



RAPPORT DE STAGE D'INITIATION

***SURVEILLANCE EN TEMPS RÉEL DES VÉHICULES;
ALERTE ET SÉCURITÉ À SIDI CHENNANE.***

FILIÈRE BIG DATA ET INTELLIGENCE ARTIFICIELLE

Remerciements :

Prière d'abord et avant tout rendue à Dieu tout- puissant de m'avoir doté de santé, de patience et de force pourachever cette formation et rédiger ce présent rapport.

Un grand merci à tout le personnel encadrant, et particulièrement **Youssef Elyamnaoui**, pour leur guide, leurs conseils avisés et leur soutien, tout au long de mon expérience professionnelle. À **M.Hadi** et **M.Ikkou**, du département électrique, pour leur disponibilité, leurs conseils avisés et leur accompagnement durant cette expérience professionnelle. À **M.Idriss Merzak**, chef de chantier, pour son soutien sur le terrain et pour m'avoir guidé dans la mise en pratique des connaissances acquises.

Leur disponibilité et leur expertise m'ont beaucoup aidé, et j'ai énormément appris grâce à eux.

Je tiens à exprimer également ma reconnaissance à la direction et à l'ensemble des équipes de l'Office Chérifien des Phosphates (OCP) pour m'avoir accueilli, m'avoir donné la possibilité de travailler dans un environnement aussi stimulant. Leur confiance et leur soutien ont été indispensables à la bonne réalisation de notre stage.

De plus, merci à l'École Nationale des Sciences Appliquées de Tétouan (ENSA Tétouan) pour son environnement d'apprentissage et de recherche, qui m'a permis d'acquérir de nouvelles compétences, de me préparer et de relever avec succès les défis de ce stage.

Dédicace :

Je dédie ce rapport de stage à mes chers parents, pour leur amour indéfectible, leur soutien sans faille et leurs sacrifices incessants. Leur confiance en mon potentiel m'a toujours donné du courage et m'a permis de croire en moi. Je leur en suis éternellement reconnaissant.

Je tiens également à remercier mes enseignants qui, par leur enseignement et leur encadrement durant mon cursus académique, m'ont permis au cours de mon parcours d'acquérir des acquis et compétences nécessaires pour la réussite de ce stage.

Les abréviations:

ML : machine learning

DL : deep learning

OCP : L'Office Chérifien des Phosphates

THT : Très Haute Teneur

HT : Haute Teneur

BTR : Basse Teneur Riche

BT : Basse Teneur

TBT : Très Basse Teneur

LSTM: Long Short-Term Memory

CNN: Convolutional Neural Networks

Introduction:

Dans le secteur industriel, et plus particulièrement au sein de grands groupes tels que l'OCP, la gestion et la maintenance d'un vaste parc de véhicules et de machines constituent des défis majeurs. Garantir la disponibilité opérationnelle des équipements tout en assurant la sécurité des employés et en minimisant les coûts est essentiel. Cependant, cette tâche complexe est souvent entravée par des pannes imprévues, des incidents de sécurité, et une utilisation sous-optimale des ressources.

Pour répondre à ces défis, les entreprises se tournent de plus en plus vers des solutions technologiques avancées, combinant l'Internet des objets (IoT), l'intelligence artificielle (IA), et des systèmes de surveillance en temps réel. L'objectif de mon stage chez OCP a été de participer à la mise en œuvre d'une application de surveillance en temps réel capable de collecter, analyser et exploiter les données des véhicules et des machines pour anticiper les défaillances, optimiser la maintenance, et renforcer la sécurité.

Ce rapport présente les différentes étapes de conception et de développement de cette application, en mettant l'accent sur l'utilisation de technologies telles que les réseaux de neurones pour la détection d'anomalies, les modèles prédictifs pour anticiper les pannes, et les protocoles de sécurité pour la gestion des accès. Il examine également les résultats obtenus et les leçons tirées de cette expérience, tout en offrant des perspectives sur l'amélioration future de ce type de système.

Table des Matières

| | |
|--|--------------|
| I. Introduction générale | 1-5 |
| 1. Introduction au Machine Learning | 1 |
| 2. Histoire du Machine Learning..... | 1-2 |
| 3. Présentation de l'OCP | 2-3 |
| 4. Problématique | 4 |
| 5. Solution proposée | 4 |
| 6. Objectifs | 5 |
| II. Observations | 6-12 |
| 1. Une coupe phosphatique..... | 6-8 |
| 2. Etapes du processus d'extraction du phosphate..... | 8-11 |
| 3. Qualités de Phosphate | 11-12 |
| III. Conception de l'application | 13-14 |
| 1. Diagramme de classe | 13-14 |
| IV. Architecture du Système Conception de l'application | 15-21 |
| 1. Composants Principaux | 14-16 |
| 2. Infrastructure | 16-17 |
| 3. Modèles de Machine Learning/Deep Learning | 17-21 |
| a. Détection d'Anomalies (AUTO-ENCODER) | 17-18 |
| b. Prédiction des Pannes (LSTM)..... | 18-19 |
| c. Reconnaissance faciale (CNN)..... | 19-21 |
| V. Réalisation | 22-31 |
| 1. Préparation de le dataset | 22-23 |
| 2. Dashboard | 23-24 |
| 3. Machine learning /Deep learning..... | 25-31 |
| a. Detection d'anomalies (AUTO-ENCODER) | 25-27 |
| b. Prédiction des Pannes (LSTM) | 28-29 |
| c. Reconnaissance faciale (CNN) | 29-31 |
| VI. Conclusion..... | 32 |

1. Introduction générale :

1. Introduction au Machine Learning :

Le Machine Learning est une sous-discipline de l'intelligence artificielle qui permet aux ordinateurs d'apprendre à partir de données sans être explicitement programmés pour chaque tâche spécifique. En analysant de grandes quantités de données, les algorithmes de Machine Learning identifient des patterns et génèrent des prédictions ou des décisions basées sur ces patterns.

Le Machine Learning se divise en trois types principaux :

1. **Apprentissage supervisé** : Les modèles sont entraînés sur des données étiquetées, où les entrées sont associées à des sorties correctes. Le but est de prédire les résultats pour de nouvelles données similaires.
2. **Apprentissage non supervisé** : Le modèle travaille avec des données non étiquetées et tente de découvrir des structures ou des groupements sous-jacents, comme dans le clustering.
3. **Apprentissage par renforcement** : Un agent apprend en interagissant avec un environnement, en recevant des récompenses ou des punitions, et en ajustant ses actions pour maximiser les récompenses au fil du temps.

Le Machine Learning est devenu un outil essentiel dans de nombreux domaines, transformant des secteurs tels que la reconnaissance d'image, la finance, la santé, et bien d'autres, grâce à sa capacité à automatiser l'analyse et à générer des insights à partir de données complexes.

2. Histoire du Machine Learning :

Bien qu'il soit aujourd'hui devenu un domaine central de l'IA (Intelligence Artificielle), les origines du Machine Learning remontent à plusieurs décennies. Son développement a été influencé par l'évolution des mathématiques, de l'informatique, et parfois même des neurosciences.

- **Années 1950 - Débuts de l'intelligence artificielle** : Le concept d'apprentissage automatique a commencé à prendre forme dans les années 1950. Alan Turing, avec son "Test de Turing", a introduit l'idée que les machines pourraient apprendre. En 1952, Arthur Samuel a développé un programme de dames capable d'apprendre à jouer par l'expérience, un des premiers exemples d'un programme capable d'amélioration par l'apprentissage.
- **Années 1960 - Premières avancées** : Dans les années 1960, les premières théories formelles du Machine Learning ont émergé. Frank Rosenblatt a introduit le perceptron, un type de réseau de neurones capable d'apprendre à classer des données. Bien que limité, le perceptron a ouvert la voie à des modèles plus complexes.
- **Années 1980 - Renaissance des réseaux de neurones** : Après une période de stagnation dans les années 1970, le Machine Learning a connu une résurgence dans les années 1980 avec la redécouverte des réseaux de neurones et l'invention de l'algorithme de rétropropagation. Cette méthode a permis d'entraîner des réseaux de neurones plus profonds, relançant l'intérêt pour ce domaine.

- **Années 1990 - L'essor des algorithmes** : Les années 1990 ont vu l'émergence d'algorithmes comme les machines à vecteurs de support (SVM) et les forêts aléatoires, qui ont permis de résoudre des problèmes plus complexes de classification et de régression. Parallèlement, l'augmentation de la puissance de calcul et l'accessibilité des données ont favorisé la croissance du Machine Learning.
- **Années 2000 - Big Data et Deep Learning** : Avec l'explosion des données (Big Data) et des avancées technologiques, le Machine Learning est devenu plus puissant et plus largement utilisé. Le Deep Learning, une branche du Machine Learning basée sur des réseaux de neurones profonds, a émergé comme une technique dominante, permettant des avancées majeures dans la reconnaissance d'image, le traitement du langage naturel, et bien plus.
- **Années 2010 à aujourd'hui - Révolution de l'IA** : Le Machine Learning est devenu un pilier de l'intelligence artificielle moderne. Il est maintenant utilisé dans une vaste gamme d'applications, des voitures autonomes aux systèmes de recommandation, en passant par les assistants virtuels. Les progrès continus en puissance de calcul, en algorithmes et en disponibilité des données ont permis au Machine Learning de transformer de nombreux secteurs.

3. Présentation de l'OCP :



Logo de l'OCP

L'Office Chérifien des Phosphates (OCP) est une entreprise marocaine de renommée mondiale, spécialisée dans l'extraction, la valorisation, et l'exportation de phosphates et de ses dérivés. Fondée en 1920, l'OCP s'est progressivement imposé comme le leader mondial du marché des phosphates, avec une expertise couvrant l'ensemble de la chaîne de valeur, depuis l'extraction du minerai brut jusqu'à la production d'engrais phosphatés.

L'OCP joue un rôle stratégique dans l'économie marocaine, étant l'un des principaux contributeurs au PIB national et un employeur majeur dans le pays. L'entreprise est également un acteur clé sur le marché international, exportant ses produits vers plus de 160 pays. Elle se distingue par son engagement en faveur de l'innovation, de la durabilité et du développement économique.

En termes d'innovation, l'OCP investit massivement dans la recherche et le développement, cherchant constamment à améliorer ses processus de production tout en réduisant son empreinte écologique. Cet engagement se traduit par des projets de transformation numérique, des initiatives de développement durable, et des partenariats avec des institutions académiques et de recherche.

En outre, l'OCP est engagé dans des projets de responsabilité sociale, visant à améliorer les conditions de vie des communautés locales, à soutenir l'éducation et à promouvoir le développement durable dans les régions où il opère.

Aujourd'hui, l'OCP continue de se positionner en tant qu'acteur clé dans l'industrie mondiale des phosphates, avec une vision claire : nourrir durablement la planète tout en contribuant au développement économique et social du Maroc.

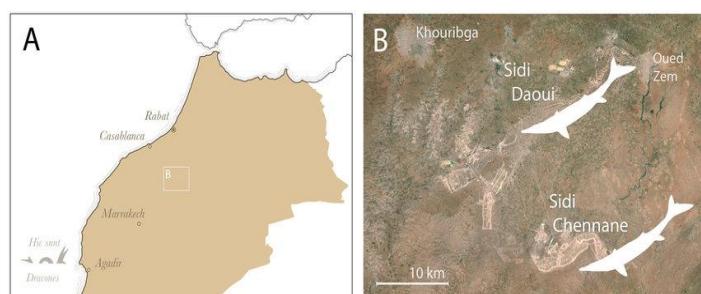
OCP Khouribga - Site Sidi Chennane

Le site Sidi Chennane, situé à Khouribga, est l'un des sites phares de l'Office Chérifien des Phosphates (OCP). Khouribga est considérée comme la capitale mondiale des phosphates en raison de ses vastes réserves de ce minerai stratégique, et le site Sidi Chennane en est un pilier essentiel.

Caractéristiques du site :

- **Exploitation minière :** Le site de Sidi Chennane est l'un des plus grands gisements de phosphates exploités par l'OCP. Il se distingue par son extraction à ciel ouvert, où des méthodes modernes d'extraction sont mises en œuvre pour optimiser la production tout en minimisant l'impact environnemental.
- **Production et traitement :** Après extraction, les phosphates sont transportés vers des installations de lavage et d'enrichissement pour être transformés en produits de haute qualité destinés à l'exportation. Le site est équipé de technologies de pointe pour assurer un traitement efficace des phosphates tout en respectant les normes environnementales les plus strictes.
- **Innovation et durabilité :** Le site Sidi Chennane est à la pointe de l'innovation en matière de production minière. L'OCP y met en œuvre des solutions technologiques avancées pour améliorer l'efficacité opérationnelle, notamment dans le domaine de la gestion des ressources et de la réduction de la consommation d'eau. L'entreprise y mène également des initiatives pour la réhabilitation des sites miniers et la protection de l'environnement local.
- **Engagement social et économique :** En plus de sa contribution économique majeure, le site de Sidi Chennane joue un rôle important dans le développement social de la région de Khouribga. L'OCP s'engage à créer des emplois locaux, à soutenir l'éducation, et à améliorer les infrastructures locales, renforçant ainsi le tissu économique et social de la région.

En résumé, le site Sidi Chennane de l'OCP à Khouribga est un acteur clé non seulement pour l'industrie marocaine des phosphates, mais aussi pour le développement durable et économique de la région. Ce site incarne la vision de l'OCP d'une exploitation minière responsable et innovante, au service du Maroc et du monde.



Localisation Map de Sidi chennane

4. Problématique :

Dans un environnement industriel tel que celui de l'OCP, la gestion efficace et sécurisée d'un parc de machines et de camions représente un défi majeur. Avec une flotte aussi vaste, il devient complexe de surveiller en continu l'état de chaque véhicule, notamment la température et la tension des pneus, le niveau de carburant, ainsi que d'autres paramètres critiques. Cette difficulté est exacerbée par la nécessité de garantir la sécurité des employés, qui dépend de la fiabilité de ces machines et de la rapidité des interventions en cas de dysfonctionnement. Comment alors mettre en place un système de gestion et de surveillance capable de garantir à la fois l'efficacité opérationnelle et la sécurité du personnel dans un tel contexte ?

5. Solution proposée :

Application de Surveillance en Temps Réel.

Pour répondre aux défis de gestion et de sécurité des machines et camions chez OCP, nous proposons la mise en place d'une application de surveillance en temps réel. Cette application permettrait de centraliser toutes les informations relatives aux véhicules et aux machines en un seul endroit, offrant ainsi une vue d'ensemble sur leur état et leurs mouvements. De plus, elle enverrait des alertes en cas de problème détecté sur un véhicule ou si un individu non autorisé tente d'accéder à une zone à risque.

Caractéristiques de l'application :

1. **Suivi en temps réel** : L'application fournirait des informations en direct sur chaque véhicule, telles que le niveau de carburant, la température du moteur, la température et la tension des pneus, ainsi que l'heure de démarrage et la durée de fonctionnement depuis le dernier démarrage. De plus, elle enregistrerait le kilométrage parcouru par chaque véhicule, permettant ainsi un suivi précis et constant de l'état de la flotte.
2. **Alertes et notifications de sécurité** : En cas d'anomalie, comme une température excessive, un niveau de carburant critique, ou une machine en inactivité prolongée, l'application émettrait des alertes immédiates pour permettre une intervention rapide. De plus, un système de sécurité intégré déclencherait des notifications si une personne non autorisée pénètre dans une zone restreinte, assurant ainsi la sécurité des employés et des installations.
3. **Gestion des accès** : L'application pourrait également être configurée pour contrôler l'accès aux zones sensibles, n'autorisant que les personnes préapprouvées. Toute tentative d'accès non autorisé serait immédiatement signalée aux responsables de la sécurité via des notifications en temps réel.

Cette solution, en intégrant des technologies avancées de surveillance et de gestion des processus industriels, permettrait d'améliorer considérablement l'efficacité opérationnelle et la sécurité au sein d'OCP. Elle faciliterait également la prise de décisions informées, basées sur des données en temps réel, et garantirait une meilleure gestion des ressources.

6. Objectifs :

Optimisation de la gestion des véhicules et des machines :

- Centraliser les informations relatives à l'état des véhicules et des machines, telles que le niveau de carburant, la température du moteur, la tension et la température des pneus, afin d'améliorer l'efficacité opérationnelle.
- Réduire les temps d'arrêt imprévus grâce à une surveillance proactive et à la détection précoce des anomalies.

Amélioration de la sécurité des employés et des installations :

- Mettre en place un système d'alerte automatique pour réagir rapidement en cas de défaillance des machines ou d'incidents liés à la sécurité.
- Surveiller l'accès aux zones sensibles pour empêcher les intrusions non autorisées et protéger les employés contre les risques potentiels.

Réduction des coûts opérationnels :

- Minimiser les coûts liés à l'entretien et à la réparation des véhicules en identifiant les problèmes avant qu'ils ne deviennent critiques.
- Optimiser l'utilisation des ressources (carburant, temps de fonctionnement des machines) pour une gestion plus rentable du parc de véhicules.

Amélioration de la prise de décision :

- Fournir aux gestionnaires des données en temps réel et des rapports détaillés pour une prise de décision plus rapide et plus éclairée.
- Utiliser l'historique des données pour identifier des tendances et planifier les opérations futures de manière plus efficace

II . Observations :

Lors de ma tournée à l'OCP (Sidi Chennane, Khouribga) j'ai observé et collecté des données pertinentes.

Avant de commencer chaque tour, il est impératif de porter le casque, le gilet et les chaussures de sécurité ;



Casque



Chaussures



Gilet

1.Une coupe phosphatique :

- ➡ J'ai eu l'opportunité de visiter et d'étudier en détail une coupe phosphatique, une formation géologique essentielle pour l'exploitation du phosphate. Cette visite avait pour objectif de comprendre la stratification géologique des gisements phosphatiques et d'acquérir une meilleure connaissance des différentes couches qui composent cette formation. La coupe phosphatique que j'ai observée se caractérise par une alternance de couches de calcaire et de phosphate, chacune jouant un rôle crucial dans l'exploitation minière.



Une coupe phosphatique

La coupe phosphatique se compose des éléments suivants :

Terrain Naturel :

- Constituant la base de la coupe, le terrain naturel est la formation géologique originelle, composée de roches mères. Cette couche sert de fondation à l'ensemble de la stratification.

Sillon B :

- Juste au-dessus du terrain naturel se trouve le "Sillon B". Il s'agit d'une épaisse bande de calcaire compacte, souvent utilisée comme une référence géologique pour distinguer le début de la séquence phosphatique. Cette couche est significative en raison de sa composition en carbonate de calcium, qui la rend dense et résistante.

Plaques de Calcaire :

- Entre le Sillon B et la première couche de phosphate, on trouve une couche de plaques de calcaire. Cette couche joue un rôle de séparation naturelle et ajoute de la stabilité à la stratification en marquant une transition entre le Sillon B et les couches phosphatiques.

Couche 1 de Phosphate :

- Au-dessus des plaques de calcaire repose la première couche de phosphate. Elle est constituée de phosphorite, riche en phosphates de calcium, et constitue la première section exploitable pour l'extraction de phosphate.

Plaques de Calcaire :

- Cette couche intermédiaire de plaques de calcaire se situe entre la première et la deuxième couche de phosphate. Elle sert de barrière naturelle, séparant les différentes couches de phosphate et apportant une stabilité structurelle à la formation.

Couche 2 de Phosphate :

- Au-dessus des plaques de calcaire, on trouve la deuxième couche de phosphate. Cette couche est similaire à la première en termes de composition, mais peut varier en épaisseur et en concentration de phosphate.

Plaques de Calcaire :

- Encore une série de plaques de calcaire se trouve entre la deuxième et la troisième couche de phosphate, continuant la séquence d'alternance.

Couche 3 de Phosphate :

- La troisième couche de phosphate, située au-dessus des plaques de calcaire, représente la dernière couche exploitable de phosphate dans cette stratification.



Coupe lithologique de la mine de sidi chennane

2. Etapes du processus d'extraction du phosphate :

→ J'ai aussi eu l'opportunité de suivre de près plusieurs étapes cruciales du processus d'extraction du phosphate, à savoir : la foration, le sautage, le décapage, le défruitage, et l'installation fixe. Ces étapes sont indispensables pour l'extraction et le traitement efficace du phosphate. Chaque étape implique l'utilisation de différentes machines, véhicules, et camions spécialisés, assurant ainsi un déroulement fluide des opérations.

Étapes et Définitions

1. Foration :

- **Définition :** La foration est la première étape de l'extraction où des trous sont percés dans la roche ou le sol pour y insérer des explosifs. Cette étape est essentielle pour préparer le terrain pour le sautage. La foration se fait généralement à l'aide de la foreuse **DML**, qui opère selon un maillage précis de **6 mètres** entre chaque trou. Chaque trou est foré à une profondeur de **10 mètres** et a un diamètre de **20 cm**.

- **Machines Utilisées :**

- Foreuse **DML** : Conçue pour percer des trous profonds et précis dans la roche selon les spécifications mentionnées.



Foreuse DML

2. Sautage :

Définition : Le sautage est l'étape qui suit la foration, où les explosifs sont insérés dans les trous forés, puis déclenchés pour fragmenter la roche. Cela permet un accès plus facile aux couches de phosphate.

Dosage des Explosifs : Un aspect important du sautage est le dosage précis des agents explosifs. Par exemple, un dosage de 1 gramme d'ammoniaque est ajouté pour chaque mètre cube de sol. Cette mesure est cruciale pour assurer une fragmentation efficace de la roche tout en minimisant les risques et les impacts environnementaux.



Mailles pour sautage

3. Décapage :

- **Définition :** Le décapage consiste à enlever les couches supérieures de stériles (roches non phosphatées) pour accéder aux couches de phosphate. Cette étape est souvent réalisée en utilisant des bulldozers et des draglines.
- **Machines Utilisées :**
 - Bulldozers : Pour pousser et enlever les matériaux de surface.
 - Draglines : Utilisées pour enlever de grandes quantités de matériaux et les transporter à une certaine distance.



Draglines



Bulldozers

4. Défruitage :

- **Définition :** Le défruitage est le processus de retrait des dernières couches de matériaux stériles juste au-dessus des couches de phosphate. Cela permet d'exposer complètement le phosphate avant l'extraction.

- **Machines Utilisées :**

- Pelle mécanique : Pour retirer les couches finales de stériles.

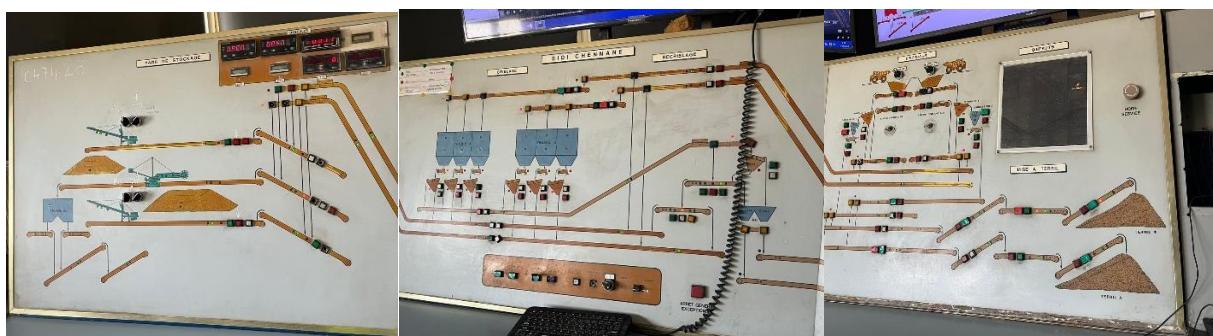


Pelle mécanique

5. Installation Fixe :

- **Définition :** L'installation fixe fait référence aux infrastructures permanentes sur le site minier, telles que les concasseurs, les tamis, et les convoyeurs, qui sont utilisées pour le traitement initial du phosphate extrait.

Schéma généralisé de l'installation fixe :



Étapes de l'Installation Fixe :

1. Transport du Phosphate :

- **Description :** Après extraction, le phosphate est transporté par des camions **Hitachi** vers l'installation fixe. Ces camions assurent le déplacement du matériau brut depuis les zones de décapage ou de défruitage jusqu'à l'installation de traitement.

2. Déchargement dans la Trémie de Réception :

- **Description :** Le phosphate brut est déchargé dans l'une des deux trémies de réception à l'installation fixe. Ces trémies servent de point d'entrée pour le phosphate brut dans le processus de traitement. Chaque trémie reçoit du phosphate, qui est ensuite acheminé vers les étapes de traitement suivantes.

3. Passage par l'Équipement de Mesure (EM1) :

- **Description** : Après le déchargement, le phosphate passe par l'**EM1**, un équipement de mesure qui évalue les caractéristiques du matériau, telles que sa granulométrie et sa qualité initiale, avant de poursuivre le traitement.

4. Épierrage :

- **Description** : Le phosphate brut est ensuite soumis à une étape d'épierrage pour éliminer les gros morceaux de matière supérieure à 90 mm. Cette étape permet de séparer les blocs de roche ou de débris qui pourraient endommager les équipements de traitement ultérieurs.
- **Traitement Supplémentaire** :
 - **Concassage** : Si la matière dépasse 90 mm, elle est envoyée vers un concasseur pour être réduite en plus petites particules. Le concassage permet de traiter les matériaux de grande taille afin qu'ils puissent être manipulés plus efficacement dans les étapes suivantes.
 - **Tamis Vibrants** : Après le concassage, le matériau est tamisé pour séparer les particules selon leur taille. Les particules appropriées sont ensuite stockées dans un **terril**, tandis que celles qui ne répondent pas aux critères sont retournées pour un traitement supplémentaire.

5. Convergence vers le Criblage :

- **Description** : Si le phosphate a une taille inférieure à 90 mm, il est transporté par un convergeur vers l'étape de criblage. Cette étape permet de trier le phosphate en fonction de différentes tailles de particules, assurant une qualité uniforme du produit fini.

6. Criblage :

- **Description** : Le criblage se fait à l'aide de deux tamis principaux :
 - **Tamis 60/30 mm** : Ce tamis trie les particules dont la taille est 30/60 mm.
 - **Tamis 30/15 mm** : Le tamis en dessous trie les particules dont la taille est 15/30 mm.
- **Résultats du Criblage** : Le phosphate criblé est ensuite stocké séparément en fonction de la taille des particules, prêt à être expédié ou traité ultérieurement.

7. Stockage Final dans les Terrils :

- **Description** : Le phosphate criblé est stocké dans des **terrils** spécifiques en fonction de sa granulométrie. Ces terrils permettent de stocker de grandes quantités de phosphate tout en préservant sa qualité avant l'expédition ou l'utilisation ultérieure.

3. Qualités de Phosphate :

Après le traitement complet dans l'installation fixe, le phosphate est classé et stocké en fonction de sa qualité. Ces différentes qualités reflètent la teneur en phosphate du matériau, déterminée lors des étapes de criblage et d'enrichissement. Il existe cinq principales catégories de qualité :

1. THT (Très Haute Teneur) :

- **Description** : Le phosphate classé comme **Très Haute Teneur (THT)** contient la plus forte concentration en phosphates de calcium. Ce type de phosphate est particulièrement recherché pour les applications nécessitant un produit de très haute qualité, tel que la production d'engrais de première catégorie.
- **Utilisation** : Principalement utilisé dans les processus industriels de haute exigence, où une teneur élevée en phosphates est cruciale.

2. HT (Haute Teneur) :

- **Description** : Le **phosphate Haute Teneur (HT)** présente une concentration légèrement inférieure à celle du THT mais reste un produit de haute qualité. Ce phosphate est largement utilisé dans l'industrie pour la fabrication d'engrais et d'autres produits dérivés.
- **Utilisation** : Convient pour une grande variété d'applications industrielles, où un équilibre entre qualité et quantité est recherché.

3. BTR (Basse Teneur Riche) :

- **Description** : Le phosphate **Basse Teneur Riche (BTR)** est de qualité moyenne, offrant une concentration en phosphates adéquate pour des applications moins exigeantes. Ce type de phosphate est souvent utilisé pour des mélanges ou des traitements qui ne nécessitent pas une très haute concentration.
- **Utilisation** : Employé dans des processus où une teneur en phosphate modérée est suffisante, comme certains types de fertilisants.

4. BT (Basse Teneur) :

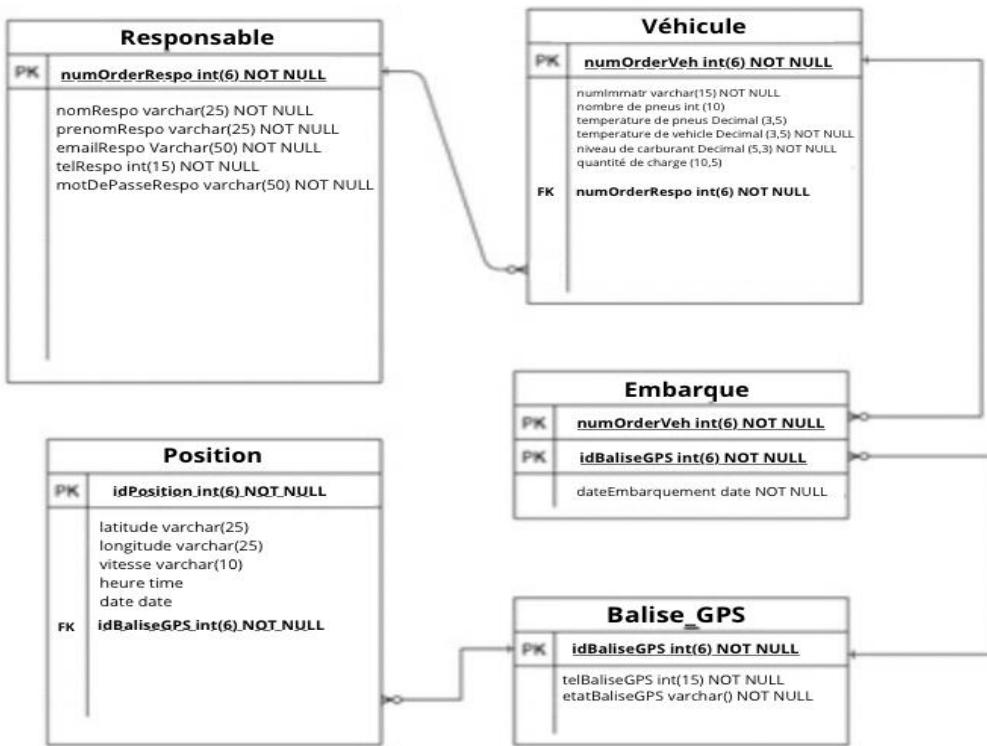
- **Description** : Le phosphate **Basse Teneur (BT)** contient une plus faible concentration en phosphates par rapport aux catégories précédentes. Bien que de moindre qualité, ce phosphate est toujours exploitable et utilisé pour des applications spécifiques.
- **Utilisation** : Utilisé principalement dans des secteurs où une haute concentration n'est pas essentielle, mais où le phosphate reste un élément nécessaire.

5. TBT (Très Basse Teneur) :

- **Description** : Le phosphate **Très Basse Teneur (TBT)** est celui qui a la plus faible concentration en phosphates. Ce matériau est généralement réservé aux usages où la qualité du phosphate n'est pas la priorité principale, mais où il est encore utile en tant que matière première.
- **Utilisation** : Souvent mélangé avec des matériaux de meilleure qualité ou utilisé dans des processus industriels de moindre exigence.

III . Conception de l'application :

- Diagramme de classe :



1. Table Responsable

- **Rôle :** Cette table représente les responsables de la gestion des véhicules. Elle contient des informations personnelles sur les responsables, telles que leur nom, prénom, adresse email, numéro de téléphone, et mot de passe.
- **Relations :**
 - Elle est liée à la table Véhicule via la clé étrangère numOrderRespo. Cela signifie qu'un responsable est assigné à un ou plusieurs véhicules.

2. Table Véhicule

- **Rôle :** Cette table stocke les informations sur les véhicules gérés par l'entreprise. Elle contient des détails tels que le numéro d'immatriculation, le nombre de pneus, les températures des pneus et du véhicule, le niveau de carburant et la quantité de charge.
- **Relations :**
 - Elle est liée à la table Responsable via la clé étrangère numOrderRespo, indiquant quel responsable est en charge de chaque véhicule.

- Elle est également liée à la table Embarque, qui gère l'embarquement des balises GPS sur les véhicules.

3. Table Position

- **Rôle :** Cette table enregistre les positions des véhicules en temps réel. Elle contient des informations comme la latitude, la longitude, la vitesse, la date et l'heure de la position enregistrée.
- **Relations :**
 - Elle est liée à la table Balise_GPS via la clé étrangère idBaliseGPS, ce qui signifie que chaque position est associée à une balise GPS spécifique.

4. Table Balise_GPS

- **Rôle :** Cette table représente les balises GPS installées sur les véhicules pour suivre leur position en temps réel. Elle contient des informations telles que le numéro de téléphone de la balise et son état (fonctionnement ou non).
- **Relations :**
 - Elle est liée à la table Position via idBaliseGPS, ce qui permet d'associer chaque position enregistrée à une balise GPS.
 - Elle est également liée à la table Embarque, qui gère le lien entre les balises et les véhicules.

5. Table Embarque

- **Rôle :** Cette table gère l'association entre les véhicules et les balises GPS. Elle enregistre la date d'embarquement d'une balise GPS sur un véhicule particulier.
- **Relations :**
 - Elle est liée à la table Véhicule via numOrderVeh, ce qui indique quel véhicule a reçu quelle balise.
 - Elle est également liée à la table Balise_GPS via idBaliseGPS, indiquant quelle balise a été installée sur quel véhicule.

IV . Architecture du Système :

1. Composants Principaux :

- **Frontend (Interface Utilisateur)**

- **Technologies** : React.js pour une interface réactive.

- **Fonctionnalités** :

- Tableau de bord (POWER BI) avec vue d'ensemble des véhicules et machines.



- Cartographie en temps réel des positions des véhicules.
 - Rapports et statistiques sur l'état et l'historique des véhicules.

- **Backend (Serveur & API)**

- **Technologies** : Python (Django).

- **Fonctionnalités** :

- Gestion des données : Collecte, stockage, et gestion des informations des véhicules.
 - API RESTful pour interaction avec le frontend.
 - Gestion des utilisateurs et des droits d'accès.
 - Algorithmes de surveillance pour la détection d'anomalies.

- **Base de Données**

- **Technologies** : PostgreSQL.

- **Fonctionnalités** :

- Stockage des données sur les véhicules, les machines, et les événements de surveillance.
 - Historique des mouvements et des alertes.



Logo PostgreSQL

- **Alerte et notifications.**
 - **Technologies** : Twilio pour les communications en temps réel.

Scénarios où Twilio serait utile :

1. **Alertes Critiques par SMS** : Lorsqu'une alerte importante est détectée (comme une intrusion dans une zone restreinte ou un problème grave avec une machine), on peut utiliser Twilio pour envoyer un SMS directement aux responsables concernés.
2. **Notifications de Backup** : En cas de défaillance de la connexion internet ou d'autres problèmes empêchant la réception des alertes via l'application web, Twilio peut servir de canal de backup pour s'assurer que les informations critiques atteignent toujours les utilisateurs.
3. **Appels Téléphoniques Automatisés** : Pour les alertes très urgentes, Twilio peut être utilisé pour déclencher un appel téléphonique automatique aux responsables avec un message pré-enregistré ou généré dynamiquement.



- **Sécurité**
 - **Technologies** : JWT (JSON Web Tokens) pour l'authentification et l'autorisation.
 - **Fonctionnalités** :
 - Chiffrement des données sensibles.
 - Gestion des accès et des permissions selon les rôles des utilisateurs.
 - Surveillance des tentatives d'accès non autorisées.

2. Infrastructure

- **Serveurs**
 - Serveurs cloud pour l'hébergement de l'application (Google Cloud).

Google Cloud Platform (GCP)

Avantages :

- **Services de Données et ML** : GCP excelle dans les services liés aux données et au machine learning. TensorFlow, BigQuery, et AutoML sont particulièrement puissants.
- **Tarification Transparente** : GCP est souvent apprécié pour sa tarification plus transparente et ses remises sur l'utilisation soutenue.
- **Performances Réseau** : Google dispose de l'un des réseaux les plus rapides et les plus sécurisés au monde.

- **Réseau**
 - Réseau sécurisé pour les communications entre les capteurs GPS, les véhicules, et le backend.
 - VPN pour les accès distants des administrateurs.
 - Réseau sécurisé pour la détection des badges RFID des personnes autorisées d'entrer à une zone précise :

1. Utilisation des Badges RFID:

- **Badges RFID:** Les personnes autorisées portent des badges contenant des cartes RFID (Radio Frequency Identification). Ces cartes peuvent être lues à distance par des capteurs.
- **Capteurs RFID :** Des capteurs sont installés aux points d'accès ou dans les zones critiques. Ils peuvent lire les informations du badge à une certaine distance .

2. Communication en Réseau :

- **Capteurs Connectés au Réseau :** Les capteurs sont connectés à un réseau centralisé qui communique avec un serveur. Lorsqu'un badge est détecté, les informations sont immédiatement envoyées au serveur pour traitement.
- **Vérification de l'Autorisation :** Le serveur vérifie dans une base de données si l'ID de la carte lue correspond à une personne autorisée à être dans la zone.
- **Capteurs et Matériel**
 - **GPS :** Capteurs GPS intégrés dans les véhicules et les machines.
 - **IoT Devices :** Capteurs pour mesurer la température, la pression, le niveau de carburant, etc.
 - **Badges RFID :** Pour le contrôle d'accès des individus dans les zones sensibles.
 - **Caméra de surveillance :** Pour le contrôle d'accès des individus dans les zones sensibles.

3. Modèles de Machine Learning/Deep Learning

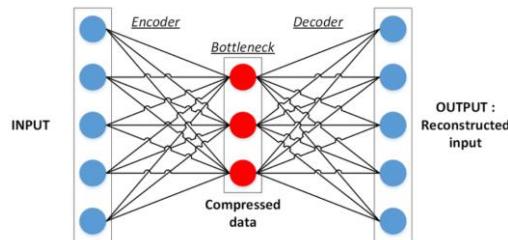
a. Détection d'Anomalies

- **Modèle :** Autoencoder.
- **Fonctionnalités :** Identifier les comportements anormaux des véhicules ou des machines (ex. panne imminente, déviation de trajectoire).

Autoencoders

- **Qu'est-ce que c'est ?**
Les autoencodeurs sont un type de réseau de neurones utilisé pour l'apprentissage non supervisé. Ils apprennent à compresser les données d'entrée en une représentation codée (latent space) et à les reconstruire. Les anomalies sont détectées en mesurant la différence entre les données d'entrée et la reconstruction (erreur de reconstruction).

- **Avantages :**
 - Bien adapté pour les données complexes avec de nombreuses caractéristiques.
 - Peut capturer des structures complexes dans les données.



Architecture d'un auto-encodeur

Un auto-encodeur a une architecture très spécifique, car les couches cachées sont plus petites que les couches d'entrée. On appelle ce type d'architecture une architecture « *bottleneck* ». On peut décomposer un auto-encodeur en trois parties :

1. **Encodeur** : L'encodeur transforme l'entrée en une représentation dans un espace de dimension plus faible appelé **espace latent**. L'encodeur compresse donc l'entrée dans une représentation moins coûteuse.
2. **Goulot d'étranglement (ou bottleneck)** : Cette partie du réseau contient la représentation compressée de l'entrée qui est sera introduite dans le décodeur.
3. **Décodeur** : cette partie doit construire l'output à l'aide de la représentation latente de l'entrée.

Tout d'abord, l'entrée passe par l'encodeur où elle est compressée et stockée dans le goulot d'étranglement, puis le décodeur décomprime ces données afin de retrouver l'entrée originale du réseau.

L'objectif principal de l'auto-encodeur est d'obtenir une sortie identique à l'entrée. L'architecture du décodeur est l'image miroir de l'encodeur. Ce n'est pas une exigence, mais c'est généralement le cas. La seule exigence est la dimensionnalité de l'entrée et de la sortie doivent être les mêmes.

b. Prédiction des Pannes

- **Modèle** : LSTM (Long Short-Term Memory).
- **Fonctionnalités** : Prévoir les pannes potentielles des véhicules en fonction des données historiques .

LSTM (Long Short-Term Memory)

- **Qu'est-ce que c'est ?**
LSTM est un type de réseau de neurones récurrent conçu pour apprendre des dépendances temporelles à long terme dans les séquences de données. Il est particulièrement utile pour les séries temporelles et les données où l'ordre des événements est important.
- **Avantages :**
 - Excellent pour modéliser des données séquentielles et des séries temporelles.
 - Peut capturer des relations temporelles complexes dans les données.

La cellule mémoire d'un LSTM est composée de plusieurs portes : une porte d'entrée, une porte de sortie et une porte d'oubli. Ces portes régulent le flux d'informations à l'intérieur de la cellule mémoire, permettant ainsi de contrôler les informations à retenir et celles à oublier. Cela donne aux LSTM la capacité de mémoriser des informations importantes sur de longues séquences et

d'ignorer les éléments moins pertinents. h_t est l'état caché habituel des RNN mais dans les réseaux LSTM nous ajoutons un deuxième état appelé c_t . Ici, h_t représente la mémoire courte du neurone et c_t représente la mémoire à long terme.

La porte d'entrée (1) détermine quelles informations doivent être mises à jour dans la cellule mémoire. Elle prend en compte l'entrée actuelle x_t ainsi que l'état précédent de la cellule mémoire h_{t-1} et génère un vecteur d'activation qui représente les informations à ajouter à la cellule c_{t-1} . Cet ajout d'informations se traduit par une opération mathématique effectuée entre ce vecteur d'activation et l'état précédent c_{t-1} .

La porte d'oubli (2) permet au LSTM de supprimer les informations obsolètes de la cellule mémoire. Elle utilise à la fois l'entrée actuelle et l'état précédent pour générer un vecteur d'activation qui détermine quelles informations doivent être oubliées. Cela se traduit également par une opération mathématique effectuée entre ce deuxième vecteur d'activation et l'état précédent c_{t-1} . C'est à partir des deux opérations précédentes sur c_{t-1} que l'on obtient l'état actuel c_t .

Enfin, la porte de sortie (3) détermine la sortie du LSTM à un instant donné. Elle utilise l'entrée actuelle x_t et l'état actuel de la cellule mémoire c_t pour générer un vecteur d'activation qui représente la sortie du LSTM. C'est ainsi que l'on obtient h_t .

La combinaison de ces trois portes permet au réseau LSTM de gérer efficacement les dépendances à long terme. Lors de la rétropropagation du gradient, les LSTM peuvent maintenir un flux d'informations constant à travers le temps, évitant ainsi le problème du « vanishing gradient » et permettant un apprentissage plus stable et plus précis.

Forget gate : $f_t = \sigma(W_{hf} h_{t-1} + W_{xf} x_t)$

Dans le cas extrême, indique à l'état de la cellule (mémoire à long-terme) les informations à oublier (multiplication par 0) ou à conserver (multiplication par 1).

Input gate : $i_t = \sigma(W_{hi} h_{t-1} + W_{xi} x_t)$

déterminer quelles informations produites par la couche tanh doivent entrer dans l'état de la cellule (donc à sauvegarder dans la mémoire à long terme).

Output gate : $o_t = \sigma(W_{ho} h_{t-1} + W_{xo} x_t)$

Indique les informations qui doivent passer au prochain état caché h_t .

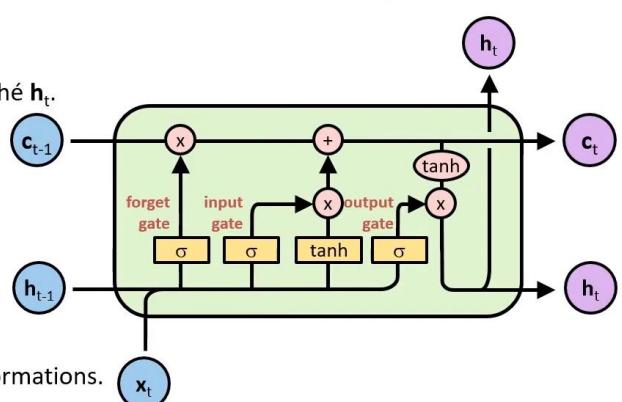
Calcul de c_t et de h_t :

$$c_t = f_t * c_{t-1} + i_t * \tanh(W_{hh} h_{t-1} + W_{xh} x_t)$$

$$h_t = o_t * \tanh(c_t)$$

L'état de la cellule c_t parcourt toute la chaîne du réseau avec des interactions linéaires mineures, ce qui leur permet de transporter efficacement les informations.

Les portes (gates) permettent de réguler intelligemment ces informations.



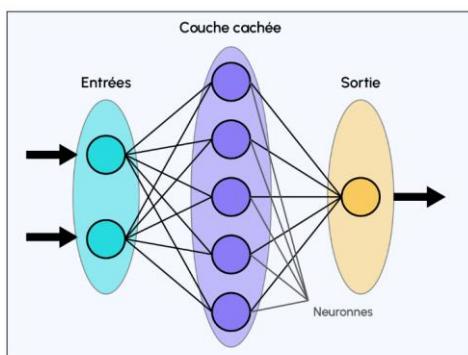
c. Reconnaissance de Modèles d'Intrusion

- **Modèle :** CNN (Convolutional Neural Networks).
- **Fonctionnalités :** Un système de surveillance par caméra sera mis en place pour capturer des images des personnes entrant dans les zones sensibles. Les caméras seront positionnées aux points d'accès stratégiques pour surveiller en temps réel.
- **Qu'est-ce que c'est ?**
Les CNN sont un type de réseau de neurones principalement utilisé pour traiter des données structurées en grille, comme les images. Cependant, ils peuvent également être appliqués à d'autres types de données, comme les séries temporelles ou les signaux des capteurs, en exploitant leurs capacités de détection de motifs.
- **Avantages :**
 - **Détection de motifs :** Les CNN sont extrêmement efficaces pour détecter des motifs complexes dans les données, ce qui les rend utiles pour identifier des anomalies dans les données des capteurs.
 - **Flexibilité :** Peut être adapté à différents types de données, notamment les signaux provenant de capteurs, les séries temporelles, etc.
 - **Efficacité :** Fonctionne bien même avec des données bruyantes, car les couches de convolution peuvent extraire les caractéristiques importantes.

Ajout d'une Surveillance par Caméra avec CNN :

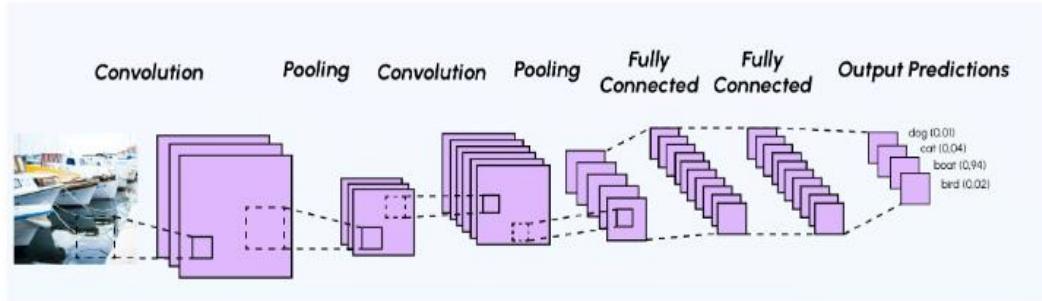
En complément des badges RFID, un système de surveillance par caméra sera mis en place pour capturer des images des personnes entrant dans les zones sensibles. Les caméras seront positionnées aux points d'accès stratégiques pour surveiller en temps réel.

- **Captures d'Images :** Lorsque quelqu'un entre dans une zone surveillée, les caméras capturent automatiquement des images ou des vidéos de la personne.
- **Analyse par CNN :** Ces images sont ensuite transmises à un serveur où un modèle de Convolutional Neural Network (CNN) analyse les images pour détecter des anomalies, comme l'absence de badge ou des comportements suspects.
 - **Reconnaissance Faciale :** Le CNN peut être entraîné pour reconnaître les visages des personnes autorisées et vérifier leur identité en complément du badge.
 - **Détection d'Anomalies :** Si le modèle CNN détecte une personne sans badge visible, ou s'il détecte un visage non autorisé, une alerte est immédiatement envoyée aux responsables de la sécurité.
- **Actions en Temps Réel :** En cas de détection d'une personne non autorisée par les caméras, le système pourrait automatiquement envoyer d'alertes.



Représentation d'un perceptron multicouche

L'algorithme que les perceptrons utilisent pour mettre à jour leurs poids (ou coefficients de réseaux) s'appelle **la rétropropagation du gradient de l'erreur**, célèbre algorithme de descente de gradient que nous verrons plus en détail par la suite.



Exemple d'architecture d'un CNN

Convolution :

- **Détection des Caractéristiques** : Le CNN applique plusieurs filtres convolutionnels à l'image pour extraire des caractéristiques comme les contours, les formes, et les textures. Par exemple, il peut détecter la forme d'un badge ou la silhouette d'une personne.

Pooling :

- **Réduction de la Taille** : Le pooling (généralement max-pooling) réduit la dimension de la carte de caractéristiques tout en conservant les informations les plus importantes. Cela rend le modèle plus efficace tout en préservant les caractéristiques essentielles.

Couches Convolutionnelles Supplémentaires :

- **Apprentissage des Motifs Complexes** : À mesure que les données passent par plusieurs couches, le réseau apprend des motifs de plus en plus complexes, comme la présence d'un badge sur une personne.

Couches Fully Connected (Dense) :

- **Décision Finale** : Les caractéristiques extraites par les couches convolutionnelles sont ensuite passées à travers des couches fully connected qui effectuent la classification finale. Le réseau décide si l'image montre une personne autorisée ou non autorisée.

Sortie :

- **Classifications et Alertes** : Si le modèle CNN détecte une personne sans badge ou avec un comportement suspect, il déclenche une alerte. Sinon, il permet l'accès.

V. Réalisation :

1. Préparation de la dataset :

On va insérer les données avec pgAdmin 4 (SQL) :

The screenshot shows the pgAdmin 4 interface with a connection to DB1/postgres@hiby. The left sidebar (Object Explorer) lists databases (public, hiby), schemas (public, hiby), and tables (e.g., cars, catalog, events, triggers, extensions, fdw, languages, publications, sequences, tables). The main pane shows a query history with multiple INSERT INTO statements for the 'vehicles' table, each specifying a vehicle name, registration number, number of tires, tire tension, and temperature. The scratch pad is empty.

```
75 ('Bulldozer', 'BULL-013', 4, 36.0, 60.0, 87.0, '2024-08-03', 'En service', 49,7128, -74.08660);
76 ('Bulldozer', 'BULL-014', 4, 29.0, 59.0, 84.0, '2024-08-04', 'En service', 49,7128, -74.08660);
77 ('Bulldozer', 'BULL-015', 4, 31.0, 61.0, 82.0, '2024-08-05', 'En maintenance', 49,7128, -74.08660);
78 ('Bulldozer', 'BULL-016', 4, 26.0, 58.0, 80.0, '2024-08-06', 'En service', 49,7128, -74.08660);
79 ('Bulldozer', 'BULL-017', 4, 29.0, 59.0, 88.0, '2024-08-07', 'En service', 49,7128, -74.08660);
80 ('Bulldozer', 'BULL-018', 4, 32.5, 62.5, 85.0, '2024-08-08', 'En service', 49,7128, -74.08660);
81
82 -- Insertion des données pour 2 Draglines
83 ✓ INSERT INTO vehicles (vehicle_name, registration_number, num_tyres, tyre_tension, tyre_temperature,
84   VALUES
85   ('Dragline', '8400', 16, 34.0, 64.0, 95.0, '2024-08-11', 'En service', 34.0522, -118.2437),
86   ('Dragline', '1570', 16, 33.0, 63.0, 92.5, '2024-08-12', 'En service', 34.0522, -118.2437);
87
88 -- Insertion des données pour 7 Hitachi
89 ✓ INSERT INTO vehicles (vehicle_name, registration_number, num_tyres, tyre_tension, tyre_temperature,
90   VALUES
91   ('Hitachi', 'HITA-004', 12, 36.5, 66.5, 80.5, '2024-08-05', 'En service', 49,7128, -74.08660),
92   ('Hitachi', 'HITA-005', 12, 35.5, 65.5, 82.5, '2024-08-07', 'En service', 49,7128, -74.08660),
93   ('Hitachi', 'HITA-006', 12, 37.0, 67.0, 78.5, '2024-08-08', 'En service', 49,7128, -74.08660),
94   ('Hitachi', 'HITA-007', 12, 35.0, 65.0, 79.5, '2024-08-09', 'En maintenance', 49,7128, -74.08660),
95   ('Hitachi', 'HITA-008', 12, 36.0, 66.0, 81.0, '2024-08-10', 'En service', 49,7128, -74.08660),
96   ('Hitachi', 'HITA-009', 12, 34.0, 64.0, 77.0, '2024-08-11', 'En service', 49,7128, -74.08660)
```

Pour obtenir cette table :

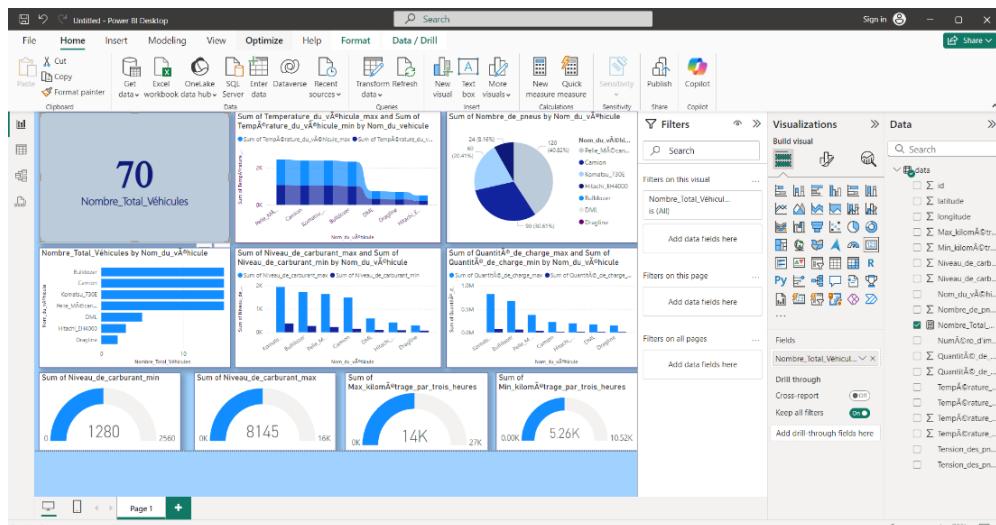
| public.vehicles/DB1/postgres@hibv | | | | | | | | | | | |
|------------------------------------|-----------------|--------------------------------------|--|-------------------|-------------------------------|-----------------------------------|-----------------------------|------------------------|---------------------------------------|------------|--|
| Data Output Messages Notifications | | | | | | | | | | | |
| | id [PK] integer | vehicle_name character varying (100) | registration_number character varying (50) | num_tyres integer | tyre_tension double precision | tyre_temperature double precision | fuel_level double precision | last_service_date date | vehicle_status character varying (50) | gpi double | |
| 1 | 1 | Dregline | 8400 | 16 | 34 | 64 | 95 | 2024-08-11 | En service | | |
| 2 | 2 | Dregline | 1570 | 16 | 33 | 63 | 92.5 | 2024-08-12 | En service | | |
| 3 | 3 | DML | DML-001 | 6 | 35 | 70 | 80 | 2024-08-01 | En service | | |
| 4 | 4 | DML | DML-002 | 6 | 34.5 | 69 | 75 | 2024-08-05 | En service | | |
| 5 | 5 | DML | DML-003 | 6 | 36 | 72 | 90 | 2024-08-10 | In maintenance | | |
| 6 | 6 | DML | DML-004 | 6 | 33 | 68 | 85 | 2024-07-28 | En service | | |
| 7 | 7 | DML | DML-005 | 6 | 35.5 | 71 | 82 | 2024-08-03 | En service | | |
| 8 | 8 | DML | DML-006 | 6 | 34 | 70.5 | 78 | 2024-08-08 | En service | | |
| 9 | 9 | DML | DML-007 | 6 | 36.5 | 73 | 88 | 2024-08-12 | In maintenance | | |
| 10 | 10 | DML | DML-008 | 6 | 33.5 | 69.5 | 76 | 2024-07-30 | En service | | |
| 11 | 11 | Camion | CAM-001 | 10 | 40 | 70 | 75 | 2024-08-01 | En service | | |
| 12 | 12 | Camion | CAM-002 | 10 | 39.5 | 69.5 | 78 | 2024-08-02 | En service | | |
| 13 | 13 | Camion | CAM-003 | 10 | 40.5 | 71 | 74 | 2024-08-03 | En service | | |
| 14 | 14 | Camion | CAM-004 | 10 | 38 | 68 | 76 | 2024-08-04 | In maintenance | | |
| 15 | 15 | Camion | CAM-005 | 10 | 39 | 69 | 73 | 2024-08-05 | En service | | |
| 16 | 16 | Camion | CAM-006 | 10 | 40 | 70.5 | 72 | 2024-08-06 | En service | | |
| 17 | 17 | Camion | CAM-007 | 10 | 38.5 | 68.5 | 77 | 2024-08-07 | En service | | |
| 18 | 18 | Camion | CAM-008 | 10 | 39.5 | 69.5 | 74.5 | 2024-08-08 | En service | | |
| 19 | 19 | Camion | CAM-009 | 10 | 40 | 70 | 75.5 | 2024-08-09 | In maintenance | | |
| 20 | 20 | Camion | CAM-010 | 10 | 39 | 69 | 76.5 | 2024-08-10 | En service | | |
| 21 | 21 | Camion | CAM-011 | 10 | 38 | 68 | 74 | 2024-08-11 | En service | | |

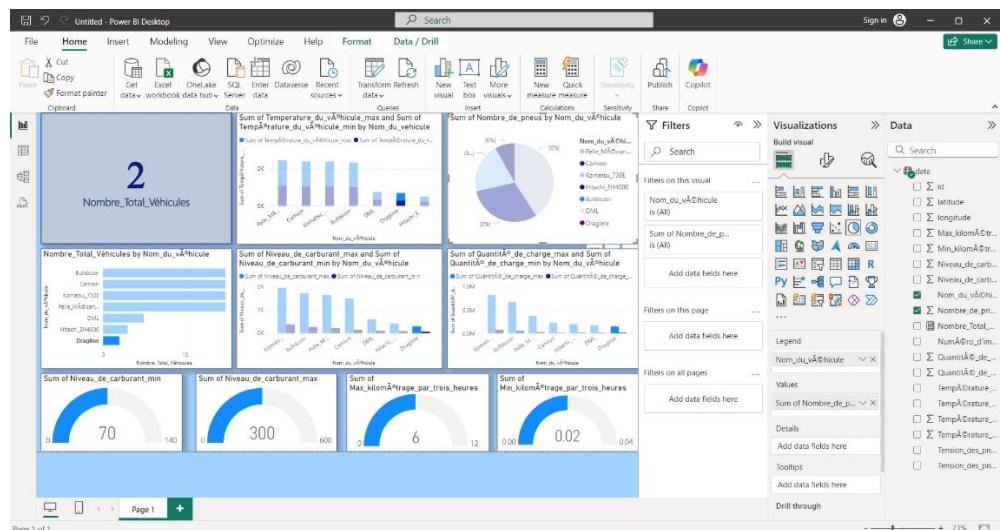
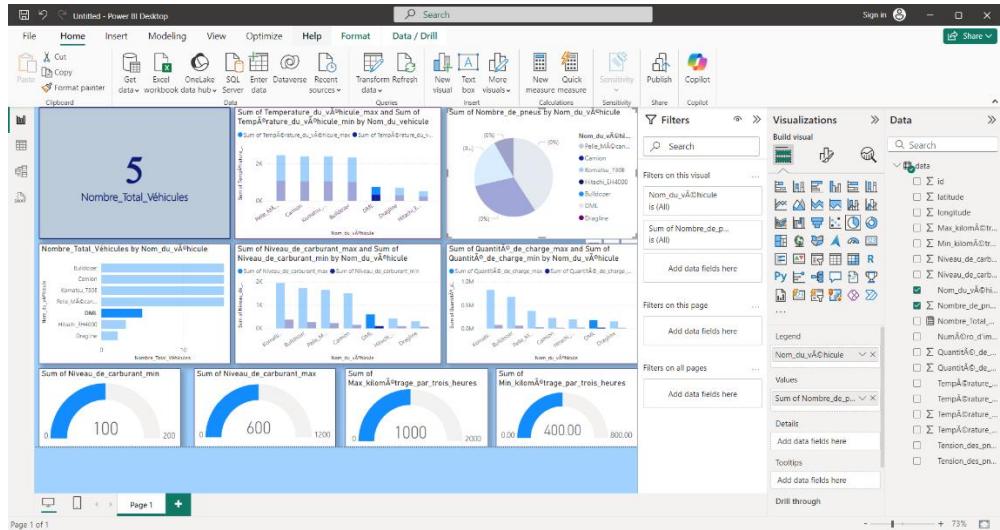
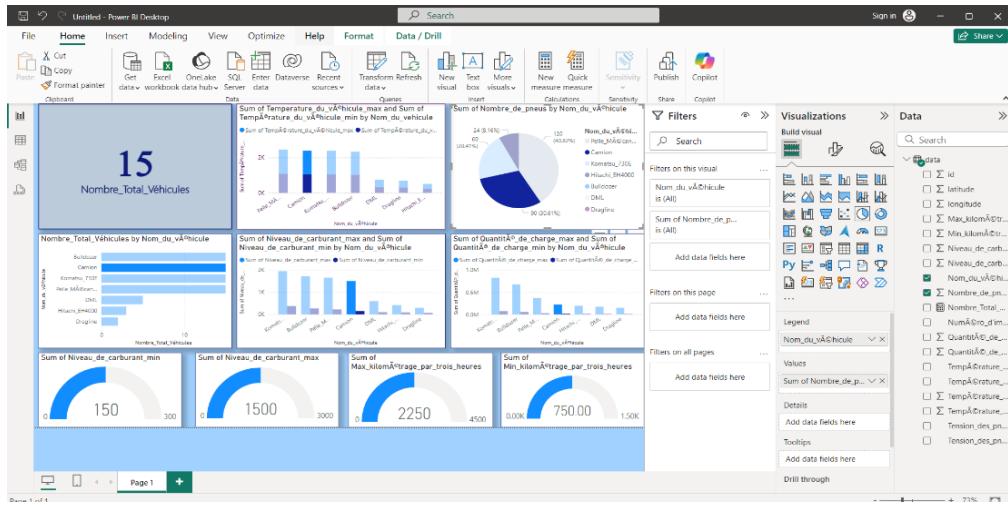
On l'enregistre sous forme Excel :

Mon dataset contient les colonnes suivantes : "Tension_des_pneus_min", "Tension_des_pneus_max", "Température_des_pneus_min", "Température_des_pneus_max", "Température_du_véhicule_min", "Température_du_véhicule_max", "Niveau_de_carburant_min", "Niveau_de_carburant_max", "Quantité_de_charge_min", "Quantité_de_charge_max", ainsi que "Min_kilométrage_par_trois_heures" et "Max_kilométrage_par_trois_heures". Ces variables permettent de surveiller et d'analyser les différents paramètres critiques des véhicules, notamment les tensions des pneus, les températures, les niveaux de carburant, les quantités de charge et les kilométrages, afin de détecter d'éventuelles anomalies ou de prévoir des pannes.

2. Dashboard :

Nous allons créer un tableau de bord interactif avec Power BI qui s'adapte dynamiquement en fonction du type de véhicule. Ce dashboard permettra de visualiser les données spécifiques à chaque catégorie de véhicule, telles que les tensions des pneus, les températures, les niveaux de carburant, et les quantités de charge, en fonction des seuils définis pour chaque type. Grâce aux fonctionnalités de filtrage et de personnalisation de Power BI, les utilisateurs pourront obtenir des insights pertinents et précis pour chaque véhicule, facilitant ainsi la prise de décision et l'optimisation des opérations en temps réel.





3. Machine learning :

3.1 AUTO-ENCODER pour la détection des anomalies :

(Chaque partie est expliquée dans les commentaires)

```
1 import pandas as pd
2
3 # Charger le fichier Excel
4 file_path = 'C:\\\\Users\\\\Hiba\\\\Desktop\\\\stage OCP\\\\dataa.xlsx' # Remplacez par le chemin correct de votre fichier
5 data = pd.read_excel(file_path)
6
7 # Afficher les premières lignes pour vérifier les données
8 print(data.head())
9
10 import numpy as np
11 # Remplacer les valeurs NULL dans les colonnes numériques par la moyenne de chaque colonne
12 # Sélectionner uniquement les colonnes numériques
13 numeric_columns = data.select_dtypes(include=[np.number]).columns
14
15 # Remplir les valeurs manquantes dans les colonnes numériques avec la moyenne
16 data[numeric_columns].fillna(data[numeric_columns].mean(), inplace=True)
17
18
19 # Pour les colonnes non numériques, remplacer les valeurs NULL par une valeur par défaut ou la plus fréquente
20 data['Nom du vÃ©hicule'].fillna('Inconnu', inplace=True)
21 data['NumÃ©ro_d\\'immatriculation'].fillna('Inconnu', inplace=True)
22
23 from sklearn.preprocessing import LabelEncoder
24
25 # Convertir les colonnes en chaînes de caractères pour un encodage cohérent
26 data['Nom du vÃ©hicule'] = data['Nom du vÃ©hicule'].astype(str)
27 data['NumÃ©ro_d\\'immatriculation'] = data['NumÃ©ro_d\\'immatriculation'].astype(str)
28
29 # Encodage des colonnes non numériques avec LabelEncoder
30 label_encoder = LabelEncoder()
31 data['Nom du vÃ©hicule'] = label_encoder.fit_transform(data['Nom du vÃ©hicule'])
32 data['NumÃ©ro_d\\'immatriculation'] = label_encoder.fit_transform(data['NumÃ©ro_d\\'immatriculation'])
33
34 from sklearn.preprocessing import MinMaxScaler
35
36 # Liste des colonnes numériques
37 numeric_columns = [
38     "Tension_des_pneus_min", "Tension_des_pneus_max",
39     "TempÃ©rature_des_pneus_min", "TempÃ©rature_des_pneus_max",
40     "TempÃ©rature_du_vÃ©hicule_min", "TempÃ©rature_du_vÃ©hicule_max",
41     "Niveau_de_carburant_min", "Niveau_de_carburant_max",
42     "QuantitÃ©_de_charge_min", "QuantitÃ©_de_charge_max",
43     "Min_kilomÃ©trage_par_trois_heures", "Max_kilomÃ©trage_par_trois_heures"
44 ]
45
46
47 # Normalisation
48 scaler = MinMaxScaler()
49 data[numeric_columns] = scaler.fit_transform(data[numeric_columns])
50
51
52 import numpy as np
53 import tensorflow as tf
54 from tensorflow.keras.models import Model
55 from tensorflow.keras.layers import Input, Dense
56 from sklearn.model_selection import train_test_split
57 from sklearn.preprocessing import StandardScaler
58
59 # Supposons que 'data' est votre DataFrame avec les colonnes numériques sélectionnées
60 numeric_columns = data.select_dtypes(include=[np.number]).columns
61 X = data[numeric_columns]
62
63 # Normaliser les données
64 scaler = StandardScaler()
65 X_scaled = scaler.fit_transform(X)
66
67 # Ensuite, vous pouvez séparer les données en ensemble d'entraînement et de test
68 X_train, X_test = train_test_split(X_scaled, test_size=0.2, random_state=42)
```

```

70     # 2. Construction du modèle Autoencoder
71     input_dim = X_train.shape[1]
72     encoding_dim = 14 # Choisissez une taille d'encodage plus petite
73     input_layer = Input(shape=(input_dim,))
74     encoded = Dense(encoding_dim, activation='relu')(input_layer)
75     encoded = Dense(encoding_dim // 2, activation='relu')(encoded)
76     encoded = Dense(encoding_dim // 4, activation='relu')(encoded)
77
78     decoded = Dense(encoding_dim // 2, activation='relu')(encoded)
79     decoded = Dense(encoding_dim, activation='relu')(decoded)
80     decoded = Dense(input_dim, activation='linear')(decoded)
81
82     autoencoder = Model(inputs=input_layer, outputs=decoded)
83     autoencoder.compile(optimizer='adam', loss='mean_squared_error')
84
85     # 3. Entraînement du modèle sur les données d'entraînement
86     history = autoencoder.fit(X_train, X_train,
87                               epochs=50,
88                               batch_size=32,
89                               shuffle=True,
90                               validation_data=(X_test, X_test),
91                               verbose=1)
92
93     # 4. Prédiction sur les données de test
94     X_test_pred = autoencoder.predict(X_test)
95

```

Les données sont envoyées par les capteurs, mais pour simplifier on va les insérer manuellement :

```

98     # Exemple de nouvelle donnée reçue en temps réel
99     new_data = pd.DataFrame({
100         "Nom_du_véhicule": ["Komatsu_730E"], # Nom du véhicule
101         "Numéro_d'immatrication": ["MM1313"], # Numéro d'immatrication
102         "Tension_des_pneus": [700], # Exemple de tension des pneus actuelle
103         "Température_des_pneus": [80], # Exemple de température des pneus actuelle
104         "Température_du_véhicule": [85], # Exemple de température du véhicule actuelle
105         "Niveau_de_carburant": [50], # Exemple de niveau de carburant actuel
106         "Quantité_de_charge": [1000], # Exemple de quantité de charge actuelle
107         "kilométrage_par_trois_heures": [260] # Exemple de kilométrage actuel par trois heures
108     })
109
110    # Fonction pour détecter si les valeurs sont normales ou non
111    def detect_normality_single_values(row):
112        vehicle = data[(data['Nom_du_véhicule'] == row['Nom_du_véhicule']) &
113                        (data['Numéro_d'immatrication'] == row['Numéro_d'immatrication'])]
114
115        if vehicle.empty:
116            return False # Véhicule non trouvé dans le dataset de référence
117
118        normality = []
119
120        # Vérifier la tension des pneus
121        if not (vehicle['Tension_des_pneus_min'].values[0] <= row['Tension_des_pneus'] <= vehicle['Tension_des_pneus_max'].values[0]):
122            normality.append("Tension des pneus normale")
123
124        # Vérifier la température des pneus
125        if not (vehicle['Température_des_pneus_min'].values[0] <= row['Température_des_pneus'] <= vehicle['Température_des_pneus_max'].values[0]):
126            normality.append("Température des pneus normale")
127
128        # Vérifier la température du véhicule
129        if not (vehicle['Température_du_véhicule_min'].values[0] <= row['Température_du_véhicule'] <= vehicle['Température_du_véhicule_max'].values[0]):
130            normality.append("Température du véhicule normale")
131
132        # Vérifier le niveau de carburant
133        if not (vehicle['Niveau_de_carburant_min'].values[0] <= row['Niveau_de_carburant'] <= vehicle['Niveau_de_carburant_max'].values[0]):
134            normality.append("Niveau de carburant normal")
135
136        # Vérifier la quantité de charge
137        if not (vehicle['Quantité_de_charge_min'].values[0] <= row['Quantité_de_charge'] <= vehicle['Quantité_de_charge_max'].values[0]):
138            normality.append("Quantité de charge normale")
139
140        # Vérifier le kilométrage par trois heures
141        if not (vehicle['Min_kilométrage_par_trois_heures'].values[0] <= row['kilométrage_par_trois_heures'] <= vehicle['Max_kilométrage_par_trois_heures']):
142            normality.append("Kilométrage par trois heures normale")
143
144
145    return normality

```

```

147 # Appliquer la fonction à la nouvelle donnée
148 normality_detected = detect_normality_single_values(new_data.iloc[0])
149
150 if normality_detected:
151     print(f"Les données du véhicule {new_data['Nom du véhicule'][0]} (immatriculation {new_data['Numéro_d\'immatriculation'][0]}) sont normales.")
152
153 else:
154     print(f"Anomalies détectées pour le véhicule {new_data['Nom du véhicule'][0]} (immatriculation {new_data['Numéro_d\'immatriculation'][0]})")
155
156     print(f"\n{new_data['Numéro_d\'immatriculation'][0]}\nAnomalies détectées pour le véhicule Komatsu_730E\n(immatriculation MMM1313)

In [61]:

```

Donc il y en a des anomalies dans la Komatsu_730 MMM1313.

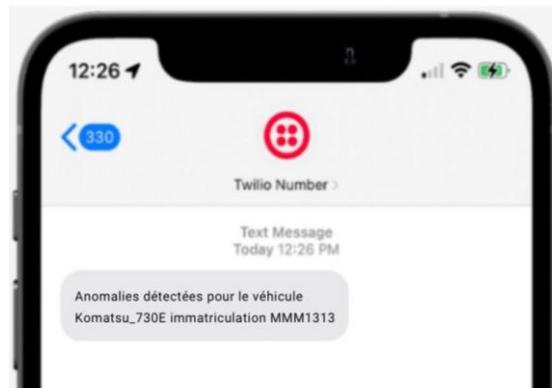
Code pour envoi de message twilio en cas d'une anomalie détectée :

```

164 import numpy as np
165 from twilio.rest import Client
166
167 # Vos identifiants Twilio
168 account_sid = 'ACc9b831ecce3f25cc1b4d6997bedacd1f'
169 auth_token = '599a8da8b7c44e3d8c918bc75ea536'
170 twilio_phone_number = '+14178052395'
171 destination_phone_number = '+212656289494'
172
173 # Créer un client Twilio
174 client = Client(account_sid, auth_token)
175
176 # Fonction pour envoyer une notification
177 def envoyer_notification(message_body, numero_destinataire):
178     message = client.messages.create(
179         body=message_body,
180         from_=twilio_phone_number,
181         to=numero_destinataire
182     )
183     print(f"Message envoyé avec SID : {message.sid}")
184
185 # Exemples de données
186 new_data = {
187     'Nom du véhicule': ['Komatsu_730E'],
188     'Numéro_d\'immatriculation': ['MMM1313']
189 }
190
191 # Exemple de données d'erreurs de reconstruction
192 # (Remplacez ceci par votre logique de détection d'erreurs)
193 X_test_pred = np.array([[1, 1, 1], [0, 0, 0], [1, 1, 1]]) # Prédictions du modèle
194 X_test = np.array([[1, 1, 1], [0, 0, 0], [1, 0, 1]]) # Données réelles
195
196 # Calcul des erreurs de reconstruction
197 reconstruction_errors = np.mean(np.abs(X_test_pred - X_test), axis=1)
198
199 # Définir le seuil d'anomalie
200 seuil_anomalie = 0.5 # Ajustez le seuil selon votre cas
201
202 # Détection des erreurs
203 erreur_detectee = np.any(reconstruction_errors > seuil_anomalie)
204
205 # Envoi de notification si une erreur est détectée
206 if erreur_detectee:
207     message_body = f"Anomalies détectées pour le véhicule {new_data['Nom du véhicule'][0]} (immatriculation {new_data['Numéro_d\'immatriculation'][0]})."
208     print(message_body)
209     envoyer_notification(message_body, destination_phone_number)
210
211

```

Message twilio :



3.2 LSTM pour la prédition des pannes :

```

1 import numpy as np
2 import pandas as pd
3 from sklearn.preprocessing import MinMaxScaler
4 from tensorflow.keras.models import Sequential
5 from tensorflow.keras.layers import LSTM, Dense
6
7 # Charger le dataset Excel
8 chemin_dataset = 'C:\\Users\\Hiba\\Desktop\\stage OCP\\dataa.xlsx'
9 df = pd.read_excel(chemin_dataset, engine='openpyxl')
10
11 # Correction des problèmes d'encodage si nécessaire
12 df.columns = df.columns.str.normalize('NFKD').str.encode('ascii', errors='ignore').str.decode('utf-8')
13
14 # Remplacer les valeurs NULL par NaN et gérer les valeurs manquantes
15 df.replace('NULL', np.nan, inplace=True)
16 df.fillna(df.select_dtypes(include=[np.number]).mean(), inplace=True)
17
18 # Assurez-vous que les colonnes à normaliser existent dans le DataFrame
19 cols_to_normalize = [
20     'Tension_des_pneus_min', 'Tension_des_pneus_max',
21     'Température_des_pneus_min', 'Température_des_pneus_max',
22     'Température_du_véhicule_min', 'Température_du_véhicule_max',
23     'Niveau_de_carburant_min', 'Niveau_de_carburant_max',
24     'Quantité_de_charge_min', 'Quantité_de_charge_max',
25     'Min_kilométrage_par_trois_heures', 'Max_kilométrage_par_trois_heures'
26 ]
27
28 # Assurez-vous que toutes les colonnes à normaliser existent dans le DataFrame
29 cols_to_normalize = [col for col in cols_to_normalize if col in df.columns]
30
31 # Normalisation des données
32 scaler = MinMaxScaler()
33 df[cols_to_normalize] = scaler.fit_transform(df[cols_to_normalize])
34
35 # Création du modèle LSTM avec activation sigmoid pour la sortie
36 model = Sequential()
37 model.add(LSTM(50, activation='relu', input_shape=(7, len(cols_to_normalize))))
38 model.add(Dense(1, activation='sigmoid'))
39
40 # Compilation du modèle
41 model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
42
43 # Préparation des données pour l'entraînement
44 X_train = []
45 y_train = []
46
47 # On suppose que chaque véhicule a plusieurs enregistrements (chaque 7 jours de données = 1 séquence)
48 for i in range(len(df) - 7):
49     X_train.append(df[cols_to_normalize].values[i:i+7])
50     # Définir les labels pour y_train (0 pour pas de panne, 1 pour panne). Ici c'est un exemple fictif.
51     y_train.append(0) # Remplacez par vos labels réels
52
53 X_train = np.array(X_train)
54 y_train = np.array(y_train)
55
56 # Entrainer le modèle
57 model.fit(X_train, y_train, epochs=100, verbose=0)
58
59 # Données fictives pour le véhicule GHI9012 sur 7 jours
60 data_fictive = {
61     'Nom du véhicule': ['Camion'] * 7,
62     'Numéro_d\'immatriculation': ['DEF5678'] * 7,
63     'Tension_des_pneus_min': [34, 34, 34, 32, 31, 30, 28],
64     'Tension_des_pneus_max': [35, 34, 34, 33, 32, 31, 29],
65     'Température_des_pneus_min': [78, 80, 81, 79, 82, 80, 83],
66     'Température_des_pneus_max': [80, 85, 87, 84, 86, 83, 87],
67     'Température_du_véhicule_min': [80, 86, 84, 82, 84, 83, 87],
68     'Température_du_véhicule_max': [85, 90, 94, 92, 90, 89, 92],
69     'Niveau_de_carburant_min': [20, 22, 21, 9, 21, 20, 19],
70     'Niveau_de_carburant_max': [70, 42, 41, 40, 70, 40, 39],
71     'Quantité_de_charge_min': [1400, 13200, 14230, 14530, 13540, 13525, 14780],
72     'Quantité_de_charge_max': [14790, 14990, 15700, 15600, 15000, 13000, 12000],
73     'Min_kilométrage_par_trois_heures': [50, 55, 53, 51, 52, 54, 50],
74     'Max_kilométrage_par_trois_heures': [150, 155, 153, 151, 152, 154, 150]
75 }
76
77 df_new_vehicle = pd.DataFrame(data_fictive)
78
79 # Normalisation des données du nouveau véhicule
80 df_new_vehicle[cols_to_normalize] = scaler.transform(df_new_vehicle[cols_to_normalize])
81
82 # Préparation des données pour la prédition
83 X_new_vehicle = df_new_vehicle[cols_to_normalize].values.reshape((1, 7, len(cols_to_normalize)))
84
85 # Faire une prédition
86 prediction = model.predict(X_new_vehicle)
87 print("Probabilité de panne pour le véhicule GHI9012 : ", prediction[0][0])
88
89

```

OUTPUT :

```
GHI9012 : ", prediction[0][0])
1/1 ━━━━━━━━ 0s 154ms/step
Probabilité de panne pour le véhicule GHI9012 : 0.18978067
```

3.3 CNN pour Surveillance par Caméra :

Le dataset est organisé sous la forme d'un dossier principal, comprenant 9 sous-dossiers. Chaque sous-dossier contient plusieurs images d'une personne spécifique, représentant un total de 9 personnes. Ces personnes sont les seules autorisées, et les images serviront à les identifier dans le cadre du projet de reconnaissance d'accès sécurisé.

| | | |
|-------------------------|-------------------|-------------|
| pins_Adriana Lima | 8/25/2024 1:52 AM | File folder |
| pins_Alex Lawther | 8/25/2024 1:52 AM | File folder |
| pins_Tom Holland | 8/25/2024 1:52 AM | File folder |
| pins_Tuppence Middleton | 8/25/2024 1:52 AM | File folder |
| pins_Ursula Corbero | 8/25/2024 1:52 AM | File folder |
| pins_Wentworth Miller | 8/25/2024 1:52 AM | File folder |
| pins_Zac Efron | 8/25/2024 1:52 AM | File folder |
| pins_Zendaya | 8/25/2024 1:52 AM | File folder |
| pins_Zoe Saldana | 8/25/2024 1:52 AM | File folder |

Exemple de Zendaya :



```

1 import os
2 import cv2
3 import numpy as np
4 from keras.models import Sequential
5 from keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout
6 from keras.utils import to_categorical
7 from sklearn.model_selection import train_test_split
8
9 # Préparation des données
10 def load_data(data_dir):
11     images = []
12     labels = []
13     # On assigne 0 à toutes les images (autorisé)
14     for person_name in os.listdir(data_dir):
15         person_dir = os.path.join(data_dir, person_name)
16         if os.path.isdir(person_dir):
17             for image_name in os.listdir(person_dir):
18                 image_path = os.path.join(person_dir, image_name)
19                 image = cv2.imread(image_path)
20                 image = cv2.resize(image, (128, 128)) # Redimensionner les images
21                 images.append(image)
22                 labels.append(0) # Assigner 0 pour toutes les images (autorisé)
23
24     # Convertir en tableaux numpy et normaliser les images
25     images = np.array(images, dtype='float32') / 255.0
26     labels = np.array(labels, dtype='float32') # Utiliser les étiquettes en format scalaire
27
28     return train_test_split(images, labels, test_size=0.2, random_state=42)
29

```

```

30 # Charger les données
31 data_train, data_test, labels_train, labels_test = load_data("C:\\\\Users\\\\Hiba\\\\Desktop\\\\stage_OCP\\\\105_classes_pins_dataset")
32
33 # Création du module CNN
34 model = Sequential([
35     Conv2D(32, (3, 3), activation='relu', input_shape=(128, 128, 3)),
36     MaxPooling2D(pool_size=(2, 2)),
37
38     Conv2D(64, (3, 3), activation='relu'),
39     MaxPooling2D(pool_size=(2, 2)),
40
41     Conv2D(128, (3, 3), activation='relu'),
42     MaxPooling2D(pool_size=(2, 2)),
43
44     Flatten(),
45     Dense(128, activation='relu'),
46     Dropout(0.5),
47     Dense(1, activation='sigmoid') # 1 sortie pour une classification binaire
48 ])
49
50 model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
51
52 # Entrainement du modèle
53 model.fit(data_train, labels_train, validation_split=0.2, epochs=20, batch_size=32)
54
55 # Évaluation du modèle
56 test_loss, test_acc = model.evaluate(data_test, labels_test)
57 print(f'test accuracy: {test_acc:.2f}')

```

```

59 # Reconnaissance faciale en temps réel
60 def recognize_face(frame, model, threshold=0.6):
61     face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')
62     gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
63     faces = face_cascade.detectMultiScale(gray, 1.1, 4)
64
65     for (x, y, w, h) in faces:
66         roi = frame[y:y+h, x:x+w]
67         roi = cv2.resize(roi, (128, 128))
68         roi = np.expand_dims(roi, axis=0)
69         roi = roi / 255.0
70
71         prediction = model.predict(roi)
72         confidence = prediction[0][0] # La sortie du modèle est un seul nombre
73
74         if confidence > threshold:
75             label = 'Personne autorisée'
76         else:
77             label = 'Personne non autorisée'
78
79         cv2.rectangle(frame, (x, y), (x+w, y+h), (255, 0, 0), 2)
80         cv2.putText(frame, label, (x, y-10), cv2.FONT_HERSHEY_SIMPLEX, 0.9, (36,255,12), 2)
81
82     return frame

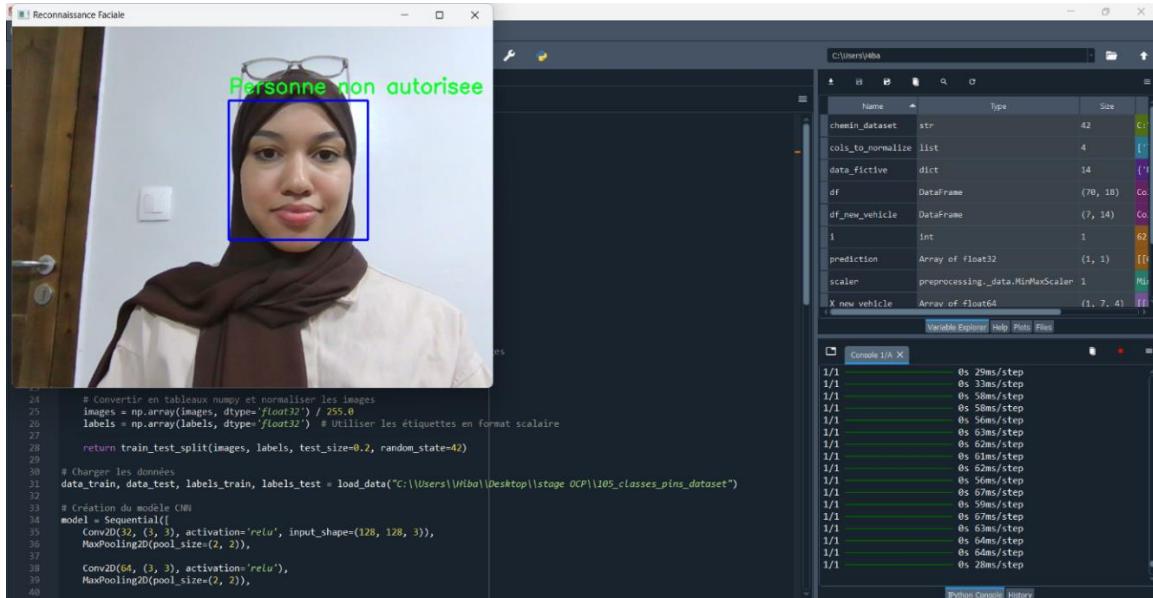
```

```

84 # Capture vidéo en temps réel
85 cap = cv2.VideoCapture(0)
86
87 while True:
88     ret, frame = cap.read()
89     if not ret:
90         break
91
92     frame = recognize_face(frame, model)
93     cv2.imshow('Reconnaissance Faciale', frame)
94
95     if cv2.waitKey(1) & 0xFF == ord('q'):
96         break
97
98 cap.release()
99 cv2.destroyAllWindows()
100

```

Puisque mon visage n'apparaît pas dans le dataset donc je suis une personne non autorisée



```
101 from twilio.rest import Client
102
103 # Configuration Twilio
104 account_sid = 'ACc9b831ecce3f25cc1b4d6997bedacd1f'
105 auth_token = '599a8da8b7ce44e3d8c918bcb75ea536'
106 client = Client(account_sid, auth_token)
107
108 # Numéros de téléphone
109 twilio_phone_number = '+14178052395'
110 recipient_phone_number = '+212656289494' # Remplacez par le numéro du destinataire
111
112 # Fonction pour envoyer un message via Twilio
113 def send_alert_message():
114     message = client.messages.create(
115         body="Alerte : Personne non autorisée détectée.",
116         from_=twilio_phone_number,
117         to=recipient_phone_number
118     )
119     print(f'Message envoyé : {message.sid}')
120
121 # Exemple d'utilisation : appeler send_alert_message() lorsqu'une personne non autorisée est détectée
```



VI . Conclusion :

Au terme de ce stage, il est évident que la gestion efficace d'un parc de machines et de camions dans un environnement industriel tel que celui de l'OCP présente des défis considérables. La complexité de surveiller en continu l'état de chaque véhicule et d'assurer la sécurité des employés nécessite une approche systématique et technologiquement avancée.

L'application de surveillance en temps réel proposée répond à ces enjeux en centralisant les informations critiques sur les véhicules et les machines, tout en offrant des fonctionnalités robustes telles que la détection des anomalies, la prédition des pannes, et la gestion des accès aux zones sensibles. Grâce à l'intégration de technologies comme React.js pour l'interface utilisateur, Django pour le backend, PostgreSQL pour la gestion des données, et des algorithmes de Machine Learning (LSTM, CNN, Autoencoder), cette solution permet de garantir à la fois l'efficacité opérationnelle et la sécurité des employés.

L'utilisation de Twilio pour les alertes critiques, couplée à des technologies de sécurité comme JWT pour l'authentification et le contrôle d'accès, renforce encore la fiabilité du système. Par ailleurs, l'infrastructure proposée sur Google Cloud Platform assure une scalabilité et une performance optimales.

En conclusion, cette solution apporte une réponse innovante et pragmatique aux problématiques rencontrées par l'OCP. Elle permet non seulement de réduire les coûts opérationnels, d'optimiser l'utilisation des ressources, mais aussi de renforcer la sécurité des employés et des installations. Les bénéfices potentiels de cette application en termes de prise de décision basée sur des données en temps réel font de cette proposition une avancée significative dans la gestion des opérations industrielles.

Ce projet de stage a été une opportunité enrichissante pour mettre en pratique des compétences techniques tout en contribuant à l'amélioration d'un processus industriel critique. La solution développée ici constitue un pas important vers une gestion plus intelligente et sécurisée des parcs de véhicules dans des environnements industriels complexes comme celui de l'OCP.

