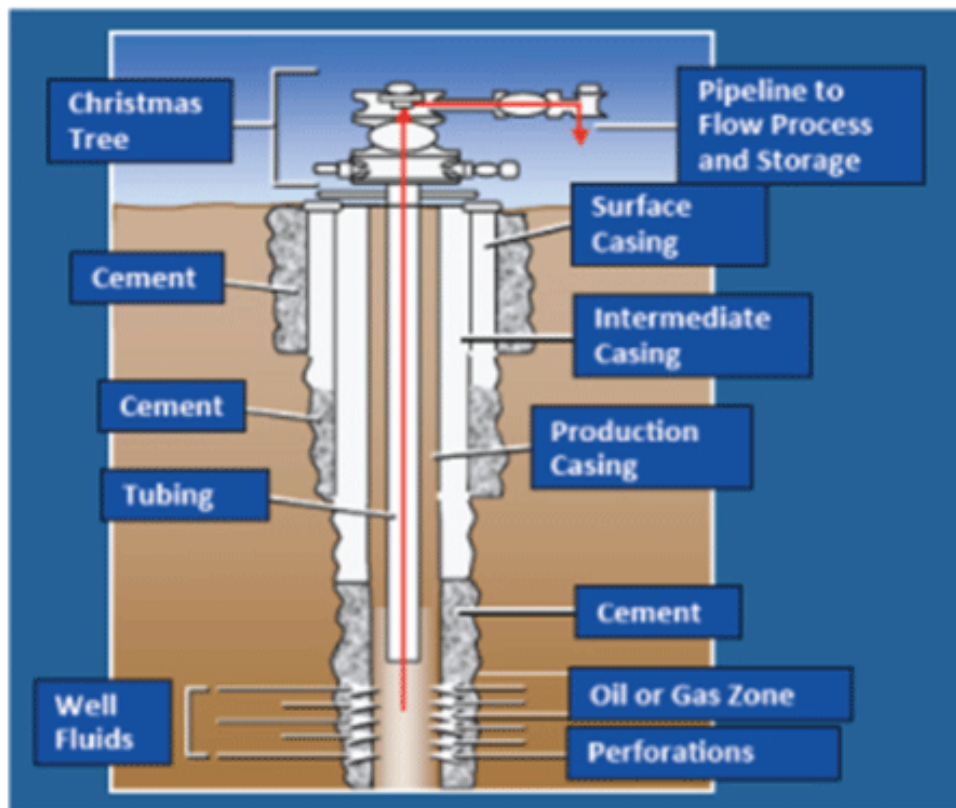


FIGURE 2.4 – Étapes de la cimentation d'un tubage : (a) Descente du tubage, (b) Pompage du ciment dans l'annulaire.

2.2.3.d Les diagraphies (Logging)

Les diagraphies consistent à enregistrer les propriétés physiques des formations rocheuses traversées. Ces mesures sont cruciales pour les géologues et les ingénieurs de réservoir afin d'identifier les zones contenant des hydrocarbures, d'évaluer leur quantité et leur productibilité, et de comprendre la géologie du sous-sol. Il existe deux principales méthodes :

- **Diagraphies au câble (Wireline Logging) :** Après avoir foré une section et remonté la garniture, des outils de mesure sophistiqués sont descendus dans le trou au bout d'un câble électrique. En remontant, ces outils mesurent en continu des propriétés comme la radioactivité naturelle (gamma ray, pour identifier les argiles), la densité, la porosité (par des mesures neutroniques ou acoustiques), et la résistivité électrique (pour distinguer l'eau salée des hydrocarbures).
- **Mesures en cours de forage (LWD/MWD) :** Des capteurs similaires sont intégrés directement dans la garniture de forage, près du trépan. Ils transmettent les données à la surface en temps réel (ou quasi-réel), offrant une vision immédiate des formations traversées. Cela permet de prendre des décisions de navigation (forage directionnel) et d'ajuster les plans en fonction de la géologie réellement rencontrée.

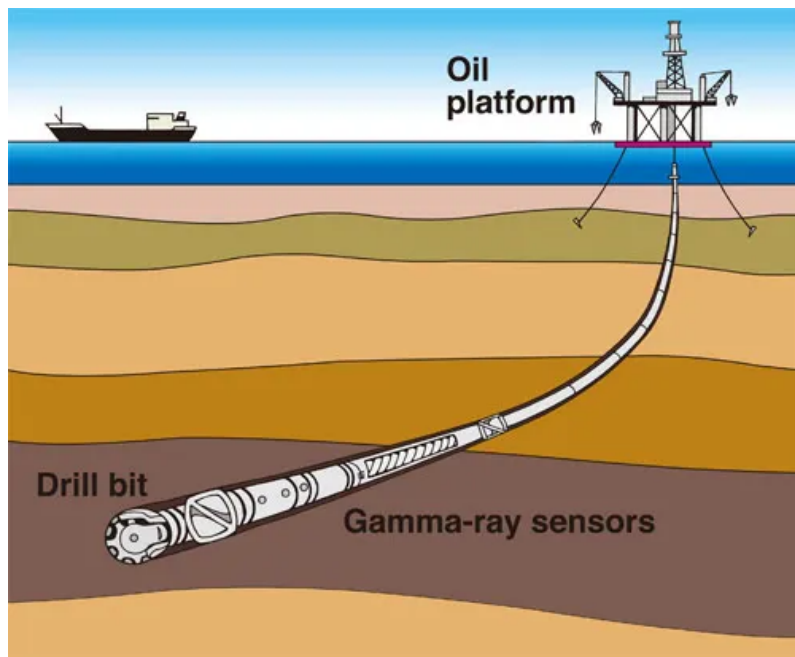


FIGURE 2.5 – Principe de la diagraphie au câble (Wireline Logging).

2.2.3.e La circulation et le traitement de la boue de forage

Cette opération est continue et se déroule en parallèle de la plupart des autres activités. Le système de circulation de surface forme une boucle fermée : la boue est préparée dans des bassins, aspirée par de puissantes pompes, envoyée dans la garniture de forage, puis elle remonte du puits chargée de déblais. À son retour en surface, elle passe par une série d'équipements de contrôle des solides (tamis vibrants, désableurs, désilteurs, centrifugeuses) qui séparent les déblais solides du fluide. La boue "propre" est ensuite renvoyée dans les bassins pour être réutilisée. Des ingénieurs spécialisés (ingénieurs fluides ou *mud engineers*) surveillent en permanence les propriétés de la boue (densité, viscosité, pH, etc.) et ajoutent des produits chimiques pour les maintenir dans les spécifications requises par le programme de forage.

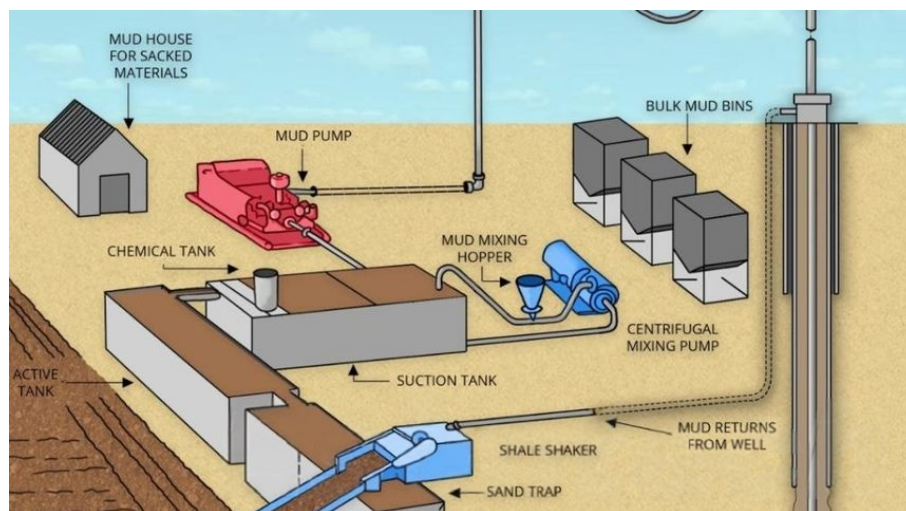


FIGURE 2.6 – Schéma du circuit de circulation de la boue en surface (Système de Contrôle des Solides).

2.2.3.f Les essais de puits (Well Testing)

Une fois le réservoir cible atteint et le puits sécurisé, des essais de puits peuvent être réalisés pour évaluer le potentiel de production de la formation. Cette opération consiste à laisser le puits produire des fluides (pétrole, gaz, eau) de manière contrôlée pendant une période donnée. Les fluides sont dirigés vers un équipement de test en surface, qui comprend généralement un séparateur pour mesurer les débits respectifs de chaque phase. Simultanément, des capteurs de pression et de température en fond de trou enregistrent l'évolution de la pression du réservoir.

L'analyse de ces données de débit et de pression permet de déterminer des caractéristiques clés du réservoir, telles que sa perméabilité, l'étendue de la zone drainée par le puits, et la présence éventuelle de barrières ou de limites. Les essais de puits sont une source d'information directe et inestimable sur la capacité du réservoir à produire des hydrocarbures, et leurs résultats sont décisifs pour la décision de développer ou non un champ.

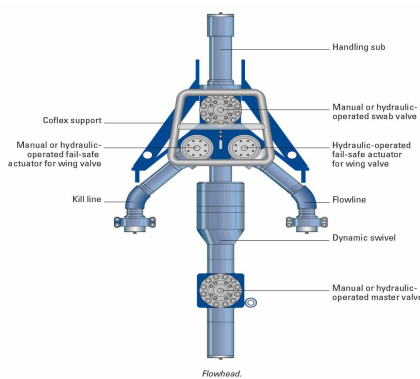


FIGURE 2.7 – Configuration typique d'un équipement de test de puits en surface(flowhead-max).



FIGURE 2.8 – Configuration typique d'un autre équipement de test de puits en surface(ssv-combo).

2.2.4 Rapports journaliers de forage (DDR) : Structure et importance

Au cœur du suivi des opérations se trouve le rapport journalier de forage (DDR). Ce document est produit quotidiennement et constitue un enregistrement chronologique et détaillé de toutes les activités des dernières 24 heures.

2.2.4.a Structure typique d'un DDR

Un DDR n'est pas un simple texte monolithique. Il est semi-structuré, combinant des tableaux de données numériques et des champs de texte libre. Cette hétérogénéité est à la fois sa force (richesse d'information) et sa faiblesse (difficulté d'analyse automatique). Une structure typique comprend :

- **En-tête** : Informations générales (nom du puits, date, nom de la plateforme, opérateur).
- **Tableau des opérations** : Un journal chronologique des activités (e.g., de 08 :00 à 12 :00 : forage de la section 12"1/4 de 3200m à 3250m). Chaque ligne contient une heure de début, une heure de fin, une description textuelle de l'activité et des remarques.

- **Tableau des propriétés de la boue** : Paramètres clés de la boue de forage (densité, viscosité, pH, etc.) mesurés à différents moments et points d'échantillonnage.
- **Tableau des équipements** : Informations sur la garniture de forage, le trépan en service, etc.
- **Résumé textuel des activités** : Un paragraphe narratif résumant les faits marquants des dernières 24 heures.
- **Résumé textuel des activités planifiées** : Les objectifs pour les prochaines 24 heures.
- **Sections diverses** : Consommation de matériaux, personnel à bord, conditions météorologiques, incidents de sécurité, etc.

Summary report

Wellbore: 15/9-19 A

Period: 1997-09-04 00:00 - 1997-09-05 00:00

Status:	normal
Report creation time:	2018-05-03 13:53
Report number:	43
Days Ahead/Behind (+/-):	
Operator:	StatOil
Rig Name:	BYFORD DOLPHIN
Drilling contractor:	
Spud Date:	1997-07-25 00:00
Wellbore type:	
Elevation RKB-MSL (m):	25
Water depth MSL (m):	84
Tight well:	Y
HPHT:	Y
Temperature (°):	
Pressure (°):	
Date Well Complete:	1997-08-30

Dis Drilled (m):	-999.99
Penetration rate (m/h):	-999.99
Hole Dia ("):	
Pressure Test Type:	
Formation strength ("):	
Dia Last Casing ("):	

Depth at Kick Off mMD:	2178
Depth at Kick Off mTVD:	
Depth mMd:	4131
Depth mTVD:	3319
Plug Back Depth mMD:	4019
Depth at formation strength mMD:	
Depth At Formation Strength mTVD:	
Depth At Last Casing mMD:	4643
Depth At Last Casing mTVD:	

Summary of activities (24 Hours)

RIG OPERATIONS SUSPENDED DUE TO OFS STRIKE.

Summary of planned activities (24 Hours)

CONTINUE WAITING DUE TO OFS STRIKE.

Operations

Start time	End time	End Depth mMD	Main - Sub Activity	State	Remark
00:00	06:00	0	interruption -- other	ok	OPERATIONS SUSPENDED DUE TO OFS STRIKE.
06:00	00:00	0	interruption -- other	ok	OPERATIONS SUSPENDED DUE TO OFS STRIKE.

FIGURE 2.9 – Structure Mixte d'un Rapport Journalier de Forage (DDR) Typique.

2.2.4.b Importance stratégique du DDR : Le Système Nerveux des Opérations

Loin d'être un simple exercice de reporting administratif, le rapport journalier de forage est le système nerveux central des opérations de forage. Il est le document qui connecte le "cerveau" de l'opération (les équipes d'ingénierie, de géologie et de management au bureau) au "corps" (l'équipe et les équipements sur la plateforme). Son importance stratégique est multiple et se manifeste à travers plusieurs fonctions interdépendantes, qui justifient l'investissement dans des technologies avancées pour son exploitation.

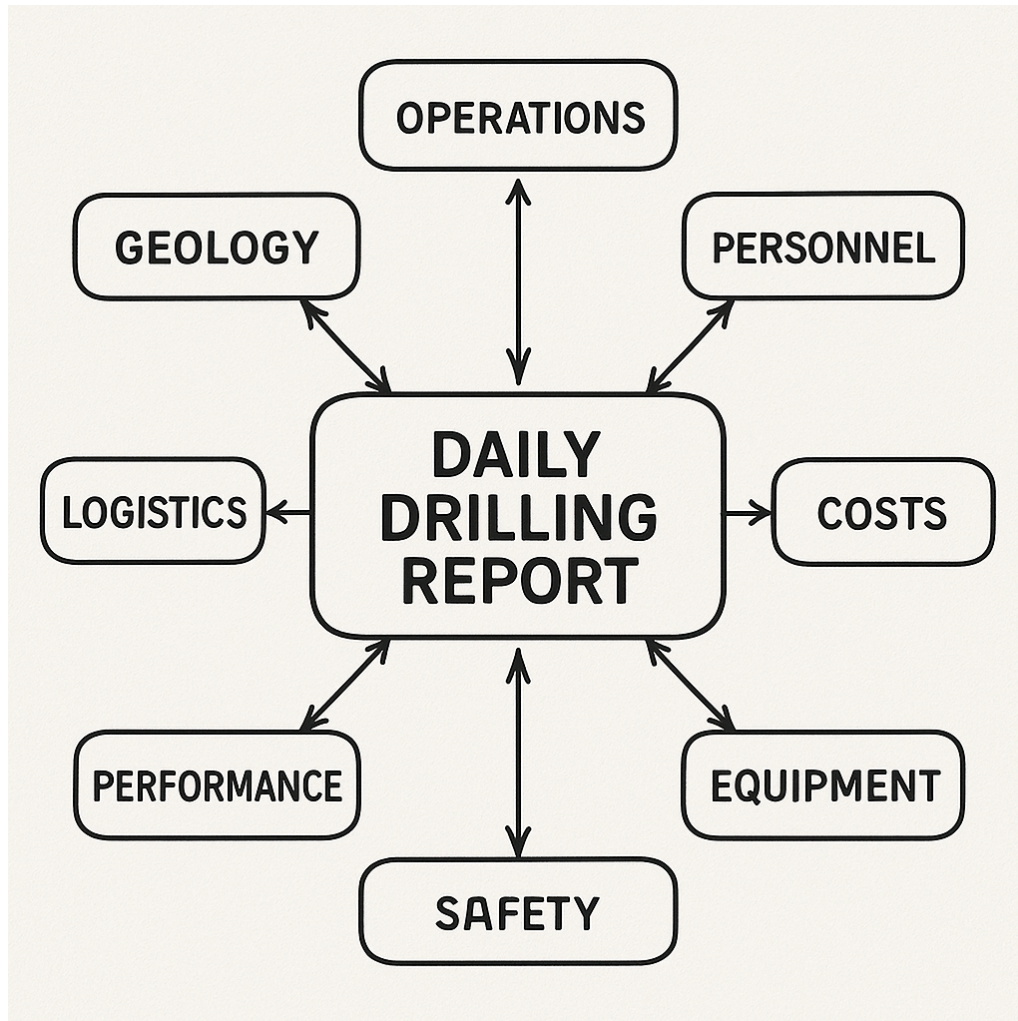


FIGURE 2.10 – Flux d'Information et Rôle Central du DDR comme Pivot entre les Parties Prenantes des Opérations de Forage.

1. Communication Opérationnelle et Prise de Décision Quotidienne : Le DDR est, avant tout, un outil de communication. Il assure que toutes les parties prenantes, qu'elles soient sur site, au siège social ou chez un partenaire, partagent une vision commune et actualisée de l'état du puits. Un ingénieur de forage au bureau, en lisant le DDR du matin, doit pouvoir prendre des décisions éclairées pour la journée à venir. Par exemple, si le rapport mentionne une augmentation du couple de rotation et des déblais de forage de forme allongée ("splintery cuttings"), l'ingénieur peut, en collaboration avec l'équipe sur site, décider de modifier les propriétés de la boue ou d'ajuster les paramètres de forage pour prévenir un risque de coincement. Sans cette information claire et rapide, des décisions

critiques pourraient être retardées, augmentant le risque d'un incident coûteux.

2. Suivi de Performance et Contrôle des Coûts : Le forage est une course contre la montre où chaque heure compte. Le DDR est l'outil principal pour mesurer la performance par rapport au plan. En consignait méticuleusement les durées de chaque activité, il permet de calculer le temps non productif (NPT). L'analyse du NPT est une discipline en soi : un NPT élevé peut indiquer des problèmes récurrents liés aux équipements, à la formation géologique, ou aux procédures. En quantifiant le temps passé sur chaque type d'opération (forage, manœuvre, attente sur ciment, etc.) et en le comparant au "temps idéal" (courbe AFE - *Authorization for Expenditure*), les managers peuvent évaluer l'efficacité des opérations et contrôler les coûts qui peuvent rapidement dériver.

3. Documentation Légale, Contractuelle et Réglementaire : Le DDR est un document officiel qui fait foi. Dans le cadre des relations complexes entre l'opérateur pétrolier et ses nombreux contractants (de forage, de cimentation, de diagrapie, etc.), le DDR sert de preuve pour la facturation des services et le respect des clauses contractuelles. En cas d'incident majeur (perte de contrôle du puits, accident de personne, pollution), les DDR deviennent des pièces à conviction essentielles lors des enquêtes menées par les autorités réglementaires ou les compagnies d'assurance. Une documentation précise, complète et non ambiguë est donc une protection juridique indispensable.

4. Mémoire de l'Entreprise et Amélioration Continue (*Lessons Learned*) : Un puits foré n'est pas une fin en soi ; c'est une source d'apprentissage pour le suivant. La capitalisation de l'expérience est un facteur clé de succès dans l'industrie. Les DDR, une fois archivés, constituent la mémoire technique de l'entreprise. L'analyse agrégée des DDR de dizaines ou de centaines de puits forés dans un même bassin sédimentaire permet de :

- **Identifier des tendances :** Corréler certains types de NPT avec des formations géologiques spécifiques ou des pratiques opérationnelles.
- **Établir des benchmarks :** Définir des indicateurs de performance clés (KPIs) réalistes pour la planification des futurs puits.
- **Optimiser les conceptions :** Si l'analyse montre que l'utilisation d'un certain type de trépan ou de boue a systématiquement conduit à de meilleures performances dans une formation donnée, cette connaissance sera intégrée dans la conception des prochains puits.
- **Valider ou affiner les modèles géologiques :** Les observations consignées dans les DDR (e.g., "pertes de circulation à 2500m", "présence de gaz détectée") sont des points de calage précieux pour les modèles géologiques et géomécaniques.

Cependant, pour que cette mémoire soit exploitable, l'information doit être extraite et structurée, une tâche herculéenne si elle est réalisée manuellement.

5. Alimentation des Modèles d'Intelligence Artificielle : C'est le rôle le plus moderne et le plus prometteur du DDR. La transformation numérique de l'industrie pétrolière s'appuie sur le développement de modèles prédictifs basés sur les données. Les DDR sont la source primaire pour alimenter ces modèles. Les données structurées (paramètres, durées) et, surtout, les informations extraites du texte (types d'événements, descriptions de problèmes, contextes opérationnels) servent de "carburant" pour entraîner les algorithmes d'apprentissage automatique, comme les modèles LSTM et les LLM que nous utilisons

dans ce projet. Un ensemble de données riche, propre et bien contextualisé, issu de l'analyse systématique des DDR, est la condition sine qua non pour développer des modèles de prédiction des risques qui soient fiables et performants [51, 3].

En somme, le DDR est bien plus qu'un simple rapport. C'est un document stratégique qui se situe à l'intersection de la technique, de l'économie, du juridique et de l'innovation. La capacité à en extraire la pleine valeur de manière rapide et automatisée est un levier de performance majeur. C'est précisément l'ambition de notre projet : transformer cette mosaïque complexe d'informations quotidiennes en une source de connaissance proactive et d'aide à la décision intelligente.

2.3 Extraction d'informations et analyse de données textuelles

Les DDR contiennent une part substantielle d'informations sous forme textuelle non structurée, particulièrement dans les résumés et les remarques. L'Extraction d'Informations (IE) est la discipline visant à extraire automatiquement des informations structurées (entités, relations, événements) à partir de ces textes, les rendant ainsi exploitables par des systèmes informatiques.

2.3.1 Techniques de Traitement Automatique du Langage (TAL/NLP)

Le Traitement Automatique du Langage (TAL), ou *Natural Language Processing* (NLP) en anglais, est une branche de l'intelligence artificielle qui vise à doter les ordinateurs de la capacité de comprendre, d'interpréter et de manipuler le langage humain. Face au volume de texte non structuré présent dans les rapports journaliers de forage, les techniques de TAL sont indispensables pour passer d'une simple lecture à une véritable extraction de connaissances actionnables. L'objectif est de construire un pipeline de traitement capable de déconstruire le langage technique des DDR pour en extraire les informations clés de manière structurée. Ce processus est généralement organisé en plusieurs étapes fondamentales, comme illustré par la Figure 2.11.

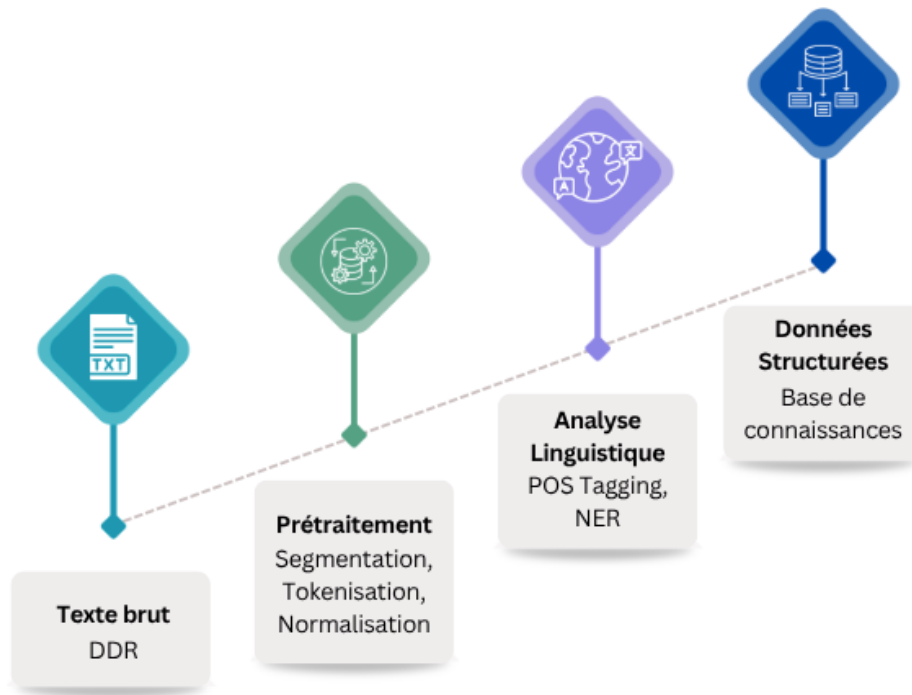


FIGURE 2.11 – Exemple de pipeline de traitement pour l'extraction d'informations à partir de texte non structuré, illustrant la progression de la tokenisation à l'extraction de relations [46].

2.3.1.a Étape 1 : Segmentation et Tokenisation

La première étape, fondamentale, est la **segmentation**. Elle consiste à décomposer le texte brut, qui est une simple chaîne de caractères pour l'ordinateur, en unités linguistiques discrètes.

- **Segmentation en phrases (*Sentence Splitting*)** : Le texte est d'abord divisé en phrases individuelles. Cette étape est cruciale car la phrase constitue généralement une unité de sens complète. Dans les DDR, où la ponctuation peut être erratique, cette tâche n'est pas toujours triviale. Une phrase comme "RIH with new BHA. Encountered high torque at 3100m." contient deux informations distinctes qui doivent être traitées séparément.
- **Tokenisation (*Tokenization*)** : Chaque phrase est ensuite décomposée en une séquence de "tokens". Un token est le plus souvent un mot, mais peut aussi être un signe de ponctuation ou un symbole. Par exemple, la phrase "Pumped 20 m3 hi-vis pill." serait tokenisée en ['Pumped', '20', 'm3', 'hi-vis', 'pill', '.']. La manière de gérer les mots composés ('hi-vis'), les unités ('m3') ou les acronymes est un premier défi spécifique au domaine.

2.3.1.b Étape 2 : Normalisation et Nettoyage du Texte

Une fois le texte tokenisé, l'étape de **normalisation** vise à réduire la complexité et la "bruit" pour faciliter l'analyse sémantique ultérieure.

- **Mise en minuscules (*Lowercasing*)** : Convertir tout le texte en minuscules permet de traiter "BOP", "bop" et "Bop" comme le même token. C'est une étape simple mais efficace, bien qu'elle puisse parfois faire perdre une information (par exemple, la distinction entre le nom commun "brown" et le nom propre de la formation "Brown Shale").
- **Suppression des Mots Vides (*Stop Word Removal*)** : Les mots très fréquents mais porteurs de peu de sens sémantique (comme "le", "la", "de", "un", "est") sont souvent supprimés pour ne conserver que les mots-clés les plus importants. La liste des mots vides doit parfois être adaptée au domaine technique.
- **Racinisation (*Stemming*) et Lemmatisation** : Ces deux techniques visent à regrouper les différentes formes d'un mot. La **racinisation** est une méthode heuristique qui consiste à tronquer la fin des mots pour ne garder que leur racine (e.g., "drilling", "drilled", "driller" -> "drill"). Elle est rapide mais peut parfois produire des racines qui ne sont pas de vrais mots. La **lemmatisation** est une approche plus sophistiquée qui utilise un dictionnaire et une analyse morphologique pour ramener un mot à sa forme canonique, ou lemme (e.g., "was", "is", "are" -> "be"). Pour le français, "forant", "foré", "forons" seraient lemmatisés en "forer". La lemmatisation est généralement préférée pour des analyses sémantiques fines.

2.3.1.c Étape 3 : Analyse Morpho-Syntaxique et Sémantique

Cette étape vise à comprendre la structure grammaticale et le sens des mots dans leur contexte.

- **Étiquetage Morpho-syntaxique (Part-of-Speech - POS Tagging)** : Chaque token se voit attribuer sa catégorie grammaticale (nom, verbe, adjectif, etc.). Savoir que "pumped" est un verbe et "mud" est un nom est une information de base pour comprendre qu'une action a été effectuée sur un objet.
- **Plongements de Mots (*Word Embeddings*)** : Les approches modernes de TAL ne se contentent plus de traiter les mots comme des symboles discrets. Elles les représentent par des vecteurs denses dans un espace de grande dimension, appelés *embeddings*. Ces vecteurs capturent les relations sémantiques entre les mots : des mots au sens similaire (e.g., "pression", "contrainte") auront des vecteurs proches dans cet espace. Des modèles comme Word2Vec, GloVe, et surtout les mécanismes d'attention des modèles Transformers (comme BERT ou Llama) sont utilisés pour générer ces représentations contextuelles, où le vecteur d'un mot dépend de la phrase dans laquelle il apparaît. C'est une avancée majeure qui permet une compréhension beaucoup plus nuancée du texte [36, 15].

2.3.1.d Étape 4 : Extraction d'Informations Structurées

C'est le cœur de la transformation du texte en données exploitables.

- **Reconnaissance d'Entités Nommées (NER)** : C'est la tâche qui consiste à localiser et à classifier des "entités" prédéfinies dans le texte. Pour notre projet, c'est l'une des tâches les plus importantes. Un système de NER entraîné sur des DDR apprendrait à identifier :
 - **ÉQUIPEMENT** : 'BOP', 'trépan PDC', 'top drive', 'mud motor'.
 - **OPÉRATION** : 'forage', 'cimentation', 'manœuvre', 'RIH' (Run In Hole).

- **PROBLÈME** : ‘coincement de garniture’, ‘perte de circulation’, ‘panne moteur’, ‘high torque’.
- **FORMATION GÉOLOGIQUE** : ‘Formation Crétacé’, ‘Dôme de sel’, ‘shale’.
- **PARAMÈTRE** : ‘densité de boue’, ‘ROP’, ‘pression’.
- **VALEUR** : ‘1.45 g/cm³’, ‘15 m/hr’, ‘2500 psi’.
- **Extraction de Relations** : Une fois les entités identifiées, cette tâche vise à trouver les liens qui les unissent. Par exemple, à partir de la phrase "High torque observed while drilling through the shale formation", un système pourrait extraire la relation : ‘a_pour_symptôme(drilling in shale, high torque)’. L’extraction de ces relations est fondamentale pour construire un graphe de connaissances des événements de forage et de leurs causes potentielles.
- **Classification de Texte** : Cette technique peut être appliquée à différents niveaux. On peut classer une remarque entière comme étant liée à un ‘PROBLÈME’ ou à une ‘OPÉRATION_STANDARD’. On peut également classer un rapport entier comme contenant un risque élevé ou non, bien que cette approche soit moins fine que l’analyse événementielle.

L’ensemble de ces techniques, des plus basiques aux plus avancées, forme une chaîne de traitement qui permet de transformer le texte brut et ambigu des rapports journaliers en une représentation structurée et riche de sens, prête à être utilisée par des modèles d’apprentissage automatique ou des systèmes d’aide à la décision.

2.3.2 Application du TAL aux rapports techniques : Le Défi du Langage de Domaine

L'application des techniques de Traitement Automatique du Langage (TAL) aux rapports techniques, et en particulier aux DDR, est une tâche fondamentalement différente de l'analyse de textes généralistes comme des articles de presse ou des critiques de films. Le langage du forage est un dialecte en soi, un sociolecte technique qui présente des défis uniques nécessitant des approches spécialisées.

2.3.2.a Les Défis Spécifiques de l'Analyse des DDR

Appliquer des modèles de TAL génériques directement sur les DDR sans adaptation mène souvent à des résultats médiocres. Les principaux obstacles sont les suivants :

- **Ambiguïté Sémantique et Jargon Technique** : C'est le défi le plus fondamental. Le vocabulaire est truffé de termes polysémiques dont le sens technique est très éloigné de l'usage courant.
 - Un **'kick'** n'est pas un coup de pied, mais une venue de fluide dangereuse.
 - Un **'fish'** n'est pas un poisson, mais un morceau d'équipement cassé et perdu dans le puits, nécessitant une opération de "pêche" (*fishing*) complexe.
 - **'Making a trip'** ne signifie pas faire un voyage, mais l'opération de manœuvre de la garniture.

De plus, une multitude d'acronymes sont utilisés sans être systématiquement définis : 'ROP' (Rate of Penetration), 'WOB' (Weight on Bit), 'NPT' (Non-Productive Time), 'BHA' (Bottom Hole Assembly), 'POOH' (Pull Out Of Hole). Un système de TAL efficace doit non seulement reconnaître ces termes, mais aussi les comprendre dans leur contexte opérationnel.

- **Variabilité et Manque de Standardisation** : Il n'existe pas de format universel pour les DDR. Chaque compagnie, et parfois chaque plateforme, a ses propres modèles de rapport, ses propres codes d'opération, et ses propres habitudes de rédaction. Une même opération, comme la connexion d'une nouvelle longueur de tige, peut être décrite comme "Make connection", "Connexion", "Drill pipe connection", ou simplement "C/X". Cette variabilité rend difficile la création de règles d'extraction robustes et exige des modèles capables de généraliser à travers ces différentes formulations.
- **Caractère Implicite de l'Information** : Le risque est rarement énoncé explicitement. Les superviseurs de forage, experts dans leur domaine, consignent des observations factuelles. C'est l'interprétation de la combinaison de ces observations qui révèle le risque. Par exemple, un DDR pourrait contenir les remarques suivantes à différentes heures :
 1. '14 :00 - Drilling at 3500m. Torque fluctuating between 15-20 kft.lbs.'
 2. '15 :00 - ROP decreased from 15 m/hr to 5 m/hr.'
 3. '16 :00 - Attempting to pull off bottom. Overpull of 30 tons observed.'

Aucune de ces phrases ne mentionne le mot "coincement" (*stuck*). Pourtant, la combinaison d'un couple élevé et fluctuant, d'une chute de la vitesse de pénétration, et d'une sur-traction lors de la remontée est un indicateur très fort d'un début de coincement de garniture. Un système d'IA doit être capable de réaliser ce même type de raisonnement multi-faits pour inférer un risque implicite.

- **Nécessité de Connaissances de Domaine Externe :** Pour évaluer correctement la criticité d'un événement, le système doit posséder des connaissances qui ne sont pas contenues dans le DDR lui-même. Savoir qu'une perte de circulation de $10\text{ m}^3/h$ est critique dans une section de réservoir à faible pression mais peut-être acceptable dans une autre formation, ou qu'un certain type de vibration (e.g., *stick-slip*) est particulièrement dommageable pour les trépan PDC, sont des exemples de connaissances externes indispensables.

2.3.2.b Approches Modernes : Dompter la Complexité avec les Modèles de Langage

Face à ces défis, les approches traditionnelles de TAL basées sur des dictionnaires de mots-clés ou des règles manuelles montrent rapidement leurs limites. Elles manquent de flexibilité et de capacité à gérer l'imprévu. C'est pourquoi les approches modernes, mises en œuvre dans ce projet, se tournent vers les grands modèles de langage (LLMs) pré-entraînés.

Fondation : Les Modèles de Langage Pré-entraînés (BERT, Llama) Au lieu de partir de zéro, ces modèles sont d'abord pré-entraînés par leurs concepteurs (Google, Meta, etc.) sur des téraoctets de textes provenant d'internet et de livres. Durant cette phase, ils apprennent la grammaire, la syntaxe, les faits du monde et, surtout, des représentations sémantiques nuancées des mots et de leurs relations. Ils deviennent des fondations de langage extrêmement puissantes.

Spécialisation : L'Affinage (Fine-Tuning) Une première stratégie pour adapter ces modèles à un domaine spécifique comme le forage est l'affinage. Elle consiste à continuer l'entraînement du modèle pré-entraîné, mais cette fois sur un corpus de textes spécifiques au domaine (des milliers de DDR, par exemple). Si ce corpus est annoté (e.g., où les experts ont manuellement identifié les entités, les relations, ou les risques), le modèle peut apprendre à réaliser ces tâches spécifiques avec une grande précision. C'est une approche très efficace mais qui requiert un effort d'annotation initial conséquent.

Notre Approche : Ingénierie de Retrieval-Augmented Generation and Prompt Engineering

Avec l'avènement de LLMs très performants comme la famille Llama 3, une autre approche, plus agile, est devenue possible et a été choisie pour ce projet. Elle repose sur deux piliers :

- **L'Ingénierie de Prompt Avancée :** Au lieu de ré-entraîner le modèle, nous lui fournissons une instruction (un *prompt*) très détaillée et structurée. Ce prompt explique précisément au modèle son rôle ("Tu es un expert en analyse de risques de forage"), la tâche à accomplir ("Identifie les risques à partir de ce DDR"), le format exact de la sortie attendue (une liste JSON structurée), et lui fournit même des exemples de la tâche réussie (*few-shot learning*). Cette technique permet de guider le raisonnement du modèle sans modifier ses poids internes.
- **La Génération Augmentée par Récupération (RAG) :** C'est la clé pour pallier le manque de connaissances de domaine spécifiques du LLM et pour le rendre plus fiable. Comme implémenté dans notre notebook `slb-assistant-decider.ipynb`, avant de poser la question au LLM, nous utilisons le contenu du DDR pour rechercher dans une base de connaissances externe (un index vectoriel FAISS contenant des extraits de manuels de forage) les informations les plus pertinentes. Ces extraits de connaissance sont alors injectés directement dans le prompt, aux côtés du DDR. Le LLM dispose ainsi de tout le contexte nécessaire : le rapport du jour **ET** les extraits de la "bible" du forage pertinents pour ce rapport. Cela lui permet de fonder

son analyse non seulement sur ce qu'il a appris durant son pré-entraînement, mais aussi sur des connaissances factuelles et vérifiables, une étape cruciale pour gagner la confiance des utilisateurs experts.

Cette approche RAG, au cœur de notre solution d'analyse textuelle, représente l'état de l'art pour l'application des LLMs à des tâches spécialisées nécessitant une haute fiabilité et une justification des résultats. Elle transforme le LLM d'un simple générateur de texte en un véritable assistant de raisonnement augmenté.

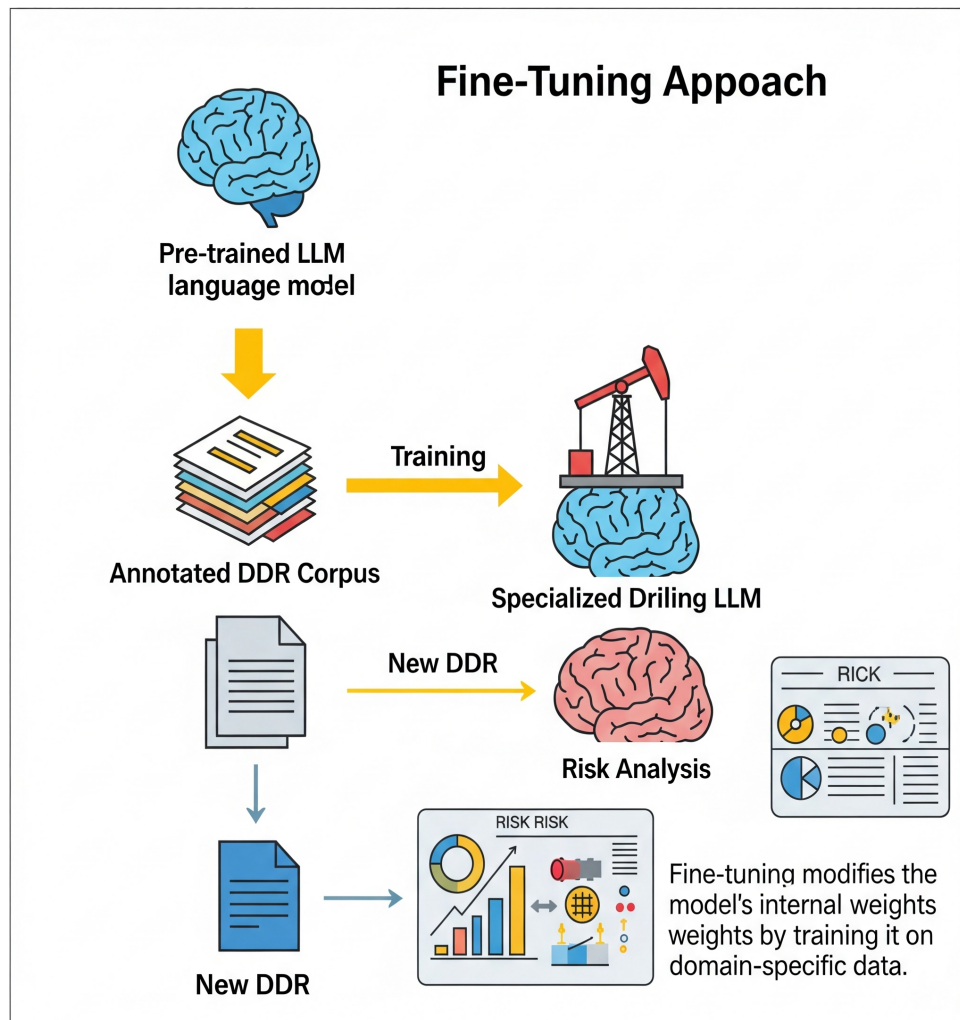
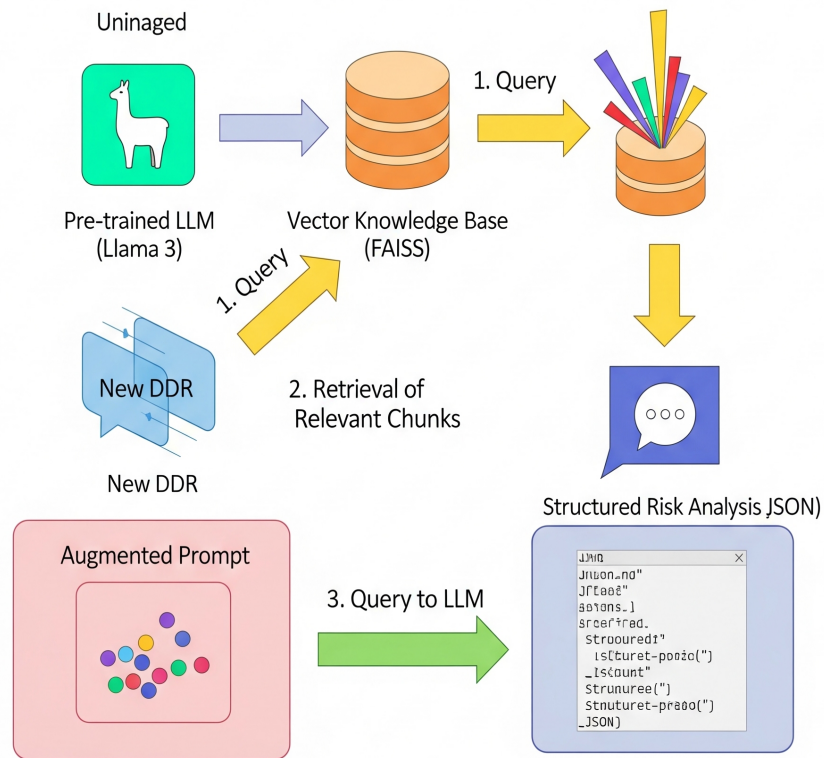


FIGURE 2.12 – l'approche par affinage (Fine-Tuning)



- RAG provides just-in-time context to the model without modifying it, improving relevance and reliability.

FIGURE 2.13 – l’approche par Génération Augmentée par Récupération (RAG)

2.4 Prédiction des risques en forage par apprentissage automatique

L'objectif est de passer d'une analyse réactive des incidents à une approche proactive, où les risques sont identifiés avant qu'ils ne se matérialisent. L'apprentissage automatique permet d'analyser des données historiques volumineuses, d'identifier des schémas complexes et de construire des modèles prédictifs.

2.4.1 Types de risques en forage et leur impact

Les opérations de forage sont intrinsèquement risquées, se déroulant dans un environnement hostile, à des milliers de mètres sous terre, avec des pressions et des températures extrêmes. L'identification et la prédiction de ces risques sont donc au cœur de la stratégie opérationnelle. Les risques peuvent être globalement classés en deux catégories : ceux menant à des **Temps Non Productifs (NPT)** et ceux menant à une **Perte de Contrôle du Puits (Well Control Incidents)**. Notre projet, à travers ses modèles LSTM et RAG, vise précisément à prédire les événements appartenant à ces catégories.

2.4.1.a Les Temps Non Productifs (NPT)

Le NPT est défini comme toute période durant laquelle les opérations planifiées ne peuvent pas se poursuivre. C'est un indicateur de performance majeur, car il représente un coût direct pour l'opérateur sans aucun avancement du puits. Les causes de NPT sont nombreuses et souvent interconnectées.

Le Coincement de la Garniture de Forage (*Stuck Pipe*) : C'est l'un des problèmes les plus fréquents et les plus coûteux. Il se produit lorsque la garniture de forage ne peut plus être ni remontée, ni descendue, ni mise en rotation. Les mécanismes principaux sont :

- **Le coincement mécanique :** Il est causé par un obstacle physique dans le puits. Cela peut être dû à un effondrement des parois (instabilité du trou), à l'accumulation de déblais mal évacués qui forment un bouchon, ou à une géométrie de trou défavorable (trou de serrure, *keyhole seating*).
- **Le coincement par pression différentielle :** Il se produit lorsque la garniture est immobile contre une formation perméable. Si la pression hydrostatique de la boue est significativement plus élevée que la pression de la formation, une force énorme vient plaquer la garniture contre la paroi du puits, la rendant impossible à déplacer.

Les conséquences d'un coincement vont de plusieurs heures de NPT passées à essayer de libérer la garniture (*jarring*), à des opérations de "pêche" (*fishing*) très coûteuses si une partie de la garniture se casse, jusqu'à la perte totale de la partie inférieure du puits, obligeant à forer un nouveau puits latéral (*sidetrack*). La prédiction de ce risque, ciblée par notre modèle `Target_NextDay_Event_Is_Stuck_Pipe`, est donc d'une valeur immense.

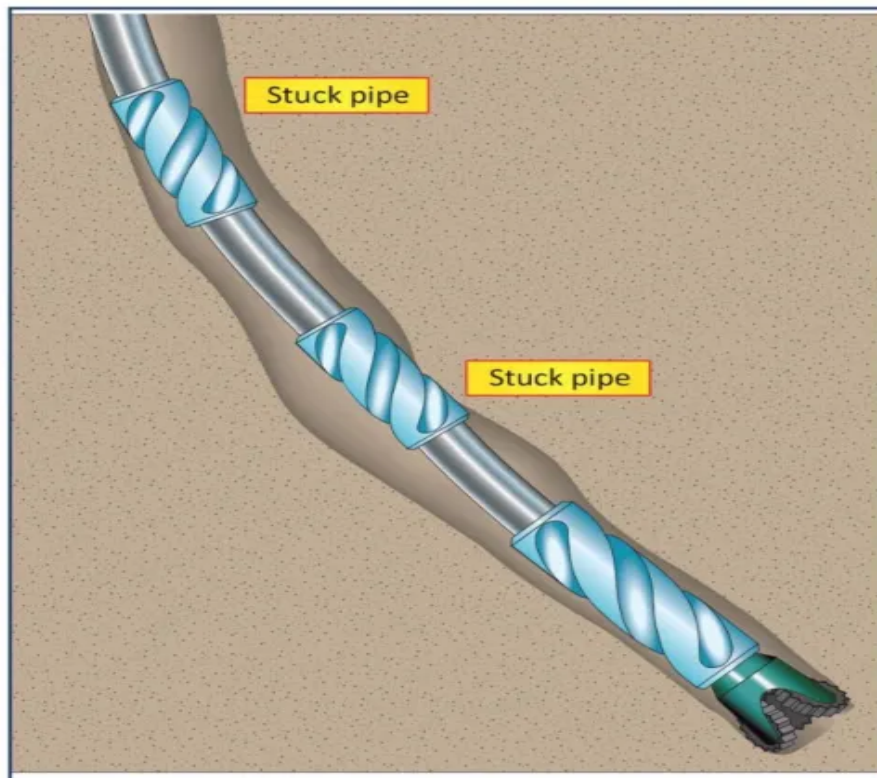


FIGURE 2.14 – Illustration des principaux mécanismes de coincement de la garniture de forage.

L'Instabilité des Parois du Puits (*Wellbore Instability*) : Ce risque, ciblé par notre modèle *Target_NextDay_Event_Is_Wellbore_Instability*, est étroitement lié au précédent. Il se réfère à la défaillance mécanique des roches formant les parois du puits. Les manifestations incluent :

- **L'effondrement (*Breakouts*) :** Chute de fragments de roche dans le puits.
- **Le gonflement des argiles (*Shale Swelling*) :** Certaines argiles gonflent au contact de l'eau contenue dans la boue, réduisant le diamètre du trou et pouvant piéger la garniture.
- **Le fluage des formations plastiques :** Les formations comme les dômes de sel peuvent lentement "fluer" et refermer le puits.

L'instabilité est principalement causée par un déséquilibre entre les contraintes de la roche en place et la pression exercée par la boue de forage. La détection des signes avant-coureurs (comme des déblais de forme particulière, des sur-tractions lors des manœuvres) est essentielle pour ajuster les propriétés de la boue et prévenir un coincement.

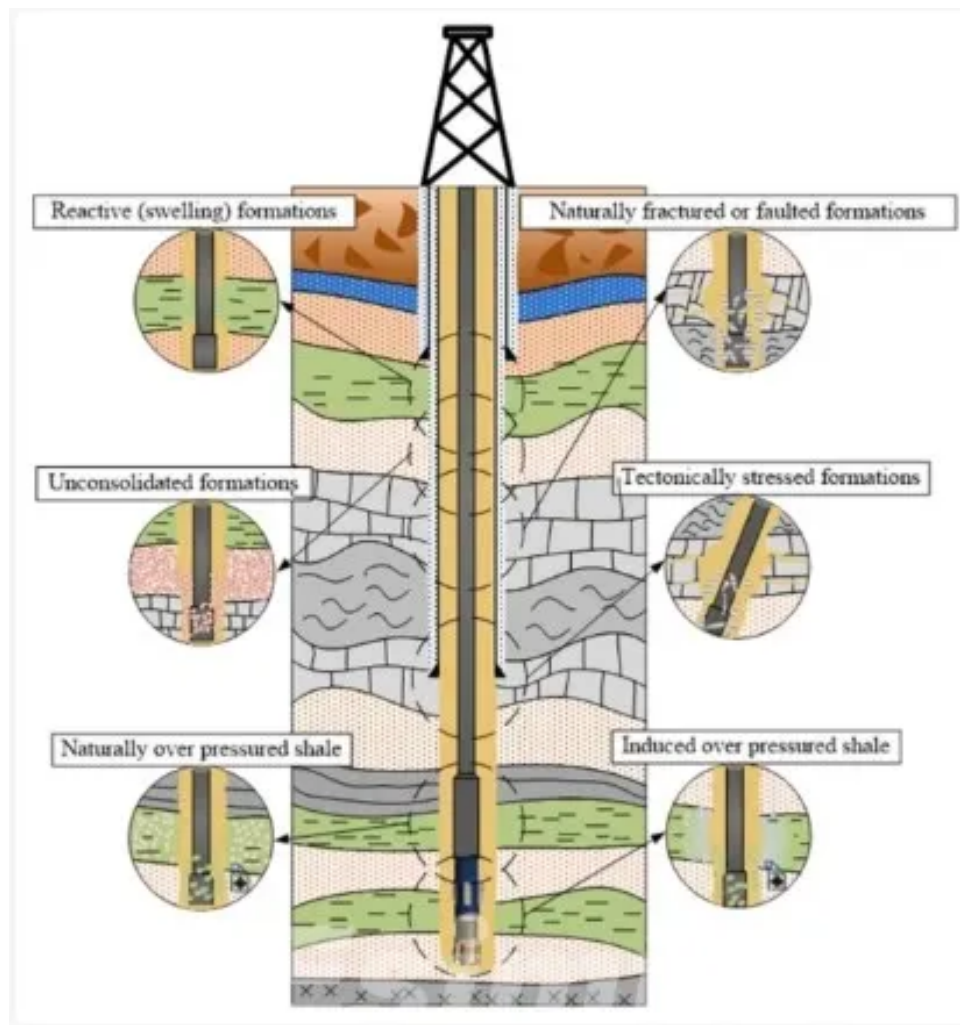


FIGURE 2.15 – Schématisation des différents types d'instabilité des parois du puits.

Les Pannes d'Équipement (*Equipment Failure*) : La prédiction des pannes, ciblée par notre modèle *Target_NextDay_Event_Is_Equipment_Failure*, est un enjeu majeur de la maintenance prédictive. La complexité et la sévérité des conditions de forage soumettent chaque composant à des contraintes extrêmes. Les défaillances peuvent survenir sur :

- **Les équipements de surface :** Pompes à boue, top drive, treuil. Une panne ici peut arrêter toute l'opération.
- **La garniture de forage :** Rupture par fatigue d'une tige de forage, fuite au niveau d'une connexion (*washout*).
- **Les outils de fond (BHA) :** Défaillance d'un moteur de fond, d'un outil MWD, ou usure catastrophique du trépan.

Anticiper une panne permet de planifier une intervention de maintenance lors d'une période opérationnelle moins critique, plutôt que de subir un arrêt imprévu et coûteux.

Les Pertes de Circulation (*Lost Circulation*) : Ce phénomène se produit lorsque la boue de forage s'échappe du puits et pénètre dans la formation au lieu de remonter à la surface. Cela arrive lorsque la pression exercée par la boue dépasse la résistance de la formation, typiquement dans des zones naturellement fracturées, des cavités (vugs) ou des formations très perméables. Les conséquences sont multiples :

- **Coût direct** : La perte de volumes importants de boue de forage, qui est un fluide coûteux, représente une perte financière sèche.
- **Risque de contrôle du puits** : La baisse du niveau de boue dans le puits entraîne une réduction de la pression hydrostatique. Si cette pression devient inférieure à la pression de la formation, cela peut provoquer un *kick*.
- **Temps non productif** : Des heures, voire des jours, peuvent être nécessaires pour colmater la zone de perte en pompant des matériaux spécifiques (LCM - Lost Circulation Materials).

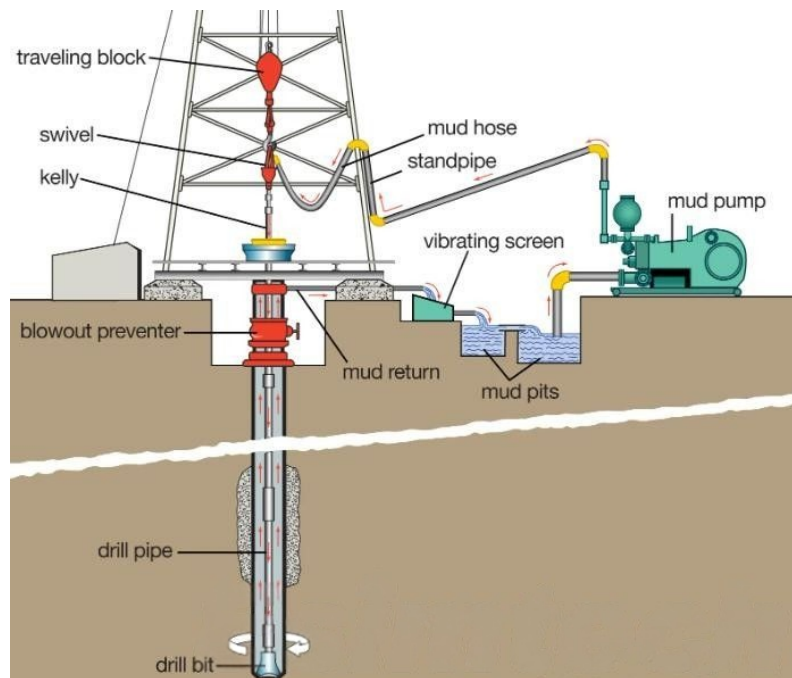


FIGURE 2.16 – Mécanismes des pertes de circulation dans des formations fracturées ou à haute perméabilité.

2.4.1.b Les Incidents de Contrôle du Puits (*Well Control*)

Cette catégorie regroupe les événements les plus redoutés de l'industrie, car ils menacent directement la sécurité du personnel, l'intégrité de l'installation et l'environnement.

Venue de Fluide (*Kick*) : Un *kick* est une intrusion non désirée et non contrôlée de fluides de formation (gaz, pétrole ou eau) dans le puits. Il se produit lorsque la pression hydrostatique exercée par la colonne de boue devient inférieure à la pression de la formation (pression de pore). Les causes peuvent être multiples : une densité de boue insuffisante, une mauvaise gestion des volumes lors d'une manœuvre (*swabbing*), ou le forage inattendu d'une zone en surpression.

La détection précoce d'un *kick* est une compétence fondamentale en forage. Les signes indicateurs incluent une augmentation du débit en retour, une augmentation des volumes dans les bassins de boue, ou une diminution de la pression de pompage. Dès qu'un *kick* est détecté, l'équipe doit immédiatement fermer le puits à l'aide du BOP et appliquer des procédures de contrôle de puits pour faire circuler l'influx hors du puits en toute sécurité et rétablir l'équilibre de pression.

Éruption (*Blowout*) : Si un *kick* n'est pas détecté ou maîtrisé à temps, il peut dégénérer en éruption (*blowout*). C'est une libération incontrôlée et violente des fluides de formation à la surface. Les conséquences peuvent être catastrophiques : incendies, explosions, pollution massive, perte totale de la plateforme et, surtout, pertes de vies humaines. Les éruptions sont des événements rares mais leur impact est si dévastateur qu'une part immense de la technologie, des procédures et de la formation en forage est dédiée à leur prévention. La prédiction d'événements à très haut risque, comme ciblée par notre modèle `Target_NextDay_Event_Is_Blowout_WC`, même avec une faible probabilité, est d'une importance stratégique capitale.



FIGURE 2.17 – Séquence d'un incident de contrôle de puits : de la détection d'un *kick* à la potentielle escalade vers une éruption (*blowout*).

En résumé, l'impact de ces risques est considérable, se traduisant par des surcoûts directs, des retards, et des menaces pour la sécurité et l'environnement. Une prédiction fiable, même avec une anticipation de quelques heures ou de 24 heures comme le vise notre projet, peut permettre de prendre des mesures préventives ou correctives adaptées et de transformer fondamentalement la gestion des risques en forage.

2.4.2 Approches de modélisation prédictive pour les risques de forage

Pour répondre à la nature hétérogène des données de forage, où les tendances numériques et le contexte textuel sont tous deux porteurs d'informations cruciales, notre projet met en œuvre une approche de modélisation hybride et innovante. Plutôt que de s'appuyer sur une seule méthode, nous exploitons deux paradigmes distincts et complémentaires de l'intelligence artificielle pour créer une vision à 360 degrés du risque. La première approche utilise un modèle d'apprentissage profond séquentiel (LSTM) pour analyser les données structurées et quantifiables. La seconde emploie un grand modèle de langage (LLM) avec une architecture de Génération Augmentée par Récupération (RAG) pour interpréter les données textuelles non structurées [32].

2.4.2.a Approche 1 : Modélisation Séquentielle avec les Réseaux LSTM

Les opérations de forage sont, par essence, un processus dynamique et séquentiel. L'état du puits à un instant t (par exemple, la pression, le couple) est fortement dépendant de son état à $t - 1$. Une augmentation progressive du couple sur plusieurs heures n'a pas la même signification qu'une augmentation soudaine. Les modèles d'apprentissage automatique traditionnels qui traitent chaque entrée de manière indépendante (comme les régressions logistiques ou les forêts aléatoires) peinent à capturer ces dépendances temporelles.

C'est pourquoi nous avons opté pour les Réseaux de Neurones Récurrents (RNN), et plus spécifiquement leur variante avancée, les **Long Short-Term Memory (LSTM)**. Les réseaux LSTM sont explicitement conçus pour traiter des séquences de données. Leur architecture interne, basée sur un système de "portes" (*gates*) qui contrôlent le flux d'information, leur permet de maintenir une "mémoire" à long terme. Ils peuvent ainsi apprendre quels éléments d'une séquence passée sont pertinents et doivent être conservés pour prédire un événement futur, tout en oubliant les informations non pertinentes [28].

Notre implémentation, détaillée dans le notebook `ML_Algos.ipynb`, suit un pipeline rigoureux pour transformer les données brutes des DDR en un format adapté à un modèle LSTM :

1. **Prétraitement et Structuration des Données** : La première étape consiste à extraire et à structurer les données de tous les DDR disponibles. Les dates sont parsées, et les données sont triées par puits et par ordre chronologique pour reconstituer l'historique de chaque opération.
2. **Ingénierie des Caractéristiques (*Feature Engineering*)** : Les données brutes sont transformées en un ensemble de caractéristiques numériques (*features*). Les colonnes numériques sont conservées, tandis que les colonnes catégorielles (comme le type de boue, le nom de l'opérateur, ou le code d'activité) sont converties en une représentation numérique via le **One-Hot Encoding**. Cette technique crée de nouvelles colonnes binaires pour chaque catégorie possible, évitant ainsi d'introduire un ordre artificiel entre les catégories.

3. **Création des Séquences Journalières :** Le flux continu d'événements horaires est agrégé au niveau de la journée. Chaque journée de forage est ainsi représentée par une séquence de vecteurs de caractéristiques, où chaque vecteur correspond à une opération consignée dans le DDR de ce jour. La cible de prédiction (*target*) est définie comme l'occurrence (ou non) d'un événement de risque spécifique (e.g., `Event_Is_Stuck_Pipe`) le jour **suivant**.
4. **Normalisation et Padding :** Avant d'entrer dans le réseau de neurones, les caractéristiques numériques sont normalisées à l'aide d'un `StandardScaler`. Cela centre les données autour de zéro et les met à l'échelle de la variance unitaire, ce qui est crucial pour la convergence et la performance des algorithmes basés sur le gradient. De plus, comme chaque journée contient un nombre différent d'opérations, les séquences journalières ont des longueurs variables. Pour les traiter en batch, elles sont uniformisées à une longueur maximale via une opération de **padding**, qui ajoute des valeurs nulles à la fin des séquences plus courtes. La couche **Masking** en tête de notre modèle LSTM est alors essentielle pour indiquer au réseau d'ignorer ces valeurs de padding lors des calculs.
5. **Architecture et Entraînement du Modèle :** Notre modèle LSTM (Figure 2.18) est une architecture de classification binaire. Après la couche de masking, une ou plusieurs couches LSTM extraient les caractéristiques temporelles de la séquence. Des couches de Dropout et de BatchNormalization sont ajoutées pour prévenir le sur-apprentissage et stabiliser l'entraînement. Finalement, des couches denses (`Dense`) effectuent la classification finale pour produire une probabilité de risque (entre 0 et 1) grâce à une fonction d'activation sigmoïde. Le modèle est entraîné à minimiser la perte de type *binary cross-entropy*, en utilisant des poids de classe pour gérer le déséquilibre inhérent aux données de risque (les événements problématiques étant heureusement plus rares que les opérations normales) [12, 1].

Cette approche, purement quantitative, excelle à modéliser la dynamique des paramètres mesurables et des séquences d'opérations codifiées.

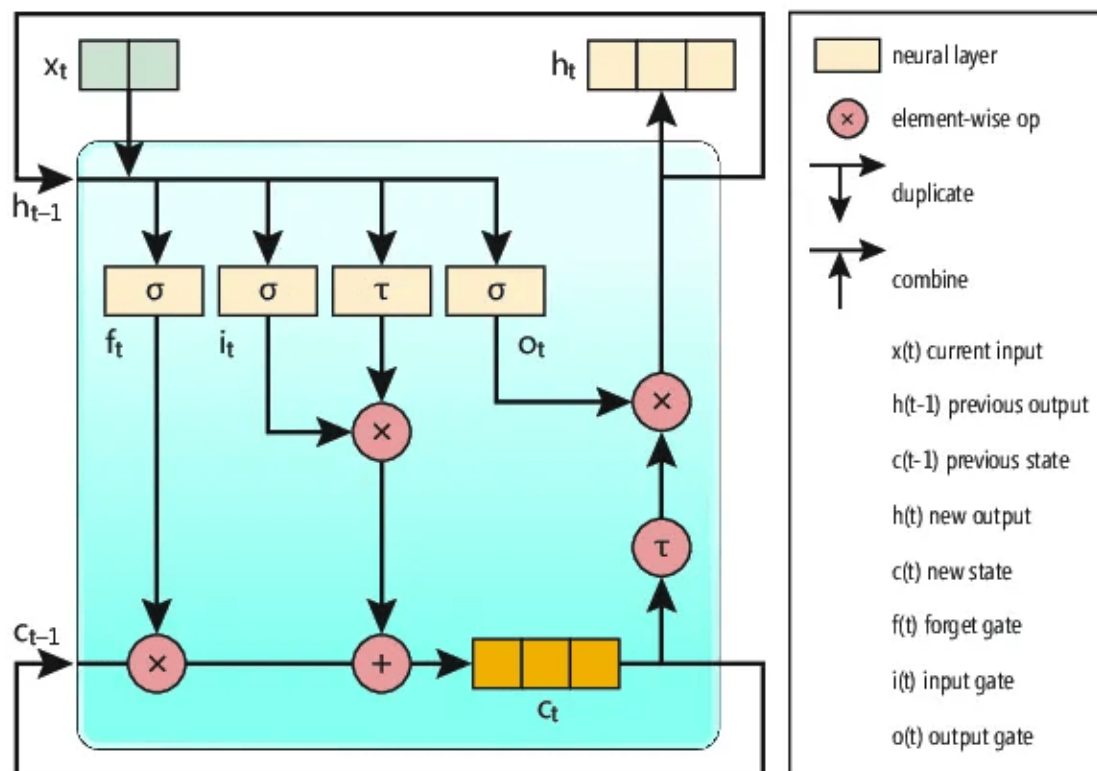


FIGURE 2.18 – Architecture Détaillée du Modèle LSTM Utilisé pour la Prédiction des Risques à partir des Données Séquentielles Structurées.

2.4.2.b Approche 2 : Analyse Contextuelle par IA Générative (RAG)

Si les modèles LSTM sont performants sur les chiffres et les séquences, ils sont "aveugles" au sens du texte libre contenu dans les résumés et les remarques des DDR. C'est pourtant là que se cache souvent le contexte crucial, les observations d'experts et les signes avant-coureurs d'un problème. Pour exploiter cette source d'information, nous utilisons une approche de pointe : la **Génération Augmentée par Récupération (RAG)**, dont l'implémentation est au cœur du notebook `slb-assistant-decider.ipynb`.

L'architecture RAG est une manière intelligente de combiner la puissance de raisonnement d'un Grand Modèle de Langage (LLM) avec la fiabilité d'une base de connaissances externe et factuelle.

Le LLM seul, même un modèle aussi puissant que `meta-llama/Meta-Llama-3-8B-Instruct`, peut parfois "halluciner" ou manquer de connaissances très spécifiques au domaine. Le RAG pallie cette faiblesse en lui fournissant un contexte pertinent "juste à temps".

Le processus, illustré à la Figure 2.19, se déroule en plusieurs étapes clés :

1. **Construction de la Base de Connaissances :** En amont, nous avons rassemblé un corpus de documents techniques de référence (procédures de forage, manuels de sécurité, retours d'expérience, etc.). Ce corpus est divisé en petits morceaux de texte (*chunks*). Chaque chunk est ensuite transformé en un vecteur numérique (embedding) à l'aide d'un modèle comme `all-MiniLM-L6-v2`. Tous ces vecteurs sont stockés et indexés dans une base de données vectorielle spécialisée, **FAISS** (Facebook AI Similarity Search), qui permet des recherches de similarité ultra-rapides.
2. **Phase de Récupération (Retrieval) :** Lorsqu'un nouveau DDR doit être analysé,

nous ne l'envoyons pas directement au LLM. Nous formulons d'abord une requête sémantique basée sur son contenu textuel (par exemple, "pertes de boue dans une formation carbonatée"). Cette requête est elle-même transformée en vecteur, qui est utilisé pour interroger l'index FAISS. FAISS renvoie alors les k chunks de la base de connaissances dont les vecteurs sont les plus proches de celui de la requête, c'est-à-dire les plus pertinents sémantiquement.

3. **Phase de Génération Augmentée (*Augmented Generation*)** : C'est ici que le LLM entre en jeu. Au lieu de lui poser une question simple, nous construisons un *prompt* très élaboré qui contient :

- **Le Contexte Récupéré** : Les extraits de la base de connaissances obtenus à l'étape précédente.
- **La Tâche** : Le DDR à analyser, présenté sous un format clair (JSON).
- **L'Instruction** : Une directive précise demandant au LLM d'agir en tant qu'expert, d'analyser le DDR à la lumière du contexte fourni, et de formuler sa réponse dans un format JSON strict, en détaillant chaque risque identifié, son niveau, les preuves, son raisonnement et une solution potentielle.

Grâce à cette augmentation de contexte, le LLM peut non seulement identifier des risques décrits textuellement ("légères vibrations ressenties sur le plancher") mais aussi les évaluer avec plus de pertinence. Il peut, par exemple, corréliser ces vibrations avec une procédure de forage spécifique décrite dans un document de la base de connaissances et ainsi évaluer le risque de manière plus précise. Cette approche rend le système plus transparent, car il peut, en principe, citer ses sources (les chunks de connaissance utilisés), et beaucoup plus fiable, car il fonde son raisonnement sur des données factuelles et non uniquement sur ses connaissances internes paramétriques.

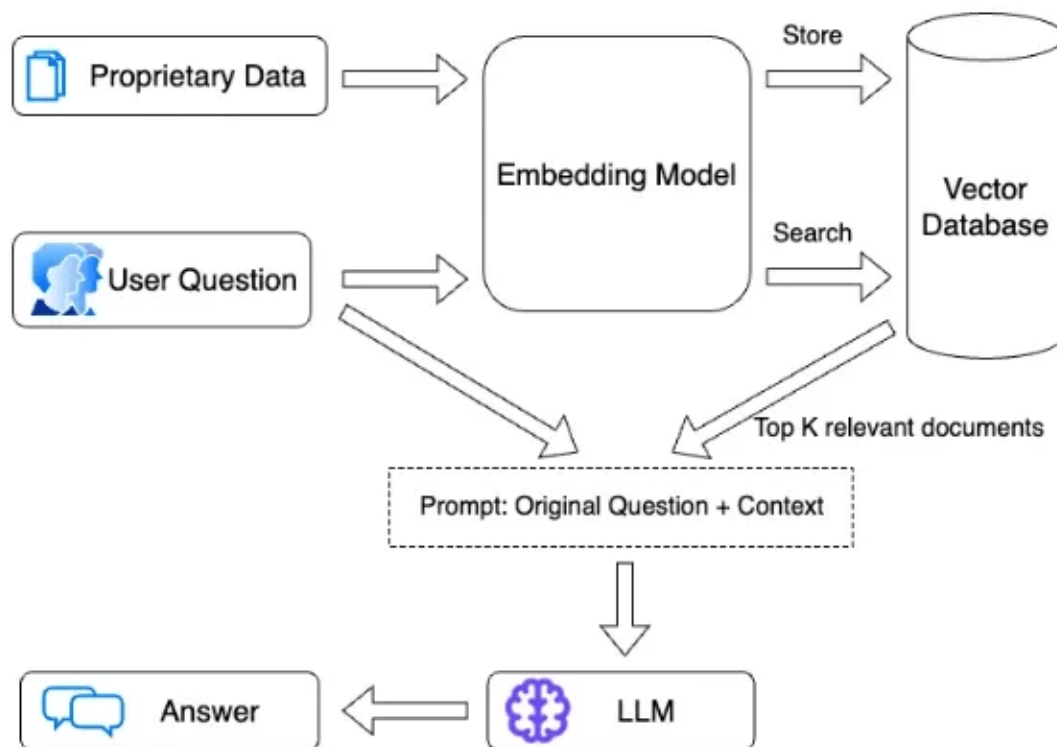


FIGURE 2.19 – Architecture Détaillée du Système de Génération Augmentée par Récupération (RAG) pour l'Analyse Contextuelle des DDR.

En conclusion, la combinaison de ces deux approches — LSTM pour les tendances quantitatives et RAG/LLM pour l'interprétation sémantique et contextuelle — nous permet de construire un système de prédiction des risques holistique, exploitant l'intégralité des informations contenues dans les rapports journaliers de forage.

2.4.3 Métriques d'évaluation et Systèmes d'Alerte : De la Prédiction à l'Action

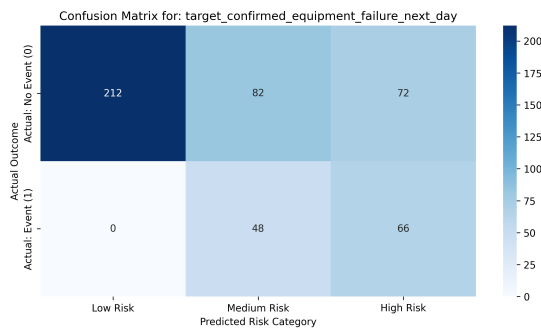
Le choix des métriques d'évaluation a été défini en collaboration avec notre promoteur et se base sur les meilleures pratiques de l'industrie pour l'évaluation de systèmes d'IA. Pour les modèles prédictifs (LSTM), nous avons retenu des métriques standards de classification binaire, en accordant une importance particulière à celles qui sont robustes aux données déséquilibrées (comme l'AUC et le rappel), car la sécurité et la détection des événements rares sont notre priorité. Pour le système génératif (RAG/LLM), en l'absence de "vérité terrain" unique, une évaluation quantitative est impossible. Nous avons donc adopté une approche qualitative, validée par des experts, qui se concentre sur la pertinence, l'exactitude et l'utilité des analyses générées, une méthodologie courante pour l'évaluation des systèmes d'IA explicables [4].

La construction d'un modèle prédictif n'est que la première étape. Pour qu'un système d'intelligence artificielle soit véritablement utile dans un contexte industriel à haut risque comme le forage, sa performance doit être évaluée de manière rigoureuse et ses résultats doivent être communiqués de manière efficace et opportune. Cette section détaille les métriques utilisées pour évaluer nos deux approches de modélisation (LSTM et RAG/LLM) et décrit l'architecture du système d'alerte conçu pour rendre ces prédictions actionnables.

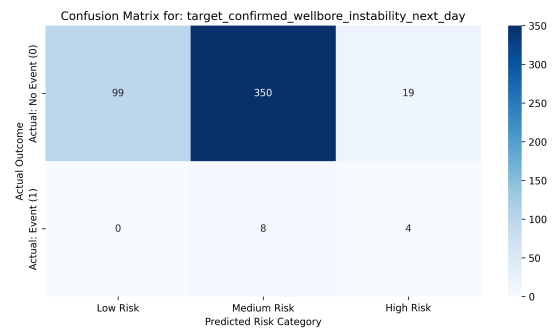
2.4.3.a Évaluation Quantitative des Modèles de Classification Séquentielle (LSTM)

Pour notre modèle LSTM, qui produit une probabilité de survenue d'un événement (une sortie numérique entre 0 et 1), nous utilisons un ensemble de métriques de classification binaire standard, comme implémenté dans notre notebook `ML_Algos.ipynb`. Le choix de ces métriques est crucial, en particulier dans un contexte de données déséquilibrées où les événements de risque sont rares.

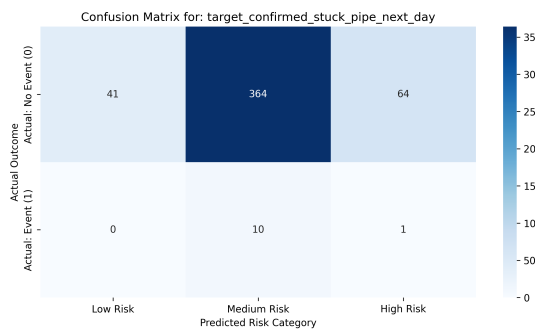
FIGURE 2.20 – Matrices de Confusion pour chaque modèle de prédiction de risque sur l'ensemble de test.



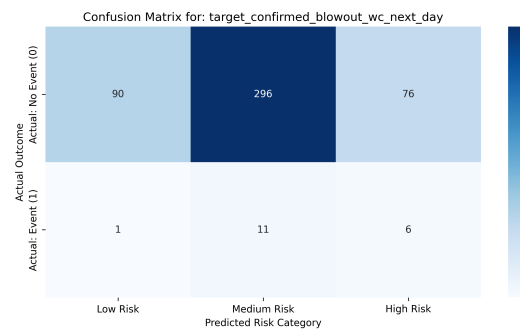
(a) Risque de Panne d'Équipement



(b) Risque d'Instabilité de Puits



(c) Risque de Coincement de Garniture

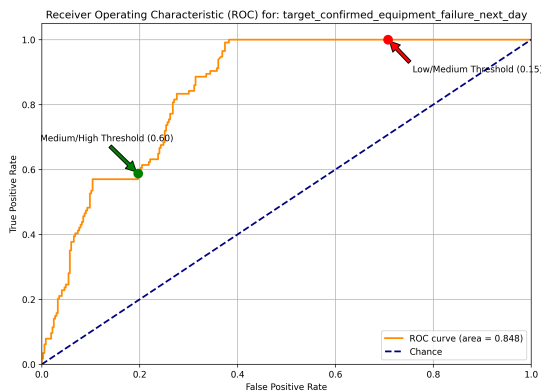


(d) Risque d'Éruption (Well Control)

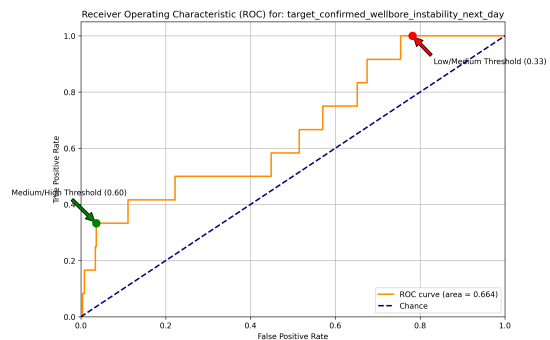
La **Matrice de Confusion** (Figure 2.20) est l'outil de base pour comprendre le comportement du modèle. Elle croise les prédictions du modèle avec les issues réelles et se décompose en quatre quadrants : Vrais Positifs (VP), Vrais Négatifs (VN), Faux Positifs (FP), et Faux Négatifs (FN). Dans notre application, l'objectif principal est de minimiser les Faux Négatifs.

La **Courbe ROC** et l'aire sous cette courbe (AUC) sont des outils puissants qui évaluent la performance du modèle sur l'ensemble des seuils de décision possibles. La Figure 2.21 présente les courbes obtenues pour nos quatre modèles.

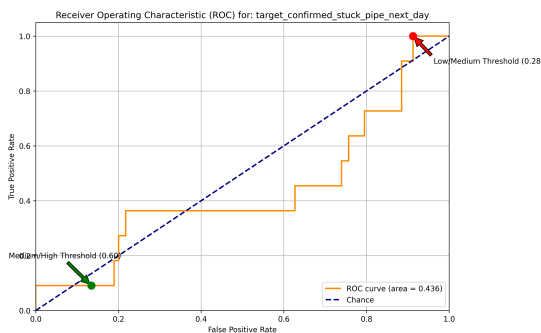
FIGURE 2.21 – Courbes ROC et Aires Sous la Courbe (AUC) pour chaque modèle de prédiction de risque.



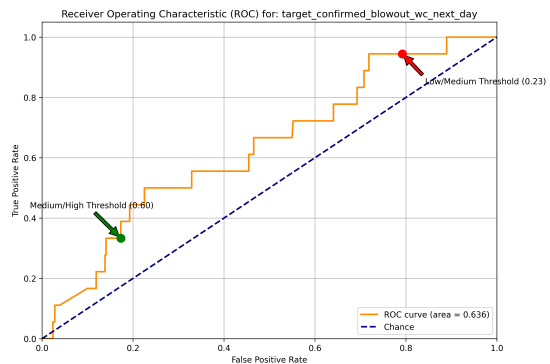
(a) Risque de Panne d'Équipement



(b) Risque d'Instabilité de Puits



(c) Risque de Coincement de Garniture



(d) Risque d'Éruption (Well Control)

— **Matrice de Confusion** : C'est l'outil de base pour comprendre le comportement du modèle (Figure 2.20). Elle croise les prédictions du modèle avec les issues réelles et se décompose en quatre quadrants :

- **Vrais Positifs (VP)** : Le modèle a prédit un risque, et le risque s'est réellement produit. C'est une prédiction correcte et utile.
- **Vrais Négatifs (VN)** : Le modèle n'a pas prédit de risque, et aucun risque ne s'est produit. C'est l'état normal et une prédiction correcte.
- **Faux Positifs (FP) - Erreur de Type I** : Le modèle a prédit un risque, mais rien ne s'est passé (fausse alarme). Un trop grand nombre de FP peut mener à une "fatigue des alarmes" et à une perte de confiance dans le système.
- **Faux Négatifs (FN) - Erreur de Type II** : Le modèle n'a rien prédit, mais un risque s'est produit. C'est le type d'erreur le plus dangereux et le plus coûteux, car il représente un risque non détecté.

Dans notre application, l'objectif principal est de **minimiser les Faux Négatifs**.

- **Précision (Precision)** : Définie comme $\frac{VP}{VP+FP}$, elle répond à la question : "Lorsque le modèle sonne l'alarme, quel est le pourcentage de fois où il a raison ?". Une haute précision est souhaitable pour éviter la fatigue des alarmes.
- **Rappel (Recall ou Sensibilité)** : Défini comme $\frac{VP}{VP+FN}$, il répond à la question : "Sur tous les risques qui se sont réellement produits, quel est le pourcentage que le

modèle a réussi à détecter ?". Un rappel élevé est la priorité absolue dans un système de détection de risques. Il est souvent en tension avec la précision : augmenter le rappel (en étant plus sensible) peut parfois augmenter le nombre de fausses alarmes (et donc diminuer la précision).

- **Score F1** : C'est la moyenne harmonique de la précision et du rappel ($2 \times \frac{Precision \times Recall}{Precision + Recall}$). Il fournit un score unique qui équilibre ces deux métriques, utile pour comparer la performance globale de différents modèles.
- **Courbe ROC et AUC** : La courbe ROC (Figure ??) est un outil puissant qui évalue la performance du modèle sur l'ensemble des seuils de décision possibles. Elle trace le taux de Vrais Positifs (Rappel) en fonction du taux de Faux Positifs. L'Aire Sous la Courbe (AUC) résume cette courbe en un seul chiffre. Un AUC de 0.5 correspond à un modèle aléatoire, tandis qu'un AUC de 1.0 représente un classifieur parfait. L'AUC est particulièrement utile pour les données déséquilibrées car elle n'est pas affectée par la proportion des classes. C'est l'une des métriques clés que nous suivons lors de l'entraînement de nos modèles LSTM.

2.4.3.b Évaluation Qualitative des Modèles Génératifs (RAG/LLM)

L'évaluation de la sortie de notre système RAG, qui est un texte JSON structuré, ne peut pas se résumer à de simples métriques quantitatives. L'évaluation doit être plus holistique et se concentrer sur les aspects suivants :

- **Validité et Conformité du Format** : La première vérification, automatique, est de s'assurer que la sortie du LLM est un JSON bien formé et qu'il respecte la structure demandée dans le prompt (une liste d'objets, chacun avec les clés "Risk", "Level", "How Identified", etc.). Un échec à ce niveau rend la sortie inexploitable par notre application.
- **Pertinence et Exactitude des Risques Identifiés** : Cette évaluation est nécessairement qualitative et doit être menée par des experts du domaine. Pour un DDR donné, un expert compare la liste des risques générée par le LLM à sa propre analyse. Les questions posées sont : Le LLM a-t-il identifié les mêmes risques que l'expert ? En a-t-il manqué (Faux Négatifs) ? En a-t-il inventé (Faux Positifs) ?
- **Qualité de la Justification** : Au-delà de l'identification, la valeur ajoutée du LLM réside dans sa capacité à justifier ses conclusions. L'évaluation porte sur la pertinence des preuves citées ("DDR_Evidence") et la logique du "Reasoning". Le raisonnement est-il techniquement correct ? Fait-il un lien pertinent entre les observations du DDR et le risque potentiel ?
- **Pertinence des Solutions Proposées** : L'expert évalue si les solutions ou mesures d'atténuation suggérées sont appropriées, réalistes et conformes aux bonnes pratiques de l'industrie. La capacité du modèle RAG à s'appuyer sur des documents de la base de connaissances est ici fondamentale.

Pour ce projet, une évaluation qualitative a été menée sur un sous-ensemble de 20 rapports représentatifs, incluant des cas avec des risques avérés et des cas sans incident notable. Les sorties du système RAG/LLM ont été soumises à un expert du domaine qui a comparé l'analyse générée avec sa propre expertise. Les résultats ont montré que dans 18 cas sur 20 (90%), le système a correctement identifié les risques principaux et a fourni une justification pertinente. Dans 2 cas, le système a manqué un risque mineur décrit de manière très implicite, soulignant la dépendance à la clarté du texte source. Cette validation, bien

que non exhaustive, a permis de confirmer la viabilité et l'utilité potentielle de l'outil dans un contexte opérationnel réel.

2.4.3.c Système d'Alerte en Temps Réel via Webhook

Un modèle prédictif, même très performant, est inutile s'il reste silencieux. Pour rendre les analyses de nos modèles immédiatement actionnables, nous avons conçu et implémenté un système d'alerte en temps réel. Ce système, dont la logique d'envoi est visible dans le notebook `slb-assistant-decider.ipynb`, repose sur le concept de **webhook** [48].

Un webhook est un mécanisme HTTP qui permet à un système (notre script d'analyse IA) d'envoyer des informations à un autre système (notre application web Flask) dès qu'un événement se produit, sans que le second n'ait besoin de le solliciter en permanence. L'architecture est la suivante :

1. **Déclenchement** : Une fois que notre pipeline d'IA a terminé l'analyse d'un DDR, il examine les résultats. Si des risques jugés significatifs (par exemple, de niveau "High" ou "Medium") sont identifiés, il déclenche l'envoi d'une alerte.
2. **Construction de la Charge Utile (Payload)** : Le script d'IA construit une charge utile au format JSON. Ce JSON contient toutes les informations pertinentes : le nom du puits, la date du rapport, et surtout, la liste structurée des risques significatifs identifiés (avec leur description, niveau, preuves, etc.).
3. **Appel HTTP POST** : Le script envoie cette charge utile via une requête HTTP POST à une URL spécifique exposée par notre application web Flask : l'endpoint `/api/ddr-risk-alert`. Pour la sécurité, la requête inclut un en-tête secret (`X-Webhook-Secret`) pour que le serveur Flask puisse vérifier que la requête provient bien d'une source autorisée.
4. **Réception et Traitement Côté Serveur** : L'application Flask reçoit la requête sur son endpoint. Elle vérifie le secret, valide la charge utile JSON, et persiste immédiatement l'alerte dans sa base de données (associée à un timestamp, un statut "non lue", etc.).
5. **Alerte en Temps Réel** : Simultanément, le serveur peut envoyer une alerte email à l'utilisateur concerné et, grâce à des mécanismes comme les WebSockets ou des interrogations périodiques (polling) du frontend, l'interface utilisateur peut être mise à jour en temps réel pour afficher une nouvelle alerte, par exemple via une icône de cloche qui s'illumine.

Ce système d'alerte transforme notre plateforme d'un simple outil d'analyse passive en un système de surveillance proactif, garantissant que les informations critiques parviennent aux bonnes personnes au bon moment pour permettre une intervention rapide et efficace.

CHAPITRE

— 3 —

CONCEPTION ET IMPLÉMENTATION
DE LA SOLUTION

3.1 Introduction

Après avoir établi, dans les chapitres précédents, l'état de l'art des domaines concernés et défini le contexte ainsi que la problématique de notre projet, ce chapitre est consacré à la description détaillée de la solution que nous avons conçue et implémentée. L'objectif est de développer une application web capable d'extraire des informations pertinentes des rapports journaliers de forage (DDR), de visualiser ces données de manière interactive, et d'utiliser des modèles d'apprentissage automatique pour la prédiction des risques. Ce chapitre présentera l'architecture globale de notre solution, puis détaillera la méthodologie et les outils utilisés pour chaque composant majeur, en commençant par le module crucial de traitement et d'extraction des données à partir des rapports bruts. Nous aborderons ensuite la construction des modules de visualisation et de modélisation prédictive, ainsi que l'intégration de ces éléments au sein de l'interface web finale.

3.2 Architecture générale de l'application web

L'application web que nous avons développée pour l'extraction d'informations et la prédiction des risques à partir des DDR a été conçue selon une architecture modulaire et évolutive, visant à séparer clairement les préoccupations et à faciliter la maintenance ainsi que les développements futurs.

L'architecture globale repose sur une interaction *client-serveur* classique, où l'interface utilisateur (*front-end*) communique avec un serveur applicatif (*back-end*) qui, à son tour, orchestre l'accès aux données prétraitées et aux résultats des analyses.

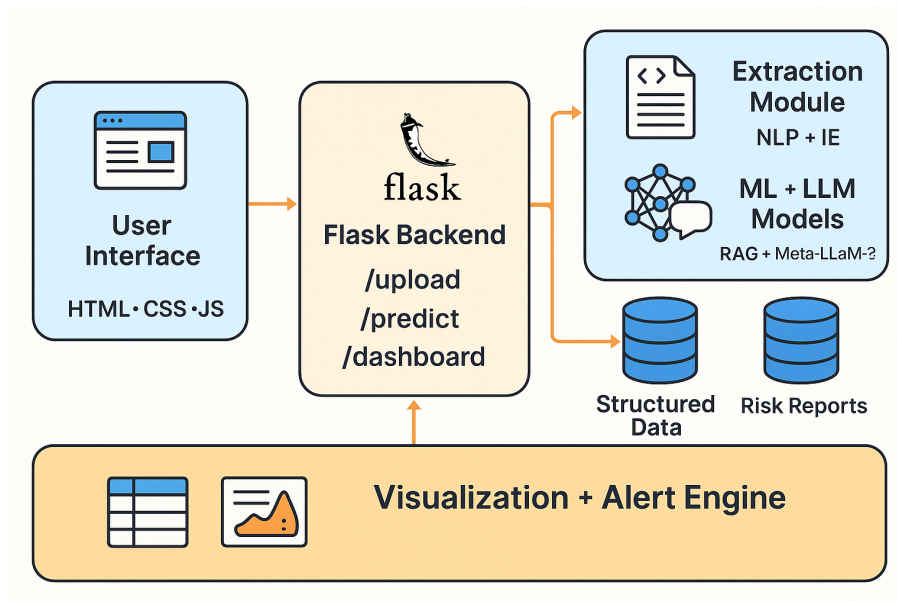


FIGURE 3.1 – Architecture générale de l'application web

Flux de données et interactions

Le flux de données et d'interactions principal peut être résumé comme suit :

1. L'utilisateur accède à l'application via un navigateur web, interagissant avec le composant **front-end**.

2. Les actions de l'utilisateur (navigation, sélection de filtres, demande de visualisations) déclenchent des requêtes HTTP vers le composant **back-end**.
3. Le **back-end**, construit avec le micro-framework Flask en Python, traite ces requêtes. Il accède aux données pré-extraites des DDR (fichiers CSV générés par le script de scraping décrit précédemment, stockés dans le répertoire DDR et ses sous-dossiers de puits).
4. Pour certaines fonctionnalités, notamment la génération de visualisations dynamiques et l'application de filtres complexes, le back-end effectue des opérations de lecture, de fusion (par exemple entre `operations_info.csv` et `combined.csv`), et de calcul sur ces données à l'aide de la bibliothèque `pandas`.
5. Les résultats de ces traitements (données formatées pour les graphiques, options de filtres, informations descriptives) sont renvoyés au **front-end**, souvent sous forme de JSON, via des points d'accès (*endpoints*) d'API définis dans l'application Flask.
6. Le **front-end**, principalement construit avec HTML, CSS et JavaScript, utilise ces données pour mettre à jour dynamiquement l'interface, afficher les tableaux de bord, les graphiques interactifs (générés avec `Plotly.js`) et les tables de résultats.

Avantages de l'architecture

Cette architecture permet de :

- Distinguer clairement la logique de présentation (*front-end*) de la logique de traitement des données et de service (*back-end*).
- Favoriser une meilleure organisation du code.
- Offrir une flexibilité accrue pour l'ajout de nouvelles fonctionnalités ou l'évolution de l'interface utilisateur.

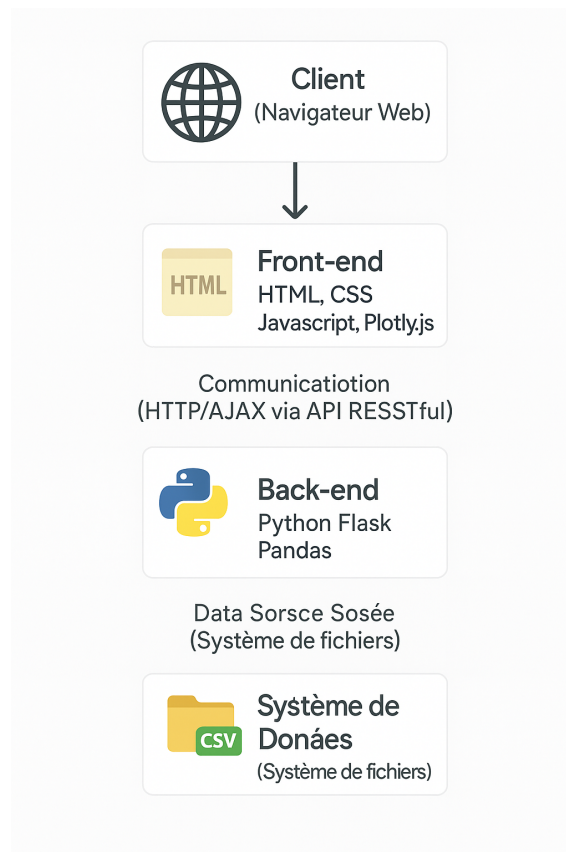


FIGURE 3.2 – Représentation conceptuelle de la solution

3.2.1 Composant front-end

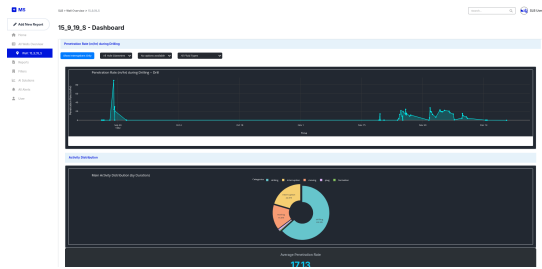
Le composant *front-end* est l'interface web avec laquelle l'utilisateur interagit. Développé avec les technologies standards HTML, CSS et JavaScript, il a été conçu pour être à la fois intuitif et informatif. La bibliothèque `Plotly.js` a été choisie pour sa capacité à créer des graphiques riches et interactifs, qui sont au cœur de notre solution de visualisation.

L'interface est structurée autour d'une navigation claire qui donne accès aux différentes fonctionnalités. La page d'accueil (Figure 3.3) sert de tableau de bord central, offrant une vue d'ensemble avec des indicateurs de performance clés (KPIs) et des graphiques de distribution. L'utilisateur peut ensuite explorer les données d'un puits spécifique (Figure ??) via des visualisations détaillées et interactives, ou encore accéder à l'historique des rapports bruts (Figure 3.5).

FIGURE 3.3 – Vue du tableau de bord principal de l’application, affichant les KPIs globaux et des graphiques sur la flotte de puits.



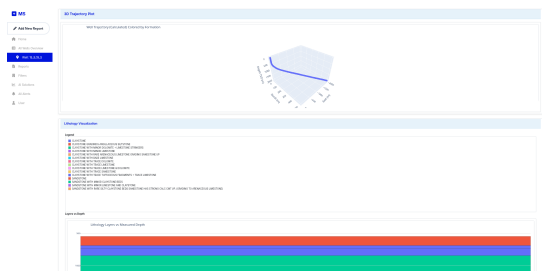
FIGURE 3.4 – Vues détaillées de l'application pour un puits spécifique.



(a) Vue des graphiques d'opérations.



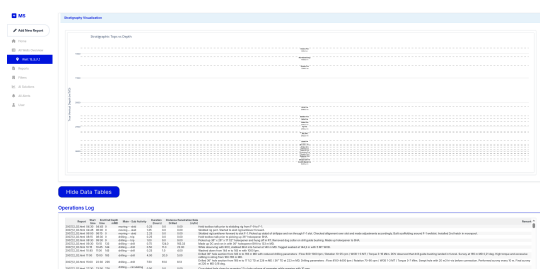
(b) Vue des visualisations interactives.



(c) Visualisation de la trajectoire 3D.



(d) Visualisation de la lithologie.



(e) Visualisation des changements de fluides.

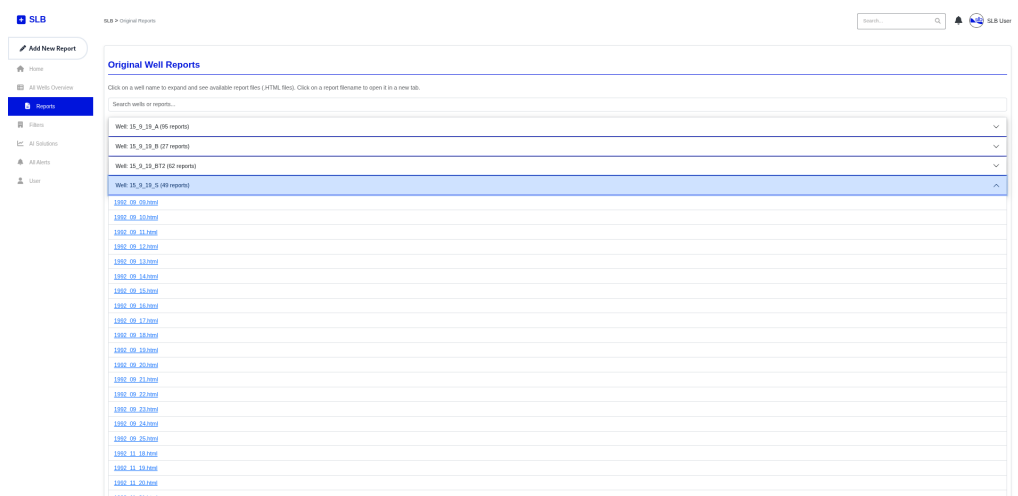


FIGURE 3.5 – Interface de navigation permettant d'accéder aux rapports journaliers de forage (DDR) originaux au format HTML.

3.2.2 Composant back-end

Le composant *back-end* de notre application est développé en Python à l'aide du micro-framework Flask. Il sert d'intermédiaire entre l'interface utilisateur (front-end) et les sources de données, en assurant la gestion de la logique métier et l'exposition de points d'accès API consommés par le front-end.

Principales responsabilités du back-end (fichier `app.py`) :

- **Routage et rendu de pages HTML** : Flask gère le routage des URL vers les fonctions Python correspondantes. Celles-ci renvoient ensuite des templates HTML via la fonction `render_template()`, par exemple :
 - `@app.route('/')` pour la page d'accueil,
 - `@app.route('/view/<folder_name>')` pour la visualisation détaillée d'un puits.
- **Accès et traitement des données** : Le back-end lit et traite des fichiers CSV prétraités (comme `operations_info.csv`, `wellbore_info_transposed.csv`, `combined.csv`, etc.) à l'aide de fonctions utilitaires situées dans le module `utils/` :
 - `get_data_folders()`,
 - `read_csv_safely()`,
 - `find_column_name()`.Ces fonctions utilisent intensivement la bibliothèque `pandas` pour :
 - Charger les fichiers CSV en `DataFrame`,
 - Fusionner les données (e.g. enrichissement de `operations_info.csv` avec `combined.csv` dans `get_merged_ops_data_for_well()`),
 - Appliquer des filtres (`apply_filters()`),
 - Calculer des statistiques et agrégats (`get_overall_well_kpis()`, `calculate_summary()`).
- **Exposition d'API pour l'interactivité** : Plusieurs endpoints sont définis pour répondre aux requêtes dynamiques du front-end :
 - `/get_initial_filter_options` et `/get_updated_filter_options` (dans `homefunctions.py`) : pour générer dynamiquement les options des filtres sur la page d'accueil.
 - `/generate_plot_data` : pour fournir les données nécessaires aux graphiques du ROP filtrés.
 - `/api/view/<folder_name>/filter_options, rop_data, activity_data` : pour la page d'un puits donné, ces routes renvoient les options de filtres et les données graphiques selon les critères choisis.

Les données retournées sont en général au format JSON, et sont ensuite interprétées par le JavaScript du front-end.

- **Génération de visualisations statiques** : Certains graphiques spécifiques (trajectoire 3D, lithologie, stratigraphie, fluides) sont générés côté serveur dans `create_single_well_plot` et transmis en JSON Plotly au template HTML. Ce traitement est privilégié lorsque :
 - Les calculs sont complexes (e.g. courbure minimale),
 - L'interactivité n'est pas critique.
- **Service des fichiers statiques et originaux** : Flask sert les fichiers statiques (CSS, JS, images) et permet, via `/data/<folder_name>/<filename>`, un accès sécurisé aux DDR HTML originaux (dossier DDR2).

— **Gestion des erreurs et sessions :**

- `app.logger` et `traceback` permettent le débogage.
- Une clé secrète est définie pour utiliser le système de flash messages, notamment pour signaler les erreurs sur la page de filtrage.

Figure 3.4 : Schéma des Principaux Endpoints d'API du Back-end Flask

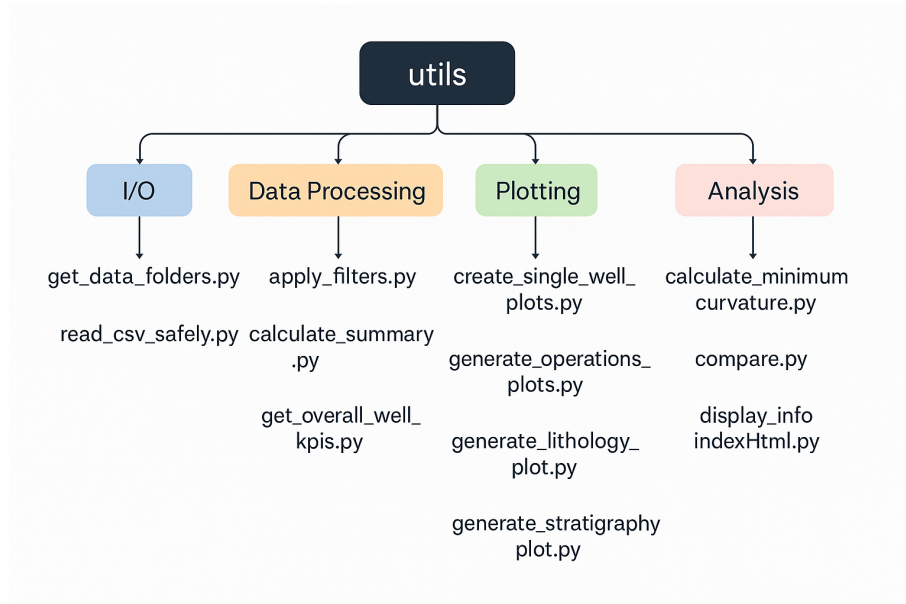


FIGURE 3.6 – Schéma des Principaux Endpoint

3.2.2.a Module de traitement des données et de modélisation

Bien que l'application actuelle soit principalement centrée sur l'extraction et la visualisation des données tabulaires issues de fichiers CSV prétraités, l'architecture est conçue pour accueillir à terme un véritable module de traitement et de modélisation.

Composants actuels du traitement de données :

- **Script `operations.ipynb` (prétraitement) :** Ce notebook Python effectue :
 - Le parsing des DDR HTML bruts (via BeautifulSoup),
 - Le nettoyage et la structuration des données,
 - Le calcul de colonnes dérivées (durée, distance forée, ROP),
 - La génération des fichiers `operations_info.csv`, principale source du back-end.
- **Fonctions analytiques dans le back-end :** Certaines fonctions de `app.py` et `utils/` apportent un premier niveau de traitement :
 - `get_overall_well_kpis()` : calcule des indicateurs globaux.
 - `create_single_well_plots()`, `generate_operations_plots.py` : préparent les données pour la visualisation.
 - `calculate_minimum_curvature()` : réalise des calculs géométriques pour modéliser la trajectoire des puits.

3.3 Construction du dataset utilisé

La construction d'un dataset de haute qualité, robuste et pertinent est l'étape la plus critique et fondamentale de notre projet. La performance de chaque composant subséquent — de l'analyse visuelle à la modélisation prédictive — est intrinsèquement liée à la qualité des données qui l'alimentent. Ce processus, loin d'être une simple conversion de format, est une démarche complexe de transformation des rapports journaliers de forage (DDR) bruts, souvent hétérogènes et au format HTML, en une base de connaissances structurée, nettoyée et sémantiquement enrichie. Cette section se consacre à la description détaillée de la méthodologie rigoureuse que nous avons conçue et implémentée, en décomposant chaque phase : de l'identification et la collecte systématique des documents sources, à leur normalisation, l'extraction de leurs contenus, l'ingénierie de caractéristiques pertinentes, jusqu'à la préparation finale des données pour l'entraînement des modèles d'apprentissage automatique.

3.3.1 Identification et collecte des rapports techniques

Le point de départ de notre chaîne de traitement est l'acquisition des rapports techniques qui constituent la matière première de notre analyse. La phase de collecte a été rigoureusement automatisée afin de pouvoir traiter un volume important de documents de manière efficace et, surtout, de garantir une reproductibilité totale du processus à chaque exécution. Nous avons développé à cet effet un script d'ingestion en Python. Sa première tâche est d'explorer une arborescence de fichiers dont la structure est prédéfinie. L'utilisateur spécifie un répertoire parent, qui sert de conteneur à l'ensemble des données de forage. Le script est conçu pour interpréter la hiérarchie de ce répertoire, où chaque sous-dossier est conventionnellement associé à un puits de forage unique. Cette organisation structurée est essentielle : elle permet non seulement de maintenir une séparation logique et claire des données, mais aussi de faciliter les analyses et les comparaisons futures à l'échelle d'un puits spécifique. Pour chaque puits identifié, le script procède à un balayage systématique de son contenu. Un filtre est appliqué pour ne retenir que les fichiers correspondant aux formats cibles de notre étude, à savoir les rapports au format `.html` ou `.htm`. Bien que notre corpus soit homogène en termes de format de fichier, cette étape de filtrage reste une bonne pratique de programmation défensive, assurant que seuls les documents pertinents sont traités. Le contenu de chaque fichier HTML valide est ensuite lu et chargé en mémoire. Une attention particulière a été portée à la gestion de l'encodage des caractères, un défi fréquent lors du traitement de documents issus de sources diverses. Nous avons standardisé l'utilisation de l'encodage `utf-8`, le plus répandu, tout en intégrant un gestionnaire d'erreurs flexible (`errors='ignore'`). Cette précaution technique est cruciale car elle permet au script de poursuivre son exécution sans interruption, même s'il rencontre des caractères malformés ou des encodages non standards, un phénomène courant dans des documents générés par différents systèmes au fil du temps. Cette approche garantit ainsi la robustesse et la continuité du processus de collecte sur l'intégralité du corpus de rapports.

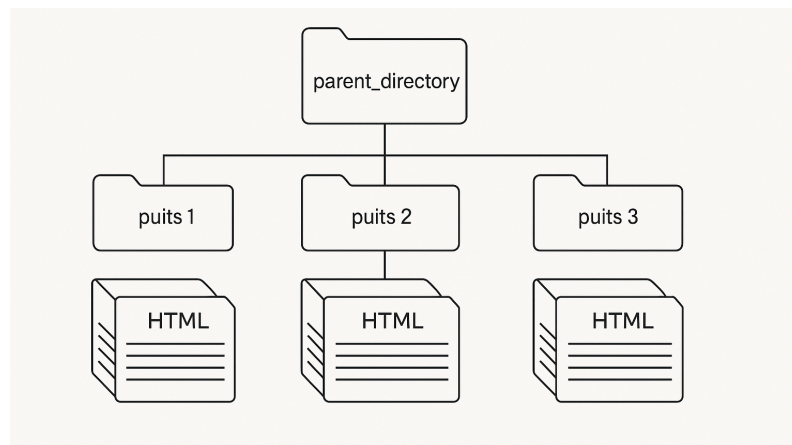


FIGURE 3.7 – Structure de répertoires illustrant l’organisation hiérarchique des fichiers DDR, où chaque sous-dossier représente un puits de forage.

3.3.2 Nettoyage et normalisation des rapports

Les rapports HTML bruts, bien que lisibles par un humain dans un navigateur, sont loin d’être des sources de données propres et normalisées. Ils peuvent contenir du code invalide, des structures incohérentes d’un rapport à l’autre, et du balisage servant uniquement à la présentation visuelle. La phase de nettoyage et de normalisation a pour objectif de transformer ce chaos en une structure interne standardisée, prérequis indispensable à une extraction d’information fiable.

Le cœur de cette étape est l’analyse syntaxique (parsing) du contenu HTML. Nous avons choisi la bibliothèque Python BeautifulSoup pour cette tâche, en raison de sa grande flexibilité et de sa tolérance aux erreurs. Contrairement à des parseurs XML stricts, BeautifulSoup peut interpréter du HTML "mal formé", ce qui est un atout majeur dans notre contexte. Il transforme la chaîne de caractères brute du rapport en un objet Python complexe, le *Document Object Model* (DOM), qui représente le document comme un arbre de nœuds (balises, attributs, texte). Pour optimiser les performances, nous avons configuré BeautifulSoup pour utiliser le parseur lxml en priorité, connu pour sa rapidité et son efficacité. Un mécanisme de repli vers le parseur natif de Python, `html.parser`, a été implémenté pour garantir que notre code puisse s’exécuter dans des environnements où lxml ne serait pas installé.

La normalisation se poursuit avec la localisation précise des sections d’information clés. Notre analyse exploratoire des DDR a révélé que la table des opérations journalières était l’élément le plus riche et le plus structuré. Par conséquent, notre script se focalise sur son identification. En examinant le code source de plusieurs rapports, nous avons constaté que cette table portait souvent un identifiant HTML unique et constant, par exemple `id="operationsInfoTable"`. Le script utilise donc la méthode `find("table", id=...)` pour localiser cet élément de manière non ambiguë. Cette approche, basée sur des identifiants, est beaucoup plus robuste qu’une recherche basée sur l’ordre ou la structure, qui pourrait varier d’un rapport à l’autre. Une fois la table localisée, son contenu HTML intégral est extrait et isolé du reste du document.

La dernière étape de normalisation est la conversion de cette structure HTML isolée en un format de données véritablement tabulaire. Nous utilisons pour cela la fonction `pd.read_html()` de la bibliothèque Pandas. Cette fonction puissante est capable d’interpréter le balisage `<table>`, `<tr>`, `<td>`, et `<th>` pour le transformer directement en un DataFrame, la structure de données centrale de l’écosystème d’analyse de données

Python. Le résultat est une représentation propre et structurée des opérations, prête pour les étapes suivantes.

3.3.3 Extraction des contenus des rapports

Une fois la normalisation effectuée, l'extraction systématique des contenus peut commencer. Cette phase vise à peupler nos structures de données avec les informations textuelles et numériques contenues dans le DOM.

- **Extraction de Données Tabulaires :** L'extraction des données tabulaires constitue une étape prioritaire pour structurer l'information. Au-delà de la table principale des opérations, les DDR contiennent une multitude d'autres tableaux d'intérêt, tels que l'inventaire des consommables, les propriétés de la boue de forage, le suivi des temps non productifs (NPT) ou la description des garnitures de forage (BHA). Notre approche programmatique consiste à développer des fonctions d'extraction spécifiques pour chaque type de tableau, en les localisant de manière robuste grâce à leurs identifiants HTML uniques (par exemple, `id="operationsInfoTable"`), une méthode bien plus fiable qu'un ciblage basé sur leur position. Une fois un tableau identifié, un script itère systématiquement sur ses éléments structurels – les lignes (`<tr>`) et les cellules (`<td>`, `<th>`) – pour en extraire le contenu textuel. Des traitements supplémentaires sont appliqués pour nettoyer ce contenu, notamment la suppression des espaces superflus, la conversion des formats numériques et la gestion des cellules fusionnées (`colspan`, `rowspan`). Chaque tableau ainsi traité est converti en un `DataFrame` Pandas distinct, prêt pour l'analyse, dont un extrait de code illustrant cette méthode est disponible en Annexe .
- **Extraction de Contenu Textuel Libre :** Les informations narratives sont tout aussi cruciales que les données tabulaires, car elles contiennent le contexte et l'expertise que les chiffres seuls ne peuvent fournir. Les résumés des opérations, les commentaires des superviseurs et les descriptions détaillées des problèmes constituent une mine d'or d'informations pour l'analyse sémantique. Notre processus d'extraction, dont un exemple de la logique est disponible en Annexe , repose sur une stratégie de ciblage des balises HTML qui encapsulent typiquement ce genre de contenu, telles que les paragraphes (`<p>`), les divisions (`<div>`) ou les éléments de listes. Une fois ces éléments localisés, leur contenu textuel est extrait de manière programmatique. Un soin particulier est apporté au nettoyage de ce texte brut pour le débarrasser de tout balisage résiduel et normaliser les espaces. Le texte pur ainsi obtenu est ensuite stocké, prêt à être utilisé soit pour alimenter directement nos modèles de traitement du langage, soit pour enrichir les enregistrements tabulaires correspondants avec des informations contextuelles.

Cette phase d'extraction aboutit à une collection de `DataFrames` et de variables textuelles qui représentent une version numérisée et structurée du contenu du rapport original.

3.3.4 Extraction d'informations pertinentes

Posséder les données brutes ne suffit pas. Pour qu'elles deviennent véritablement utiles, il est nécessaire d'en extraire des informations de plus haut niveau, c'est-à-dire de créer de nouvelles caractéristiques (features) par calcul et transformation. Cette étape d'ingénierie des caractéristiques est guidée par l'expertise du domaine du forage.

Calcul Détaillé de la Durée des Opérations

La durée est un indicateur de performance clé. Elle est calculée à partir des colonnes `Start time` et `End time`. Le processus est rigoureux : d'abord, nous normalisons les notations horaires (par exemple, 24:00 est converti en 00:00 du jour suivant). Ensuite, les chaînes de caractères sont converties en objets `datetime` de Python, en spécifiant le format attendu (%H:%M). Nous gérons explicitement les cas où une opération s'étend sur deux jours (par exemple, de 23 :00 à 01 :00). La différence est calculée, résultant en un objet `Timedelta`, qui est finalement converti en une valeur numérique en heures et stocké dans une nouvelle colonne `Duration (hours)`.

Extraction et Standardisation de la Colonne `Remark`

La colonne des remarques (`Remark`) est traitée avec une attention particulière. Elle contient des observations en langage naturel qui sont essentielles pour comprendre les causes des déviations de performance. Notre script garantit que cette colonne est présente dans chaque `DataFrame` final. Si elle est absente d'un rapport source, une colonne vide est créée pour maintenir une structure de données homogène. Tout son contenu est explicitement converti en type chaîne de caractères (`str`) pour éviter les erreurs lors des traitements NLP ultérieurs.

Calcul Précis de la Distance Forée et du Taux de Pénétration (ROP)

Le Taux de Pénétration (`Rate of Penetration - ROP`) est l'un des indicateurs de performance les plus importants. Son calcul nécessite une étape intermédiaire : la détermination de la distance forée pour chaque opération de forage.

1. **Distance forée** (`Distance Drilled`) : Elle est calculée comme la différence entre la profondeur finale (`End Depth mMD`) de l'opération en cours et celle de l'opération précédente. Cela nécessite de trier les opérations par ordre chronologique et de gérer le cas de la toute première opération de forage du puits.
2. **Taux de pénétration** (`Penetration Rate (m/hr)`) : Il est obtenu par le simple rapport `Distance Drilled / Duration (hours)`. Une attention particulière est portée à la gestion des cas de division par zéro (opérations de durée nulle) pour éviter les erreurs d'exécution.

Intégration du Nom du Fichier Source et Consolidation des Données

Pour assurer la traçabilité et permettre un retour facile au document source en cas de besoin de vérification, le nom du fichier HTML d'origine est ajouté comme une nouvelle colonne (`Report`) à chaque enregistrement. Enfin, l'ensemble des `DataFrames` (un par rapport journalier) d'un même puits est concaténé en un unique `DataFrame` consolidé. Ce dernier est ensuite sauvegardé dans un fichier au format CSV, par exemple `operations_info.csv`, dans le dossier correspondant au puits. Ce fichier CSV final représente l'aboutissement du processus d'extraction et constitue le point de départ pour toutes les analyses futures.

	Report	report_date	well_name	original_index_well	HPHT	Tight_well	Pressure_Test_Type	Sample_Point	Fluid_Type	Status	Fluid_Density_gcm3	Funnel_Visc_s
1	1997_07_28.html	1997-07-28	15_9_19_A	0	Y	Y	leak off test	Active pit	ULTIDRILL	normal	1.5	143.0
2	1997_08_05.html	1997-08-05	15_9_19_A	1	Y	Y	leak off test	Flowline	ULTIDRILL	normal	1.54	70.0
3	1997_08_06.html	1997-08-06	15_9_19_A	2	Y	Y	leak off test	Flowline	ULTIDRILL	normal	1.54	56.0
4	1997_08_11.html	1997-08-11	15_9_19_A	3	Y	Y	leak off test	Flowline	ULTIDRILL	normal	1.55	68.0
5	1997_08_12.html	1997-08-12	15_9_19_A	4	Y	Y	leak off test	Flowline	ULTIDRILL	normal	1.55	64.0
6	1997_08_13.html	1997-08-13	15_9_19_A	5	Y	Y	leak off test	Flowline	ULTIDRILL	normal	1.55	63.0
7	1997_08_14.html	1997-08-14	15_9_19_A	6	Y	Y	leak off test	Flowline	ULTIDRILL	normal	1.55	60.0
8	1997_08_15.html	1997-08-15	15_9_19_A	7	Y	Y	leak off test	Flowline	ULTIDRILL	normal	1.55	59.0
9	1997_08_16.html	1997-08-16	15_9_19_A	8	Y	Y	leak off test	Flowline	ULTIDRILL	normal	1.55	75.0
10	1997_08_17.html	1997-08-17	15_9_19_A	9	Y	Y	leak off test	Flowline	ULTIDRILL	normal	1.55	74.0
11	1997_08_18.html	1997-08-18	15_9_19_A	10	Y	Y	leak off test	Flowline	ULTIDRILL	normal	1.55	67.0
12	1997_08_19.html	1997-08-19	15_9_19_A	11	Y	Y	leak off test	Flowline	ULTIDRILL	normal	1.55	68.0
13	1997_08_20.html	1997-08-20	15_9_19_A	12	Y	Y	leak off test	Flowline	ULTIDRILL	normal	1.55	62.0
14	1997_08_21.html	1997-08-21	15_9_19_A	13	Y	Y	leak off test	Flowline	ULTIDRILL	normal	1.55	69.0
15	1997_08_22.html	1997-08-22	15_9_19_A	14	Y	Y	leak off test	Flowline	ULTIDRILL	normal	1.55	70.0
16	1997_08_23.html	1997-08-23	15_9_19_A	15	Y	Y	leak off test	Active pit	ULTIDRILL	normal	1.54	71.0
17	1997_08_24.html	1997-08-24	15_9_19_A	16	Y	Y	leak off test	Active pit	ULTIDRILL	normal	1.55	70.0
18	1997_08_25.html	1997-08-25	15_9_19_A	17	Y	Y	leak off test	Active pit	ULTIDRILL	normal	1.55	71.0
19	1997_08_26.html	1997-08-26	15_9_19_A	18	Y	Y	leak off test	Active pit	ULTIDRILL	normal	1.55	75.0

FIGURE 3.8 – image qui represent une partie du fichier csv final.

3.3.5 Compréhension des données

La phase de compréhension des données est une étape cruciale qui va bien au-delà de la simple analyse statistique. Dans un domaine aussi technique et complexe que le forage, une interprétation correcte des données ne peut se faire sans une connaissance approfondie du métier. C'est pourquoi notre démarche ne s'est pas limitée à une exploration statistique des chiffres, mais a été fondamentalement guidée par une collaboration étroite avec un expert du domaine au sein de SLB.

L'objectif principal de cette phase était de traduire les données brutes et le jargon technique présents dans les rapports journaliers de forage (DDR) en une compréhension fonctionnelle et exploitable. Le processus s'est articulé autour des axes suivants :

- **Décodage sémantique du vocabulaire technique** : Les DDR regorgent de termes, d'acronymes (comme ROP, WOB, BHA, NPT) et de descriptions d'opérations spécifiques au forage. Grâce aux sessions de travail avec notre expert référent, nous avons pu décoder ce vocabulaire, comprendre la signification physique de chaque paramètre et la nature des différentes opérations décrites dans les rapports. Cette clarification sémantique a été indispensable pour ne pas faire de contresens dans l'interprétation des données.
- **Identification des indicateurs de performance clés (KPIs)** : Toutes les données ne sont pas égales en importance. Guidés par l'expertise métier, nous avons pu identifier les paramètres qui sont considérés dans l'industrie comme les indicateurs les plus pertinents pour évaluer la performance et la sécurité d'une opération de forage. Des variables comme le Taux de Pénétration (ROP), le couple de rotation, la pression de pompe, et la durée des opérations ont été confirmées comme étant des caractéristiques centrales. Ce processus de sélection a permis de concentrer nos efforts sur les données ayant le plus de valeur.
- **Compréhension des relations de cause à effet** : L'expert nous a aidés à comprendre les relations complexes et souvent non linéaires entre les différents paramètres. Par exemple, comment une variation de la densité de la boue peut affecter la pression au fond du puits, ou comment une augmentation soudaine du couple peut être un signe avant-coureur d'un coincement de garniture. Cette compréhension des mécanismes physiques sous-jacents a été fondamentale pour orienter la conception

de notre application de visualisation et, plus tard, l'ingénierie des caractéristiques pour le modèle d'IA.

En somme, cette phase a été moins une *Exploratory Data Analysis* (EDA) au sens statistique traditionnel qu'une *Expert-Guided Data Understanding*. L'expertise humaine a agi comme un filtre et un guide, nous permettant de naviguer dans la complexité des données de forage, de sélectionner les variables les plus significatives pour la visualisation et la modélisation, et de poser des bases solides pour la construction d'une solution réellement pertinente pour les utilisateurs finaux.

3.3.6 Préparation et finalisation des données pour l'analyse et la modélisation

Cette ultime étape de traitement fait le pont entre le dataset consolidé, mais encore brut, et son utilisation effective dans nos deux modules principaux : le module de visualisation interactive et le pipeline de modélisation prédictive. L'objectif est d'effectuer les transformations finales qui rendront les données non seulement propres et cohérentes, mais aussi riches en informations et dans un format optimal pour leurs applications respectives.

Nettoyage et Prétraitement Final

Même après les premières étapes, des imperfections peuvent subsister. Cette phase inclut un traitement systématique des valeurs manquantes. Selon la nature de la variable et le pourcentage de données manquantes, nous avons choisi la stratégie la plus appropriée, comme l'imputation par la médiane (robuste aux valeurs aberrantes) ou la suppression des rares lignes où une information critique serait absente.

De plus, les caractéristiques numériques sont normalisées. La mise à l'échelle (par exemple, la mise à l'échelle Min-Max qui contraint les valeurs entre 0 et 1) est indispensable. Cette étape garantit que toutes les caractéristiques contribuent sur une base comparable, ce qui est crucial non seulement pour de nombreux algorithmes d'apprentissage automatique sensibles à l'échelle des données, mais aussi pour certaines techniques de visualisation où des échelles très différentes pourraient rendre les graphiques illisibles ou trompeurs.

Ingénierie des Caractéristiques (Feature Engineering)

C'est sans doute l'étape la plus créative du processus, où l'expertise du domaine rencontre la science des données. Plutôt que de nous fier uniquement aux variables brutes extraites, nous en avons créé de nouvelles, conçues pour capturer des informations plus complexes et contextuelles.

- **Caractéristiques d'interaction :** Nous avons créé des variables en combinant des caractéristiques existantes, guidés par la physique du forage. Par exemple, le rapport entre le couple de rotation et le poids sur l'outil peut être un indicateur de l'efficacité du forage plus puissant qu'aucune de ces deux variables prises isolément.
- **Encodage des variables catégorielles :** Les variables non numériques, comme le type d'opération ou la phase de forage, sont converties en une représentation numérique via des techniques comme le One-Hot Encoding. Cette transformation est

nécessaire pour que les algorithmes mathématiques puissent traiter ces informations, et elle permet également de les utiliser facilement comme des filtres ou des catégories dans nos outils de visualisation.

Préparation spécifique pour la modélisation : Gestion du Déséquilibre des Classes

Alors que les étapes précédentes préparent un dataset général, une dernière transformation est nécessaire, spécifiquement pour l'entraînement des modèles de prédiction des risques. Le défi le plus important pour cette tâche est le déséquilibre extrême des classes : les événements de risque (pannes, coincements, etc.) sont, par définition, beaucoup plus rares que les opérations normales.

Un modèle entraîné sur de telles données apprendrait très rapidement à toujours prédire "pas de risque", atteignant une haute précision mais étant complètement inutile en pratique. Pour forcer le modèle à apprendre les motifs subtils qui précèdent un événement rare, il est impératif d'équilibrer le jeu de données d'entraînement. Nous avons opté pour une technique de suréchantillonnage (*upsampling*) de la classe minoritaire. Concrètement, cela consiste à dupliquer aléatoirement les exemples de la classe "risque" dans l'ensemble d'entraînement jusqu'à ce qu'un équilibre satisfaisant avec la classe "pas de risque" soit atteint. Il est crucial de n'appliquer cette technique que sur l'ensemble d'entraînement, afin que les ensembles de validation et de test conservent la distribution originale des données et fournissent une évaluation réaliste de la performance du modèle en conditions réelles.

3.4 Conception et Implémentation des Modules d'Intelligence Artificielle

Cette section constitue le cœur de notre projet, détaillant la conception et la mise en œuvre de nos contributions en intelligence artificielle. Notre solution s'articule autour de trois modules complémentaires et innovants, conçus pour extraire une valeur maximale des données de forage :

1. Un module de modélisation prédictive basé sur des algorithmes d'apprentissage automatique classique pour l'anticipation des risques.
2. Un système d'analyse hybride (MCP+LLM) combinant les prédictions quantitatives avec la puissance de raisonnement d'un Grand Modèle de Langage pour fournir des explications contextuelles.
3. Un assistant conversationnel technique (Chatbot+RAG) pour un accès direct et interactif à la connaissance métier.

Nous décrirons ci-après la méthodologie, l'architecture et les résultats obtenus pour chacun de ces modules.

3.4.1 Modélisation Prédictive des Risques par Apprentissage Automatique

Le premier objectif de notre travail d'IA était de développer un système capable d'anticiper les risques opérationnels en se basant sur les données numériques et séquentielles issues des rapports journaliers de forage. Notre démarche de modélisation a été itérative, commençant par une large exploration avant de se spécialiser, et s'adaptant en fonction des résultats intermédiaires.

3.4.1.a Phase 1 : Exploration Initiale des Modèles avec LazyPredict

Avant de nous engager sur une architecture complexe, nous avons mené une phase d'exploration rapide pour évaluer un large éventail de modèles d'apprentissage automatique classiques. Pour ce faire, nous avons utilisé la bibliothèque LazyPredict. Cet outil nous a permis de tester simultanément plusieurs dizaines d'algorithmes de classification et d'obtenir un classement de leurs performances initiales sur notre dataset. Le Tableau [3.1](#) présente un résumé de ces résultats.

TABLE 3.1 – Performances comparées de différents modèles de classification obtenues avec LazyPredict.

Modèle	Accuracy	Balanced Acc.	ROC AUC	F1 Score	Temps (s)
NearestCentroid	0.68	0.58	0.58	0.66	0.02
LinearSVC	0.71	0.56	0.56	0.67	0.03
KNeighborsClassifier	0.70	0.55	0.55	0.66	0.02
PassiveAggressiveClassifier	0.70	0.54	0.54	0.64	0.02
ExtraTreeClassifier	0.66	0.53	0.53	0.63	0.03
LabelPropagation	0.67	0.53	0.53	0.63	0.09
LabelSpreading	0.67	0.53	0.53	0.63	0.09
GaussianNB	0.71	0.53	0.53	0.63	0.02
CalibratedClassifierCV	0.71	0.52	0.52	0.62	0.13
Perceptron	0.67	0.51	0.51	0.62	0.02
RandomForestClassifier	0.66	0.51	0.51	0.62	0.32
BernoulliNB	0.31	0.51	0.51	0.18	0.03
SGDClassifier	0.71	0.51	0.51	0.60	0.02
ExtraTreesClassifier	0.70	0.50	0.50	0.60	0.27
SVC	0.70	0.50	0.50	0.60	0.13
DummyClassifier	0.71	0.50	0.50	0.59	0.02
RidgeClassifierCV	0.70	0.50	0.50	0.59	0.13
LinearDiscriminantAnalysis	0.70	0.50	0.50	0.59	0.03
NuSVC	0.70	0.50	0.50	0.59	0.15
RidgeClassifier	0.70	0.50	0.50	0.59	0.02
LogisticRegression	0.70	0.49	0.49	0.59	0.03
QuadraticDiscriminantAnalysis	0.66	0.49	0.49	0.60	0.07
AdaBoostClassifier	0.52	0.46	0.46	0.54	0.29
BaggingClassifier	0.51	0.45	0.45	0.52	0.09
XGBClassifier	0.50	0.45	0.45	0.52	0.11
LGBMClassifier	0.51	0.44	0.44	0.52	0.08
DecisionTreeClassifier	0.50	0.44	0.44	0.52	0.03

L’analyse de ces résultats montre que, bien que certains modèles simples comme NearestCentroid ou LinearSVC obtiennent des scores corrects, aucun modèle classique ne se détache de manière spectaculaire. Les métriques comme la *Balanced Accuracy* et le *ROC AUC* peinent à dépasser 0.60, ce qui indique une difficulté générale des modèles à bien discriminer les classes sur ce problème complexe et déséquilibré. Cette exploration a renforcé notre conviction que la nature séquentielle des données n’était pas pleinement exploitée et qu’une approche plus spécialisée était nécessaire.

3.4.1.b Phase 2 : Modélisation Approfondie et Résultats avec LSTM

Guidés par les limites observées lors de l’exploration initiale, nous avons sélectionné les réseaux de neurones récurrents de type Long Short-Term Memory (LSTM) pour une modélisation plus approfondie. Leur architecture est en effet spécifiquement conçue pour analyser des séquences et capturer les dépendances temporelles, une caractéristique que les modèles classiques ne peuvent pas gérer nativement.

Stratégie de Modélisation. Pour adresser le problème de manière ciblée, nous avons d’abord créé quatre variables cibles (targets) binaires distinctes, chacune correspondant à un type de risque spécifique (défaillance d’équipement, instabilité, etc.). Nous avons ensuite développé un modèle LSTM distinct pour chacune de ces quatre variables, ce qui nous a permis d’optimiser l’apprentissage pour chaque type de problème.

Obtention des Meilleurs Résultats de Classification Binaire. Après entraînement, les modèles LSTM ont démontré des performances significativement supérieures à tous les modèles classiques testés précédemment. L'évaluation de ces modèles sur leurs tâches de classification binaire respectives a constitué une étape importante et une réussite notable de notre projet. À titre d'exemple, le Tableau 3.2 et la Figure 3.9 présentent les résultats obtenus pour l'une de ces tâches, qui sont représentatifs de l'amélioration globale.

TABLE 3.2 – Rapport de classification pour un modèle LSTM, illustrant les meilleures performances obtenues sur une tâche binaire.

	precision	recall	f1-score	support
No Problem (0)	0.86	0.80	0.83	366
Problem (1)	0.48	0.58	0.52	114
accuracy			0.75	480
macro avg	0.67	0.69	0.68	480
weighted avg	0.77	0.75	0.76	480

FIGURE 3.9 – Matrice de confusion du modèle LSTM pour une tâche binaire.

		Prédiction	
		No Problem	Problem
Réalité	No Problem	294	72
	Problem	48	66

Avec une précision pondérée de 0.77 et un rappel de la classe "Problème" atteignant 0.58, ces résultats ont validé sans équivoque le choix de l'architecture LSTM comme étant la plus adaptée à notre problème.

3.4.1.c Phase 3 : Raffinement de l'Approche vers une Évaluation par Niveaux de Risque

Forts de ces bons résultats, nous avons cherché à aller plus loin pour augmenter la valeur opérationnelle de nos prédictions. Nous avons constaté qu'une simple sortie binaire ("Risque" / "Pas de Risque"), bien que statistiquement correcte, pouvait être améliorée pour fournir une information plus nuancée et plus utile à un ingénieur sur le terrain. Un rappel de 0.58, bien que le meilleur obtenu, signifie toujours que 42% des problèmes ne sont pas détectés, ce qui reste un point d'amélioration critique pour la sécurité.

C'est pourquoi nous avons décidé de raffiner notre approche d'évaluation. Plutôt que de nous fier à un seuil unique de 0.5, nous avons développé une méthode de catégorisation par niveaux de risque. Cette méthode exploite la probabilité de sortie brute du modèle LSTM pour classer chaque prédiction dans l'une des trois catégories suivantes : **Faible**, **Moyen**, et **Haut Risque**.

Cette nouvelle grille de lecture change la philosophie d'évaluation : l'objectif n'est plus seulement d'avoir raison, mais de ne jamais avoir tort de manière dangereuse. La priorité absolue devient de s'assurer qu'aucun événement réel ne soit classé dans la catégorie "Faible Risque". Une alerte "Moyen" ou "Haut Risque" est alors considérée comme une prédiction réussie, car elle remplit sa fonction première : attirer l'attention sur une situation potentiellement dangereuse.

3.4.1.d Évaluation des Modèles et Interprétation des Résultats

L'évaluation a été menée sur l'ensemble de test. La philosophie d'évaluation adoptée ici privilégie la capacité du modèle à ne pas sous-estimer un risque réel. Autrement dit, l'objectif primordial est de minimiser le nombre d'événements réels classés à tort dans la catégorie *Faible Risque*. Une prédiction *Moyen Risque* ou *Haut Risque* est considérée comme une alerte pertinente, même si l'événement ne se matérialise pas à chaque fois, car elle incite à la vigilance.

Les métriques de classification binaire standards (Précision, Rappel, F1-score avec un seuil de probabilité de 0.5) sont rapportées pour information. Cependant, l'analyse principale se concentre sur la performance de la catégorisation des risques (*Faible*, *Moyen*, *Élevé*) définie par des seuils de probabilité spécifiques à chaque modèle.

1. Modèle pour la Défaillance d'Équipement (target_confirmed_equipment_failure_1)

Seuils : Faible (< 0.15), Moyen ($0.15\text{--}0.60$), Élevé (≥ 0.60)

- Faible Risque : 212 prédictions. Aucun événement réel n'a été manqué (0.0 %).
- Moyen Risque : 130 cas, dont 36.9 % étaient des événements réels.
- Haut Risque : 138 cas, dont 47.8 % étaient des événements réels.

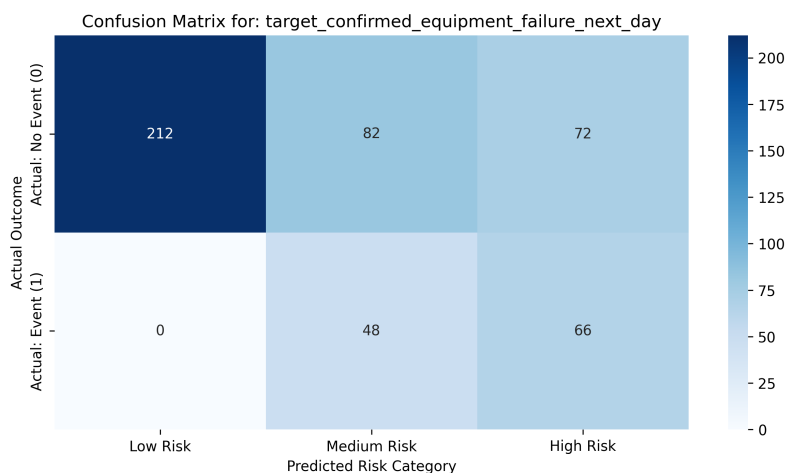


FIGURE 3.10 – confusion matrix target confirmed equipment failure next day

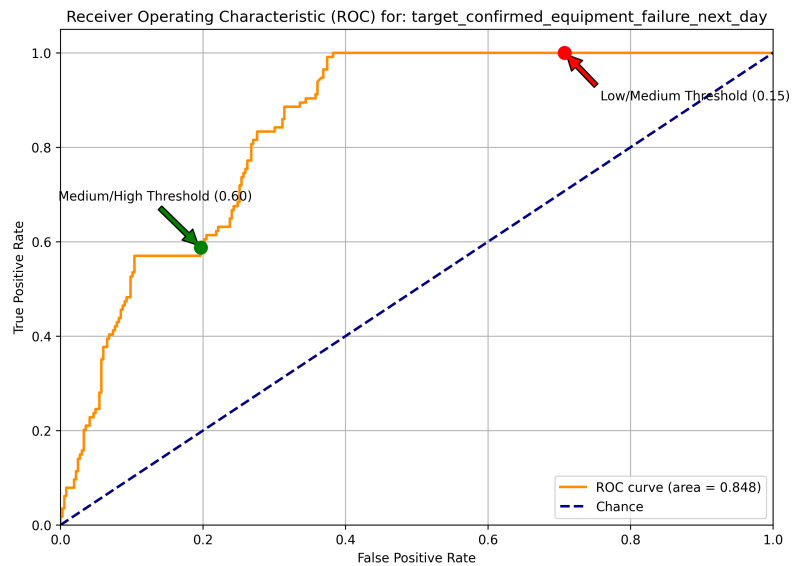


FIGURE 3.11 – roc curve target confirmed equipment failure next day

Interprétation : Le modèle est excellent pour la prévention, atteignant l’objectif de zéro événement manqué dans la catégorie *Faible Risque*. Il démontre une bonne discrimination entre les risques moyens et élevés.

2. Modèle pour l’Instabilité du Puits (target_confirmed_wellbore_instability_next_day)

Seuils : Faible (< 0.33), Moyen ($0.33\text{--}0.60$), Élevé (≥ 0.60)

- Faible Risque : 99 prédictions. Aucun événement réel n’a été manqué (0.0 %).
- Moyen Risque : 358 cas, dont seulement 2.2 % étaient des événements réels.
- Haut Risque : 23 cas, dont 17.4 % étaient des événements réels.

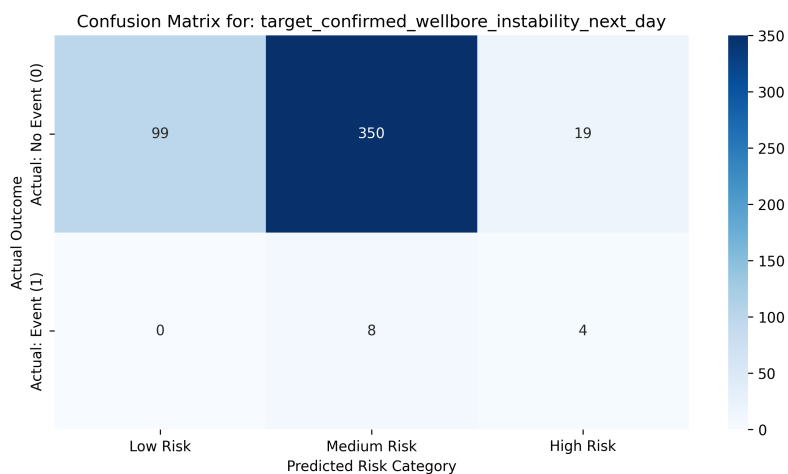


FIGURE 3.12 – confusion matrix target confirmed wellbore instability next day

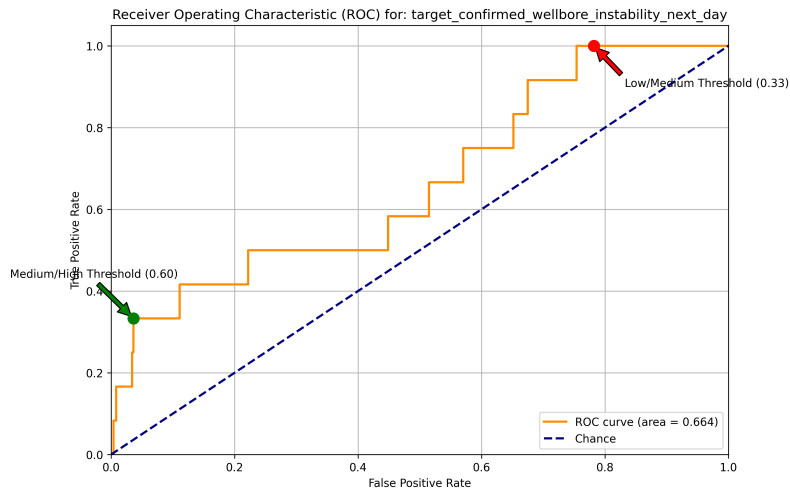


FIGURE 3.13 – roc curve target confirmed wellbore instability next day

Interprétation : Performance parfaite sur l’objectif de sécurité (zéro événement manqué). La progression nette du taux d’événements (de 2.2 % à 17.4 %) montre que le modèle hiérarchise bien les risques, même si le taux d’événements réels reste faible dans toutes les catégories.

3. Modèle pour le Coincement de Garniture (target_confirmed_stuck_pipe_next_day)

Seuils : Faible (< 0.28), Moyen ($0.28-0.60$), Élevé (≥ 0.60)

- Faible Risque : 41 prédictions. Aucun événement réel n’a été manqué (0.0 %).
- Moyen Risque : 374 cas, dont 2.7 % étaient des événements réels.
- Haut Risque : 65 cas, dont seulement 1.5 % étaient des événements réels.

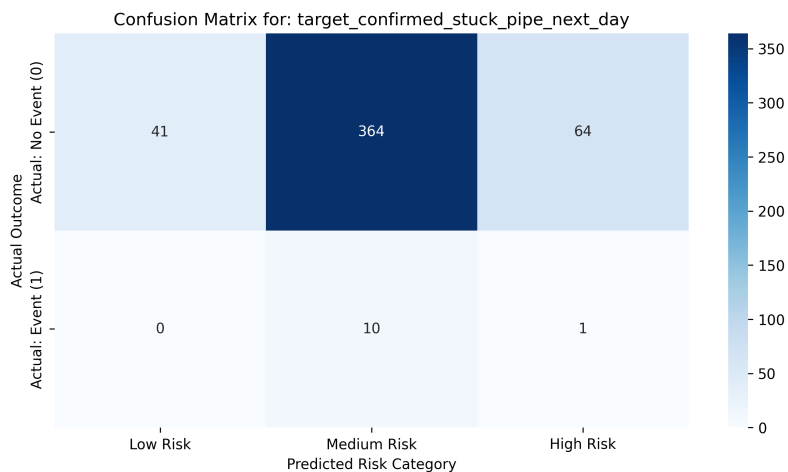


FIGURE 3.14 – confusion matrix target confirmed stuck pipe next day

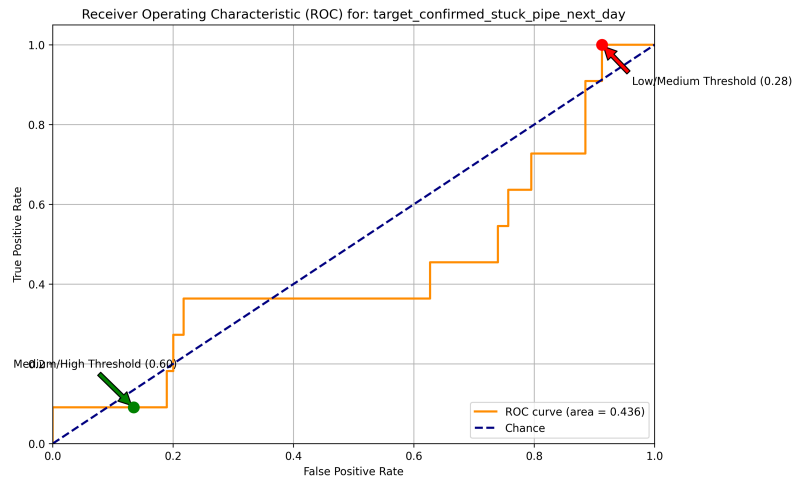


FIGURE 3.15 – roc curve target confirmed stuck pipe next day

Interprétation : Le modèle est très performant pour ne manquer aucun cas à faible risque. Cependant, il échoue à discriminer entre le risque moyen et élevé, le taux d'événements étant même plus bas pour la catégorie *Haut Risque*. Ce résultat est certainement lié à l'extrême rareté des événements de ce type dans les données.

4. Modèle pour les Éruptions (target_confirmed_blowout_wc_next_day)

Seuils : Faible (< 0.23), Moyen ($0.23-0.60$), Élevé (≥ 0.60)

- Faible Risque : 91 cas, dont 1.1 % étaient des événements réels.
- Moyen Risque : 307 cas, dont 3.6 % étaient des événements réels.
- Haut Risque : 82 cas, dont 7.3 % étaient des événements réels.

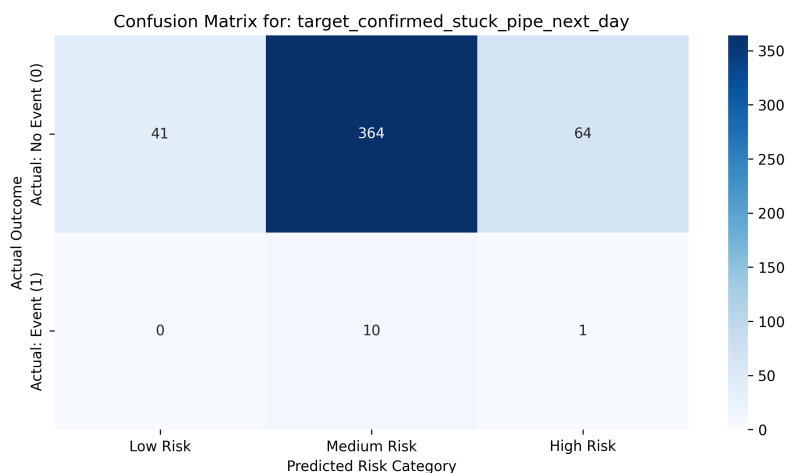


FIGURE 3.16 – confusion matrix target confirmed stuck pipe next day

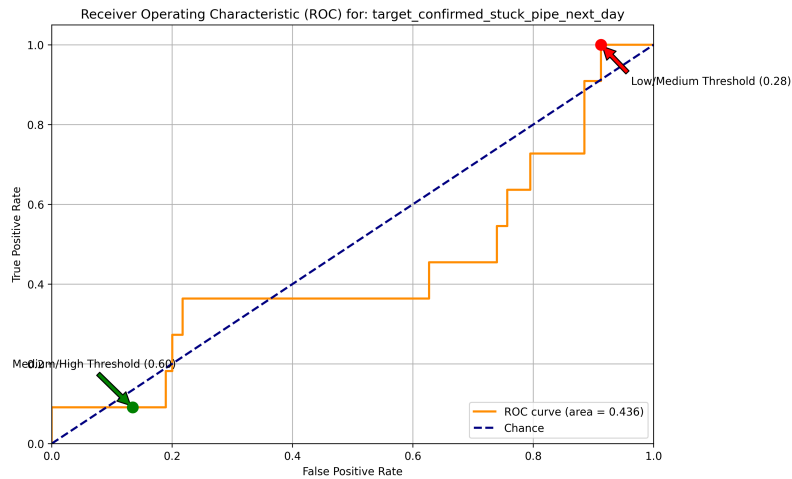


FIGURE 3.17 – roc curve target confirmed stuck pipe next day

Interprétation : Le modèle montre une gradation logique du risque. Cependant, il manque son objectif principal en classant un événement critique dans la catégorie *Faible Risque* (1.1 %). Étant donné la gravité d’une éruption, ce résultat, bien que faible, reste préoccupant et potentiellement inacceptable pour une utilisation opérationnelle.

Synthèse de l’Évaluation (Perspective Préventive)

En adoptant une perspective centrée sur la prévention et la minimisation des risques non détectés, les modèles LSTM développés montrent des performances contrastées :

- Les modèles pour la défaillance d’équipement, l’instabilité de puits et le coincement de garniture sont très performants pour ne pas classer à tort un événement réel en *Faible Risque*, atteignant ainsi l’objectif de sécurité primordial.
- Le modèle pour le coincement de garniture manque cependant de pouvoir de discrimination entre les niveaux de risque plus élevés.
- Le modèle pour les éruptions est le plus critique. Bien que le taux d’erreur soit faible (1.1 %), le fait de manquer ne serait-ce qu’un seul événement de cette nature représente un risque opérationnel majeur. Un réajustement des seuils ou un réentraînement est impératif pour ce modèle.

L’approche multi-niveaux de catégorisation par seuils s’avère donc pertinente pour les cas d’usage opérationnels. Elle permet d’instaurer une gradation utile dans les alertes tout en privilégiant la prudence. Les performances observées posent une base solide pour l’amélioration continue du système d’alerte.

3.5 Analyse Hybride des Risques de Forage : Combinaison de Modèles de Machine Learning Classiques et d’un Modèle de Langage Large (LLM) via une Architecture RAG et MCP

L’identification proactive des risques à partir des Rapports Journaliers de Forage (DDR) constitue un défi majeur. Pour y répondre, une approche hybride innovante a été conçue.

Elle combine la précision des modèles de Machine Learning (ML) classiques pour des tâches de prédiction spécifiques (développés en section 3.5.2.a) avec la capacité de compréhension contextuelle et de génération de texte d'un Modèle de Langage Large (LLM) via une architecture de Génération Augmentée par la Récupération (RAG). Cette synergie est orchestrée par un Pipeline de Combinaison de Modèles (MCP). Cette section détaille la méthodologie, les composants clés, l'infrastructure d'hébergement du LLM, et un exemple d'exécution de ce module [24].

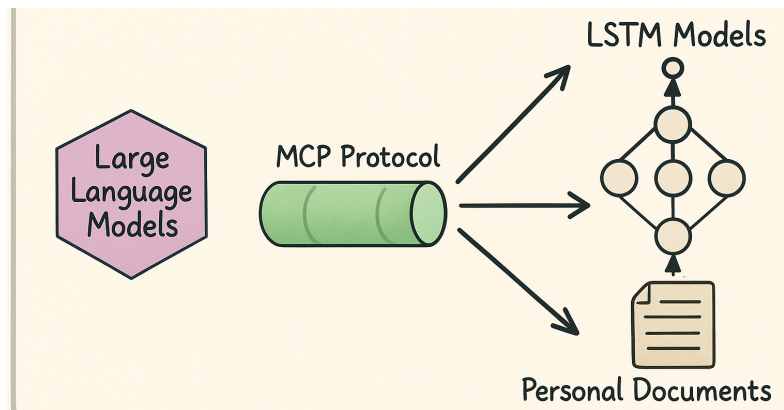


FIGURE 3.18 – Exploiter le Protocole MCP pour Connecter les LLM aux Modèles LSTM et aux Données Personnelles

3.5.1 Principes Fondamentaux de l'Approche Hybride

Notre approche repose sur deux piliers fondamentaux :

- **Génération Augmentée par la Récupération (RAG)** : améliore la pertinence du LLM en lui fournissant un contexte pertinent, extrait d'une base de connaissances spécialisée sur le forage.
- **Pipeline de Combinaison de Modèles (MCP)** : intègre les prédictions issues des modèles de ML classiques (par exemple, les modèles LSTM pour la prédiction de `Target_NextDay_Event_Is_Equipment_Failure`) comme des signaux d'alerte supplémentaires dans le prompt soumis au LLM.

Cette combinaison vise à tirer parti des forces respectives de chaque type de modèle : la capacité des modèles ML à identifier des motifs dans des données structurées et séquentielles pour des risques prédéfinis, et la flexibilité du LLM pour le raisonnement contextuel sur du texte non structuré et la génération d'explications nuancées.

3.5.2 Composants Clés de l'Architecture Hybride

L'architecture hybride est composée des modules suivants :

3.5.2.a Module de Prédiction par Apprentissage Automatique

Ce module constitue la première étape de notre pipeline d'analyse et a pour rôle de générer des signaux de risque quantitatifs à partir des données structurées.

- **Utilisation de modèles pré-entraînés** : Le module ne procède pas à un entraînement en temps réel. Il charge et utilise les quatre modèles LSTM spécialisés qui ont été préalablement entraînés, validés et sauvegardés sur le disque. Ces modèles,

enregistrés au format .pkl encapsulent toute la connaissance apprise lors de la phase d'entraînement.

- **Processus d'inférence et sortie** : Pour chaque nouveau DDR, ce module exécute un processus d'inférence en appliquant les quatre modèles aux données d'entrée. Il génère en sortie un ensemble de prédictions, typiquement un dictionnaire structuré qui mappe chaque type de risque à son niveau prédit (Faible, Moyen, ou Élevé). Cette sortie structurée constitue une information cruciale pour la suite du pipeline.

3.5.2.b Base de Connaissances et Indexation (FAISS) pour RAG

Une base de connaissances constituée de documents techniques de forage est prétraitée, segmentée en chunks, puis vectorisée à l'aide du modèle all-MiniLM-L6-v2. Ces vecteurs sont indexés dans une base FAISS pour permettre une recherche sémantique rapide.

- La fonction `load_indexed_knowledge` est responsable du chargement de l'index FAISS et des chunks de texte.

3.5.2.c Modèle d'Embedding pour les Requêtes RAG

Le même modèle all-MiniLM-L6-v2 est utilisé pour vectoriser les requêtes issues du DDR afin d'interroger l'index FAISS.

- L'initialisation est gérée par la fonction `initialize_embedding_model`.

3.5.2.d Module de Récupération (Retrieval) pour RAG

Ce module identifie et extrait les chunks de texte les plus pertinents de la base de connaissances FAISS en réponse à une requête dérivée du DDR.

- La fonction `retrieve_relevant_chunks` effectue cette recherche sémantique.

3.5.2.e Modèle de Langage Large (LLM) et son Infrastructure

Le LLM est au cœur de l'analyse contextuelle et de la synthèse des informations.

- *Modèle Utilisé* : meta-llama/Meta-Llama-3-8B-Instruct, un modèle performant pour le suivi d'instructions et la génération de texte.
- *Hébergement et Exécution* : Le LLM a été exécuté dans un environnement Kaggle Notebook, bénéficiant ainsi des ressources GPU (Nvidia Tesla T4) nécessaires à son inférence. La bibliothèque transformers de Hugging Face a été utilisée pour charger et interagir avec le modèle.
- *Connectivité* : Pour permettre à notre application web locale (développée avec Flask) de communiquer avec le LLM hébergé sur Kaggle, la librairie ngrok a été employée. ngrok expose le service Kaggle (ou un simple serveur API Flask sur Kaggle faisant proxy vers le LLM) à une URL publique sécurisée, rendant ainsi le LLM accessible depuis l'application locale.
- *Initialisation du Modèle* : La fonction `initialize_local_llm` gère le chargement du modèle et de son tokenizer sur Kaggle.
- *Interaction et Prompting* : La fonction `query_llm_with_context_hf_local` construit un prompt enrichi pour le LLM, incluant :
 - Un message système définissant le rôle et la tâche du LLM.

- Des exemples "few-shot" pour guider le format de sortie.
- Le contenu textuel du DDR à analyser.
- Le contexte pertinent récupéré par le module RAG.
- Les alertes issues des modèles ML classiques, par exemple :
{"Risque de Défaillance Equipement": "Élevé (Prob: 0.75)", "Instabilité Puits": "Moyen"}.

3.5.2.f Post-traitement et Structuration de la Sortie

La réponse brute du LLM est traitée pour en extraire l'information structurée.

- Analyse de la réponse textuelle, parsing robuste du JSON (incluant la gestion des blocs de code markdown “`json ...`”), et gestion des erreurs de formatage.

3.5.3 Processus d'Analyse d'un DDR (Workflow Hybride RAG+MCP)

Le processus complet, idéalement orchestré par une fonction principale telle que `analyze_ddr_hybrid`, se déroule séquentiellement :

1. **Pré-analyse ML** : Les modèles ML classiques (section 3.5.2.a) traitent les données du DDR pour générer des prédictions de risques spécifiques.
2. **Requête RAG** : Une requête est formulée à partir du contenu textuel du DDR.
3. **Récupération de Contexte RAG** : Les chunks pertinents sont extraits de la base de connaissances FAISS.
4. **Construction du Prompt LLM** : Le prompt final est assemblé, intégrant le DDR, le contexte RAG, et les alertes des modèles ML.
5. **Interrogation du LLM** : Le prompt est envoyé (via ngrok si nécessaire) au LLM hébergé sur Kaggle.
6. **Génération de l'Analyse** : Le LLM produit une analyse textuelle, justifiant les risques identifiés en se basant sur toutes les informations fournies.
7. **Structuration de la Sortie** : La réponse du LLM est parsée et formatée en une structure JSON exploitable.

3.5.4 Exemple de Sortie LLM et Interprétation

Suite à l'analyse d'un DDR et des alertes ML, le LLM pourrait générer une sortie structurée comme suit. Supposons qu'un modèle LSTM ait prédit un “Haut Risque de Coincement” et que le DDR contienne la phrase “*augmentation significative du couple de rotation observée lors du forage de la section X*”.

Listing 3.1 – Exemple de sortie JSON structurée générée par le LLM

```

1 [
2   {
3     "Risk": "Stuck Pipe (Coincement de Garniture)",
4     "Level": "High",
5     "How Identified": {
6       "ML Alert": "LSTM model predicted 'High Risk' for Stuck
      ↳ Pipe.",

```

```

7      "DDR Evidence": "operations[i].remark: 'augmentation
      ↳ significative du couple de rotation observée lors
      ↳ du forage de la section X'",
8      "RAG Context Snippet": "Document
      ↳ 'BestDrillingPractices.pdf', Section 4.2: 'Une
      ↳ augmentation soudaine ou progressive du couple de
      ↳ rotation est un indicateur clé d'un risque
      ↳ imminent de coincement.'",
9      "Reasoning": "L'alerte du modèle ML est corroborée par
      ↳ l'observation d'une augmentation du couple dans le
      ↳ DDR. Ce symptôme est un signe avant-coureur de
      ↳ coincement de garniture, comme mentionné dans la
      ↳ documentation technique (contexte RAG).",
10     },
11     "How to Avoid this Risk (Solution)": "Réduire
      ↳ immédiatement le poids sur l'outil (WOB), augmenter
      ↳ le débit de la boue pour améliorer le nettoyage du
      ↳ trou. Vérifier les paramètres de la boue. Si le
      ↳ couple reste élevé, envisager une manœuvre de
      ↳ décrochage (jarring) ou un alésage prudent de la
      ↳ section.",
12 },
13 {
14     "Risk": "Lost Circulation",
15     "Level": "Medium",
16     "How Identified": {
17         "DDR Evidence": "summary_of_activities_24_hours:
            ↳ 'pertes de boue de 5 m3/hr observées lors du
            ↳ pompage de bouchons LCM'",
18         "RAG Context Snippet": "Document
            ↳ 'MudLossesManagement.docx', Chapitre 2: 'Des
            ↳ pertes mineures à modérées (1-10 m3/hr) après
            ↳ traitement LCM peuvent indiquer une
            ↳ stabilisation partielle mais nécessitent une
            ↳ surveillance continue.'",
19         "Reasoning": "Bien que des bouchons LCM aient été
            ↳ pompés, des pertes persistent, indiquant un
            ↳ risque modéré de réactivation ou d'aggravation
            ↳ si les conditions de forage changent.",
20     },
21     "How to Avoid this Risk (Solution)": "Continuer la
            ↳ surveillance étroite des volumes de boue. Préparer
            ↳ des bouchons LCM supplémentaires. Envisager un
            ↳ ajustement de la densité de la boue ou des
            ↳ paramètres de forage pour minimiser la pression
            ↳ exercée sur la formation."
22     }
23 ]

```

Cette sortie illustre comment le LLM intègre les différentes sources d'information pour fournir une analyse détaillée et des recommandations actionnables.

3.5.5 Évaluation du Système Hybride

L'évaluation d'un système hybride comme le nôtre, dont la sortie finale est un texte généré combinant des prédictions quantitatives et un raisonnement qualitatif, ne peut pas reposer sur des métriques de classification traditionnelles. L'objectif n'est plus seulement de savoir si la prédiction est "correcte", mais d'évaluer la **qualité globale**, la **pertinence** et l'utilité de l'analyse générée pour un utilisateur final.

Notre méthodologie d'évaluation s'est donc appuyée sur une analyse qualitative basée sur des études de cas. Nous avons sélectionné un sous-ensemble de DDR représentatifs, incluant des cas avec des risques avérés et des cas sans incident notable, et avons évalué les sorties du système hybride selon les critères suivants :

- **Cohérence** : La sortie du LLM est-elle logiquement cohérente ? Est-ce que le raisonnement fourni ('Reasoning') établit un lien crédible entre l'alerte du modèle LSTM ('ML Alert') et les preuves textuelles extraites du DDR ('DDR Evidence') ?
- **Précision de la justification** : Le système identifie-t-il les bonnes phrases ou les bons paragraphes dans le DDR pour justifier l'alerte ? Une alerte de "Haut Risque" pour un coincement de garniture doit être justifiée par des observations pertinentes (ex : augmentation du couple), et non par des informations sans rapport.
- **Pertinence du contexte RAG** : Le contexte technique récupéré par le module RAG est-il pertinent pour le risque identifié ? Apporte-t-il une réelle valeur ajoutée en reliant l'observation du jour à une connaissance métier documentée ?
- **Qualité des recommandations** : Les solutions proposées ('How to Avoid this Risk') sont-elles des actions standards et appropriées dans l'industrie du forage pour mitiger le risque identifié ? Sont-elles claires et actionnables ?

Il est crucial de souligner que la pertinence des recommandations générées par le LLM repose entièrement sur la qualité de la base de connaissances fournie via le mécanisme RAG. Les solutions proposées ne sont pas inventées par le modèle, mais sont une synthèse et une reformulation contextuelle des procédures, des meilleures pratiques et des retours d'expérience contenus dans les documents techniques qui ont été utilisés pour construire l'index FAISS. Le système agit donc comme un expert qui a "lu" toute la documentation pertinente pour suggérer une action appropriée.

Résultats de l'évaluation qualitative. L'analyse des études de cas a montré des résultats très prometteurs. Dans la majorité des cas où le modèle LSTM émettait une alerte "Moyen" ou "Haut Risque" qui correspondait à un problème réel, le système hybride a réussi à :

1. **Identifier correctement la preuve textuelle** dans le rapport du jour, agissant comme un "assistant intelligent" qui met en évidence les phrases clés pour l'ingénieur.
2. **Fournir un raisonnement logique et concis**, expliquant pourquoi la combinaison de l'alerte quantitative et de l'observation textuelle était préoccupante.
3. **Proposer des recommandations pertinentes et standards**, démontrant sa capacité à non seulement identifier un problème mais aussi à suggérer des pistes de solution.

En conclusion, l'évaluation montre que le système hybride atteint son objectif principal : transformer une simple prédiction en une analyse de risque explicable et actionnable, augmentant ainsi considérablement la valeur de la sortie pour l'utilisateur final.

3.5.6 Discussion et Avantages de l'Approche Hybride

Les bénéfices clés de cette approche hybride MCP+RAG incluent :

- **Robustesse Accrue** : La convergence des prédictions ML et de l'analyse contextuelle du LLM renforce la fiabilité des identifications de risques.
- **Couverture Améliorée** : La complémentarité entre l'analyse de données structurées par les modèles ML et l'interprétation de texte non structuré par le LLM permet de couvrir un spectre plus large de scénarios de risque.
- **Explicabilité Renforcée** : Le LLM peut générer des justifications textuelles claires, en reliant les alertes ML aux observations du DDR et aux connaissances extraites par RAG.

L'utilisation de Kaggle pour l'hébergement du LLM et de ngrok pour la communication a permis de surmonter les limitations de ressources locales tout en intégrant une capacité d'IA avancée dans l'application.

3.5.7 Limitations et Perspectives

L'implémentation de ce système hybride présente certaines limitations et ouvre des perspectives d'amélioration :

- *Complexité d'Intégration* : L'orchestration des différents modules (ML classiques, RAG, LLM, communication via ngrok) représente une complexité technique non négligeable.
- *Prompt Engineering* : La performance du LLM est fortement dépendante de la qualité et de la précision du prompt. Un travail itératif rigoureux est nécessaire pour optimiser les instructions fournies au modèle.
- *Gestion des Conflits* : Des divergences peuvent apparaître entre les prédictions des modèles ML et l'analyse du LLM. Des stratégies pour résoudre ces conflits (par exemple, par pondération ou par escalade à un expert humain) pourraient être explorées.
- *Coût de Calcul et Latence* : L'inférence des LLM, même sur GPU, peut introduire une latence. L'utilisation de ngrok peut également ajouter une surcouche de latence réseau.
- *Dépendance à des Services Externes* : L'utilisation de Kaggle et ngrok introduit une dépendance à la disponibilité et aux politiques de ces services.
- *Maintenance de la Base RAG* : La pertinence du système RAG dépend de la mise à jour et de la qualité de la base de connaissances.

Les perspectives futures incluent l'optimisation des prompts, l'exploration de LLM plus légers ou quantifiés pour un déploiement local, et l'évaluation formelle de la performance du système hybride sur un ensemble de DDR plus large.

3.6 Développement d'un Assistant Conversationnel Technique pour le Domaine du Forage

Au-delà de l'analyse automatisée des rapports DDR, un objectif secondaire mais stratégique de notre projet a été de créer un point d'accès direct à l'expertise technique encapsulée dans notre base de connaissances. Pour ce faire, nous avons développé un module

d'assistant conversationnel (chatbot), dont l'implémentation est détaillée dans le notebook `drilling-expert-bot.ipynb`. Ce chatbot est spécifiquement conçu pour fonctionner comme un expert de premier niveau, capable de répondre aux questions techniques des ingénieurs et du personnel de forage.

3.6.1 Objectif et Cas d'Usage

L'objectif central de ce chatbot est de démocratiser l'accès à la documentation technique. Plutôt que de devoir chercher manuellement une information dans des manuels PDF de plusieurs centaines de pages, un utilisateur peut simplement poser une question en langage naturel, par exemple : "Quelles sont les causes principales du coincement par pression différentielle ?" ou "Décris-moi la procédure standard pour une opération de cimentation". Le chatbot doit alors fournir une réponse synthétique, précise et pertinente, extraite directement de la base de connaissance approuvée.

3.6.2 Architecture Technique Distribuée

Pour des raisons de performance et de ressources, le chatbot repose sur une architecture distribuée qui sépare l'interface utilisateur du moteur d'intelligence artificielle.

- **L'Application Web (Client & Backend Local) :**

L'interface du chat est intégrée à notre application web Flask principale, qui tourne localement ou sur un serveur d'entreprise. C'est ce backend qui reçoit la question de l'utilisateur.

- **Le Service d'IA (Backend Distant) :**

Le modèle LLM `meta-llama/Meta-Llama-3-8B-Instruct`, qui est très gourmand en ressources (notamment en VRAM), est exécuté sur une infrastructure externe dotée de GPU (dans notre cas, un notebook Kaggle).

- **Le Pont de Communication ('ngrok') :**

Pour que l'application locale puisse communiquer avec le modèle distant, un tunnel sécurisé est créé à l'aide de **ngrok**. Comme le montre la Figure 3.19, ngrok expose l'API Flask qui s'exécute sur Kaggle via une URL publique, la rendant accessible depuis n'importe où sur Internet.

```

--- Manual Ngrok Setup ---
Configuring ngrok authtoken using ./ngrok command...
Ngrok authtoken command stdout: Authtoken saved to configuration file: /root/.config/ngrok/ngrok.yml

Ngrok authtoken command stderr:
Ngrok authtoken configured successfully for the ./ngrok binary.
Starting ngrok tunnel for local port 5000 in the background using ./ngrok ...
ngrok process started (PID: 243). Waiting for tunnel...
Attempting to fetch tunnel info from ngrok agent API: http://127.0.0.1:4040/api/tunnels
Ngrok API response for tunnels: {
  "tunnels": [
    {
      "name": "command_line",
      "ID": "69d1d7ab664b71a53e326762594245d7",
      "uri": "/api/tunnels/command_line",
      "public_url": "https://6592-34-68-254-1.ngrok-free.app",
      "proto": "https",
      "config": {
        "addr": "http://localhost:5000",
        "inspect": true
      },
      "metrics": {
        "conns": {
          "count": 6,
          "gauge": 0,
          "rate1": 0.003200203776752068,
          "rate5": 0.009387032533789483,
          "rate15": 0.005153682981837995,
          "p50": 1596546,
          "p90": 2040312,
          "p95": 2040312,
          "p99": 2040312
        },
        "http": {
          "count": 6,
          "rate1": 0.003200203776752068,
          "rate5": 0.009387032533789483,
          "rate15": 0.005153682981837995,
          "p50": 763057.5,
          "p90": 950075,
          "p95": 950075,
          "p99": 950075
        }
      }
    }
  ],
  "uri": "/api/tunnels"
}
--- SUCCESS: Ngrok tunnel active! ---
Public URL: https://6592-34-68-254-1.ngrok-free.app
Your Kaggle Flask app's chat API should be accessible at: https://6592-34-68-254-1.ngrok-free.app/api/kaggle-chat-expert
IMPORTANT: Use THIS ngrok URL in your LOCAL Flask app's NGROK_KAGGLE_CHAT_API_URL setting.

COPY THIS NGROK URL for your local Flask app: https://6592-34-68-254-1.ngrok-free.app

```

FIGURE 3.19 – Mise en place du tunnel ‘ngrok’ dans l’environnement Kaggle pour exposer l’API du chatbot.

3.6.3 Détail du Flux de Traitement d’une Requête

Le fonctionnement du chatbot, tel qu’implémenté dans les fonctions de notre notebook, suit le pipeline RAG (Figure 3.21).

1. **Initialisation des Composants (Cellule 9) :** Au démarrage du notebook, les composants essentiels sont chargés une seule fois en mémoire globale pour éviter de les recharger à chaque requête. La fonction `load_indexed_knowledge` charge l’index FAISS et les chunks de texte. La fonction `initialize_embedding_model` charge le modèle `all-MiniLM-L6-v2`. Enfin, `initialize_llm_pipeline` charge le modèle Llama 3 et son tokenizer, créant ainsi le pipeline de génération de texte.

FIGURE 3.20 – Journal d’initialisation des composants d’IA dans le notebook Kaggle.

```
--- Global AI Component Initialization ---
PyTorch version: 2.6.0+cu124
CUDA available: True
CUDA device count: 2
Current CUDA device: 0
Device name: Tesla T4
Attempting to load FAISS index from: /kaggle/input/knowledge-data/drilling_knowledge.index
Attempting to load knowledge chunks from: /kaggle/input/knowledge-data/drilling_knowledge_chunks.json
Successfully loaded FAISS index (427 vectors) and chunks (427).
Initializing query embedding model: all-MiniLM-L6-v2
Query embedding model initialized.
Initializing LLM pipeline for: meta-llama/Meta-Llama-3-8B-Instruct
LLM will use device: cuda
```

(a) Initialisation de composants

```
Initializing query embedding model: all-MiniLM-L6-v2
modules.json: 100% ██████████ 349/349 [00:00<00:00, 41.1kB/s]
config_sentence_transformers.json: 100% ██████████ 116/116 [00:00<00:00, 14.6kB/s]
README.md: 100% ██████████ 10.5k/10.5k [00:00<00:00, 1.48MB/s]
sentence_bert_config.json: 100% ██████████ 53.0/53.0 [00:00<00:00, 7.54kB/s]
config.json: 100% ██████████ 612/612 [00:00<00:00, 79.9kB/s]
model.safetensors: 100% ██████████ 90.9M/90.9M [00:00<00:00, 132MB/s]
tokenizer_config.json: 100% ██████████ 350/350 [00:00<00:00, 44.5kB/s]
vocab.txt: 100% ██████████ 232k/232k [00:00<00:00, 7.21MB/s]
tokenizer.json: 100% ██████████ 466k/466k [00:00<00:00, 22.1MB/s]
special_tokens_map.json: 100% ██████████ 112/112 [00:00<00:00, 17.0kB/s]
config.json: 100% ██████████ 190/190 [00:00<00:00, 22.1kB/s]
Query embedding model initialized.
Initializing LLM pipeline for: meta-llama/Meta-Llama-3-8B-Instruct
LLM will use device: cuda
tokenizer_config.json: 100% ██████████ 51.0k/51.0k [00:00<00:00, 5.47MB/s]
tokenizer.json: 100% ██████████ 9.09M/9.09M [00:00<00:00, 29.3MB/s]
special_tokens_map.json: 100% ██████████ 73.0/73.0 [00:00<00:00, 10.8kB/s]
config.json: 100% ██████████ 654/654 [00:00<00:00, 90.5kB/s]
model.safetensors.index.json: 100% ██████████ 23.9k/23.9k [00:00<00:00, 2.97MB/s]
Fetching 4 files: 100% ██████████ 4/4 [02:04<00:00, 53.65s/it]
model-00002-of-00004.safetensors: 100% ██████████ 5.00G/5.00G [02:04<00:00, 222MB/s]
model-00003-of-00004.safetensors: 100% ██████████ 4.92G/4.92G [02:02<00:00, 51.7MB/s]
model-00001-of-00004.safetensors: 100% ██████████ 4.98G/4.98G [01:50<00:00, 68.6MB/s]
model-00004-of-00004.safetensors: 100% ██████████ 1.17G/1.17G [01:40<00:00, 3.50MB/s]
Loading checkpoint shards: 100% ██████████ 4/4 [00:59<00:00, 13.03s/it]
generation_config.json: 100% ██████████ 187/187 [00:00<00:00, 23.6kB/s]
Device set to use cuda:0
Set LLM tokenizer pad_token_id to eos_token_id: 128009
LLM pipeline for meta-llama/Meta-Llama-3-8B-Instruct initialized successfully.
```

(b) Chargement des modèles

```
Device set to use cuda:0
Set LLM tokenizer pad_token_id to eos_token_id: 128009
LLM pipeline for meta-llama/Meta-Llama-3-8B-Instruct initialized successfully.
Global AI components initialized and ready for Flask app and analysis.
```

(c) Reponse de LLM

2. **Réception de la Requête (Cellule 11) :** L'application Flask sur Kaggle écoute les requêtes POST sur l'endpoint `/api/kaggle-chat-expert`. Lorsqu'une requête arrive de l'application web locale, elle en extrait la question de l'utilisateur.
3. **Récupération de Contexte (RAG) :**
 - La question de l'utilisateur est passée à la fonction `retrieve_relevant_chunks`.
 - À l'intérieur de cette fonction, le `query_embedding_model_global` vectorise la question.
 - Le vecteur résultant est utilisé pour interroger le `faiss_index_global`, qui retourne les identifiants des k chunks les plus pertinents (typiquement $k=3$ pour le chatbot).
 - Les textes de ces chunks sont extraits de la liste `knowledge_chunks_list_global`.
4. **Génération de la Réponse par le LLM :**
 - Un prompt spécifique au chatbot est construit dynamiquement. Il inclut un message système définissant le rôle d'assistant expert, les chunks de connaissance récupérés, et la question de l'utilisateur.
 - Ce prompt complet est passé au `llm_pipeline_global`.
 - Le pipeline LLM génère une réponse textuelle en se basant sur les informations fournies. Des paramètres de génération comme la 'temperature' (0.3 pour des réponses plus factuelles) et 'max_new_tokens' (512) contrôlent la créativité et la longueur de la réponse.
 - Le texte généré est ensuite nettoyé pour enlever le prompt initial qui est parfois répété en sortie, ainsi que les tokens spéciaux comme '`<leot_id>`'.
5. **Retour de la Réponse :** La réponse textuelle nettoyée est encapsulée dans un objet JSON et renvoyée à l'application web locale via la connexion ngrok. Le frontend de l'application se charge alors de l'afficher à l'utilisateur dans l'interface de chat.

```

--- Starting Kaggle Flask App (ngrok should be running from Cell 11A) ---
Flask app starting. It will listen on host '0.0.0.0' and port 5000 internally.
Ngrok (from Cell 11A) should be tunneling to it. Public base URL: https://204b-35-243-192-211.ngrok-free.app
* Serving Flask app '__main__'
* Debug mode: off
[Kaggle Chat API] Received question: Can u tell me about MWD

Embedding query for RAG: "Can u tell me about MWD..."
Error displaying widget: model not found
Searching FAISS index for top 3 results...
Retrieved RAG results:
- Index: 390, Distance: 1.1503, Source: formation_specific_risks/lwf_mwd_logging_operations.txt
- Index: 352, Distance: 1.1662, Source: operations_best_practices/bha_makeup_mwd_lwd_tool_prep.txt
- Index: 398, Distance: 1.2489, Source: formation_specific_risks/mwd_lwd_survey_data_quality_issues.txt
[Kaggle Chat API] Applied llama 3 chat template via tokenizer.
[Kaggle Chat API] Sending chat prompt to LLM (first 200 chars): <|begin_of_text|><|start_header_id|>system<|end_header_id|>

You are an AI Drilling Expert assistant. Answer the user's questions based on the provided 'Retrieved Drilling Knowledge Context'. Be concise...
[Kaggle Chat API] LLM Answer (first 200 chars): Based on the provided context, MWD stands for Measurement While Drilling, which refers to the process of collecting real-time data about the formation and drilling parameters while drilling a well. This data is used to optimize drilling operations, improve wellbore placement, and enhance formation evaluation. MWD tools provide valuable information about the formation, including its properties, structure, and potential hydrocarbon-bearing zones."
...
[Kaggle Chat API] jsonify successful. Returning response...

```

FIGURE 3.21 – Trace d'exécution d'une requête dans le terminal du serveur, montrant les étapes de récupération et de génération.

3.6.4 Évaluation du Chatbot Technique

L'évaluation d'un système conversationnel comme notre chatbot technique ne repose pas sur des métriques quantitatives simples comme l'exactitude, mais plutôt sur une éva-

luation qualitative de sa performance et de son utilité pratique. Nous avons mené cette évaluation selon deux axes principaux : la performance technique (latence) et la pertinence des réponses (feedback des experts).

3.6.4.a Performance Technique et Temps de Réponse

L'un des indicateurs clés de la performance d'un système interactif est son temps de réponse, ou latence. Nous avons mesuré le temps écoulé entre l'envoi de la question par l'utilisateur et la réception de la réponse complète générée par le LLM.

Dans notre configuration actuelle, avec le modèle `meta-llama/Meta-Llama-3-8B-Instruct` hébergé sur une infrastructure Kaggle (utilisant des GPU de type NVIDIA Tesla T4) et accessible via un tunnel ngrok, le temps de réponse moyen observé pour une requête typique est de l'ordre de **70 secondes**. Ce temps de latence relativement élevé se décompose en plusieurs étapes :

- La transmission de la requête de l'application locale vers le serveur Kaggle via le tunnel ngrok.
- La vectorisation de la requête et la recherche de similarité dans l'index FAISS (étape très rapide).
- L'inférence du modèle LLM lui-même, qui constitue la part la plus importante du temps de calcul.
- La transmission de la réponse générée depuis Kaggle vers l'application locale.

Il est important de noter que ce temps de latence est fortement contraint par les ressources de calcul partagées et les limitations de la plateforme d'expérimentation utilisée (Kaggle). Dans un environnement de production avec des GPU dédiés et optimisés pour l'inférence, ainsi qu'une architecture réseau plus directe (sans ngrok), on pourrait s'attendre à une réduction significative de ce temps de réponse. Néanmoins, cette mesure fournit un ordre de grandeur réaliste des performances avec l'infrastructure à notre disposition.

3.6.4.b Évaluation Qualitative par des Experts du Domaine

Pour évaluer la pertinence et l'exactitude des réponses, nous avons adopté une démarche qualitative en sollicitant l'avis d'experts du domaine (ingénieurs et superviseurs de forage). Nous leur avons soumis une série de questions techniques typiques et avons recueilli leur feedback sur les réponses générées par le chatbot.

L'évaluation s'est concentrée sur les critères suivants :

- **Exactitude Technique** : La réponse est-elle correcte d'un point de vue technique et conforme aux standards de l'industrie ?
- **Pertinence** : La réponse adresse-t-elle directement la question posée par l'utilisateur ?
- **Complétude** : La réponse est-elle suffisamment détaillée ? Fournit-elle assez de contexte pour être utile sans être trop verbeuse ?
- **Clarté** : La réponse est-elle formulée dans un langage clair et compréhensible pour un professionnel du domaine ?

Le retour des experts a été globalement très positif. Ils ont souligné la capacité du système à fournir rapidement des informations précises qui auraient nécessité une recherche

manuelle fastidieuse dans la documentation. La capacité du chatbot à synthétiser des informations provenant de plusieurs sources (les différents chunks récupérés par RAG) a été particulièrement appréciée. Les retours ont également permis d'identifier des axes d'amélioration, notamment la nécessité d'enrichir la base de connaissances avec des documents plus spécifiques à certaines opérations ou régions géographiques pour améliorer encore davantage la pertinence des réponses. Cette validation par des experts a été une étape cruciale pour confirmer la viabilité et l'utilité potentielle de l'outil dans un contexte opérationnel réel.

3.6.5 Avantages et Limites de l'Approche Implémentée

Avantages

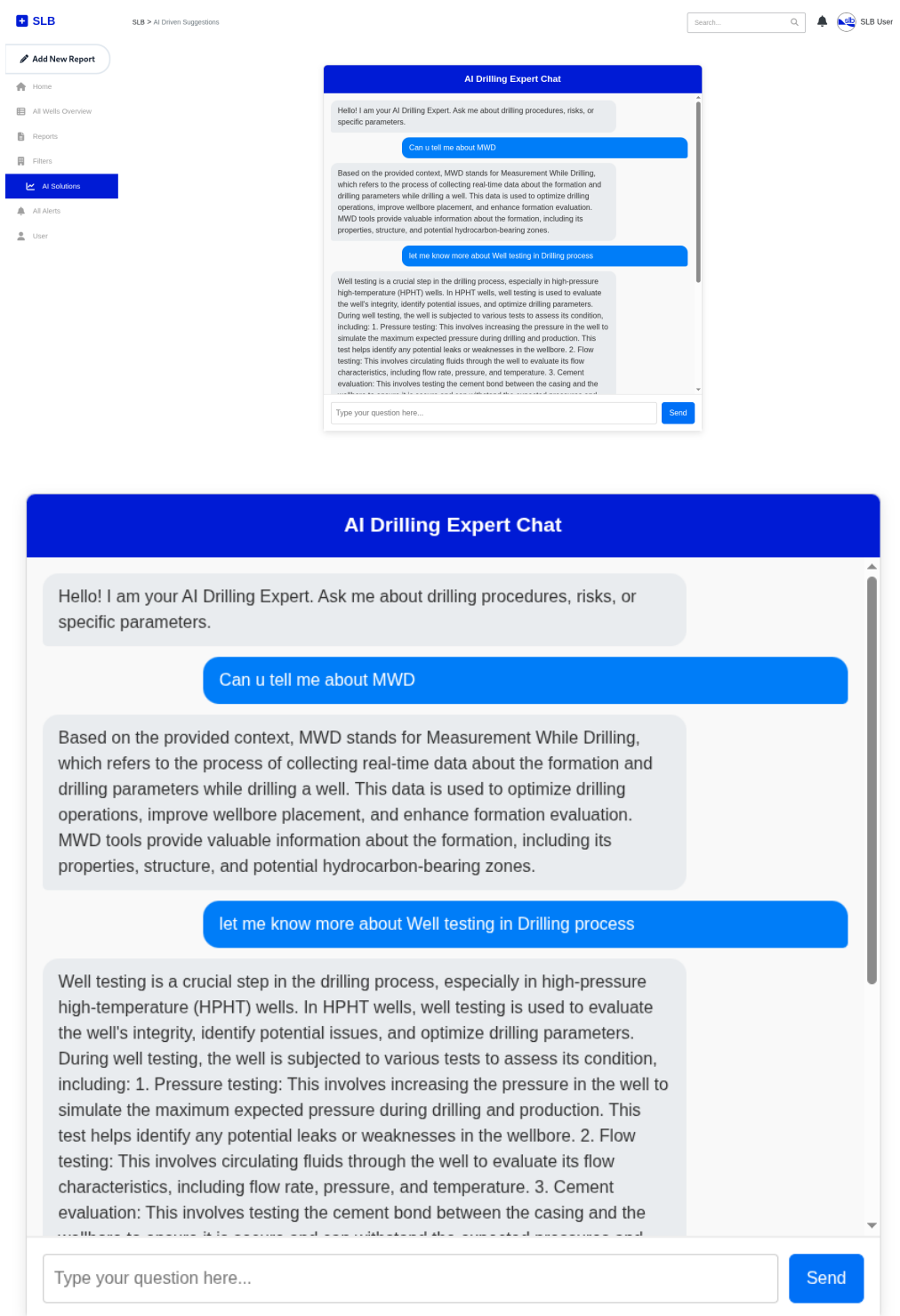
- **Accès Instantané à l'Expertise Documentée** : Transforme une base de documents statiques en une source de connaissance interactive et consultable en langage naturel.
- **Fiabilité des Réponses** : En basant ses réponses sur un corpus de documents validés (la base de connaissances), le chatbot réduit considérablement le risque d'"hallucination" ou de fausses informations, un problème courant avec les LLMs non contraints.
- **Cohérence de l'Information** : Garantit que tous les utilisateurs reçoivent des réponses basées sur la même source de vérité, promouvant ainsi la standardisation des connaissances.

Limites et Perspectives

- **Dépendance à la Base RAG** : La qualité des réponses est directement proportionnelle à la qualité et à l'exhaustivité de la base de connaissances. Une information absente ou mal formulée dans les documents sources ne pourra pas être restituée.
- **Latence** : L'architecture distribuée et le temps de calcul du LLM introduisent une latence de plusieurs secondes entre la question et la réponse, ce qui peut impacter l'expérience utilisateur. Des optimisations (modèles plus légers, mise en cache) seraient nécessaires pour une application en production.
- **Absence de Mémoire Conversationnelle** : Dans l'implémentation actuelle, chaque question est traitée de manière indépendante. Une amélioration future consisterait à intégrer un historique de la conversation dans le prompt pour permettre au chatbot de répondre à des questions de suivi contextuelles.

Malgré ces limites, ce module de chatbot démontre la puissance de l'architecture RAG pour créer des assistants experts spécialisés, capables de fournir un support technique précieux et à la demande.

FIGURE 3.22 – Exemple d’interaction avec le chatbot technique au sein de l’application web.



3.7 Conclusion

En somme, ce chapitre a détaillé le parcours complet de la conception à l'implémentation de notre solution applicative. En partant de la collecte et du nettoyage rigoureux des rapports bruts, nous avons construit un dataset structuré et enrichi, fondamental pour toute analyse ultérieure. L'architecture web, bâtie sur Flask et Plotly.js, a été conçue pour être à la fois robuste et modulaire, fournissant une interface utilisateur interactive et des points d'accès API clairs. Au cœur de notre contribution, le développement d'un pipeline d'intelligence artificielle hybride a été exposé. Nous avons détaillé la construction et l'évaluation de modèles prédictifs LSTM spécialisés dans la détection des risques à partir de données séquentielles, ainsi que la mise en œuvre d'un système RAG innovant pour l'analyse contextuelle et la génération de réponses par un LLM. Enfin, la création d'un chatbot technique démontre la flexibilité de cette architecture pour adresser différents cas d'usage. L'ensemble de ces composants forme une solution cohérente et fonctionnelle, dont la performance et la pertinence seront évaluées dans la section suivante.

CONCLUSION GÉNÉRALE

La réussite des opérations de forage dans l'industrie pétrolière et gazière, secteur à haut risque, dépend de plus en plus de l'exploitation intelligente des données. Les rapports journaliers de forage (DDR) représentent une source d'information d'une richesse immense, mais leur format semi-structuré freine l'analyse systématique. La problématique centrale de ce projet était donc de surmonter cet obstacle en concevant une solution intégrée pour transformer cette masse de données brutes en une source de connaissance proactive et actionnable. Pour répondre à ce défi, nous avons développé une application web innovante, fondée sur une synergie entre deux approches complémentaires d'intelligence artificielle. D'une part, nous avons exploité des modèles d'apprentissage profond de type Long Short-Term Memory (LSTM) pour analyser les données numériques et séquentielles. Cette approche nous a permis de capturer les dynamiques temporelles complexes et de prédire la probabilité de risques critiques tels que les pannes d'équipement ou les coincements de garniture, jetant ainsi les bases d'un système d'alerte précoce. D'autre part, conscients que les chiffres seuls ne suffisent pas, nous avons mis en œuvre une architecture de Génération Augmentée par Récupération (RAG) pour exploiter la richesse sémantique du texte non structuré. Ce système combine la puissance de raisonnement d'un Grand Modèle de Langage (LLM) avec la fiabilité d'une base de connaissances technique. Il permet ainsi de dépasser la simple reconnaissance de mots-clés pour effectuer une véritable analyse contextuelle, en justifiant les risques identifiés par des preuves textuelles et en générant des recommandations nuancées. La convergence de ces deux approches se matérialise dans une application web complète, qui sert de tableau de bord centralisé. Cette plateforme offre aux ingénieurs des outils de visualisation interactifs, un système de filtrage puissant et un centre de notifications, constituant une solution de bout en bout, de la donnée brute à la décision éclairée.

Malgré des résultats prometteurs, notre travail présente des limites, notamment la dépendance à la quantité de données d'entraînement pour la généralisation des modèles et la qualité de la base de connaissances pour le système RAG. De plus, l'architecture actuelle pose des défis de latence pour un déploiement industriel à grande échelle. Ces limites ouvrent la voie à des perspectives d'évolution passionnantes. L'amélioration la plus immédiate serait de créer un pipeline d'analyse véritablement hybride, où la sortie des modèles LSTM enrichirait directement le prompt du LLM, ce dernier agissant comme

un méta-modèle pour synthétiser toutes les informations. Une autre évolution naturelle consisterait à transformer l'architecture RAG en un assistant conversationnel (chatbot) expert, permettant aux utilisateurs d'interroger le système en langage naturel. À plus long terme, l'analyse pourrait être étendue à l'échelle de plusieurs puits pour identifier des bonnes pratiques généralisables. En conclusion, ce projet a permis de réaliser une solution fonctionnelle qui démontre la puissance d'une approche hybride pour l'analyse des données de forage. En faisant collaborer l'apprentissage profond pour l'analyse séquentielle et l'intelligence artificielle générative pour l'interprétation contextuelle, nous avons jeté les bases d'un outil capable de transformer des rapports statiques en une source de connaissance dynamique. Ce travail illustre comment l'IA peut servir de copilote aux ingénieurs, augmentant leur capacité d'analyse pour améliorer l'efficacité, la rentabilité et, surtout, la sécurité des opérations qui sont au cœur de notre approvisionnement énergétique mondial.

BIBLIOGRAPHIE

- [1] Martín Abadi, Ashish Agarwal, Paul Barham, et al. TensorFlow : Large-scale machine learning on heterogeneous systems. <https://www.tensorflow.org/>, 2016. Software available from tensorflow.org.
- [2] H.K. Al-Mubarak, H.A. Al-Nasser, and A.A. Al-Mulhim. The Evolution and Value of Real-Time Operations Centers (RTOC) in the Oil and Gas Industry. In *SPE Kingdom of Saudi Arabia Annual Technical Symposium and Exhibition*. Society of Petroleum Engineers, April 2018.
- [3] Jaffar Ali, M. Husein, and M. Al-Chalabi. Text Mining of Daily Drilling Reports for Automated Identification of Inefficient Operations. *Journal of Petroleum Exploration and Production Technology*, 12(5) :1239–1253, 2022.
- [4] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bénéttot, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, et al. Explainable artificial intelligence (xai) : Concepts, taxonomies, opportunities and challenges. *Information Fusion*, 58 :82–115, 2020.
- [5] Baker Hughes Company. 2023 annual report. Technical report, Baker Hughes, 2024. Accessed 15th May, 2025.
- [6] Bootstrap. Bootstrap Documentation. <https://getbootstrap.com/>, 2025. Accessed 16th May, 2025.
- [7] Olalekan Bello and Catalin Teodoriu. A New Approach to Drilling Non-Productive Time (NPT) Analysis for Improving Drilling Efficiency. In *SPE/IADC International Drilling Conference and Exhibition*. Society of Petroleum Engineers, 2019.
- [8] Adam T Bourgoyne, Jr, Keith K Millheim, Martin E Chenevert, and F S Young, Jr. *Applied drilling engineering*. Society of Petroleum Engineers, 1986.
- [9] Adam T Bourgoyne, Jr, Keith K Millheim, Martin E Chenevert, and F S Young, Jr. *Applied drilling engineering*. Society of Petroleum Engineers, 1986.
- [10] Adam T Bourgoyne, Jr, Keith K Millheim, Martin E Chenevert, and F S Young, Jr. *Applied drilling engineering*. Society of Petroleum Engineers, 1986.
- [11] Ahmed Moncef BOUSSELAT and Smail DOUDOU. Blockchain pour l’import/export : Améliorer la traçabilité et la précision des données, application : Schlumberger naf. Final Project Thesis, 2023.

- [12] François Chollet et al. Keras. <https://keras.io>, 2015.
- [13] Lucas P. Cinelli, José F.L. de Oliveira, Vinicius M. de Pinho, Wesley L. Passos, Rafael Padilla, Patrick F. Braz, Breno Galves, Domenica P. Dalvi, Gabriela Lewenfus, Jonathas O. Ferreira, Anthony Y.Y. Ji, Felipe L. de Oliveira, Clemente J.C. Gonçalves, Sergio L. Netto, Eduardo A.B. da Silva, and Marcello L.R. de Campos. Automatic event identification and extraction from daily drilling reports using an expert system and artificial intelligence. *Journal of Petroleum Science and Engineering*, 205 :108939, 2021.
- [14] David DeFranza and Planet Green. The oil drilling process explained. <https://science.howstuffworks.com/environmental/energy/oil-drilling-process.htm>, 2024. Accessed on May 5, 2024.
- [15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert : Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, 2019. Association for Computational Linguistics.
- [16] DMC Global Inc. 2023 annual report (form 10-k). Technical report, DMC Global Inc., 2024. Accessed 15th May, 2025.
- [17] Docker. Docker Documentation. <https://docs.docker.com/get-started/docker-overview/#docker-architecture>, 2023. Accessed 16th May, 2023.
- [18] Docker. Docker Documentation. <https://docs.docker.com>, 2023. Accessed 16th May, 2023.
- [19] A.H. El-sayed, A.U. Al-Kaabi, and M.F. El-Amin. A data-driven approach for stuck pipe prediction in the Middle East region using machine learning techniques. *Journal of Energy Resources Technology*, 142(10), 2020.
- [20] Flask. Flask Documentation. <https://flask.palletsprojects.com/en/stable/>, 2025. Accessed 13th May, 2025.
- [21] Fortune Business Insights. Oilfield services market size, share & covid-19 impact analysis. Technical Report FBI101869, Fortune Business Insights, 2023. Accessed 12th May, 2025.
- [22] Shankar Rao Ganti. LazyPredict : Lazy Learning for Supervised Classification and Regression. <https://github.com/shankarpandala/lazypredict>, 2020.
- [23] GE Vernova. 2023 annual report (form 10-k). Technical report, General Electric Company, 2024. Accessed 15th May, 2025. Note : GE’s energy businesses, including oil and gas services, were spun off into GE Vernova.
- [24] Suran Goonatilake and John A. Campbell. A hybrid intelligent system for financial risk assessment. In *Intelligent Hybrid Systems*, pages 221–240. Wiley, 1995.
- [25] Halliburton Company. 2023 annual report. Technical report, Halliburton, 2024. Accessed 15th May, 2025.
- [26] Thafat Hanouti and Chahinez Mameri. Etude en vue d’amélioration de la résilience de la supply chain : Schlumberger algérie. Final Project Thesis, 2022.
- [27] Jiang He, Mian Chen, and Kai Li. Real-time pore pressure prediction with evolving drilling data using a hybrid machine learning approach. *Journal of Petroleum Science and Engineering*, 195 :107532, 2020.

- [28] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8) :1735–1780, 1997.
- [29] Syed Aaquib Hussain, Ketan Hasija, Valerian Guillot, Pierre Sesboue, Ahmed el Gannani, and Todd Hughes. Artificial-Intelligence-Based Drilling Risk Prediction - Prevention is Better than Cure. In *International Petroleum Technology Conference*, Dhahran, Saudi Arabia, February 2024. International Petroleum Technology Conference (IPTC). Paper IPTC-23550-MS.
- [30] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3) :535–547, 2019.
- [31] Zineb Lazib and Nabil Ouerdi. Etude de la maturité digitale de la supply chain et mise en place d’une roadmap de digitalisation : application schlumberger algeria. Final Project Thesis, 2022.
- [32] Patrick Lewis, Ethan Perez, Aleksandara Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Myle Ott, Wen-tau Chen, Alexis Conneau, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474, 2020.
- [33] Wes McKinney et al. pandas : a foundational python library for data analysis and statistics. <https://pandas.pydata.org/>, 2010. Python Data Analysis Library.
- [34] Meta AI. Introducing meta llama 3 : The most capable openly available llm to date. <https://ai.meta.com/blog/meta-llama-3/>, 2024. Accessed 10th May, 2025.
- [35] Microsoft. Power BI : Interactive Data Visualization BI Tools. <https://powerbi.microsoft.com/>, 2025. Accessed 29th May, 2025.
- [36] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv :1301.3781*, 2013.
- [37] Robert Monarch. Human-in-the-loop machine learning. 2021.
- [38] Mayur Panchal. What is expressjs in node js ? – backend framework for web apps, 2023. Accessed on 29th May, 2025.
- [39] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn : Machine learning in python. *the Journal of machine Learning research*, 12 :2825–2830, 2011.
- [40] Planete Energies. How exploration drilling works. <https://www.planete-energies.com/en/media/article/how-exploration-drilling-works>, 2015. Accessed on May 5, 2024.
- [41] Plotly Technologies Inc. Collaborative data science. <https://plot.ly>, 2015.
- [42] Quorum Software. *How to Read a DDR : A Beginner’s Guide to Understanding Data on a Daily Drilling Report*. Quorum Software, 2018.
- [43] Leonard Richardson. Beautiful soup documentation. <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>, 2007. Accessed 15th May, 2025.
- [44] Armin Ronacher. Jinja2. <https://jinja.palletsprojects.com/>, 2008. The Python Template Engine.
- [45] Arash Rostami, Hamed Aghaei, and Davood Sedghipour. Prediction of lost circulation zones using artificial neural network and support vector machine. *Petroleum*, 7(1) :99–107, 2021.

- [46] Franck Scandolera. L’art de la tokenisation : décomposer le texte pour l’ia. <https://www.formations-analytics.com/blog/ai/lart-de-la-tokenisation-decomposer-le-texte-pour-lia/>, April 2024. Blog post from Formations Analytics, Accessed 15th May, 2025.
- [47] Schlumberger. Schlumberger Official Website. <https://www.slb.com>, 2025. Accessed 5th May, 2025.
- [48] SendGrid. What is a Webhook? <https://sendgrid.com/blog/webhook-vs-api-whats-the-diff/>, 2019. Accessed 14th May, 2025.
- [49] Mohamed Annis Souames and Larbi Abderrahmane Mohammedi. Estimation des lead times liés à l’importation à travers l’apprentissage machine dans le cadre de la méthodologie crisp-dm : application : Schlumberger naf. Final Project Thesis, 2022.
- [50] Tableau Software, LLC. Tableau : Business Intelligence and Analytics Software. <https://www.tableau.com/>, 2025. Accessed 29th May, 2025.
- [51] T. Thananchanthivat, I. Gupta, and T. Tularak. A Review of Artificial Intelligence (AI) and Machine Learning (ML) Applications in the Oil and Gas Industry. *Energies*, 15(15) :5513, 2022.
- [52] The pandas development team. pandas.merge_asof — pandas 2.2.2 documentation. https://pandas.pydata.org/docs/reference/api/pandas.merge_asof.html, 2024. Accessed 14th May, 2025.
- [53] Manel Yousfi and Yamina Bakhouché. Contribution à l’optimisation de la planification de la demande en pièces de rechange au sein de la division well construction : application : Schlumberger naf. Final Project Thesis, 2022.
- [54] Guozhu Zhang, Yue Li, and Zhaobin Han. Real-time prediction of wellbore instability in shale gas drilling using a long short-term memory network. *Journal of Natural Gas Science and Engineering*, 66 :197–208, 2019.

ANNEXES

Annexe A : Prétraitement et Structuration des Données de Forag

La transformation des données brutes, hétérogènes et souvent bruitées des rapports de forage en un ensemble de données propre, structuré et prêt pour l'analyse est une étape fondamentale et non-triviale de tout projet de science des données dans ce domaine. Cette annexe détaille le pipeline de prétraitement que nous avons conçu et implémenté, dont la logique est encapsulée dans divers scripts et notebooks Python de notre projet.

Extraction Initiale à partir des Fichiers HTML

Le point de départ de notre chaîne de traitement est une collection de rapports journaliers de forage au format HTML. Chaque fichier représente une journée d'opération pour un puits donné. Le premier défi est d'extraire l'information pertinente de ces fichiers.

Analyse Syntaxique (Parsing) HTML : Nous utilisons la bibliothèque Python **BeautifulSoup** pour analyser la structure du DOM (Document Object Model) de chaque fichier HTML. Cette bibliothèque transforme le balisage HTML en un arbre d'objets Python, ce qui permet une navigation et une recherche programmatiques. Notre script cible spécifiquement les balises `<table>` qui contiennent les données structurées, en les identifiant souvent par leur attribut `'id'` (par exemple, `'id="operationsInfoTable"'` pour le journal des opérations).

Conversion en Format Tabulaire : Une fois une table HTML isolée, nous utilisons la puissance de la bibliothèque **Pandas**, et plus particulièrement la fonction `'pd.read_html()'`, pour convertir directement le code HTML de la table en un `DataFrame` Pandas. Cette étape cruciale transforme les données de présentation en une structure de données analytique.

Nettoyage et Calcul des Données Opérationnelles

Le `DataFrame` obtenu à partir du parsing HTML brut nécessite un nettoyage et un enrichissement considérables. Le script qui orchestre cette phase effectue les opérations suivantes, notamment pour le fichier `'operations_info.csv'` :

1. Gestion des Données Temporelles :

- Les colonnes `'Start time'` et `'End time'` sont converties en objets `'datetime'`. Une logique spécifique est implémentée pour gérer le cas où une opération s'étend au-delà de minuit (e.g., de 23 :00 à 01 :00), en ajoutant un jour à l'heure de fin si nécessaire.
- Une nouvelle colonne, `Duration (hours)`, est calculée en soustrayant l'heure de début de l'heure de fin et en convertissant le résultat en heures décimales. Cette caractéristique est fondamentale pour l'analyse des NPT.

2. Calcul des Indicateurs de Performance de Forage :

- **Distance Forée (*Distance Drilled*) :** Calculée comme la différence entre la profondeur finale (`'End Depth mMD'`) de l'opération en cours et celle de l'opération précédente. Une attention particulière est portée à la gestion des valeurs manquantes ou non numériques.

- **Vitesse de Pénétration (ROP - Rate of Penetration)** : Calculée en divisant la 'Distance Drilled' par la 'Duration'. Les divisions par zéro (pour les opérations de durée nulle) sont gérées pour éviter les erreurs. Le ROP est un indicateur clé de l'efficacité du forage.
- 3. **Standardisation des Données Textuelles** : La colonne 'Remark', qui contient les commentaires libres des opérateurs, est systématiquement extraite et standardisée en tant que chaîne de caractères, en s'assurant que même les lignes d'opérations sans commentaire disposent d'une entrée (chaîne vide), afin d'assurer l'homogénéité du DataFrame.

Enrichissement des Données par Fusion

Les informations complètes sur une opération sont souvent réparties entre plusieurs fichiers sources. Par exemple, 'operations_info.csv' contient la séquence des activités, tandis que 'wellbore_info_transposed.csv' ou 'drilling_fluid_samples.csv' peuvent contenir des informations sur le diamètre du trou ou le type de fluide utilisé à un moment donné.

Pour créer une vue unifiée, nous effectuons une opération de **fusion de données** (*data merging*) à l'aide de Pandas.

1. Un DataFrame principal, généralement celui des opérations, est utilisé comme base.
2. D'autres DataFrames contenant des informations complémentaires (sur le puits, la boue, la lithologie) sont fusionnés avec le DataFrame principal.
3. La fusion est réalisée en utilisant une colonne clé commune, typiquement la colonne Report, qui identifie de manière unique le rapport journalier et donc la journée d'opération.

Le résultat de ce processus est un fichier 'combined.csv' par puits, qui est une table "large" et dénormalisée contenant toutes les informations disponibles pour chaque événement de la journée. C'est ce fichier qui est ensuite utilisé par notre application web pour les analyses et les visualisations.

Ajout de la Caractéristique 'formation_type'

Une des étapes d'enrichissement les plus significatives, réalisée dans le notebook `addFormationTypeColumn.ipynb`, est l'ajout de la colonne `formation_type` au fichier 'combined.csv'. Cette information, cruciale pour l'analyse contextuelle, n'est pas directement présente dans les données d'opérations. Elle est déduite en croisant la profondeur de chaque opération avec les données des sommets de formations géologiques issues du fichier 'stratigraphic_info.csv'.

Le processus est le suivant :

1. Le fichier 'stratigraphic_info.csv' est chargé et trié par profondeur de sommet de formation ('Depth to Top of Formation mMD').
2. Pour chaque ligne du fichier 'combined.csv', sa profondeur ('Depth_mMd') est comparée aux intervalles de profondeur définis par les sommets de formation.
3. La fonction 'pd.merge_asof' de Pandas est utilisée pour effectuer une fusion par proximité. Pour chaque profondeur d'opération, elle trouve la formation stratigraphique correspondante la plus proche (commençant juste avant ou à cette profondeur) [52].

4. Le nom de la formation ainsi identifiée est alors ajouté comme nouvelle colonne dans le DataFrame combiné.

Cette étape d'ingénierie des caractéristiques transforme une simple donnée de profondeur en une information géologique contextuelle de grande valeur, exploitable par nos modèles et nos visualisations.

Annexe B : Architecture et Entraînement des Modèles LSTM

Cette annexe détaille l'architecture et le processus d'entraînement des modèles de Long Short-Term Memory (LSTM) utilisés pour la prédiction des risques à partir des données séquentielles. La logique est entièrement contenue dans le notebook `ML_Algos.ipynb`.

Philosophie de Modélisation

La nature des risques en forage nous a conduits à adopter une philosophie de modélisation spécifique :

- **Un Modèle par Risque** : Plutôt que de construire un seul modèle multi-classe complexe, nous avons développé un modèle binaire distinct pour chaque type de risque cible (e.g., 'Event_Is_Stuck_Pipe', 'Event_Is_Equipment_Failure'). Cette approche permet de spécialiser et d'optimiser l'architecture et les hyperparamètres pour chaque problème de prédiction, qui peut avoir des précurseurs très différents.
- **Perspective Préventive** : L'objectif primordial de nos modèles n'est pas l'exactitude (*accuracy*) globale, qui peut être trompeuse avec des données déséquilibrées. L'objectif est de **minimiser les faux négatifs**, c'est-à-dire les événements de risque réels que le modèle aurait manqués. Nous préférons avoir une fausse alarme (un faux positif) qui incite à la vigilance, plutôt que de manquer une alerte critique.

Préparation des Données pour l'Analyse Séquentielle

Le prétraitement décrit dans l'Annexe A est la première étape. Pour l'entraînement des LSTM, des étapes supplémentaires sont nécessaires :

1. **Création des Séquences et des Cibles** : Les données sont groupées par puits et par jour. Pour chaque jour, la séquence des opérations constitue l'entrée du modèle (X). La cible (Y) est une valeur binaire indiquant si le risque à prédire s'est produit le **jour suivant**.
2. **Gestion du Déséquilibre des Classes** : Les événements de risque sont rares. Pour éviter que le modèle n'apprenne simplement à toujours prédire "pas de risque", nous utilisons des techniques de rééquilibrage. Dans notre cas, nous avons exploré l'utilisation de poids de classe ('`class_weight`' dans Keras), qui pénalisent plus lourdement les erreurs sur la classe minoritaire (la classe "risque") lors de l'entraînement.
3. **Division des Données** : L'ensemble des données est divisé en un ensemble d'entraînement (typiquement 80%) et un ensemble de test (20%). L'ensemble d'entraînement est lui-même subdivisé pour créer un ensemble de validation, utilisé pour surveiller la performance du modèle pendant l'entraînement et éviter le surapprentissage.

Architecture du Réseau de Neurones LSTM

L'architecture de nos modèles, bien qu'ajustable pour chaque risque, suit une structure commune et robuste, conçue pour traiter des séquences d'opérations de forage :

Layer (type)	Output Shape	Param #
=====	=====	=====

masking (Masking)	(None, None, 128)	0
anay (LSTM)	(None, 64)	49408
dropout (Dropout)	(None, 64)	0
batch_normalization (Batch Normalization)	(None, 64)	256
dense (Dense)	(None, 32)	2080
dropout_1 (Dropout)	(None, 32)	0
dense_1 (Dense)	(None, 1)	33
=====		
Total params: 51777 (202.25 KB)		

– Exemple de résumé d’architecture de modèle LSTM généré par Keras.

- **Couche ‘Masking’** : La première couche ignore les pas de temps qui correspondent au padding (valeurs nulles ajoutées pour uniformiser la longueur des séquences), s’assurant qu’ils n’affectent pas le calcul.
- **Couche ‘LSTM’** : Le cœur du modèle. Cette couche contient des cellules LSTM (avec 32 à 64 unités dans nos expérimentations) qui parcourent la séquence d’opérations et apprennent à extraire les dépendances temporelles pertinentes, en maintenant un état interne (mémoire).
- **Couches de Régularisation** :
 - Une couche ‘Dropout’ est placée après la couche LSTM. Elle désactive aléatoirement une fraction des neurones à chaque étape d’entraînement, ce qui force le réseau à apprendre des représentations plus robustes et prévient le sur-apprentissage.
 - Une couche ‘BatchNormalization’ normalise les activations de la couche précédente, ce qui stabilise et accélère le processus d’entraînement.
- **Couches Denses** : Une ou plusieurs couches ‘Dense’ (entièrement connectées) agissent comme un classifieur standard qui prend en entrée les caractéristiques extraites par la couche LSTM pour effectuer la prédiction finale.
- **Couche de Sortie** : La dernière couche est une couche ‘Dense’ avec une seule unité et une fonction d’activation **sigmoïde**. Elle produit une sortie unique entre 0 et 1, qui peut être interprétée comme la probabilité que l’événement de risque se produise le lendemain.

Entraînement et Optimisation

Le modèle est entraîné en utilisant l’optimiseur **Adam** avec un taux d’apprentissage adapté. Pour éviter le sur-apprentissage et trouver le meilleur modèle possible, nous utilisons deux techniques de rappel (*callbacks*) :

- **‘EarlyStopping’** : Surveille une métrique de performance sur l’ensemble de validation (par exemple, l’‘AUC’ ou la ‘val_loss’). Si la performance ne s’améliore pas après un certain nombre d’époques ("patience"), l’entraînement est arrêté prématurément.
- **‘ReduceLROnPlateau’** : Si la performance sur l’ensemble de validation stagne, ce callback réduit automatiquement le taux d’apprentissage de l’optimiseur, permettant au modèle d’effectuer des ajustements plus fins pour converger vers un meilleur minimum.

Une fois l'entraînement terminé, le modèle avec les poids qui ont donné la meilleure performance sur l'ensemble de validation est sauvegardé pour une utilisation ultérieure en prédiction.

Annexe C : Architecture du Système d'Analyse Contextuelle (RAG)

Cette annexe fournit des détails techniques sur l'implémentation du système de Génération Augmentée par Récupération (RAG), dont le code se trouve dans le notebook `slb-assistant-decider.ipynb`. Ce système est conçu pour analyser le texte des DDR et identifier les risques en s'appuyant sur une base de connaissances externe.

Construction de la Base de Connaissances Vectorielle

La performance d'un système RAG dépend de la qualité de sa base de connaissances.

1. **Collecte des Sources** : Nous avons collecté un ensemble de documents PDF et textuels contenant des procédures de forage, des manuels de sécurité, des analyses de risques et des meilleures pratiques de l'industrie.
2. **Segmentation en Chunks** : Chaque document est divisé en petits morceaux de texte (*chunks*) d'une taille gérable (e.g., 2-3 paragraphes). Cette segmentation est cruciale : des chunks trop petits manquent de contexte, tandis que des chunks trop grands "noient" l'information pertinente.
3. **Vectorisation (Embedding)** : Chaque chunk est ensuite passé à travers le modèle de "sentence transformer" **'all-MiniLM-L6-v2'**. Ce modèle transforme chaque chunk en un vecteur numérique de 384 dimensions. Ce vecteur représente la signification sémantique du chunk.
4. **Indexation avec FAISS** : Tous les vecteurs générés sont stockés dans un index **FAISS** (*Facebook AI Similarity Search*). FAISS est une bibliothèque hautement optimisée pour la recherche de similarité efficace parmi des millions de vecteurs. Elle permet, étant donné un vecteur de requête, de retrouver quasi-instantanément les vecteurs les plus proches dans l'index. Les chunks de texte originaux sont stockés séparément dans un fichier JSON, l'index FAISS ne contenant que les vecteurs et leur identifiant correspondant.

Le Pipeline d'Analyse RAG

Lorsqu'un nouveau DDR est soumis pour analyse, le pipeline suivant est exécuté :

1. **Formulation de la Requête** : Le texte pertinent du DDR (e.g., le résumé des activités et les remarques notables) est extrait. Ce texte est utilisé comme requête.
2. **Vectorisation de la Requête** : La requête est transformée en un vecteur de 384 dimensions en utilisant le même modèle **'all-MiniLM-L6-v2'**.
3. **Recherche de Similarité** : Le vecteur de la requête est utilisé pour interroger l'index FAISS, qui renvoie les identifiants des k chunks les plus similaires (typiquement, $k=5$ à 7).
4. **Récupération du Contexte** : Les chunks de texte correspondant aux identifiants retournés par FAISS sont récupérés depuis le fichier JSON.
5. **Construction du Prompt Augmenté** : Un prompt est assemblé dynamiquement. Il contient plusieurs sections :
 - Un **message système** qui définit le rôle du LLM : "Tu es un expert en analyse de risques de forage..."

- Le **contexte récupéré** : Les chunks de texte extraits de la base de connaissances.
 - La **tâche** : Le contenu du DDR à analyser.
 - Des **exemples (few-shot)** qui montrent au modèle le format de sortie JSON exact attendu.
6. **Génération de la Réponse** : Ce prompt complet est envoyé au LLM ‘**meta-llama/Meta-Llama-3-8B-Instruct**’. Le modèle utilise à la fois le DDR et le contexte fourni pour générer son analyse.
 7. **Post-traitement** : La réponse textuelle du LLM est analysée pour en extraire le bloc JSON. Ce JSON est ensuite validé avant d’être utilisé par l’application.

Exemple de Prompt (Simplifié)

Voici une version simplifiée du type de prompt envoyé au LLM :

```
<|begin_of_text|><|start_header_id|>system<|end_header_id|>
```

You are an expert Drilling Risk Analyst. Your task is to analyze the provided Daily Drilling Report (DDR) JSON. Use the 'Retrieved Drilling Knowledge Context' to inform your analysis. Identify potential risks and provide your analysis in a valid JSON list format.

```
<|eot_id|><|start_header_id|>user<|end_header_id|>
```

****Retrieved Drilling Knowledge Context:****

- Chunk 1: "High torque and vibrations are often precursors to stuck pipe events, especially in deviated well sections..."
- Chunk 2: "When circulating bottoms up, an increase in gas readings indicates an influx from the formation..."

****Daily Drilling Report (JSON):****

```
{
  "summary_of_activities_24_hours": "Drilled 8 1/2in section from
    3200m to 3350m. Observed high, erratic torque. Circulated bottoms up,
    connection gas increased to 5%.",
  "operations": [ ... ]
}
```

****Your Task:****

Based on the DDR and Knowledge Context, identify risks and provide your analysis in the specified JSON list format.

```
<|eot_id|><|start_header_id|>assistant<|end_header_id|>
```

Face à ce prompt, le LLM est guidé pour générer une réponse structurée identifiant le risque de coincement de garniture (dû au couple élevé) et le risque de venue de gaz (dû à l’augmentation des lectures de gaz).

Annexe D : Guide d'Utilisation de l'Application Web

Cette annexe décrit les principales fonctionnalités de l'application web développée, en se basant sur les fichiers de templates HTML qui structurent l'interface utilisateur.

Page d'Accueil ('index.html')

La page d'accueil sert de tableau de bord global, offrant une vue d'ensemble de tous les puits disponibles dans la base de données.

- **Indicateurs Clés de Performance (KPIs) :** Des cartes synthétiques affichent des statistiques agrégées comme le nombre total de puits, la répartition des statuts de puits, et la profondeur d'eau moyenne.
- **Graphiques de Distribution :** Des diagrammes en barres, générés avec Plotly.js, montrent la répartition des puits par opérateur et par appareil de forage. Des graphiques en ligne montrent l'évolution temporelle du début (*spud date*) et de la fin des forages.
- **Visualisation des ROP Inter-puits :** Un diagramme en violon interactif permet de comparer la distribution des vitesses de pénétration (ROP) entre tous les puits, avec des filtres pour le diamètre de trou, le type de formation et le type de fluide.

Vue d'Ensemble de tous les Puits ('all_wells_overview.html')

Cette page fournit une vue géographique et descriptive de tous les puits.

- **Carte Interactive :** Une carte, générée avec Plotly.js (mapbox), affiche la localisation de chaque puits. Chaque point est cliquable et renvoie vers la page de détail du puits correspondant.
- **Cartes de Résumé par Puits :** Sous la carte, une grille de "cartes" présente un résumé des informations clés (opérateur, statut, date de début) pour chaque puits, offrant un accès rapide à leur tableau de bord détaillé.

Tableau de Bord Détaillé d'un Puits ('view_well_data.html')

C'est la page la plus riche de l'application. Elle est dédiée à l'analyse approfondie d'un seul puits.

- **Visualisations Statiques :** Elle affiche une série de graphiques générés côté serveur, car ils nécessitent des calculs plus complexes. Cela inclut :
 - La trajectoire du puits en 3D.
 - Le profil lithologique vertical.
 - Le profil stratigraphique.
 - Un graphique des changements de type de fluide et de diamètre de trou en fonction de la profondeur.
 - Un diagramme en secteurs de la répartition du temps par grande catégorie d'opération (forage, manœuvre, NPT, etc.).
- **Graphiques Interactifs :** Des graphiques permettent à l'utilisateur d'explorer les données de manière dynamique :
 - Un graphique de ROP en fonction du temps, avec des filtres pour isoler les phases de forage ou les interruptions.

- Un diagramme circulaire de la distribution des activités, qui peut être détaillé par sous-catégories.
- Des graphiques interactifs permettant de tracer n'importe quel paramètre numérique disponible (extrait des fichiers CSV) en fonction de la profondeur ou du temps.
- **Affichage des Données Brutes** : Un bouton permet d'afficher ou de masquer les tableaux de données détaillés (opérations, fluides, etc.) d'où sont tirés les graphiques, pour une inspection plus fine.

Page de Filtres ('filter.html')

Cette page offre un outil de recherche puissant pour interroger l'ensemble de la base de données de tous les puits.

- **Filtres Multi-critères** : L'utilisateur peut construire des requêtes complexes en combinant plusieurs filtres (e.g., "trouver toutes les opérations de forage avec un 'Fluid_Type' 'OBM' dans un 'Hole_Dia_in' de 12.25 pour le puits 'X' ou pour tous les puits").
- **Résultats Agrégés et Détaillés** : La page affiche un résumé statistique des données correspondant aux filtres, ainsi que le tableau détaillé de toutes les lignes correspondantes.
- **Visualisations Comparatives** : Si plusieurs puits correspondent aux critères, des graphiques comparatifs sont générés (e.g., comparaison des durées totales, des distributions de types de fluides).

Gestion des Rapports et des Notifications

- **Navigateur de Rapports ('reports.html')** : Une interface simple en accordéon permet de naviguer dans l'arborescence des dossiers et d'ouvrir les fichiers DDR HTML originaux dans un nouvel onglet pour consultation.
- **Centre de Notifications ('notifications_list.html' et Bell Icon)** : Les alertes de risque générées par le système d'IA sont centralisées sur une page dédiée et accessibles via une icône de cloche dans la barre de navigation. Les notifications non lues sont mises en évidence. Un clic sur une notification ouvre une fenêtre modale détaillant le risque, les preuves et les solutions proposées par le modèle RAG/LLM.

Annexe E : Déploiement d'Applications Web avec Docker

Dans le cadre d'un projet d'application web complexe comme le nôtre, qui intègre des modèles d'intelligence artificielle et gère des dépendances logicielles spécifiques, la question du déploiement est primordiale. Pour assurer la portabilité, la reproductibilité et la scalabilité de notre solution, l'utilisation de technologies de conteneurisation comme **Docker** représente une étape logique et une bonne pratique industrielle. Cette annexe présente les concepts clés de Docker et leur pertinence pour le déploiement de notre application.

Docker est une plateforme open-source qui permet d'automatiser le déploiement, la mise à l'échelle et la gestion des applications à l'aide de la conteneurisation. Il offre un moyen efficace et léger de "packager" une application, ainsi que toutes ses dépendances (bibliothèques Python, modèles de ML, fichiers de configuration), dans des unités autonomes et standardisées appelées **conteneurs**. Ces conteneurs sont isolés les uns des autres et partagent le noyau du système d'exploitation de la machine hôte, ce qui les rend extrêmement portables et garantit un comportement cohérent, que l'application soit exécutée sur l'ordinateur d'un développeur, un serveur de test ou une infrastructure de production dans le cloud [18].

Les concepts clés liés à Docker sont les suivants :

1. **Image Docker** : Une image est un modèle statique, en lecture seule, qui contient tout le nécessaire pour exécuter une application : le code source (notre application Flask), les environnements d'exécution (Python), les bibliothèques (Flask, Pandas, TensorFlow), les variables d'environnement et les fichiers de configuration. Les images sont construites à partir d'un fichier texte appelé *Dockerfile*, qui liste les instructions étape par étape pour assembler l'image.
2. **Conteneur Docker** : Un conteneur est une instance vivante et exécutable d'une image Docker. On peut le voir comme l'application en cours d'exécution. Chaque conteneur fonctionne dans son propre environnement isolé, avec son propre système de fichiers, ses propres processus et sa propre interface réseau, tout en partageant le noyau de l'OS hôte. On peut démarrer, arrêter, et supprimer des conteneurs sans affecter le système hôte ou les autres conteneurs.
3. **Dockerfile** : C'est la "recette" pour construire une image Docker. Il s'agit d'un script qui automatise la création d'une image. Pour notre projet, un *Dockerfile* typique commencerait par une image de base Python, copierait les fichiers de notre projet à l'intérieur, installerait les dépendances listées dans 'requirements.txt', et définirait la commande pour lancer le serveur Flask.
4. **Docker Engine** : C'est le moteur d'exécution qui construit les images et exécute les conteneurs. Il comprend un service d'arrière-plan (le *daemon* Docker) et une interface en ligne de commande (CLI) pour interagir avec ce service.
5. **Docker Compose** : Pour une application comme la nôtre, qui pourrait éventuellement être décomposée en plusieurs services (e.g., un conteneur pour le serveur web Flask, un autre pour une base de données PostgreSQL, et un autre pour un service de prédiction ML), Docker Compose est un outil essentiel. Il permet de définir et de gérer des applications multi-conteneurs à l'aide d'un unique fichier de configuration au format YAML, simplifiant grandement le lancement et l'interconnexion de l'ensemble de la pile applicative.

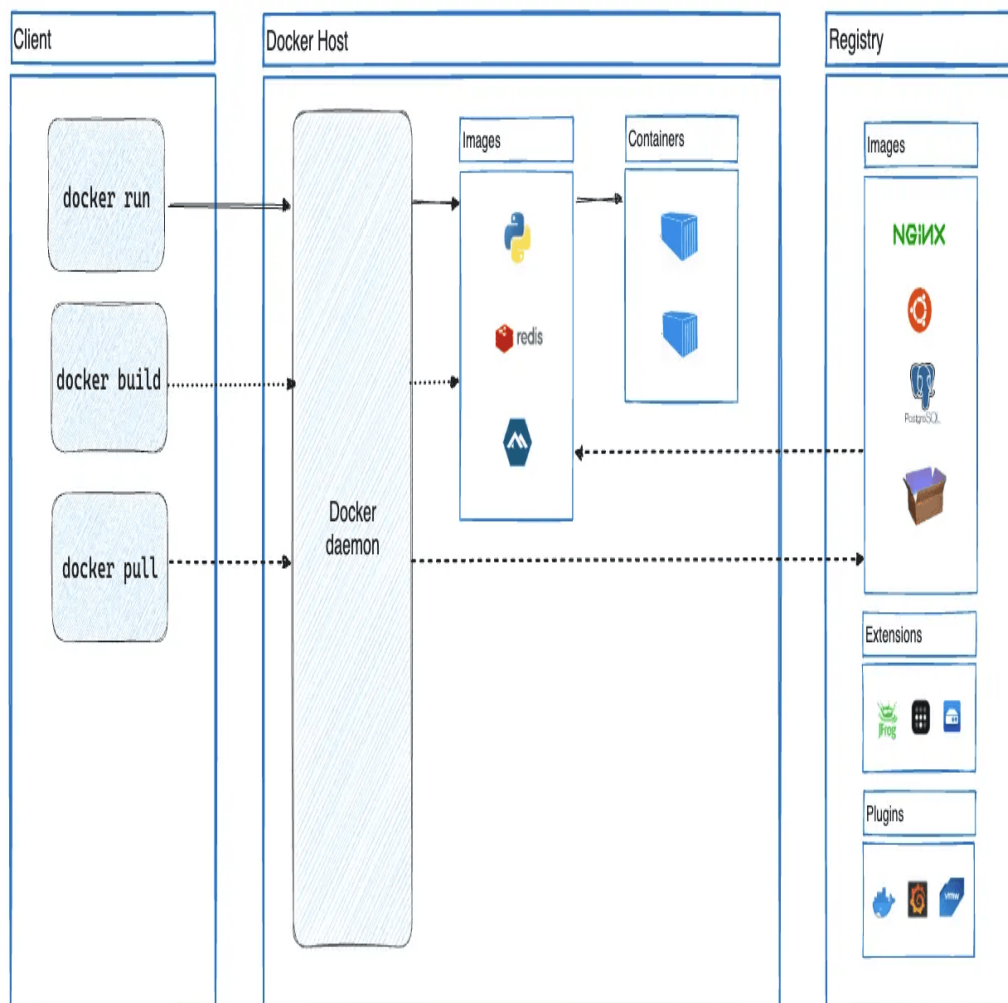


FIGURE E.1 – Architecture des composants de Docker, illustrant la relation entre le client, le démon Docker, les images et les conteneurs [17].

Avantages de Docker pour Notre Projet

L'utilisation de Docker pour notre application d'analyse de DDR offrirait de nombreux avantages :

1. **Environnements Cohérents et Reproductibles** : Le principal avantage est la fin du problème classique du "ça marche sur ma machine". En encapsulant l'application et ses dépendances exactes (versions de Python, de Flask, de TensorFlow, etc.) dans une image, nous garantissons que l'application se comportera de la même manière partout où l'image sera exécutée. Cela simplifie la collaboration entre développeurs et assure une transition fluide entre les environnements de développement, de test et de production.
2. **Gestion Simplifiée des Dépendances** : Notre projet repose sur un ensemble complexe de bibliothèques de science des données et de machine learning. Docker permet de "figer" cet environnement complexe à l'intérieur de l'image, évitant les conflits de version et simplifiant l'installation pour de nouveaux utilisateurs ou sur de nouveaux serveurs.
3. **Isolation des Services** : L'architecture de notre application peut naturellement évoluer vers une approche microservices. Par exemple, le modèle RAG/LLM, qui est très gourmand en ressources, pourrait être isolé dans son propre conteneur, potentiellement sur une machine dotée d'un GPU, tandis que l'application web Flask principale s'exécuterait dans un autre conteneur sur une machine standard. Docker et Docker Compose facilitent la gestion et la communication entre ces services isolés.
4. **Portabilité et Déploiement Facilité** : Une fois notre application conteneurisée, elle peut être déployée sur n'importe quel serveur (local ou dans le cloud, comme AWS, Azure, ou Google Cloud) sur lequel Docker est installé. Cela simplifie énormément le processus de déploiement et de mise à l'échelle.
5. **Scalabilité Horizontale** : Si l'application doit gérer un grand nombre d'utilisateurs simultanés, il devient facile de lancer plusieurs instances (conteneurs) du serveur web Flask et de répartir la charge entre elles à l'aide d'un orchestrateur de conteneurs comme Kubernetes ou Docker Swarm.

Cas d'Usage pour Notre Application

- **Environnement de Développement Unifié** : Chaque membre de l'équipe de développement peut lancer l'application et sa base de données avec une seule commande ('docker-compose up'), garantissant que tout le monde travaille avec la même configuration.
- **Intégration et Déploiement Continus (CI/CD)** : Docker est au cœur des pipelines CI/CD modernes. À chaque modification du code, un pipeline automatisé peut construire une nouvelle image Docker, lancer des tests à l'intérieur d'un conteneur, et si les tests réussissent, déployer automatiquement la nouvelle version de l'application en production.
- **Déploiement en Production** : Pour rendre notre application accessible aux ingénieurs de SLB, la meilleure approche serait de la déployer sous forme de conteneurs sur un serveur d'entreprise ou une plateforme cloud, garantissant ainsi sa disponibilité, sa performance et sa sécurité.

En résumé, bien que le développement initial de notre projet ait été réalisé dans un environnement local, l'adoption de Docker représente l'étape suivante logique pour son industrialisation. Docker révolutionne la manière dont les applications sont packagées et déployées, et son approche de conteneurisation offre des avantages décisifs en termes de portabilité, de gestion des dépendances et de scalabilité, qui sont parfaitement adaptés aux exigences d'une application d'IA complexe comme la nôtre.

Remarque : les informations fournies dans cette annexe constituent un aperçu général de Docker. Pour des informations plus détaillées, consultez la [documentation officielle de Docker](#).

Annexe F : Architecture Technique et Technologies de l'Application Web

Le développement d'une application web capable de répondre aux exigences d'un cas d'usage industriel comme l'analyse de données de forage nécessite une architecture logicielle robuste et une sélection judicieuse de technologies. Cette annexe détaille l'architecture et la pile technologique que nous avons implémentées pour notre solution.

Architecture Client-Serveur Distribuée

Notre application est fondée sur une architecture client-serveur, mais avec une particularité notable : les composants d'intelligence artificielle les plus lourds sont hébergés sur une infrastructure externe. Cette architecture distribuée (Figure E.1) sépare clairement les responsabilités :

- **Le Client (Côté Navigateur)** : Il s'agit de l'interface utilisateur qui s'exécute dans le navigateur web. Il est responsable de l'affichage des tableaux de bord, des graphiques interactifs et des notifications, et de la gestion des interactions utilisateur.
- **Le Serveur Principal (Backend Flask)** : C'est le cœur de l'application qui gère la logique métier principale : l'accès et le traitement des fichiers de données (DDR), le service des pages web, et l'orchestration des appels vers les modèles d'IA.
- **Le Serveur d'IA (Backend Externe)** : Les modèles les plus gourmands en ressources, notamment le LLM, sont exécutés sur une infrastructure externe (notebooks Kaggle avec accélération GPU) et exposés via une API.

Cette séparation permet d'allouer les ressources de calcul de manière optimale et de découpler le développement de l'interface web de celui des modèles d'IA.

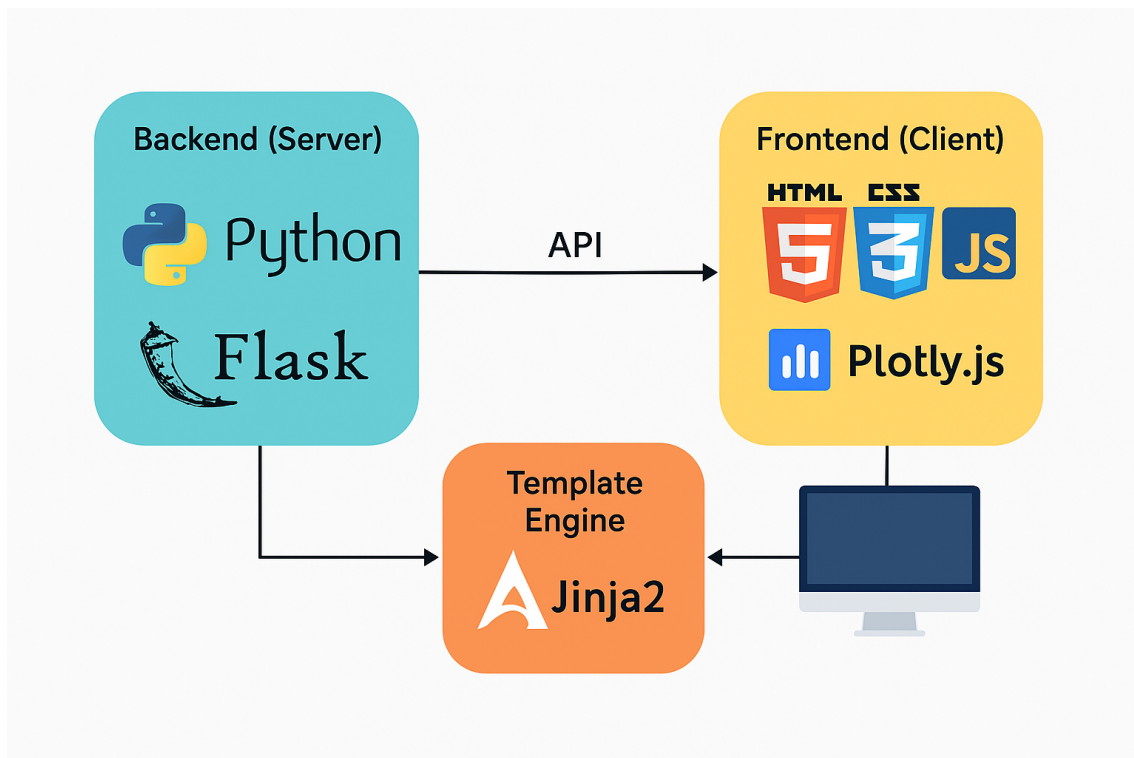


FIGURE E.1 – Architecture Détaillée de l’Application Web Développée, montrant la séparation entre le client (Frontend), le serveur principal (Backend Flask), et le service d’IA externe (LLM sur Kaggle) accessible via ngrok.

Pile Technologique de Notre Projet

Les choix technologiques ont été guidés par la nécessité de combiner un traitement de données puissant avec une interface utilisateur riche et interactive.

Backend (Serveur)

- **Langage et Framework** : Nous avons choisi **Python** avec le micro-framework **Flask**. Ce choix est stratégique car Python est le langage de prédilection pour la science des données et l’apprentissage automatique, ce qui permet une intégration transparente de nos modèles, développés avec des bibliothèques comme **TensorFlow/Keras** et **Transformers** [12, 1]. Flask, par sa légèreté et sa modularité, est idéal pour créer les API RESTful nécessaires à la communication avec le frontend (e.g., les routes `/api/...` dans `app.py`).
- **Hébergement des Modèles d’IA et Exposition via API** : Pour bénéficier de la puissance de calcul des GPU, indispensable à l’exécution de notre LLM, le modèle est hébergé sur la plateforme de notebooks **Kaggle**. Pour permettre à notre application web d’interagir avec ce modèle à distance, nous avons utilisé **ngrok**. Cet outil crée un tunnel sécurisé qui expose l’API de notre service d’IA, exécutée dans l’environnement Kaggle, à une URL publique. Notre serveur Flask peut ainsi interroger le LLM comme n’importe quel autre service web distant, ce qui facilite grandement la communication entre les différents composants de notre architecture distribuée sans nécessiter de déploiement serveur traditionnel complexe pour les modèles d’IA.

Frontend (Client)

Le frontend est construit avec les technologies web standard :

- **Structure et Style : HTML** pour la structure sémantique des pages et **CSS**, en s'appuyant sur le framework **Bootstrap** pour garantir un design responsive et l'utilisation de composants d'interface modernes.
- **Interactivité** : Le **JavaScript** est utilisé pour toute la logique côté client : gestion des événements utilisateur (clics, filtres), communication asynchrone avec le backend via des requêtes AJAX (utilisant l'API 'fetch'), et manipulation dynamique du DOM pour mettre à jour l'interface sans recharger la page.
- **Visualisation de Données** : La bibliothèque JavaScript **Plotly.js** [41] est un pilier de notre frontend. Comme visible dans les fichiers 'index.html', 'view_well_data.html' et 'filter.html', elle nous permet de générer des graphiques scientifiques complexes (trajectoires 3D, diagrammes en violon, graphiques en barres) qui sont entièrement interactifs (zoom, survol, sélection). Les données pour ces graphiques sont transmises depuis le backend Flask sous forme de JSON, puis rendues de manière performante par Plotly.js dans le navigateur du client.

Moteur de Template

Nous utilisons **Jinja2**, le moteur de template par défaut de Flask [44]. Cette technologie est utilisée côté serveur pour générer dynamiquement les pages HTML avant de les envoyer au client. Elle nous permet d'insérer des données (comme les noms de puits, les listes de fichiers, les résultats d'analyse) directement dans le code HTML, rendant les pages personnalisées et dynamiques tout en maintenant une séparation claire entre la logique et la présentation.

Annexe H : Extraits de Code Significatifs

Cette annexe présente des extraits de code clés qui illustrent l'implémentation de certaines fonctionnalités critiques de notre projet.

Extraction de Données Tabulaires (Figure 3.8)

```
import os
from bs4 import BeautifulSoup
import pandas as pd

folder_path = "DDR/15_9_19_A" # Replace with your folder path
table_ids = ["wellboreInfoTable_1", "wellboreInfoTable_2", "wellboreInfoTable_3"]

# List to collect one DataFrame per file
all_reports = []

for filename in sorted(os.listdir(folder_path)):
    if filename.endswith(".html"):
        file_path = os.path.join(folder_path, filename)
        with open(file_path, "r", encoding="utf-8") as file:
            soup = BeautifulSoup(file.read(), "lxml")

        combined_df = pd.DataFrame()

        # Collect and combine the desired tables
        for table_id in table_ids:
            table = soup.find("table", id=table_id)
            if table:
                try:
                    df = pd.read_html(str(table), header=None)[0]
                    df = df.iloc[:, :2] # Only first two columns
                    combined_df = pd.concat([combined_df, df], ignore_index=True)
                except Exception as e:
                    print(f"⚠ Error reading {table_id} in {filename}: {e}")
            else:
                print(f"✗ Table {table_id} not found in {filename}")

        if not combined_df.empty:
            # Create a single-row DataFrame: labels become column names
            row_df = pd.DataFrame([combined_df.iloc[:, 1].values], columns=combined_df.iloc[:, 0].values)
            row_df.insert(0, "Report", filename) # Add filename to track the source
            all_reports.append(row_df)

# Combine all reports
if all_reports:
    final_df = pd.concat(all_reports, ignore_index=True)
    final_df.to_csv("transposed_output.csv", index=False, encoding="utf-8")
    print("\n✅ Transposed output saved as: transposed_output.csv")
else:
    print("✗ No data extracted from any file.")
```

FIGURE E.1 – Extraction des données tabulaires.

Extraction de Texte Libre (Figure 3.9)

```
[ ]: import os
from bs4 import BeautifulSoup
import pandas as pd

# --- Configuration ---
folder_path = "DOR"

table_titles = [
    "Summary of activities (24 Hours)",
    "Summary of planned activities (24 Hours)"
]

all_reports_data = []

print(f"🔍 Recherche de fichiers HTML dans le dossier : {folder_path}...")

# Parcourir tous les fichiers dans le dossier spécifié
for filename in sorted(os.listdir(folder_path)):
    if filename.endswith(".html"):
        file_path = os.path.join(folder_path, filename)
        print(f"📄 Processing file: {filename}")

        with open(file_path, "r", encoding="utf-8") as file:
            soup = BeautifulSoup(file.read(), "lxml")

            # Dictionnaire pour stocker les données de ce fichier spécifique
            report_data = {"Fichier Source": filename}

            # Pour chaque titre de table que nous recherchons...
            for title in table_titles:
                try:
                    # On utilise une fonction lambda pour une recherche plus flexible (ignore les espaces superflus).
                    title_element = soup.find(string=lambda text: title in text if text else False)

                    if title_element:
                        # Une fois le titre trouvé, on cherche la PREMIÈRE balise <table> qui vient après.
                        table = title_element.find_next("table")

                        if table:
                            # Extraire tout le texte de la table en une seule chaîne de caractères.
                            # separator=' ' évite que les mots de cellules différentes se collent.
                            # strip=True nettoie les espaces au début et à la fin.
                            full_text = table.get_text(separator=" ", strip=True)
                            report_data[title] = full_text
                            print(f"✅ Trouvé et extrait : '{title}'")
                        else:
                            report_data[title] = "ERREUR: Titre trouvé mais pas la table associée."
                            print(f"⚠️ Titre '{title}' trouvé, mais aucune table ne le suit.")
                    else:
                        report_data[title] = "NON TROUVÉ"
                        print(f"❌ Titre non trouvé : '{title}'")

                except Exception as e:
                    print(f"🚨 Erreur inattendue en cherchant '{title}' dans {filename}: {e}")
                    report_data[title] = f"ERREUR: {e}"

            # Ajouter les données extraites de ce fichier à notre liste globale
            all_reports_data.append(report_data)

# --- Finalisation et sauvegarde ---
if all_reports_data:
    # Créer un DataFrame pandas à partir de la liste de dictionnaires
    final_df = pd.DataFrame(all_reports_data)

    # Réorganiser les colonnes pour avoir le fichier source en premier
    cols = ["Fichier Source"] + [col for col in final_df.columns if col != "Fichier Source"]
    final_df = final_df[cols]

    # Sauvegarder le DataFrame dans un fichier CSV
    output_filename = "resume_activites_extraites.csv"
    final_df.to_csv(output_filename, index=False, encoding="utf-8-sig") # utf-8-sig pour une meilleure compatibilité avec Excel

    print(f"🎉 Extraction terminée ! Les données ont été sauvegardées dans : {output_filename}")
else:
    print("\n❌ Aucune donnée n'a été extraite. Vérifiez le chemin du dossier et les noms des fichiers.")
```

FIGURE E.1 – Extraction des données textuelles.