

**UNIVERSITE DE
YAOUNDE I**

**ECOLE NATIONALE SUPERIEURE
POLYTECHNIQUE DE YAOUNDE**



Modélisation d'un Parcours opportuniste d'itinéraire

RAPPORT D'ADMINISTRATION RESEAU



Noms & prénoms	Matricule	
TCHAKONTE TCHIEGUE CEDRICK	20P533	20%
TCHATCHUENG DJOUM GILLES BORIS	20P221	20%
TCHIENDJO WETIE PAUL	22P138	20%
TCHINO CHATCHUENG LEA CALLIE	20P367	20%
TENE FOGANG ZACHARIE IGOR (chef)	20P540	20%

Sous la supervision De : Pr. Ing THOMAS DJOTIO

Année Académique 2023/2024

TABLE DES MATIERES

TABLES DES FIGURES.....	3
REMERCIEMENTS.....	4
INTRODUCTION GENERALE	5
PARTIE I : MODELISATION D'UN PARCOURS D'ITINERAIRE OPPORTUNISTE	6
I. PRÉSENTATION GÉNÉRALE DU PROBLÈME.....	7
A. Définition des termes-clés.....	7
B. Problème	8
II. ETAT DE L 'ART DU PROBLEME	9
A. Quelques notions mathématiques.....	9
A.1. Généralité sur les graphes.....	9
a. Notations.....	10
b. Représentation	11
i. Généralités sur les lois de probabilités	12
ii. Choix des outils et modèles utilisées.....	13
PARTIE II : IMPLEMENTATION DU PROJET PARCOURS OPPORTUNISTE D'ITINERAIRE D'UN TAXI.....	22
I. PRÉSENTATION DU PROJET	23
I.1. Contexte du projet	23
I.2. Problématique du projet.....	23
I.3. Objectif du projet	23
II. ANALYSE DU PROJET	23
II.1. Rapport d'enquête	24
II.2. Modèle d'apprentissage supervisée.....	26
II.4.1. Vue générale du Machine Learning	26
II.4.2. Modèle d'apprentissage supervisée	27
III. RÉALISATION DU PROJET.....	31
III.1. COLLECTE DES DONNEES SUR LE RESEAU ROUTIER DE YAOUNDE.....	31

III.1. Présentation de l'environnement de projet.....	35
III.1.1. Présentation des outils utilisés pour le frontend.....	35
a) Outils de Programmation	35
b) Les fonctionnalités implémentées et Objectifs :.....	36
III.1.2. Présentation de l'environnement du backend	41
III.2. Méthode comptage des véhicules pour l'analyse du trafic au	
niveau des intersections	42
III.2.1. COMPTAGE MANUEL.....	42
III.2.2. COMPTAGE AUTOMATIQUE	43
III.2.2.1. Capteurs non-programmables	43
III.2.2.1. Capteurs programmables.....	47
III.3. Caractéristique du jeu de données.....	51
III.4. Modèles et algorithmes.....	52
III. CHALLENGES ET PERSPECTIVES	56
IV.1. Challenges.....	56
IV.2. Perspectives	56
CONCLUSION.....	58
BIBLIOGRAPHIE	59

TABLES DES FIGURES

Figure 1: illustration de l'objectif	8
Figure 2 : Les sept ponts de Königsberg. Euler démontra qu'il est impossible de traverser une et une seule fois tous les ponts et introduisit la notion de graphe.	9
Figure 3: Un exemple de graphe	10
Figure 4 graphe représentant un réseau routier(fictif) (ici 288 nœuds et 674 arcs).....	14
Figure 5 : Architecture d'un modèle d'apprentissage supervisé [9]	28
Figure 6 : Diagramme circulaire de représentation des taux de véhicules utilisés par jour	32
Figure 7 : Cartographie de la ville de Yaoundé	33
Figure 8 : Cartographie de la ville de Yaoundé	34
Figure 9 : Framework SAAS.....	35
Figure 10 : Django Framework.....	35
Figure 11 : page de sélection des rôles.....	37
Figure 12 : page d'authentification.....	37
Figure 13 : page d'alerte densité de trafic en Baisse.....	38
Figure 14 : contrôle des capteurs	39
Figure 15 : page du calcul d'itinéraires optimales	40
Figure 16 : page du système d'analyse des capteurs.....	41
Figure 17 : Fonctionnement du capteur OVS-01 GT sensor	44
Figure 18 : Karrus.....	45
Figure 19 : FLIR TrafiCam.....	46
Figure 20 : FLIR TrafiOne 156.....	46
Figure 21 : Piezoelectric Traffic Sensor	47
Figure 22 : Capteur de proximité à ultrasons	48
Figure 23 : Capteur de proximité photoélectrique	49
Figure 24: Capteur de proximité rectangulaire	50
Figure 25 : classification report du modèle SVM et KNN.....	53
Figure 26 : classification report du modèle logistic regression et gradient Boosting classifier	53
Figure 27 : classifictaion report du modele Decission Tree Classifier et Random Forest Classifier	54

REMERCIEMENTS

Voilà que le temps est arrivé de vous présenter cette étude sur la *modélisation du parcours opportuniste d'itinéraire* laquelle nous a donnée l'opportunité de travailler avec beaucoup de questionnement, de remise en question et laisser libre à notre observation du monde extérieur et de la psychologie humaine. Nous tenons à vous remercier sincèrement professeur Thomas Djotio.

Nous remercions également tous les chauffeurs taxis de la ville de Yaoundé qui ont bien voulu partager leurs expériences et apporter leurs contributions afin que nous puissions mieux appréhender la chose.

Nous tenons également à remercier nos camarades de la 4GI (en respectant leur souhait de rester discret) pour leurs idées, leurs appréciations et leurs recommandations.

Nous avons pris un énorme plaisir en faisant ce travail et nous l'avons réalisé avec beaucoup d'amour et de passion, pas seulement pour obtenir une bonne note, mais parce que nous avons cru à ce projet. Pendant toute sa durée, de multiples aléas difficiles à gérer sont advenus, mais nous avons été si bien entourées qu'aucun de ces obstacles n'a pu mettre en question la réalisation du projet. Aujourd'hui, si on nous demandait de tout recommencer, sans aucune hésitation, la réponse serait positive. Nous espérons que le résultat sera au niveau des espérances et des efforts de tous les gens qui ont contribué à cette entreprise, et des attentes du professeur ainsi qu'aux vôtres. Nous vous en laissons juge...

INTRODUCTION GENERALE

L'utilisation des réseaux de transport est conditionnée par l'efficacité et la simplicité de ceux-ci. En réponse à une mobilité exacerbée, volontaire ou subie, l'offre de transport se développe et motive tout à la fois, en un cycle continu, des déplacements encore plus exigeants. De manière complémentaire, la mobilité est bousculée par le développement de plus en plus de chemin reliant deux positions. Si la technologie permet de suivre cette évolution dans les services d'information aux voyageurs, il reste toujours à satisfaire des attentes déterminées par des usages en constante évolution. C'est de ce point de vue que l'obtention de chemins multiples pour relier une origine à une destination est un facteur qui n'est plus à négliger, surtout dans des réseaux de transport denses et comportant de nombreuses lignes de circulation. Une liberté dans le choix laissé à l'utilisateur du réseau réduit les sentiments d'exclusion, d'incompréhension ou d'anxiété qui peuvent survenir face à une application logicielle ou sur internet et qui effectuent des choix arbitraires de façon autoritaire. De plus, cela permet de vérifier la qualité de l'offre de transport, car plus il existe de moyens différents pour effectuer un trajet dans un intervalle de temps donné, plus il est difficile de savoir lequel serait le plus judicieux. Notre devoir s'intéresse au calcul de telles alternatives par le biais de l'énumération par coût croissant des chemins entre deux points en tenant compte de leur attractivité, puis par le filtrage de ceux-ci suivant des critères, dans le but d'optimiser les gains tout en proposant le chemin le plus optimal. Notre devoir est donc structuré de la façon suivante : dans un premier temps nous nous attarderons sur la présentation générale du problème (contexte et situations), puis nous rappellerons quelques notions de mathématiques nécessaires pour la formalisation, ensuite nous expliquerons le choix des outils/modèles mathématiques utilisés et enfin nous proposerons une formalisation du problème, quelques méthodes de résolution de celui-ci et une évaluation de celles-ci.

PARTIE I : MODELISATION D'UN PARCOURS D'ITINERAIRE OPPORTUNISTE

I. PRÉSENTATION GÉNÉRALE DU PROBLÈME

Afin de mieux comprendre notre thème, il est nécessaire de définir quelques termes-clés :

A. Définition des termes-clés

- **Parcours** : C'est un chemin qui relie deux points ou plus.
- **Itinéraire** : c'est une succession de parcours.
- **Parcours opportuniste d'itinéraire** : C'est un itinéraire qui peut être modifié en fonction des circonstances. C'est une situation de recherche opérationnelle.
- **Modélisation** : est la création d'un modèle, c'est-à-dire d'une représentation simplifiée d'un objet ou d'un système.
- **Le plus court chemin multi objectif** est un itinéraire qui minimise un ensemble d'objectifs, tels que la distance, le temps, le coût et maximiser le gain.
- **Itinéraire de substitution** : est un itinéraire alternatif utilisé lorsqu'un itinéraire original est inaccessible ou indisponible.
- **Itinéraire de déviation** : est un itinéraire utilisé pour éviter un obstacle ou une zone dangereuse.
- **Un problème de recherche opérationnelle (RO)** : est un problème de décision qui peut être formulé comme un problème d'optimisation. Un problème d'optimisation consiste à trouver la solution qui maximise ou minimise une fonction objectif donnée, tout en respectant un ensemble de contraintes.

Aux vues des définitions ci-dessus, nous pouvons donc dire que l'expression « *modélisation du parcours opportuniste d'itinéraire* » fait référence à la description de la situation (la création d'un modèle) qui permet de déterminer le meilleur itinéraire pour se rendre d'un point A à un point B en utilisant des opportunités qui se présentent le long du chemin. Le cadre ici est celui du trafic routier où le particulier(taxi) cherche à maximiser son gain sur l'itinéraire choisi.

Le problème de parcours d'itinéraire opportuniste est donc un problème de recherche opérationnelle qui consiste à trouver le meilleur itinéraire pour un véhicule en tenant compte des opportunités d'arrêts en cours de route. Ce type de problème est souvent rencontré dans les domaines de la logistique et du transport, où il est important de trouver des itinéraires efficaces qui permettent de maximiser le nombre de clients desservis. C'est également un problème d'optimisation multi objectif où nos objectifs sont le gain et le temps.

B. Problème

Le problème du parcours opportuniste d'itinéraire est assez complexe et difficile à formaliser. Il existe néanmoins de nombreux modèles pour le décrire mais étant trop complexe car nécessitant une certaine expertise des mathématiques appliquées, nous avons opté à concevoir notre propre modèle.

Pour concevoir le modèle (qui sera expliqué dans la partie II.B) de notre problème, il était nécessaire de répertorier les agents intervenants dans ce dernier. Ces agents sont : le client (potentiel passager), le taxi et les nœuds du réseau. Par la suite, nous nous sommes appesantis essentiellement sur les questions suivantes :

- 1- Comment représenter le réseau routier ? le taxi ? le client ?
- 2- Quels sont les stratégies les plus employées par les taxis ?
- 3- Comment déterminer les nœuds à forte attractivité ?
- 4- Comment savoir à un nœud les destinations les plus prisées ?
- 5- Comment évaluer le cout (ici, en dépense) et le gain du taxi au cours de son itinéraire ?
- 6- Comment tenir compte des fluctuations qui peuvent advenir sur le réseau ?
- 7- Etant à un nœud, quelles sont les destinations les plus susceptibles pour un taxi ?
- 8- A quelle condition un taxi doit-il accepter ou non de porter un client ?

L'analyse de l'image ci-dessous permet de comprendre pourquoi les questions ci-dessus.

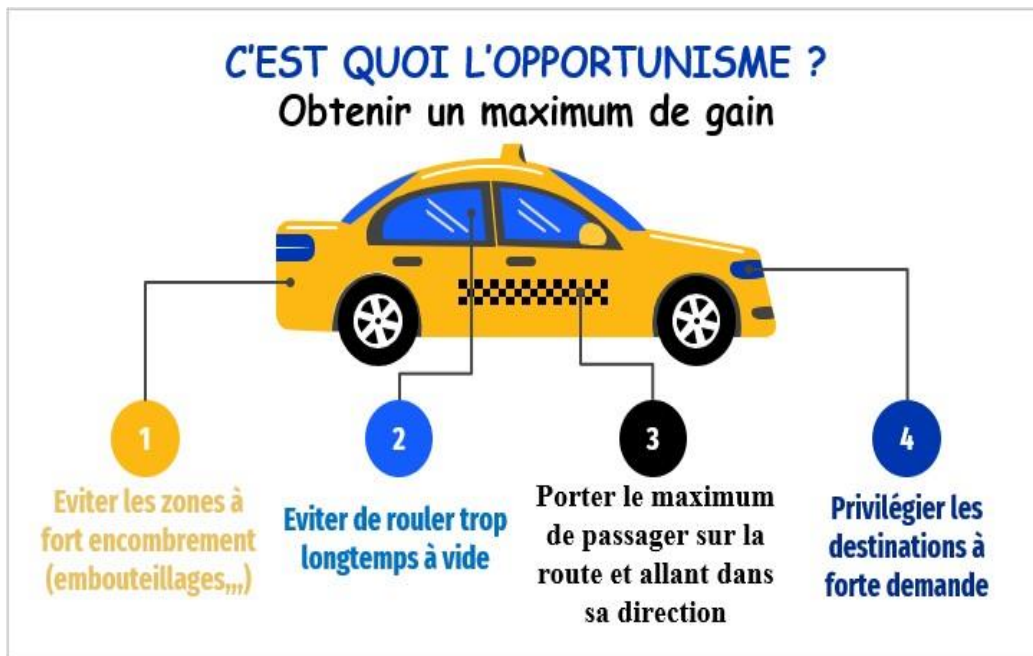


Figure 1: illustration de l'objectif

II. ETAT DE L 'ART DU PROBLEME

A. Quelques notions mathématiques

Cette partie a pour but de rappeler les bases requises sur les graphes et les lois de probabilité pour situer nos travaux. Cependant avant de les présenter, nous décrivons leurs bases communes. Dans un premier temps nous présentons les généralités autour des graphes puis quelques lois de probabilités.

A.1. Généralité sur les graphes

Dans un article publié en 1736, Leonhard Euler démontre qu'il est impossible d'emprunter une et une seule fois les sept ponts de Königsberg (aujourd'hui Kaliningrad). Pour cela il introduit une structure de données qui sera appelée plus tard *graphe*.

Un graphe est une structure de données très simple utilisée dans de nombreux domaines tels que les télécommunications, la planification, l'électronique, les transports ou encore la théorie de la complexité.

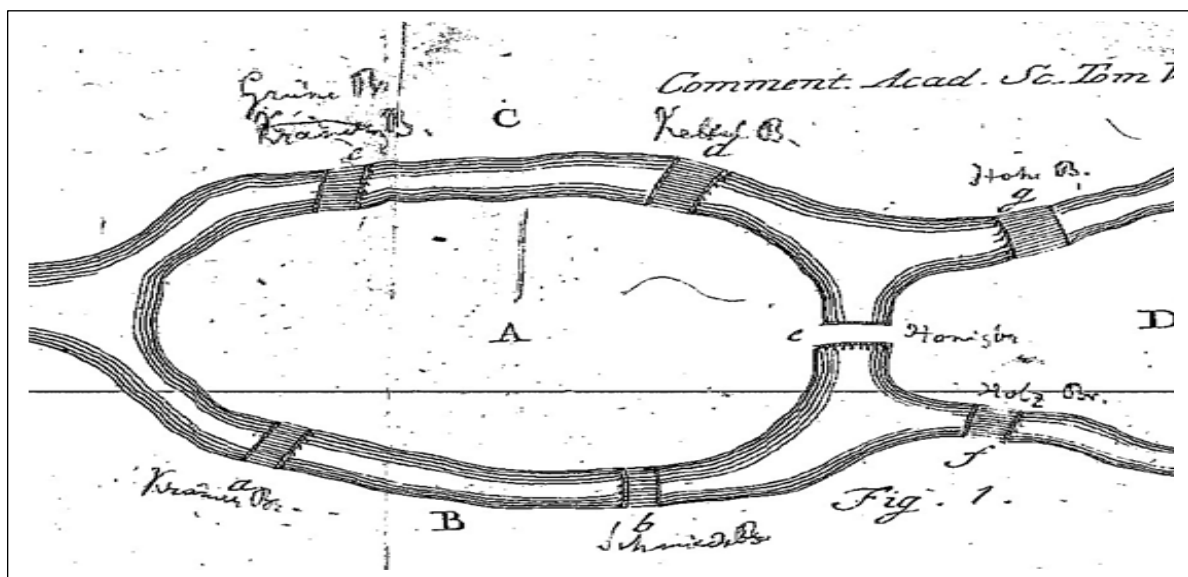


Figure 2 : Les sept ponts de Königsberg. Euler démontra qu'il est impossible de traverser une et une seule fois tous les ponts et introduisit la notion de graphe.

Google Scholar référence près de cinq millions d'articles comportant le mot « graphe » dans le titre. Face à l'étendue du domaine il ne serait donc pas possible de présenter l'ensemble

de la théorie des graphes. Nous nous focaliserons donc dans cet état de l'art sur le problème particulier du plus court chemin.

a. Notations

En simplifiant à l'extrême, un **graphe** définit l'existence d'une relation entre objets tels qu'une ligne entre deux stations de métro, une relation d'amitié dans un réseau social ou encore une rue entre deux carrefours.

D'un point de vue mathématiques, **un graphe** est un objet mathématique discret établissant des relations entre un ensemble d'éléments. Il permet d'abstraire des situations dans lesquelles des liens entre composants peuvent être définis. Dans notre cas, la relation nouée entre un ensemble de positions géographiques et la façon de se déplacer de l'une à l'autre modélise les réseaux de transport, qui peuvent donc être transcrits en graphes afin d'effectuer des calculs de cheminement dessus.

- **Nœuds et arcs**

Les objets sont appelés **nœuds** (nodes) ou **sommets** (vertices) et les relations **arcs** (edges) ou **arêtes** (links). Soit N l'ensemble des $n = |N|$ nœuds où $||$ désigne le cardinal d'un ensemble et A l'ensemble des $m = |A|$ arcs. Puisqu'un arc relie toujours exactement deux nœuds (mais deux nœuds ne sont pas nécessairement reliés puisque certains nœuds peuvent être **déconnectés** puisque reliés par aucune arête à d'autres nœuds), on a $A \subseteq N \times N$. **Formellement**, $G(N, A)$ définit donc un graphe.

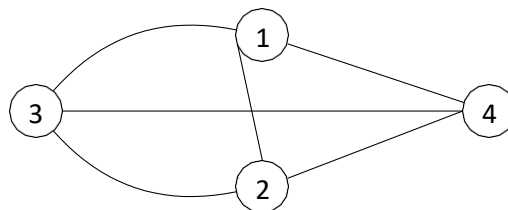


Figure 3: Un exemple de graphe

- **Être orienté ou pas**

Lorsque les relations sont symétriques, le graphe est dit non-orienté. Formellement, un graphe est non-orienté lorsque $\forall (u, v) \in A, (v, u) \in A$. Un réseau informatique sera généralement non-orienté puisque deux ordinateurs peuvent communiquer dans les deux sens. À l'opposé, un réseau routier est orienté pour modéliser les rues à sens unique. Il est habituel (mais pas systématique) d'utiliser les termes nœud et arc pour les graphes orientés et sommet et arête pour les graphes non orientés.

- **Poids sur les arcs**

Dans de nombreux problèmes, il est souhaitable de pouvoir qualifier une relation. Pour cela à chaque arc est associé un poids qui le décrit. Dans un réseau social il peut définir la nature de la relation (ami, famille, collègue) et dans un réseau routier la longueur d'une rue. Parfois le terme coût est utilisé. En associant à chaque arc $\forall a \in A$ un poids $ca \in C$, on obtient un graphe $G(N, A, C)$ qui est dit **valué**.

Même si dans les cas les plus courants les poids sont constants, il est possible d'utiliser des fonctions pour représenter les poids.

- **Successeurs et prédécesseurs**

La fonction Γ^+ définit pour chaque nœud les **successeurs**, soit $\Gamma^+(u) = \{v \mid (u, v) \in A\}$. Symétriquement on définit les **prédécesseurs** $\Gamma^-(u) = \{v \mid (v, u) \in A\}$. Lorsque le graphe est symétrique les prédécesseurs et successeurs d'un nœud sont les mêmes. Plus généralement, deux nœuds sont dits **adjacents** lorsqu'ils sont reliés par un arc.

- **Degré et densité**

Le degré d'un nœud est le nombre d'arcs adjacents à ce nœud et la densité d'un graphe est le rapport entre le nombre d'arcs **m** sur le nombre de nœuds **n**, soit $\frac{m}{n}$.

b. Représentation

Pour représenter un graphe informatiquement, généralement deux modélisations sont utilisées.

- **Matrice d'adjacence**

Une matrice M de dimension $n \times n$ représente un graphe de la manière suivante : $M_{uv} = 1$ s'il existe un arc du nœud u au nœud v et 0 sinon. Plutôt que 1, il est possible d'utiliser le

poids de l'arc pour marquer son existence. Si cette représentation permet de facilement tester l'existence d'un arc en $O(1)$, obtenir l'ensemble des successeurs d'un nœud est en $O(n)$. Enfin, la mémoire nécessaire pour stocker la matrice est en $O(n^2)$.

- **Liste d'adjacence**

Dans cette représentation, un vecteur de n éléments contient pour chaque nœud la liste de ses successeurs. Cette représentation nécessite donc moins de mémoire qu'une matrice d'adjacence et obtenir la liste des successeurs d'un nœud est en $O(1)$. Lorsque le degré des nœuds est indépendant de la taille du graphe — comme dans un réseau routier — alors la mémoire nécessaire est en $O(n)$. Cette représentation est donc préférée pour les graphes grands et peu denses.

i. Généralités sur les lois de probabilités

Soit $(\Omega, \mathcal{T}(\Omega))$ un espace probabilisable.

- **Variable aléatoire**

Une application $X: \Omega_1 \rightarrow \Omega_2$ est dite mesurable si $\forall A \in \mathcal{T}(\Omega_2), X^{-1}(A) \in \mathcal{T}(\Omega_1)$. On dit alors que X est une variable aléatoire à valeurs dans Ω_2 .

- **Loi de probabilité**

C'est une règle qui associe une probabilité à chaque issue possible d'une expérience aléatoire.

- **Loi de Poisson**

C'est une loi (de paramètre λ) utilisée pour modéliser le nombre d'événements rares qui se produisent dans un intervalle de temps donné ou dans un espace donné. Elle est souvent utilisée pour modéliser des phénomènes tels que les arrivées de clients dans un système de files d'attente. Son paramètre λ représente à la fois l'espérance et la variance de la distribution. Il indique le nombre moyen d'événements qui se produisent dans un intervalle de temps.

- **Loi exponentielle**

C'est une distribution (paramètre λ) continue qui modélise l'attente (intervalle de temps) entre des événements rares et indépendants qui se produisent de manière aléatoire. Elle est souvent utilisée pour modéliser des durées de vie, des temps d'attente ou des délais.

Dans cette loi, le paramètre λ est l'inverse de la moyenne ($1/\mu$) de la distribution, où μ est la durée moyenne entre les événements.

- **Loi log-normale**

La loi log-normale est une distribution continue dont le logarithme est distribué selon une loi normale, qui elle, également connue sous le nom de loi de Gauss ou de loi en cloche, est souvent utilisée pour modéliser des variables continues. Dans le contexte des chauffeurs de taxi, la loi normale peut être utilisée pour modéliser la distribution de la demande de clients à un arrêt.

La loi log-normale quant à elle peut être utilisée pour modéliser certaines variables liées à cette activité. Celle-ci est souvent associée avec un paramètre appelé **shift**, ce dernier représente la valeur à laquelle la distribution est décalée. IL indique en général la valeur à partir de laquelle les observations positives commencent.

ii. Choix des outils et modèles utilisées

(B.1) A la question 1 posée en I.B, nous avons décidé que nous représenterons les agents intervenant dans notre problème comme suit :

Réseau routier

Il convient de modéliser notre réseau par un graphe **valué** noté **G** constitué de :

- **Nœuds** : qui représentent les points d'arrêts, de ramassage du taxi, les ronds-points et les carrefours du monde physique. Nous noterons l'ensemble des nœuds par **S**.
- **Arcs orientés** (chaque n'est parcouru que dans un sens) : ceux-ci ici représentent les routes reliant deux nœuds adjacents. L'ensemble des arcs sera noté **A**.
- **Le descriptif des arcs** : c'est une grandeur binaire (0 ou 1) qui servira à qualifier l'état de l'arc (praticable ou pas) au cours d'un intervalle de temps. L'ensemble de ces valeurs sur les arcs sera noté par **C**.

Ainsi **G** = (**S**, **A**, **C**).

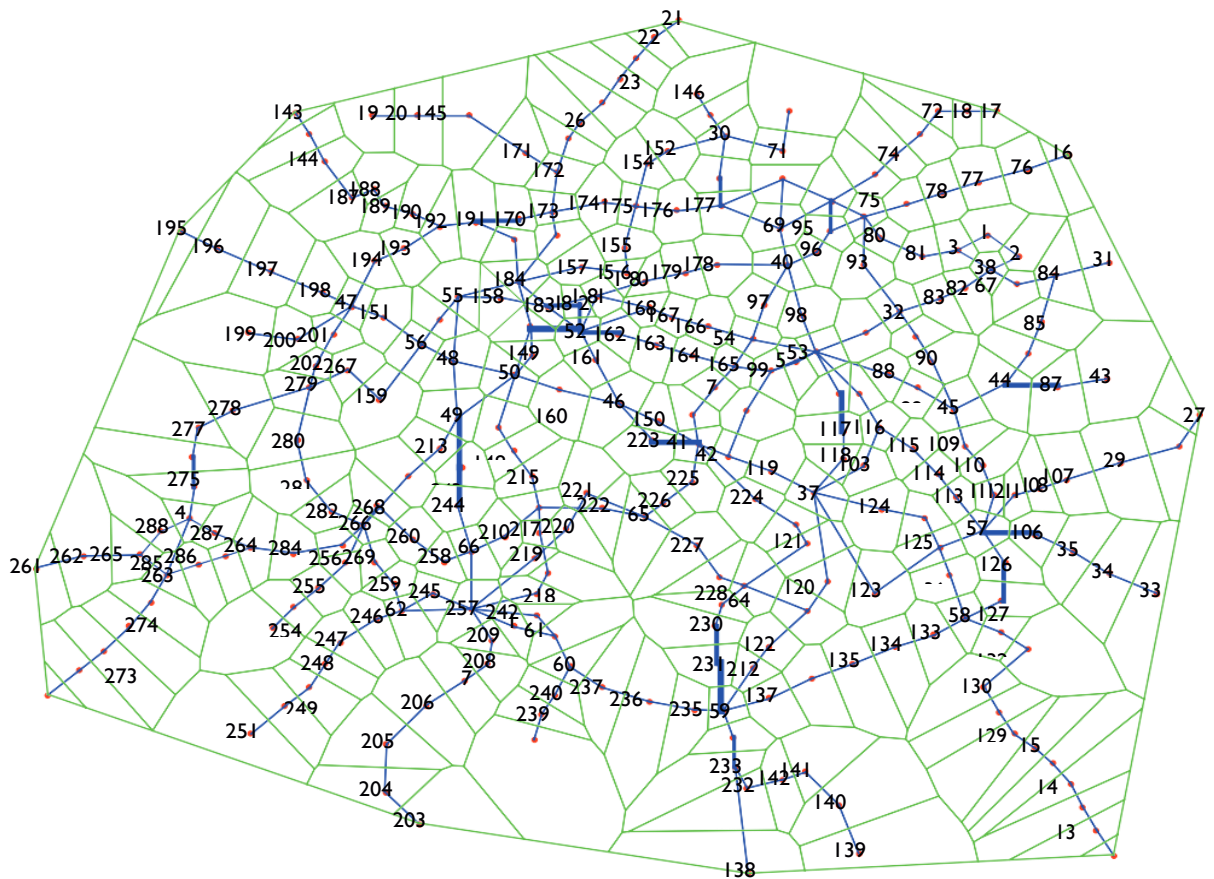


Figure 4 graphe représentant un réseau routier(fictif) (ici 288 nœuds et 674 arcs).

➤ Le Taxi

Nous avons décidé de le représenter par un vecteur T dont les composantes sont les suivantes :

- Sa position
- Le nombre de place libre qui est $4 - M$, où M est le nombre de passagers à bord.
- La liste L de destination à desservir
- Sa vitesse à chaque instant V_t
- sum_L représentant la somme cumulative que le chauffeur reçoit des clients sur son trajet.

➤ Le client

Pour ce dernier il convient de le représenter par :

- Sa position d'origine n_o

- Sa destination n_d
- Son tarif tar_o^d
- Le nombre de places qu'il va prendre.

(B.2) Pour ce qui est de la question 2 de la partie I.B, nous avons mené une étude sur le terrain au cours de laquelle nous avons interrogé de nombreux chauffeurs taxis, sur leur méthode de travail. A la suite de celle-ci, nous avons recensé deux principaux cas de figures (scénario / stratégie) :

- Celui qui ne travaille que dans une zone prédéfinie $Z_t \subset G$.
- Celui dont le trajet est fonction de la destination de son premier passager. Nous noterons que celui-ci n'a pas de zone de parcours prédéfinie mais il privilégie les destinations à forte demande.

Ces scénarios nous serviront par la suite lors de l'application de certains algorithmes.

(B.3) A la question de savoir comment déterminer l'attractivité d'un nœud, cela revient à connaître si un nœud est susceptible d'être plus fréquenté qu'un autre pendant un intervalle de temps de données. Il faut donc entre en mesure de déterminer le nombre d'apparition à un nœud pendant un intervalle de temps souhaitant quitter ce nœud. Ce processus comme expliqué en i (page 12) suit généralement une loi de poisson c'est-à-dire le temps qui sépare l'arrivée de deux clients successifs est une variable aléatoire distribuée selon une loi exponentielle de densité :

$$g(t) = \lambda e^{-\lambda t}$$

Le paramètre λ est le nombre moyen de clients arrivant par unité de temps (donc $1/\lambda$ est la moyenne de t , mais aussi de l'écart-type). Là encore, il y a bien sûr un paramètre λ par nœud du réseau. Nous noterons désormais λ le vecteur ligne de tous ces paramètres de nœud.

(B.4) Pour répondre à la question 3 de la partie I.B, il aurait fallu mener une étude statistique préalable à longue durée sur des échantillons dans laquelle, pour un nœud donné on aurait recueilli les différentes destinations des clients se trouvant à ce nœud et ainsi mesurer les tendances (destinations les plus prisées) par intervalle horaire. Par manque de données et les contraintes imposées par notre condition, nous avons admis que nous possédions ces données. Il devient donc aisé de répondre à la question souhaitée. Ainsi lorsqu'un client apparaît à un

nœud, sa destination est déterminée en tirant au hasard selon la probabilité de chaque destination pour une origine donnée. La matrice dont chaque ligne correspondant à une origine, représente la loi de probabilité de chaque destination (en colonne) est appelée **Matrice Origine-Destination** (matrice O-D) et elle sera notée M . Puis que chaque ligne représente une loi de probabilité discrète, la somme de ces éléments sur chaque ligne vaut 1. Si on désigne par $\mathbf{1}$ le vecteur colonne de dimension égal au nombre de nœud et composé uniquement de 1.

$$M\mathbf{1} = \mathbf{1} \quad (\text{II.1})$$

Avant de passer à la prochaine question, il est intéressant de faire le bilan (vectoriellement) par unité de temps et pour chaque nœud i , entre le nombre de clients apparaissant à ce nœud, à savoir λ_i et ceux qui apparaissent aux autres nœuds du réseau et qui désirent arriver à savoir $\sum_j \lambda_j M_{ji} = (\lambda M)_i$. Le bilan de tous les nœuds s'écrit donc vectoriellement :

$$\mathbf{b} = \lambda M - \lambda \quad \text{ou encore } \mathbf{b} = \lambda (\mathbf{M} - \mathbf{I}) \quad (\text{II.2})$$

Avec \mathbf{I} matrice identité de même taille que M .

Les composantes positives de ce vecteur indiquent les nœuds « récepteur » c'est-à-dire les nœuds les plus réceptifs en demande et les négatives indiquent les nœuds « émetteur ».

Le calcul de λ nécessite une étude statistique sur un échantillon de données.

Par exemple, pour une ville comme Yaoundé, le centre-ville sera considéré comme une zone réceptrice et les périphéries des zones émettrices.

(B.5) A propos de la question 4 de la partie I.B, qui était de savoir comment évaluer le cout et le gain du taxi au cours de son itinéraire, il est important de noter que le cout ici représente la consommation du taxi en carburant durant son itinéraire. Comme la dépense en litre en carburant est fonction du véhicule et de la nature du carburant (gasoil, super etc.), il serait très complexe d'explicitier une relation générique de cette dépense en fonction de la distance. Il est plus aisé de standardiser cette dépense. En notant $conso_o^d$ la consommation en carburant du taxi sur le trajet (ij) , on peut alors déduire :

$$sum_i^j = P_{std} \times conso_i^j + \xi_i^j \quad (\text{II.3})$$

$$\text{Où } \left\{ \begin{array}{l} \text{conso}_i^j = k \times \text{dist}(o, d), k \text{ dependant du type de vehicule} \\ \xi_i^j: \text{il represente la somme gagnée par le taxi sur la trajet } (ij) \\ P_{std}: \text{c'est le prix standart d'un litre de carburant} \end{array} \right.$$

(B.6) On s'intéresse maintenant à la caractérisation de la densité/flux du trafic routier et des circonstances pouvant advenir sur la route ; Pour se faire, on définit les notions suivantes :

- **Temps moyen sur un arc** : comme son nom l'indique, il représente le temps en moyenne mis par un véhicule pour traverser l'arc reliant deux nœuds, cette moyenne étant prise sur une durée conséquente pour minimiser au maximum les aléas les plus imprévisibles de la route. Ce temps sera la moyenne des temps de parcours aléatoires sur les arcs. La modélisation habituelle de temps de parcours aléatoire, est faite à l'aide de la loi log-normale avec **shift**. Le shift est une constance s rajoutée à la variable aléatoire distribuée selon la loi log-normale, afin que son support soit $[s ; +\infty[$ au lieu de $[0 ; +\infty[$ (donc s est une durée minimale de temps de parcours. La loi log-normale est une loi de densité :

$$f(t) = \frac{e^{\frac{-(\log t - \mu)^2}{2\sigma^2}}}{t\sigma\sqrt{2\pi}} \quad (\text{II.4})$$

Où μ et σ sont respectivement la moyenne et l'écart-type du logarithme de la variable aléatoire. La moyenne et la variance de t données :

$$(\text{II.5}) \quad \text{moyenne} = e^{\mu + \sigma^2} \quad \text{variance} = (e^{\sigma^2} - 1)e^{2\mu + \sigma^2}$$

Il faut donc se donner un triplet de paramètre (μ, σ, s) par un arc du réseau.

Lors des calculs de plus courts chemins pour construire les itinéraires prévisionnels des taxis, on ne se sert que des temps moyens de parcours des arcs (voir (II.5) en rajoutant s à la moyenne). En fait, nous supposons avoir précalculé tous les plus courts chemins pour toutes les paires de nœud du graphe a été stockée une fois pour toutes. On la note :

$$\mathbf{M}_{\text{tempsMoyen}} = (\mathbf{M}_{ij})_{1 \leq i, j \leq |G|} \quad \text{Où} \quad \mathbf{M}_{ij} = e^{\mu + \sigma^2}$$

Le calcul de ces plus courts chemins peut se faire en utilisant les algorithmes de Dijkstra version modifiée, de JOHNSON (combinaison de l'algorithme de Bellman Ford et de Dijkstra).

De même, nous avons stocké la **matrice des stratégies**, pour toute paire ordonnées (i, j) de nœuds (origine et destination d'un trajet), l'élément (i, j) de cette matrice donne le nœud k vers lequel il faut se diriger pour rejoindre j à partir de i par le plus court chemin c'est-à-dire :

$$\mathbf{M}_{\text{strategie}} = (\mathbf{M}_{ij})_{1 \leq i, j \leq |G|}$$

avec $M_{ij} = \{k \in \text{suivant}(i)/(i,k,\dots,j) \equiv \text{plus-court-chemin}(i,j)\}$

- **Etiquette d'état sur les arcs** : voir la représentation du réseau routier. Ce paramètre peut être vu comme une sorte de variable de décision qui permet d'autoriser la prise en compte d'un arc. Si cette étiquette prend la valeur a un instant t, il sera impératif de refaire tous les calculs ci-dessus pour la réactualisation de nos matrices. Ce qui montre de plus, le niveau de difficultés de la modélisation. Afin de déterminer si une route doit être prise en compte, on définit les notations suivantes :

Une vitesse seuil $V_{s_{ij}}$ par arcs définie par :

$$V_{s_{ij}} = \min\{v_{lim}^T, \forall \text{ vehicule } T \text{ circulant sur l'arc}(i,j)\}$$

Où v_{lim}^T , est la limite des vitesses acceptables pour T roulant sur l'arc (i,j).

Par définition, si la vitesse a l'instant t d'un taxi se trouve devenir inférieure a cette vitesse seuil alors nous considérons que cet arc est impraticable.

(B.7) Etant à un nœud, quelles sont les destinations les plus susceptibles pour un taxi ? La réponse à cette question n'est pas très facile. Cette question est surtout plus importante pour un taxi vide se trouvant à un nœud mais ne pouvant porter aucun client (par exemple, celui qui ne travaille que dans une zone prédéfinie donc tous les clients sortent de cette zone). En effet, la question de savoir vers quel nœud, nous allons bien pouvoir le rediriger afin de maximiser son gain est primordiale. Pour cela, nous construisons une matrice de type Origine-Destination (matrice O-D) que nous notons **P**. Chaque ligne i de cette matrice est associée à un nœud i représentant la position actuelle du taxi vide dans le réseau et chaque colonne j représente l'attractivité d'un nœud j pour y aller chercher des clients (son poids P_{ij}) lorsqu'on se trouve en i. La somme des poids dans chaque ligne peut accessoirement être normalisé a 1 comme dans une matrice O-D. Nous avons conclu que le poids P_{ij} doit prendre en compte, de façon croissante λ_j et de façon décroissante à d_{ij} séparant i de j. On obtient donc :

$$P_{ij} = \lambda_j e^{-\alpha d_{ij}}$$

Le coefficient α est un coefficient ajustable pour donner plus ou moins d'importance à l'un des deux critères. Sa détermination nécessite des simulations et la mise en pratique des outils de machine Learning. De même pour la valeur des lambda.

Ainsi, on choisit donc de rediriger le taxi systématiquement le nœud vers lequel P_{ij} est maximum.

En plus de cela, d'autres contraintes doivent être respectées. Supposons que l'étape précédente me redirige vers le nœud k :

Deux scenarios sont possibles

- **Scenario 1** : Le taxi arrive en k toujours vide (sans avoir porter des personnes au préalable sur son itinéraire ik) : dans ce cas, il faudrait que

$$sum_k^j > conso_i^k + conso_k^j$$

Cependant, le choix de k ne le permet pas toujours.

- **Scenario 2** : Le taxi a porte des personnes sur son itinéraire (i, k) :

(B.8) Acceptation et refus de clients

La formulation de ce problème d'acceptation/refus fait intervenir un "seuil", c'est-à-dire un paramètre qui limite les détours subis par tous les clients, à la fois les passagers déjà à bord et le nouveau candidat. L'étude en simulation vise à déterminer, par des simulations répétées, une valeur raisonnable pour ce seuil. Un seuil restrictif permet d'obtenir une bonne qualité moyenne des trajets effectués par les clients acceptés. Cependant, s'il est trop sévère, cela conduira à des refus trop fréquents de nouveaux candidats, ce qui entraîne l'allongement des temps d'attente et éventuellement un taux d'abandon élevé des clients potentiels qui se lassent d'attendre un taxi et sont souvent refusés.

Dans cette démarche, une étude de cas basée sur des données fictives, mais réalistes, est utilisée. Tout d'abord, il est nécessaire de décrire toutes les données quantitatives utilisées dans cette étude.

Les données quantitatives peuvent inclure des informations telles que :

- **La demande de taxi** : le nombre de clients potentiels, leur répartition géographique, les heures de pointe, etc.
- **Les distances et temps de trajet** : la distance entre les différents points de prise en charge et de dépose des clients, les temps de parcours moyens entre ces points, les conditions de circulation, etc.
- **Les données sur les passagers** : le nombre de passagers déjà à bord du taxi, leur destination, etc.

PROPOSITION : ALGORITHME

L'algorithme que nous proposons ici est une heuristique basée sur notre propre analyse du comportement routier dans notre pays et de la méthode de Dijkstra afin de déterminer si on peut accepter ou non un client.

Notations

t_0 : Date actuelle de modification (date de la rencontre)

n_0 : Nœud de la position actuelle

n_i^0 : Nœud d'origine du passage ri

n_i^d : Nœud de destination du passager i

C : Indice du client candidat

n_c^d : Nœud de destination du candidat (son nœud d'origine)

$L = \{n_1, n_2, \dots, n_{M'}\}$: Liste ordonnée des nœuds de passagers (destination à desservir) de l'itinéraire futur (après n_0) du véhicule avant acceptation du candidat.

$l = LU\{n_c^d\}$: Liste non ordonnée des destinations à desservir (après n_0) si le candidat est accepté ; L'un des objets de l'algorithme est d'ordonner l ;

J : L'ensemble des indices des éléments de l ; autrement dit, $J = \{1, \dots, M\}$ où M est le cardinal de l (selon que n_c^d appartient ou n'appartient pas à L , $M=M'$ ou $M'+1$).

I : Ensemble des indices des passagers des véhicules et du candidat ($I = IU\{c\}$)

$d: I \rightarrow J$: Application qui sélectionne la destination des clients c'est-à-dire que $\forall i \in I, n_i^d = n_{d(i)} \in l$;

$P(j) : j \in J$ Nombre de passagers qui descendent du nœud J ; autrement dit $P(j)$ est le cardinal de l'ensemble $d^{-1}(j)$

$\delta(a, b)$: durée du trajet direct pour aller du nœud a au nœud b dans le graphe avec les temps de parcours moyens ; cette matrice des plus courtes distances en temps entre paire de nœud du graphe évaluée avec les temps de parcours moyens des arcs est pré-calculée.

t_i^0 : date à laquelle le passager est entré dans le taxi (pour $i=c$, on a évidemment $t_c^0 = t_0$)

t_j^p : date prévisionnelle d'arrivée à la destination $n_j \in L$ avant que le candidat ne soit accepté.

$$t_j^p = t_0 + \delta(n_0, n_1) + \sum_{k=1}^{j-1} \delta(n_k, n_{k+1})$$

s : seuil de détour acceptable par rapport au trajet direct ; l'un des buts des simulations est d'optimiser ensuite ce paramètre s pour obtenir le meilleur compromis entre temps d'attente initial des candidats – ou taux de rejet de ceux-ci- et la durée moyenne de leur trajet effectif rapportée au trajet une fois qu'ils sont acceptés.

t_j^{lim} : heure limite d'arrivée à la destination $n_j \in l$: cette limite doit traduire le seuil de détour acceptable pour tous les passagers qui descendent au nœud j ; mais elle ne peut être inférieure à heure prévisionnelle d'arrivée à n_j ; compte tenu de la situation actuelle, même si le candidat n'est pas accepté dans le taxi : en effet, les temps de parcours étant aléatoires, il est possible que la contrainte de détour pour certains passagers soit déjà compromise ; il ne faut pas que l'acceptation d'un nouveau client aggrave cette situation, mais il ne faut pas non plus que le nouveau client soit rejeté pour une violation de contrainte qui ne lui est pas imputable. On adopte donc la formule suivante :

$$t_j^{lim} = \begin{cases} \max(t_j^p, \min(t_i^0 + s * \delta(n_i^0, n_c^d))) & \text{si } n_j \in l \\ t_0 + s * \delta(n_0, n_c^d) & \text{si } n_j = n_c^d \text{ et si } n_c^d \text{ n'appartient pas à } L \end{cases}$$

Les notations précédentes permettent de définir les données du problème. On va maintenant introduire les notations permettant d'écrire une formulation du problème sous forme

dynamique. Le problème consiste essentiellement à trouver un ordre de parcours de la liste **I** tel que la contrainte d'heure limite soit respectée à chaque étape de l'itinéraire et que le critère constitué par la somme de ces dates de passages à toutes les étapes du parcours (pondérés par le nombre de client descendant à chacune de ces étapes) soit minimisé. Si aucune solution admissible n'est trouvée, le candidat doit être rejeté. Sinon, on obtient une liste ordonnée de l'ensemble **I**.

Le parcours de **I** nécessite **M** mouvements entre les **M+1** nœuds de l'ensemble $LU\{n_0\}$.

Ces étapes seront indexées par un indice **k** allant de 0 à **M**. On note :

- **x(k)**: la suite ordonnée des nœuds de **I** successivement visités; on complète cette définition par $x(0)=n_0$.
- **u(k)**, **k** allant de 0 à **M-1**: le choix du prochain nœud à visiter une fois qu'on a atteint **x(k)**; par conséquent, $x(k+1)=u(k)$
- **ε(k)**: L'ensemble des nœuds déjà parcouru à l'étape **k**; on considère que $\varepsilon(0) = \phi$: évidemment $\varepsilon(k+1) = \varepsilon(k) \cup \{u(k)\}$
- **t(k)**: heure à laquelle on atteint **x(k)**; $t(0)=t_0$ et $t(k+1)=t(k) + \delta(x(k), u(k))$.

Algorithme

Commencer par calculer les t_j^{lim}

Créer des listes ordonnées l_m pour **m** appartenant à $\{1, \dots, M\}$ avec tous les ordres possibles pour les nœuds d'itinéraire. Poser $c^*=+\infty$ et $m^*=0$.

Pour chaque valeur de **m**,

Poser $x(0)=n_0$ (le nœud actuel) et $c(0)=0$

Pour $k=0 \dots M-1$,

i. Prendre pour $x(k+1)$ le $(k+1)$ ième élément de l_m

ii. Calculer $t(k+1)$

iii. Vérifier si $t(k)$ est inférieure à t_j^{lim} , si calculer aussi $c(k+1)=c(k)+p(x(k+1))*t(k+1)$

Si $c(k+1) < c^*$, passer à la prochaine valeur de **k**.

Sinon, passer à la prochaine valeur de **m** ;

iv. si $t(k) > t_j^{lim}$, passer à la prochaine valeur de **m**.

Fin de boucle sur **k** : si $c(M) < c^*$, remplacer c^* par $c(M)$ et

m^* par valeur courante de **m**.

Fin de boucle sur **m** : si $m^*=0$, le client est rejeté. Sinon, le client est accepté et l'ordre optimal est celui de la liste l_{m^*} .

PARTIE II : IMPLEMENTATION DU PROJET PARCOURS OPPORTNISTE D'ITINERAIRE D'UN TAXI

I. PRÉSENTATION DU PROJET

I.1. Contexte du projet

Au Cameroun, la recherche de la route la plus efficace pour un chauffeur taxi peut avoir un impact sur son gain d'argent. Les conditions de circulation dans notre pays sont complexes et les tarifs sont variables en fonction de l'heure, du trafic et de la distance. De ce fait l'optimisation du choix d'un parcours d'itinéraire pour les chauffeurs taxis revêt une importance cruciale. Dans ce contexte, réalisation d'un projet de visant à optimiser le parcours d'itinéraire au Cameroun tout en maximisant le gain d'argent est d'une grande pertinence. Ce projet nécessite donc une collecte de données, une modélisation mathématique faite dans la première partie de ce rapport, des modèles et des algorithmes de Machine Learning et du développement logiciel pour fournir des recommandations à temps réel aux taximen. Mais dans la suite nous limiterons à un seul aspect de ce projet, qui est la recommandation d'un itinéraire en tenant compte du trafic au niveau des carrefours.

I.2. Problématique du projet

Le secteur des transports des véhicules au Cameroun, fait face à de nombreux problèmes dont le plus récurrent est l'affluence du trafic. Ce problème affecte de nombreux automobiles, en particulier les chauffeurs de taxi, qui ont pour but de maximiser leur gain tout en minimisant le coût en carburant, en temps...

I.3. Objectif du projet

Notre solution a pour but de faire une meilleure recommandation d'itinéraire en fonction de l'affluence du trafic à un carrefour aux différents chauffeurs taxis du Cameroun afin de maximiser au mieux leur gain d'argent et de minimiser au maximum le coût en carburant, le coût en temps ... De ce fait, nous nous sommes tournées vers le Machine Learning pour résoudre ce problème.

II. ANALYSE DU PROJET

Afin de comprendre au mieux le projet qui nous a été proposé nous nous devons de faire une interview aux différents acteurs principaux du problème, notamment les chauffeurs de taxi.

II.1. Rapport d'enquête

Nous avons interviewé trois chauffeurs de taxi dans la ville de Yaoundé. Ci-dessous, les différentes questions qui leur ont été posées pour une meilleure compréhension du problème.

Questions :

1. Quelles sont vos préférences d'itinéraires dans la ville de Yaoundé ?
2. Pourquoi est-ce que vous préférez procéder ainsi ?
3. Avez-vous des préférences d'itinéraires en fonction des horaires ?
4. Si oui pourquoi ? Sinon pourquoi ?
5. Qu'est-ce que vous pensez des dépôts et des courses ?
6. D'après vous, sur quelle base vos collègues taximen ont l'habitude de choisir un parcours par rapport à un autre ?
7. Comment est-ce que vous procédez pour vous sortir des embouteillages ou les éviter ?
8. Dans la zone obili-cradat quels sont les différents lieux de ramassage ?
9. Et dans cette zone quelles sont les destinations habituelles ?

Les réponses de ces chauffeurs de taxi aux questions posées sont respectivement les suivantes :

→ Réponses du taximan n°1 :

1. *“J'aime bien la zone d'ekounou et généralement je me déplace pour aller vers des zones de carrefour assez proches”*
2. *“J'aime cette zone car la distance est courte et le tarif assez conséquent. Je me déplace de carrefour en carrefour parce que ce sont des zones d'affluences. Je sais que si j'ai un passager qui s'arrête près d'un carrefour ou va vers une destination située après un carrefour alors une fois arrivé au carrefour, je trouverai forcément des clients qui vont vers des destinations similaires à celle de mon passager ou même si je veux faire un retour je trouverai toujours des clients qui iront dans mon sens.”*

3. *“Moi je ne travaille qu’en soirée, et généralement je travaille dans la zone de Melen-Ekounou-Odza. ”*
4. *“Car la distance est assez courte et les tarifs assez conséquents en soirée.”*
5. *“D’après mon expérience je gagne plus en faisant de simple déplacement qu’en privilégiant les dépôts sauf en cas de renégociation du tarif normal. En ce qui concerne les courses, j’en ressort généralement plus perdant que gagnant lorsque j’essaie de comparer à de simples déplacements.”*
6. *“Moi je pense que la plupart choisissent leur parcours en fonction des passagers qui font des meilleures offres.”*
7. *“En ce qui concerne la zone nous savons tous qu’il est fortement embouteillé autour de 18h - 20h mais il y a deux chemins qui permettent facilement d’arriver à Ekounou ; Celui passant par la poste et celui passant par carrefour Atangana Mballa. Lorsque je suis bloqué par les embouteillages sur l’un des chemins, je m’arrête et j’essaie de tourner pour passer vers l’autre chemin.”*

→ Réponse du taximan n°2 :

1. *“Je n’ai pas de zone de préférence. Tout dépend du tarif du premier passager et de la distance pour arriver à sa destination.”*
2. *“Je procède ainsi parce que je pense que de cette façon je pourrais emmagasiner le maximum d’argent.”*
3. *“Non je n’ai aucune préférence je vais où le vent m’emporte.”*
4. *“Parce qu’une fois que j’ai choisi le premier passager, je suis entièrement dépendant de lui. Lorsque je suis à vide je me dirige vers des zones d’affluences pour porter des clients.”*
5. *“Les dépôts et les courses sont de bonnes offres. Généralement lorsque j’ai des propositions de dépôt ou de courses, j’ai déjà des passagers à bord. Dans ce cas s’il s’agit d’une course, je propose au nouveau passager de me permettre de laisser l’autre passager avant de faire la course ou si c’est une offre de dépôt, je le propose de le porter malgré le fait qu’il y ait un client à bord, s’il refuse je continue.”*
6. *“Chaque taxi man a sa façon de voir les choses, certains ont des zones de préférences et d’autres pas.”*

7. *“Si je constate que les embouteillages persistent, je peux proposer aux passagers de descendre et s’il refuse on continue notre parcours ensemble car pour moi chaque acceptation de clients est un contrat.”*

→ **Réponses du taximan n°3** : (à la question 8 et 9)

8. *“Ça dépend des heures par exemple le matin comme nous pouvons citer cradat, devant polytechnique, carrefour obili.”*
9. *“Le soir ça se densifie plus car il y a plusieurs points comme carrefour emia, cradat encore, château, cité-U, encore polytechnique avec les étudiants qui rentrent, CHU.”*
10. *“Les destinations sont nombreuses à l’exemple d’emana, etoudi nlongkak, mangui, odza, ekounou, omnisport, ngouso, olembe.”*

Nous pouvons constater grâce à cette interview que chacun taximan a ses préférences en termes d’itinéraires optimales pour au mieux maximiser son gain durant une journée de travail. Nous remarquons aussi qu’il y’a des heures de pointes, des zones de pointes, et des heures où le trafic est vraiment dense et chaque taximan a sa manière de sortir de cela ou de l’éviter, mais n’a vraiment pas de moyens de mieux le prédire.

II.2. Modèle d’apprentissage supervisé

II.4.1. Vue générale du Machine Learning

L’**Intelligence Artificielle (IA)** est définie comme “une série de techniques mises en œuvre pour permettre aux machines d’imiter une forme d’intelligence réelle” (Larousse [7]). Cette définition a le mérite de laisser à entendre que l’IA ne correspond pas à un domaine limité, mais plutôt à un ensemble de domaines. Ainsi, le **Machine Learning** correspond à un sous domaine de l’IA, car il nous permet de disposer d’algorithmes permettant de résoudre des tâches qui étaient préalablement uniquement résolues par l’humain. Le **Deep Learning** quant à lui, est un sous-domaine du Machine Learning : l’état de l’art en termes de traitement d’images, de reconnaissance vocale, et de génération (de texte, d’image...) étant atteint en très grande majorité par des techniques propres au deep learning, dont la spécificité est d’utiliser des réseaux de neurones construits sur une abstraction du fonctionnement du cerveau humain. Enfin, le machine learning se divise en trois catégories distinctes :

- **Le supervised learning (apprentissage supervisé)**: qui consiste à utiliser des données préalablement annotées (labellisées), afin de permettre au modèle d’apprendre une

fonction de prédiction, qui va permettre soit de classer des données dans des catégories (tels que la détermination de l'espèce d'une plante à partir de ses caractéristiques), soit d'effectuer une régression (tels que la prédiction du prix d'une maison par exemple).

- ***L'unsupervised learning (apprentissage non supervisé)*** : qui quant à lui, consiste à utiliser des données non annotées, et dont le but est majoritairement d'extraire des classes d'individus présentant des caractéristiques communes, soit, de généraliser des caractéristiques présentes dans le jeu de données [8].
- ***Le reinforcement learning (apprentissage par renforcement)*** : qui représente une classe d'algorithmes de machine learning consistant à fortifier la connaissance d'un agent de son environnement à travers un processus d'essai/erreur, permettant à l'agent de récolter des informations sur les actions à prendre dans le futur.

Nous nous intéresserons lors de ce projet à priori à l'apprentissage supervisé qui est le mieux adapté dans notre contexte et ses différents modèles.

II.4.2. Modèle d'apprentissage supervisé

L'apprentissage supervisé est une sous-catégorie de l'apprentissage automatique et de l'intelligence artificielle. Il est défini par son utilisation d'ensembles de données étiquetés pour former des algorithmes qui permettent de classer les données ou de prédire les résultats avec précision. L'objectif est de donner un sens aux données dans le contexte d'une question spécifique. L'apprentissage supervisé est utilisé pour des problèmes de classification et de régression, comme la détermination de la catégorie à laquelle appartient un article de presse, ou la prévision du volume des ventes pour une date future donnée, ou encore pour de recommandation d'itinéraires en fonction du trafic sur un carrefour.

L'apprentissage supervisé est fait en utilisant une **vérité**, c'est-à-dire qu'on a une connaissance préalable de ce que les valeurs de sortie pour nos échantillons devraient être. Par conséquent, le but de ce type d'apprentissage est d'apprendre une fonction qui, compte tenu d'un échantillon de données et de résultats souhaités, se rapproche le mieux de la relation entre les entrées et les sorties observables dans les données.

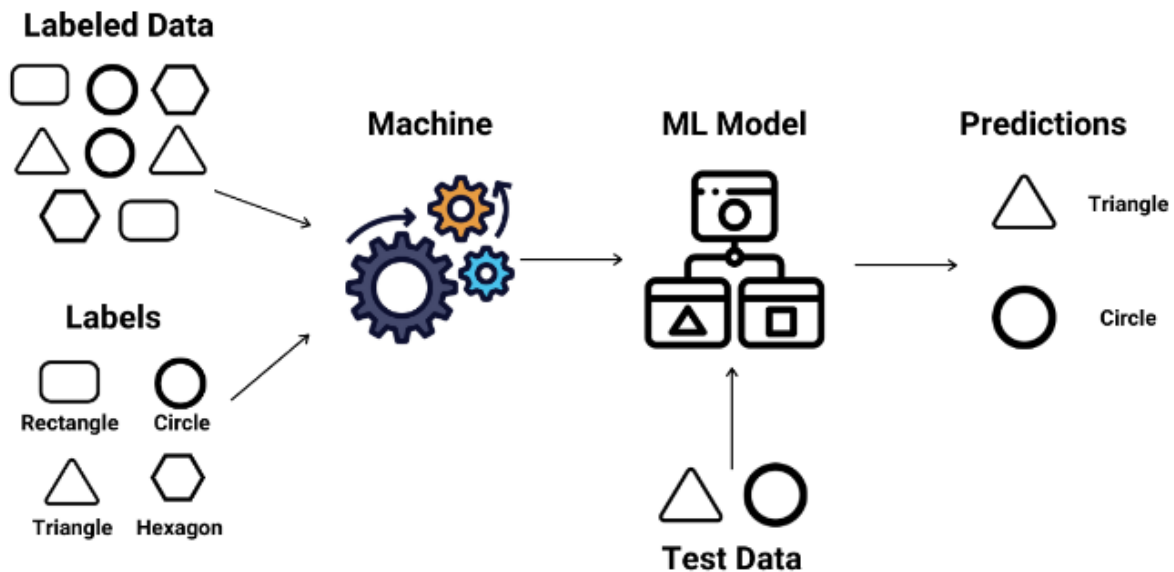


Figure 5 : Architecture d'un modèle d'apprentissage supervisé [9]

Dans l'apprentissage supervisé, on a deux types d'algorithmes :

- **Les algorithmes de régression** : qui cherchent à prédire une valeur continue, une quantité.
- **Les algorithmes de classification** : qui cherchent à prédire une classe/catégorie.

Dans l'apprentissage supervisé, les données d'entraînement fournies aux machines fonctionnent comme le superviseur qui apprend aux machines à prédire correctement la sortie. Il applique le même concept qu'un élève apprend dans la supervision de l'enseignant.

L'apprentissage supervisé, c'est lorsque l'on a des variables d'entrée x et une variable de sortie y et qu'on utilise un algorithme pour apprendre la fonction de mappage de l'entrée à la sortie.

$$Y=f(x)$$

Cela se fait sur plusieurs étapes.

Divers algorithmes et techniques de calcul sont utilisés dans l'apprentissage supervisé. Ci-dessous des explications succinctes de certaines des méthodes d'apprentissage les plus

couramment utilisées, généralement calculées à l'aide de programmes tels que R ou Python [10] :

- **Réseaux neuronaux** : Principalement utilisés pour les algorithmes d'apprentissage en profondeur, les réseaux neuronaux traitent les données d'entraînement en imitant l'interconnectivité du cerveau humain par des couches de nœuds. Chaque nœud est composé d'entrées, de pondérations, d'un biais (ou d'un seuil) et d'une sortie. Si cette valeur de sortie dépasse un seuil donné, elle "déclenche" ou active le nœud, en transmettant les données à la couche suivante du réseau. Les réseaux neuronaux apprennent cette fonction de mappage par un apprentissage supervisé, en s'ajustant sur la base de la fonction de perte par le processus de descente de gradient. Lorsque la fonction de coût est égale à zéro ou proche de zéro, la précision du modèle peut être considérée comme assez fiable pour générer la bonne réponse.
- **Classification naïve bayésienne** : La classification naïve bayésienne est une approche de classification qui adopte le principe d'indépendance conditionnelle de classe du théorème de Bayes. Ainsi, la présence d'une caractéristique n'a pas d'incidence sur la présence d'une autre caractéristique dans la probabilité d'un résultat donné, et chaque prédicteur a un effet égal sur ce résultat. Il existe trois types de classifieurs bayésiens naïfs : bayésien naïf multinomial, bayésien naïf de Bernoulli et bayésien naïf gaussien. Cette technique est principalement utilisée dans la classification des textes, l'identification des courriers indésirables et les systèmes de recommandation.
- **Régression linéaire** : La régression linéaire est utilisée pour identifier la relation entre une variable dépendante et une ou plusieurs variables indépendantes. Elle sert généralement à faire des prédictions sur les résultats futurs. Lorsqu'il n'existe qu'une seule variable indépendante et une seule variable dépendante, on parle de régression linéaire simple. Lorsque le nombre de variables indépendantes augmente, on parle de régression linéaire multiple. Pour chaque type de régression linéaire, la régression linéaire tente de tracer une ligne de meilleur ajustement, qui est calculée par la méthode des moindres carrés. Cependant, contrairement aux autres modèles de régression, cette ligne est droite lorsqu'elle est tracée sur un graphique.
- **Régression logistique** : Alors que la régression linéaire est utilisée lorsque les variables dépendantes sont continues, la régression logistique est choisie lorsque la variable dépendante est catégorique, c'est-à-dire qu'elle a des résultats binaires, tels que « vrai » et « faux » ou « oui » et « non ». Si les deux modèles de régression cherchent à

comprendre les relations entre les entrées de données, la régression logistique est principalement utilisée pour résoudre les problèmes de classification binaire, tels que l'identification des courriers indésirables.

- **Machine à vecteurs de support (SVM) :** Une machine à vecteurs de support est un modèle d'apprentissage supervisé, développé par Vladimir Vapnik, couramment utilisé à la fois pour la classification et la régression des données. Cela dit, elle est généralement utilisée pour les problèmes de classification, en construisant un hyperplan où la distance entre deux classes de points de données est maximale. Cet hyperplan s'appelle la limite de décision. Elle sépare les classes de points de données (par exemple, des oranges comparées à des pommes) de chaque côté du plan.
- **k plus proches voisins :** L'algorithme k plus proches voisins, également appelé algorithme k-NN, est un algorithme non paramétrique qui classe les points de données en fonction de leur proximité et de leur association avec d'autres données disponibles. Cet algorithme part du principe que des points de données similaires peuvent être trouvés à proximité des uns des autres. Par conséquent, il cherche à calculer la distance entre les points de données, généralement par la distance euclidienne, puis il attribue une catégorie basée sur la catégorie ou la moyenne la plus fréquente. Sa facilité d'utilisation et son faible temps de calcul en font un algorithme privilégié par les spécialistes de données, mais au fur et à mesure que le jeu de données de test augmente, le temps de traitement s'allonge, ce qui le rend moins intéressant pour les tâches de classification. Cet algorithme est généralement utilisé pour les moteurs de recommandation et la reconnaissance d'images.
- **Forêt aléatoire :** Une forêt aléatoire est un autre algorithme flexible de machine Learning supervisé utilisé à des fins de classification et de régression. La « *forêt* » fait référence à une collection d'arbres de décision non corrélés qui sont ensuite fusionnés pour réduire la variance et créer des prédictions de données plus précises.
- **Le Boosting :** est une méthode d'apprentissage supervisé consistant à bâtir une prédiction fiable en agrégeant les réponses d'apprenants de base, c'est-à-dire d'estimateurs tout juste meilleurs que le hasard. Cette famille d'algorithmes de machine Learning construit séquentiellement des apprenants de base, encore appelés faibles ou weak learners. A chaque itération, le nouvel estimateur favorise son apprentissage sur les erreurs du précédent et s'y ajoute pour finalement obtenir un strong learner.

III. RÉALISATION DU PROJET

La réalisation de notre implémentation (telle qu'évoquée à la partie I) nécessitant beaucoup de données, nous avons opté de changer d'approche. Nous avons décidé de concevoir une application web basée sur un modèle de Machine Learning pour faire des prédictions sur la densité(qualité) d'un trafic routier au niveau des carrefours/ronds-points. Compte tenu de la difficulté de la collecte des données dans de grands réseaux, nous nous sommes limités dans l'étude de quelques tronçons de la ville de Yaoundé.

III.1. COLLECTE DES DONNEES SUR LE RESEAU ROUTIER DE YAOUNDE

Le périmètre directement concerné couvre une superficie totale d'environ 700 km² dont 304 km² dans les limites administratives de la ville de Yaoundé.

La mobilité urbaine à Yaoundé comme dans la majorité des agglomérations à forte croissance urbaine en Afrique est dominée par des véhicules de petite capacité comme les taxis collectifs et mototaxis. Ce phénomène a comme conséquence une augmentation des externalités négatives comme la congestion, les accidents, la pollution, et des coûts monétaires élevés à supporter par les usagers des transports.

La structure urbaine de Yaoundé a comme conséquences une augmentation de la distance et du temps de déplacement, et une multiplication des véhicules de petite capacité. Le Plan Déplacement Urbain 2010, estime que plus de 86% du volume quotidien du trafic à Yaoundé est effectué par des véhicules de petites capacités avec au plus cinq sièges. Seulement 2% des véhicules sont des minibus (9 à 12 places) et autobus avec plus de 30 sièges.

Selon le plan de mobilité urbaine soutenable pour la communauté urbaine de la ville de Yaoundé, on estime qu'en 2020, ce sont quelques 5 millions de déplacements qui sont réalisés quotidiennement par les habitants de Yaoundé, dont 36% sont des déplacements de quartiers, des déplacements des enfants pour aller à l'école ou encore des adultes pour se rendre aux marchés, essentiellement à pied, ou en motos-taxis. Et 64% sont des déplacements de plus longue portée, pour aller au travail, ou étudier, et sont réalisés en taxis, en motos, en voitures.

Bien que les taxis soient bien nombreux, ils représentent environ 40% des véhicules, parce qu'ils sont utilisés intensément tout au long de la journée. Le taux de remplissage moyen est de 3 passagers et ce sont quelques 2 000 000 déplacements qui utilisent quotidiennement les taxis, à comparer aux 430 000 utilisant les voitures, les 80 000 utilisant les motos.



Figure 6 : Diagramme circulaire de représentation des taux de véhicules utilisés par jour

Cependant, plusieurs problèmes de mobilité apparaissent à Yaoundé : la voirie n'est pas adaptée à la forte proportion de la marche à pied ; la voie ferrée n'est pas structurée pour favoriser la mobilité urbaine par rail. Plus de 74 % des véhicules de petite capacité immatriculés ont plus de 15 ans d'âge. L'offre de transports en commun existant reste insuffisante et inefficace pour satisfaire la demande. Le rapport estime que plus de 86% du volume quotidien du trafic à Yaoundé est effectué par des véhicules de petite capacité (taxis et voitures personnelles). Seulement 2% des véhicules sont des minibus (9 à 12 sièges) et autobus avec plus de 30 sièges. Le trafic horaire représente en moyenne 4% du trafic journalier. L'heure de pointe du matin est située entre 7 et 8 heures ; elle illustre la forte intensité de transport pour des motifs professionnels et éducatifs, et celle du soir de 17 à 19 heures et représente 7% du trafic total. Le trafic entre 22 heures et 6 heures du matin représente 7,5% du trafic total. Suivant les enquêtes sur le réseau routier de la ville aux heures de pointe, il apparaît que le trafic est dense à Yaoundé avec un plancher de 810 et un plafond de 3 530 véhicules à l'heure de pointe. Ceci induit un engorgement et des embouteillages de plus en plus persistants qui allongent les temps de déplacement, favorisant ainsi une consommation accrue d'énergie, et une émission de gaz d'échappement. Le temps moyen des déplacements tous modes relevé par les enquêtes est de 29 minutes. Pour un aller et retour, il est d'environ 60 minutes.

D'après la communauté urbaine de Yaoundé(CUY), la ville dénombre plus de 11222 rues. Aucun chiffre n'est communiqué concernant le nombre de carrefour de la ville.

La figure ci-dessous nous présente la cartographie de la ville de Yaoundé :

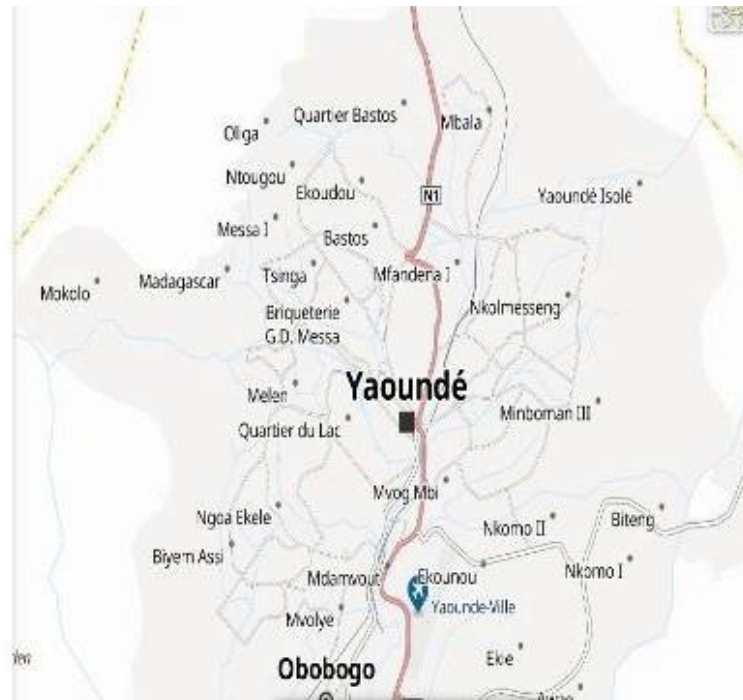


Figure 7 : Cartographie de la ville de Yaoundé

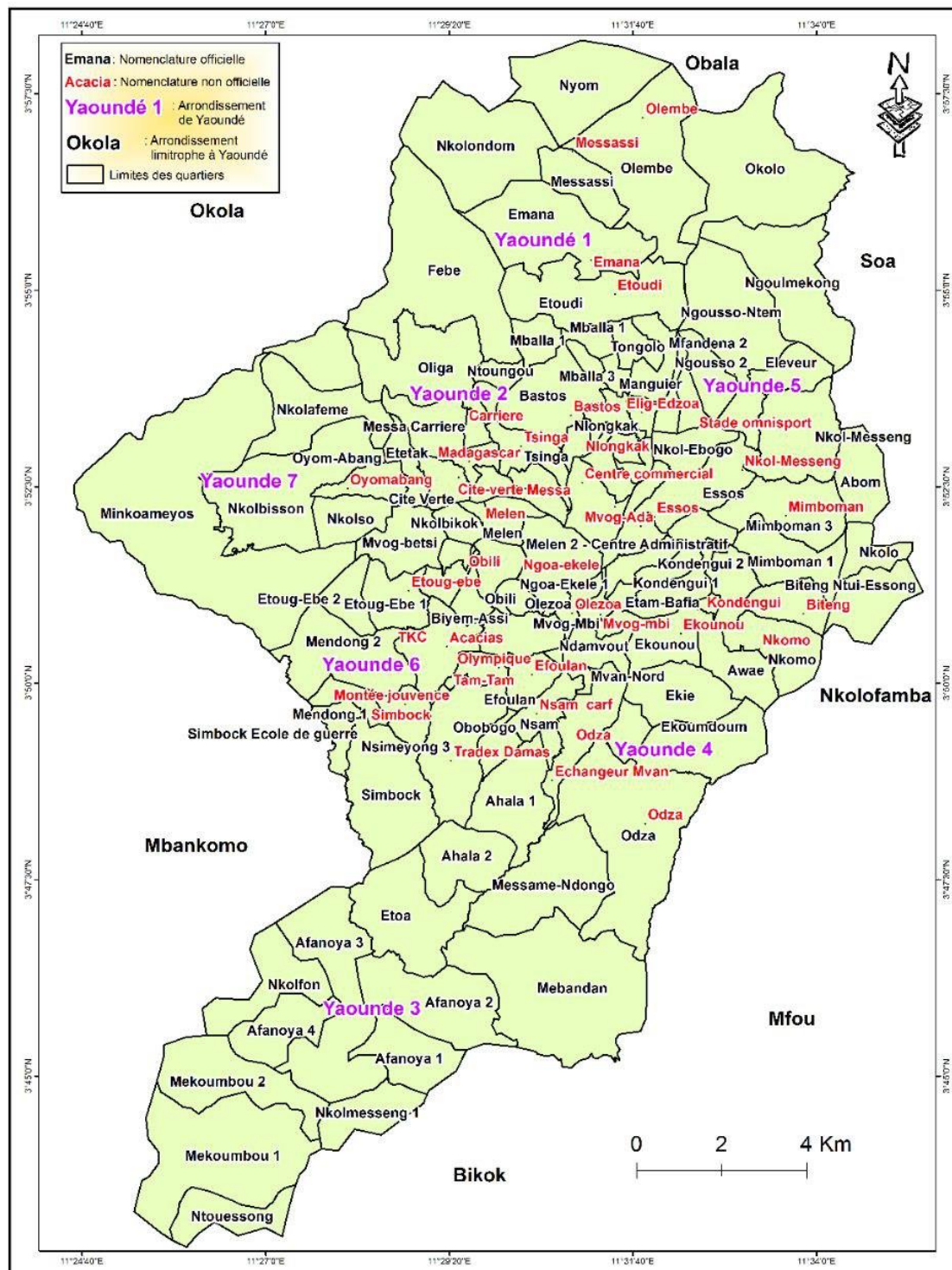


Figure 8 : Cartographie de la ville de Yaoundé

III.1. Présentation de l'environnement de projet

III.1.1. Présentation des outils utilisés pour le frontend

a) Outils de Programmation

Côté Front-End :

- **HTML/CSS** : Utilisés pour structurer et styliser les pages web.
- **JavaScript** : Utilisé pour ajouter des fonctionnalités interactives et dynamiques à l'interface utilisateur.
- **Bootstrap** : Un Framework CSS qui offre des composants et des styles prédéfinis pour un développement web réactif et esthétiquement plaisant.
- **SASS/LESS** : Des préprocesseurs CSS qui permettent d'écrire du CSS de manière plus structurée et modulaire.



Figure 9 : Framework SAAS

Côté Backend :

- **Django REST Framework** : Une extension de Django qui facilite la création d'API RESTful pour communiquer avec des clients front-end.
- **Django Templates** : Le moteur de templates intégré à Django, utilisé pour générer des pages web dynamiques en combinant le code HTML avec des variables et des boucles.
- **Django Channels** : Une bibliothèque qui permet de développer des fonctionnalités temps réel en utilisant des websockets et des protocoles de communication asynchrone.
- **Django Debug Toolbar** : Un outil de développement qui fournit des informations détaillées sur les requêtes, les temps de réponse, les requêtes de base de données, etc.



Figure 10 : Django Framework

Pour une rapide implémentation de l'interface, nous avons jugé bon d'utiliser le Framework Django car facilitant la communication avec le modèle implémenté et la logique backend.

Voici donc quelques raisons supplémentaires qui nous ont poussé à nous servir du paradigme de Django au centre de notre implémentation :

- Django est conçu pour favoriser la productivité des développeurs en offrant une structure de projet claire et des conventions de développement bien définies.
- Il prend en charge la gestion des sessions utilisateur, l'authentification, les autorisations et la gestion des cookies.
- Django propose une API robuste pour la manipulation et le traitement des images, ainsi que pour la génération de fichiers PDF.
- Il offre des outils pour la gestion des tâches asynchrones, ce qui permet de traiter des opérations longues en arrière-plan.
- Django est accompagné d'une riche documentation et d'une communauté active, ce qui facilite l'apprentissage et la résolution des problèmes.
- Il est utilisé par de nombreuses entreprises et organisations renommées dans le monde entier pour développer des applications web de grande envergure.
- Django est connu pour sa stabilité, sa performance et sa capacité à gérer des charges de trafic élevées.
- Il offre des fonctionnalités avancées telles que la gestion des flux de données (streaming), le cache, la gestion des fichiers statiques et bien plus encore.
- Django permet également d'intégrer des applications tierces et des bibliothèques externes pour étendre les fonctionnalités de base du Framework.
- Il suit les meilleures pratiques de sécurité et propose des mécanismes pour se protéger contre les attaques courantes telles que l'injection SQL et les attaques CSRF.
- Django est un choix populaire pour le développement web en raison de sa maturité, de sa fiabilité et de sa capacité à accélérer le processus de développement.

b) Les fonctionnalités implémentées et Objectifs :

Lors de la conception de notre application, Nous avons déployé diverses fonctionnalités pour répondre aux besoins spécifiques de notre projet. Voici un aperçu détaillé des fonctionnalités incluses :

➤ Système d'authentification:

- Nous avons mis en place un système d'authentification robuste qui permet aux utilisateurs de s'inscrire, de se connecter et de gérer leur compte.

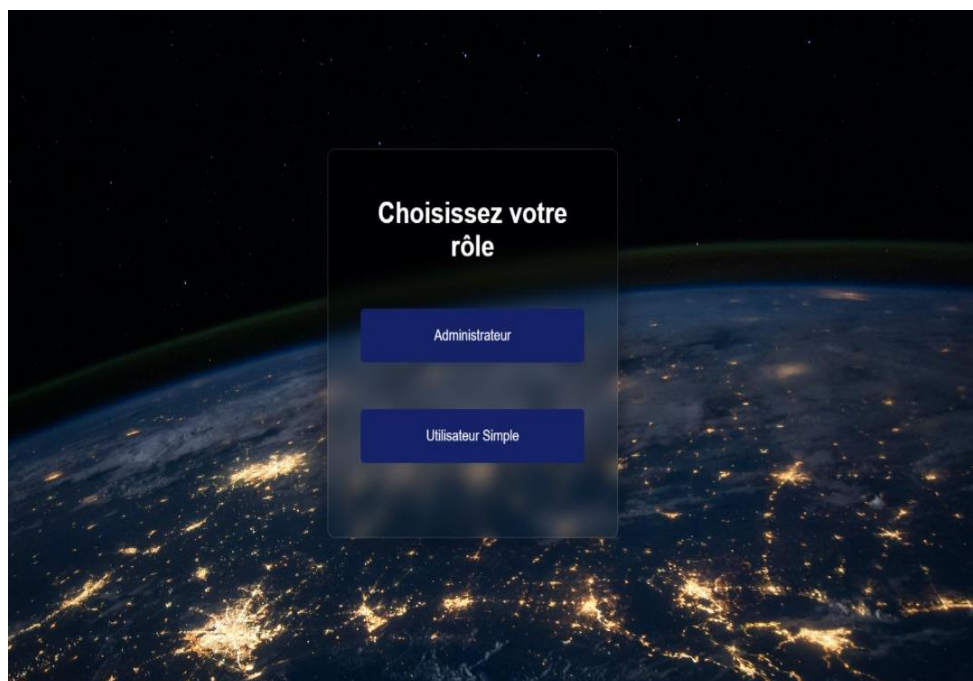


Figure 11 : page de sélection des rôles

- Nous avons également intégré un rôle d'administrateur qui possède des privilèges supplémentaires, tels que la possibilité de supprimer ou de modifier des données sensibles, les historiques des utilisateurs de l'application.

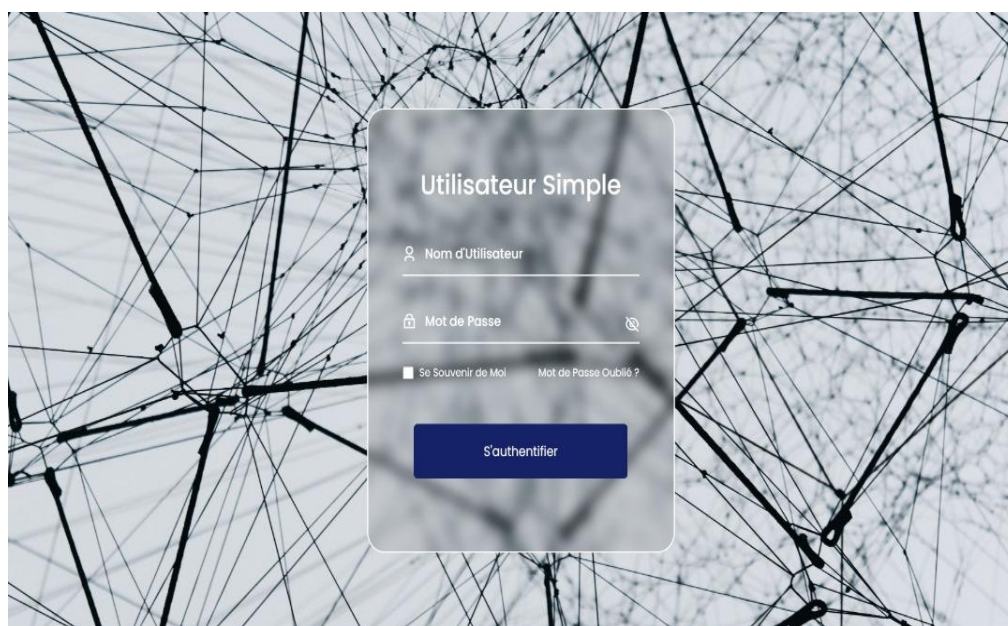


Figure 12 : page d'authentification

➤ **Gestion et maintenance des capteurs :**

La gestion et la maintenance des capteurs sont des aspects essentiels dans notre implémentation, compte tenu de la nécessité de recueillir des données sur de longues périodes. Voici des détails supplémentaires sur cette fonctionnalité :

- L'application intègre un module spécifique dédié à la gestion et à la maintenance des capteurs installés dans les carrefours. Ces capteurs ont été conçus dans le but précis de collecter en temps réel les données des véhicules qui passent par ces intersections.
- Une interface utilisateur intuitive a été développée afin de permettre aux utilisateurs autorisés d'accéder facilement aux données collectées par les capteurs. Grâce à cette interface, les utilisateurs peuvent visualiser les données en temps réel, mais également accéder à l'historique complet des données collectées sur de longues périodes.
- La visualisation des données collectées peut se faire sous différentes formes, telles que des tableaux, des graphiques. Cela permet aux utilisateurs d'analyser et d'interpréter les tendances, les schémas de trafic et les variations saisonnières au fil du temps.
- En outre, l'interface offre également la possibilité de surveiller l'état et la performance des capteurs. Cela inclut l'anticipation des dysfonctionnements éventuels, la génération de rapports de maintenance pour assurer un fonctionnement optimal des capteurs.

La collecte de données sur de longues périodes est essentielle pour obtenir des informations précises et fiables sur les patterns de trafic, les heures de pointe, les variations saisonnières, etc. Ces données historiques sont précieuses pour la prise de décisions stratégiques en matière de planification urbaine, d'optimisation des infrastructures et de gestion du trafic.

En résumé, grâce à la fonctionnalité de gestion et de maintenance des capteurs, notre application permet également d'assurer une collecte continue et fiable des données des capteurs sur de longues périodes, offrant ainsi aux utilisateurs une vision complète et détaillée de l'évolution du trafic dans les carrefours concernés.

VIGILANCE VIEW		
Accueil Carrefour Surveillance Historique Calcul d'itinéraires		
Alertes Densité de Trafic en Baisse		
Nom du Carrefour	Densité du Carrefour	Activité
Carrefour Mvan	1	Actions
Carrefour Camière	2	Actions
Carrefour Ehoulan	1	Actions
Carrefour Etoudi	6	Actions
Carrefour Biyem Assi	3	Actions
Carrefour Meec	5	Actions
Carrefour Obili	10	Actions
Carrefour Awae	5	Actions
Carrefour Odza	5	Actions
Carrefour Golf	8	Actions
Carrefour Nkomo	7	Actions

Figure 13 : page d'alerte densité de trafic en Baisse

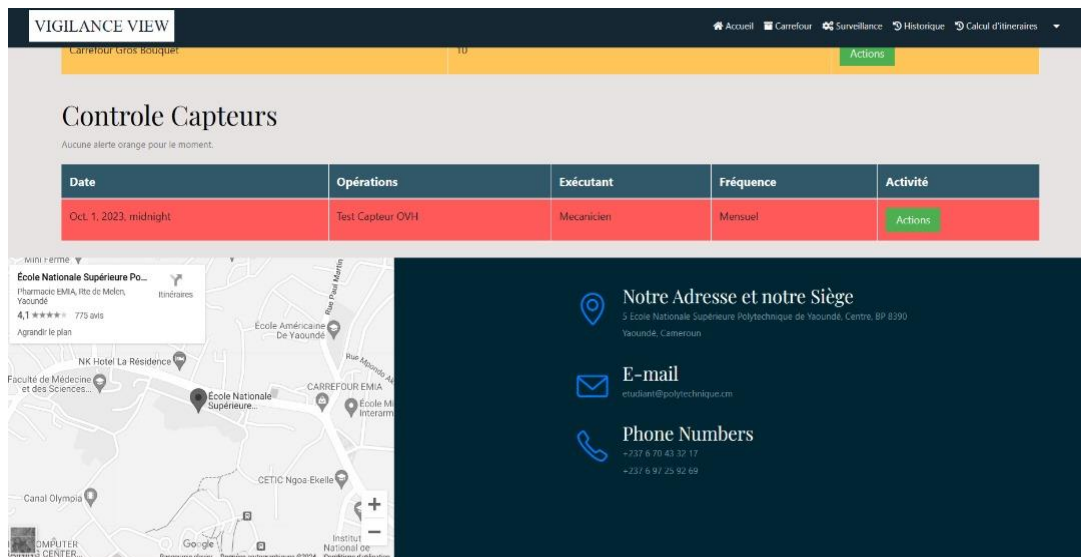


Figure 14 : contrôle des capteurs

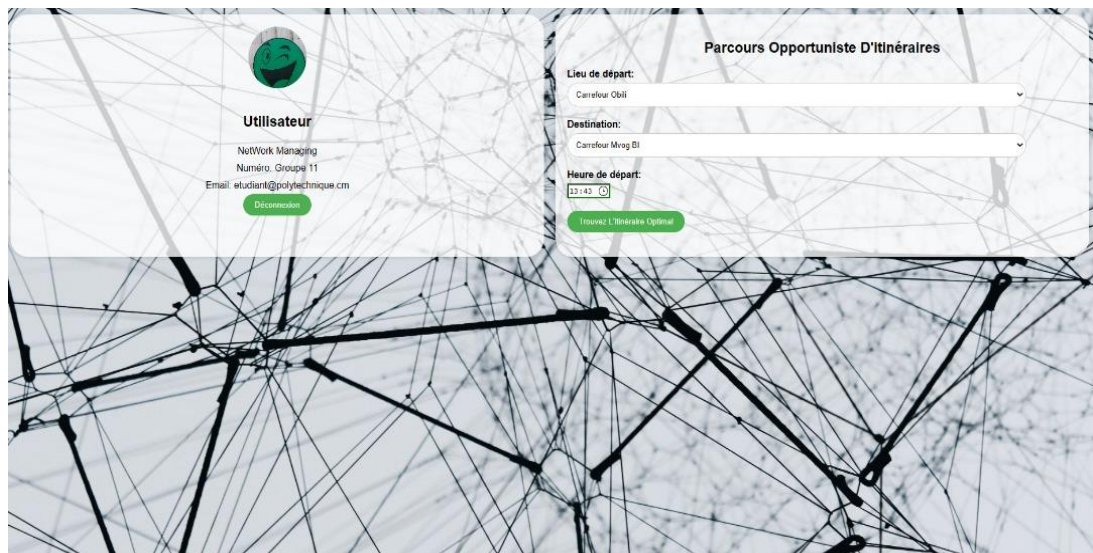
➤ Calcul d'itinéraire optimal

Dans le but d'optimiser les déplacements des taxis lorsqu'ils transportent des clients, nous avons incorporé dans notre application, un algorithme de calcul d'itinéraire optimal. Lorsqu'un taxi se voit attribuer une course, cet algorithme effectue une analyse minutieuse des conditions de circulation, afin de déterminer le trajet le plus efficace permettant au chauffeur de livrer le client dans les délais les plus brefs.

Grâce à cette fonctionnalité, les trajets sont optimisés au maximum, ce qui se traduit par une réduction significative des temps d'attente tant pour les chauffeurs que pour les clients. En effet, l'algorithme sélectionne avec précision les itinéraires les plus appropriés, en prenant en compte divers facteurs. Ainsi, les taxis peuvent atteindre plus rapidement leur destination.

Cette fonctionnalité d'optimisation des itinéraires contribue à améliorer l'expérience globale des utilisateurs. Les clients bénéficient d'une prise en charge rapide et d'une arrivée à destination plus rapide, ce qui accroît leur satisfaction. De plus, les chauffeurs de taxi peuvent effectuer davantage de courses dans un laps de temps donné, améliorant ainsi leur rentabilité et leur efficacité opérationnelle.

En somme, grâce à l'algorithme de calcul d'itinéraire optimal intégré dans notre implémentation, les déplacements des taxis sont grandement facilités. Les trajets sont optimisés, les temps d'attente sont réduits et l'expérience des



Itineraire à suivre : ! [['Carrefour obili', 'Carrefour simbock', 'Carrefour etoug ebe', 'Carrefour biyem-assi', 'Carrefour mvog bi']]

Figure 15 : page du calcul d'itinéraires optimales

Aperçu avec OpenStreetMap :

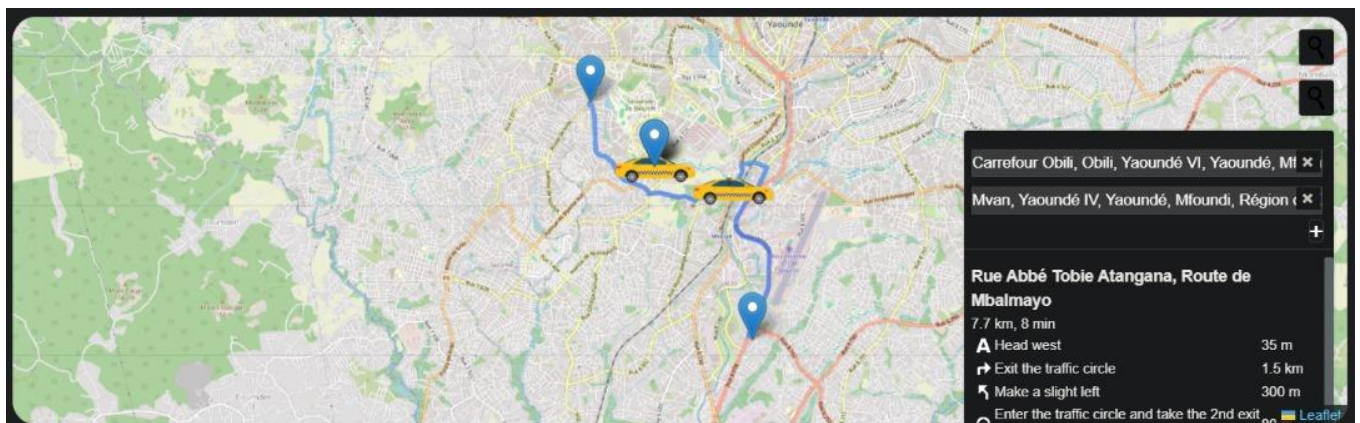


Figure : Itinéraire Optimal avec OpenStreetMap

➤ **Système d'analyse des données des capteurs :**

En effet, dans notre implémentation, nous y avons ajouté un système d'analyse qui exploite pleinement les données collectées par les capteurs. Cette fonctionnalité offre aux utilisateurs la possibilité de générer des statistiques détaillées basées sur ces données, ce qui

leur permet de visualiser clairement les carrefours présentant une densité de trafic élevée et ceux affichant une densité plus faible.

Grâce à ces informations précieuses, les autorités compétentes disposent d'une base solide pour prendre des décisions éclairées en matière de planification urbaine, d'optimisation des infrastructures et de gestion du trafic. Les statistiques générées permettent d'identifier les points névralgiques où la circulation est la plus dense, ce qui peut orienter les efforts d'aménagement et de régulation du trafic dans ces zones spécifiques.

En analysant les données des capteurs, il est possible de déterminer avec précision les heures de pointe, les tendances saisonnières et les schémas de circulation dominants. Ces informations permettent d'anticiper les besoins en termes d'infrastructures routières, de signalisation et de régulation du trafic. Par conséquent, les autorités peuvent prendre des mesures appropriées pour améliorer la fluidité de la circulation, réduire les embouteillages et optimiser l'efficacité du réseau routier.

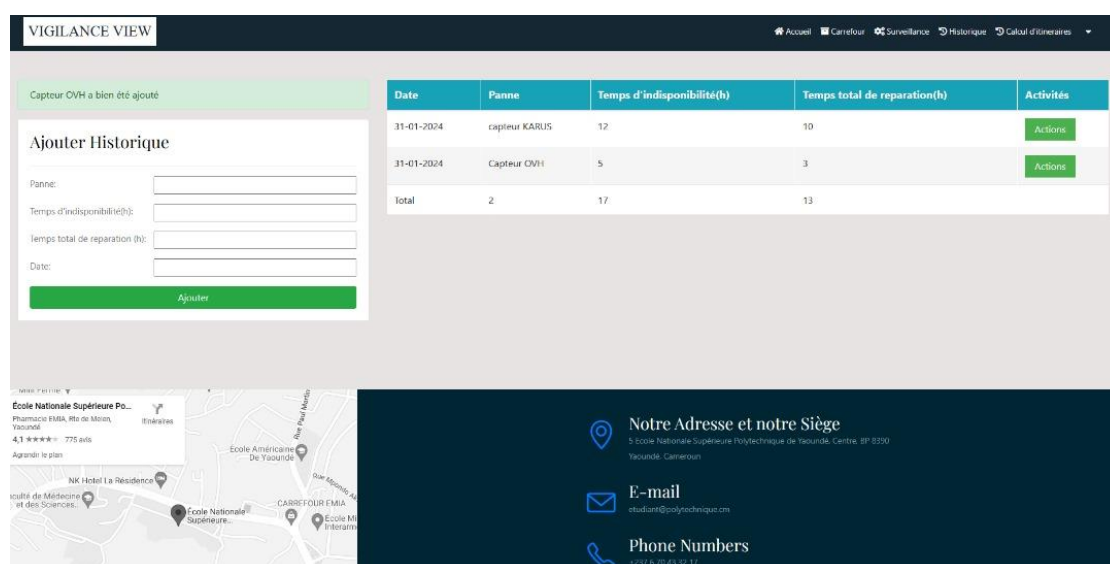


Figure 16 : page du système d'analyse des capteurs

En somme, notre application offre une gamme complète de fonctionnalités, allant de l'authentification et de la gestion des utilisateurs à la collecte de données des capteurs, en passant par l'optimisation des itinéraires et l'analyse statistique des données. Ce faisant, elle contribue à améliorer l'efficacité des opérations de transport, à optimiser la circulation routière et à faciliter la prise de décisions basées sur des données fiables et précises.

III.1.2. Présentation de l'environnement du backend

Pour notre implémentation nous avons utilisé l'IDE **jupyter**, qui a pour particularité de permettre d'exécuter votre code instructions par instructions selon notre bon vouloir ce qui

permet une meilleure maintenabilité en cas d'erreur. Cette IDE est l'outil de développement par excellence du Machine Learning et de l'analyse de données avec python.

Le Machine Learning étant une discipline nécessitant beaucoup de donnée, il est impératif de trouver les méthodes de collecte de données (notamment le nombre de véhicules traversant un carrefour ou un axe routier donné pendant un intervalle de temps t). Pour cela, il existe plusieurs méthodes.

III.2. Méthode comptage des véhicules pour l'analyse du trafic au niveau des intersections

Afin de bien réaliser ce projet nous devons avant tout penser à la collecte de données. Un compte de trafic s'effectue de deux façons(stratégies) :

- Manuellement
- Automatiquement : avec installation d'un équipement électronique d'enregistrement (capteurs).

III.2.1. COMPTAGE MANUEL

Une méthode simple mais précise de comptage du trafic consiste à compter manuellement les véhicules.

Une personne utilise soit un compteur électronique portatif, soit enregistre les données à l'aide d'une feuille de pointage ou regarde une vidéo de la route et compte à partir de là.

Selon des études similaires, le comptage manuel des véhicules est précis à 99% sur la période de comptage.

Avec les comptages manuels, un petit échantillon de données est prélevé généralement sur moins d'une journée et les résultats sont extrapolés pour le reste de l'année ou de la saison. C'est à ce moment que des erreurs sont introduites, car le petit échantillon est rarement exactement représentatif de l'année entière ou même de la semaine.

III.2.2. COMPTAGE AUTOMATIQUE

III.2.2.1. Capteurs non-programmables

a. Vision par ordinateur

De toute évidence, le comptage manuel demande beaucoup de travail. Des systèmes sont désormais disponibles qui analysent automatiquement les images vidéo lorsque les voitures passent devant les caméras, détectant les véhicules avec une précision similaire à celle des personnes qui regardent la vidéo.

L'analyse vidéo locale minimise l'utilisation de la bande passante. Les décomptes de trafic sont téléchargés en temps réel sans fil via Internet, afin que les ingénieurs du trafic puissent visualiser les décomptes en direct et historiques à partir de leurs navigateurs Web.

Les décomptes sont facilement vérifiés simplement en regardant la vidéo et en vérifiant les décomptes automatiques.

Le comptage du trafic vidéo s'effectue généralement en continu, tout au long de l'année, et donne des chiffres précis. Cette méthode de détection des véhicules est souvent utilisée dans les villes intelligentes : les données de comptage sont mises à disposition sur le cloud partout où elles sont nécessaires. Utile pour surveiller les embouteillages, le stationnement et évaluer les initiatives de contrôle du trafic. Un logiciel basé sur l'IA permet de compter et de classer le trafic – camions, voitures, tracteurs, vélos, etc.

b. Comptage pneumatique de tubes routiers

Un ou plusieurs tuyaux en caoutchouc sont tendus sur la route et connectés à une extrémité à un enregistreur de données. L'autre extrémité du tube est scellée.

Lorsqu'une paire de roues heurte le tube, la pression de l'air dans le tube écrasé active l'enregistreur de données qui enregistre l'heure de l'événement.

Une paire de tubes peut être étendue sur plusieurs voies de circulation. L'enregistreur de données peut établir la direction du véhicule en enregistrant sur lequel d'une paire de tubes le véhicule passe en premier. Cela présente l'inconvénient que si deux véhicules traversent les tubes en même temps, la direction ne peut pas être déterminée avec précision. Si deux voitures sont très proches l'une de l'autre lorsqu'elles traversent les tubes, le système peut les considérer comme un seul véhicule à plusieurs essieux.

Les vendeurs revendiquent une précision de 99 %. Des études montrent cependant que l'erreur absolue d'un décompte typique de 15 minutes était en moyenne plus proche de dix pour cent. Cela suggère que le niveau d'inexactitude est masqué par les erreurs de comptage positives et négatives qui s'annulent mutuellement.

c. Capteur micro-onde/radar

Les dispositifs de détection de micro-ondes Doppler transmettent un signal continu de rayonnement micro-ondes de faible énergie sur une zone cible, puis analysent le signal réfléchi. Le détecteur enregistre un changement dans la fréquence des ondes se produisant lorsque la source micro-ondes et le véhicule sont en mouvement l'un par rapport à l'autre. Cela permet à l'appareil de détecter les véhicules en mouvement.

Le radar est capable de détecter des objets éloignés et de déterminer leur position et leur vitesse de déplacement. Avec la détection de véhicule, un dispositif dirige des ondes radio haute fréquence sur la chaussée pour déterminer le délai du signal de retour, calculant ainsi la distance jusqu'au véhicule détecté. Ne convient pas aux embouteillages denses.

Exemple de capteur doppler : OVS-01 GT sensor.

Ce capteur est conçu pour détecter de façon fiable les véhicules à l'arrêt et en mouvement jusqu'à 20 km/h.

Coût 214 775 FCFA

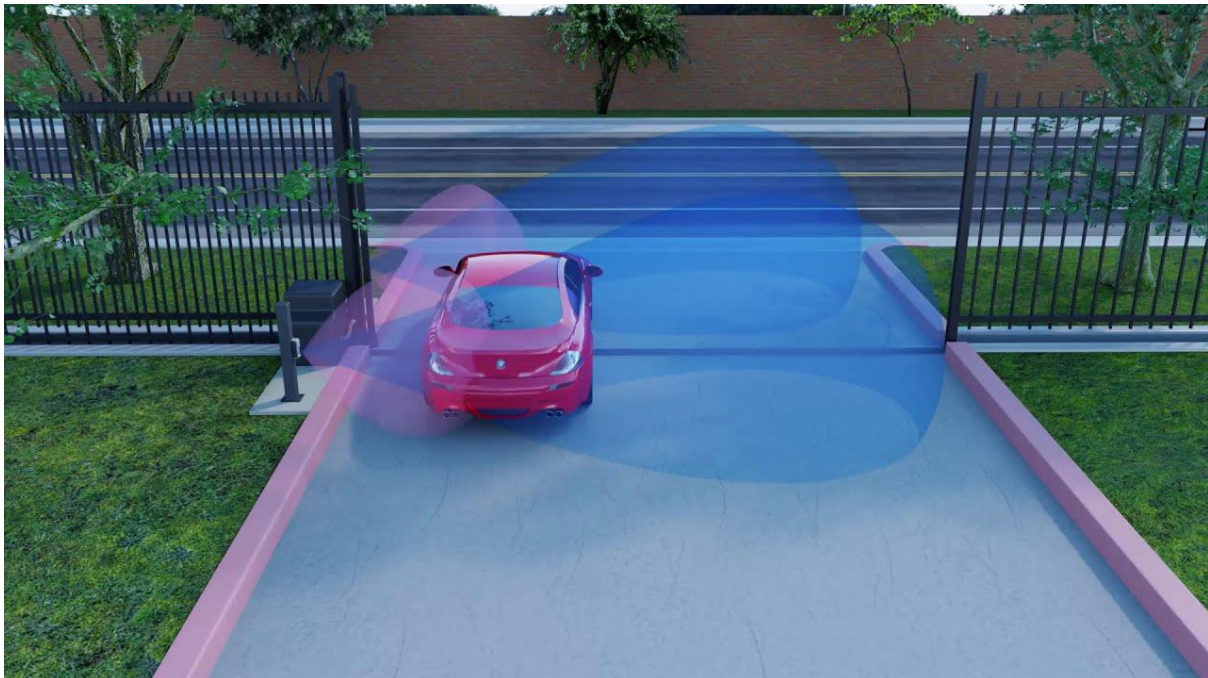


Figure 17 : Fonctionnement du capteur OVS-01 GT sensor

d. Capteur magnétique

Pour mesurer plus efficacement le volume du trafic, le mieux est d'utiliser un capteur magnétique. Ces capteurs détectent les changements dans le champ magnétique terrestre lorsqu'un objet métallique passe.

Il est capable de détecter toutes les différentes parties du véhicule qui passent dessus. Dans certains cas, ces capteurs peuvent enregistrer des informations routières comme la classification d'un véhicule.

Exemple : **karrus**

Karrus distribue et intègre le système de comptage à magnétomètre sans fil Sensys. Cette technologie sans câble et autonome en énergie permet de déployer rapidement, économiquement et avec une maintenance limitée des réseaux de capteurs de trafic dans les zones dépourvues d'instrumentation ou trop compliqués à instrumenter par des technologies câblés.



Figure 18 : Karrus

e. Détecteur acoustique

Détecteur acoustique, ces capteurs sont montés sur un poteau pointant vers la route et détectent les véhicules en fonction de leur bruit de passage. Bien que ceux-ci ne fonctionnent pas aussi bien sur les routes très fréquentées, ils peuvent être installés dans les virages où les voitures ralentissent naturellement.

Exemple : **FLIR TrafiCam**

FLIR TrafiCam est un capteur de présence de véhicule qui combine une caméra CMOS et un détecteur vidéo en un seul appareil. Le FLIR TrafiCam détecte les véhicules en mouvement et stationnaires aux intersections signalisées. Grâce aux sorties de détection.

Coût : 296 450 fcfa



Figure 19 : FLIR TrafiCam

Exemple : **TrafiOne Modèle : FLIR TrafiOne 156**

FLIR TrafiOne est un détecteur tout-en-un pour la surveillance de la circulation et la commande dynamique des feux tricolores. Le FLIR TrafiOne emploie des technologies d'imagerie thermique et le Wi-Fi pour adapter les feux tricolores à la détection de présence de véhicules, vélos et piétons, même dans la nuit la plus sombre ou dans des conditions météorologiques difficiles. Ce capteur génère également des données haute résolution pour mesurer les durées des trajets pour différents modes de transport et pour améliorer la circulation. TrafiOne comprend également une caméra vidéo HD pour une aide visuelle supplémentaire.



Figure 20 : FLIR TrafiOne 156

f. Capteur piezoélectrique

Un capteur piézoélectrique collecte des données lorsqu'une voiture passe dessus. Ces capteurs sont installés dans une rainure de la route et les voitures qui passent dessus génèrent un signal de tension. Un dispositif de comptage est fixé à proximité et les données peuvent être collectées localement ou par modem.

Exemple : **Piezoelectric Traffic Sensor**

Le capteur de trafic piézoélectrique RoadTrax BL Brass Linguini (BL) est conçu pour une installation permanente ou temporaire dans ou sur la surface de la route pour collecter des données sur le trafic



Figure 21 : Piezoelectric Traffic Sensor

III.2.2.1. Capteurs programmables

Nous allons citer dans la suite quelques capteurs de proximité qui pourront être programmés pour recueillir les données sur le trafic routier.

Voici quelques différentes étapes pour programmer ces capteurs qui seront cités dans la suite :

- Sélectionnez le capteur approprié : Choisissez le capteur en fonction de la grandeur physique que vous souhaitez mesurer. Assurez-vous qu'il est compatible avec votre système de développement ou votre microcontrôleur.
-
- Connectez le capteur : Selon le type de capteur, vous devrez peut-être le connecter à votre système à l'aide de fils ou de connecteurs spécifiques. Consultez la documentation du capteur pour connaître les détails de connexion.
- Alimentez le capteur : Les capteurs nécessitent généralement une alimentation électrique pour fonctionner. Assurez-vous de fournir la tension ou le courant approprié au capteur selon les spécifications fournies.
- Programmez le microcontrôleur : Si vous utilisez un microcontrôleur ou un

système de développement, vous devrez écrire le code pour lire les données du capteur.

- Calibrez le capteur (facultatif) : Certains capteurs peuvent nécessiter une calibration pour obtenir des mesures précises. Suivez les instructions du fabricant pour calibrer le capteur si nécessaire.
- Traitez les données : Une fois que vous avez lu les données du capteur, vous pouvez les traiter selon vos besoins. Cela peut inclure des calculs, des comparaisons, des enregistrements ou l'activation de certaines actions en fonction des valeurs mesurées.
- Testez et validez : Assurez-vous de tester et de valider les performances du capteur dans différentes conditions pour vous assurer qu'il fournit des mesures précises et fiables.

Listes des différents capteurs

- **Capteur de proximité à ultrasons** UGT58, UGT59 séries



Figure 22 : Capteur de proximité à ultrasons

Description du capteur

- Détecteurs à ultrasons -Type M18 Cube en boîtier métallique.
 - Détection sans contact indépendante de la couleur, de la transparence ou de la nature de la surface de l'objet.
 - Boîtier peu encombrant approprié pour des espaces limités.
 - Réglage intuitif de la zone de détection par appui sur une touche •Avec sortie de commutation programmable et sortie analogique évolutive •Boîtier en métal carré robuste avec filetage M18 latéral.
- **Capteur de proximité photoélectrique** PT430470



Figure 23 : Capteur de proximité photoélectrique

Description du capteur

Les capteurs optiques fonctionnent sans contact. Ils détectent les objets indépendamment de leurs caractéristiques (par exemple, forme, couleur, structure de surface, matériau). Le principe de fonctionnement de base est basé sur l'émission et la réception de la lumière. Il existe trois versions différentes :

- Le capteur à faisceau barrage se compose de deux dispositifs distincts, un émetteur et un récepteur qui sont alignés l'un par rapport à l'autre. Si le faisceau lumineux entre les deux dispositifs est interrompu, la sortie de commutation intégrée dans le récepteur change d'état.
- Avec le système de rétro-réflexion, l'émetteur et le récepteur se trouvent dans un seul appareil. Le faisceau lumineux émis est réfléchi vers le récepteur par un réflecteur qui doit être monté à l'opposé du dispositif. Dès que le faisceau lumineux est interrompu, la sortie de commutation intégrée dans l'appareil change d'état.
- Avec le capteur à réflexion diffuse, l'émetteur et le récepteur sont dans un seul appareil. Le faisceau lumineux émis est réfléchi par l'objet à détecter.

➤ Capteur de proximité rectangulaire



Figure 24: Capteur de proximité rectangulaire

Description du capteur

La technologie de proximité est utilisée depuis des décennies pour détecter un vide insuffisant dans les boîtes de conserve. Cependant, l'utilisation récente et généralisée des extrémités bimétalliques Easy Open (EZO) (extrémités en acier avec languette en aluminium) a représenté un défi pour la technologie traditionnelle. Le détecteur de proximité TapTone Twin est conçu pour détecter et rejeter les boîtes de conserve en acier à vide faible ou sans vide avec des extrémités EZ-Open (tirettes) à des vitesses de production allant jusqu'à 525 pieds par minute. Le nouveau détecteur de proximité double combine les inspections de Proximity-1 et Proximity-2 avec un algorithme amélioré offrant une meilleure résolution sur le profil du couvercle. TapTone propose le capteur de proximité double sur les systèmes PRO Série et T550. (Non supporté par l'IHM).

Ses caractéristiques sont :

- Vitesse de ligne jusqu'à 2,000 cpm, précision de détection jusqu'à 0.005 in (0.127mm).
- Fenêtre de capture de profil de signal réglable pour un meilleur traitement du signal.
- Quatre algorithmes pour calculer le profil, la hauteur, l'arête et le contour - Paramètres de gain et de décalage programmables pour améliorer la plage d'inspection.
- Les limites de rejet ont suivi automatique dans le contrôleur s'adaptent aux fluctuations de température et de pression au cours de la production Double capteur de proximité - Vitesses de ligne jusqu'à 2 000 cpm, précision de détection jusqu'à 0,127 mm (0,005 in).
- Fenêtre de capture du profil du signal réglable pour un meilleur traitement du signal.

- Quatre algorithmes pour calculer le profil, la hauteur, l'inclinaison et le contour - Paramètres de gain et de décalage programmables pour améliorer la plage d'inspection.
- Interface utilisateur requise : Peut être utilisé avec une interface T550 ou Pro Séries.

Estimation de l'installation des capteurs dans la ville de Yaoundé

Etant donné qu'on dénombre environ 1200 rues et en prenant 50 carrefours dans la ville de Yaoundé en positionnant 2 capteurs de types dans chaque rue on les estimations suivantes :

- Pour le capteur ovs : on a une estimation du cout total de 535 460 000 FCFA.
- Pour le capteur flir trafiCam on a une estimation du cout total de 731 480 000 FCFA.

III.3. Caractéristique du jeu de données

Notre jeu de données est inspiré de celui du trafic routier dans 04 carrefours de l'Angleterre durant la période du premier janvier 2015 au 31 Décembre 2017. En effet, ces données ont été récoltées à partir des capteurs placés au niveau de 04 carrefours préalablement choisis par des chercheurs britanniques dans le but d'effectuer une analyse de données afin d'interpréter quels sont les facteurs qui influencent la densité d'un trafic routier, mais aussi de prédire la densité du trafic pour les jours à venir.

En ce qui concerne les données que nous avons utilisées, nous pouvons dire qu'ils sont inspirés de ceux du dataset du trafic routier de l'Angleterre parce que, nos données sont structurées de la même manière plus précisément au niveau des types de données de chaque colonne, les étiquettes des colonnes et également la structure de données de chaque ligne. Les lignes de notre dataset sont structurées de la façon suivante : la colonne **DateTime** va du 1er Janvier 2019 au 31 Décembre 2023 en bloc de 10 car nous avons 10 dix jonctions (carrefours) ce qui nous fait au total 569 736 lignes. De plus, nos données sont supposées représenter ceux de la ville de Yaoundé et les étiquettes 1,2,3,4,5,6,7,8,9,10 de la colonne **Junction** représentent respectivement les carrefours suivants :

- ❖ Carrefour obili
- ❖ Carrefour biyem-assi
- ❖ Carrefour melen
- ❖ Carrefour simbock
- ❖ Carrefour etoug ebe

- ❖ Carrefour mvog bi
- ❖ Carrefour mvan
- ❖ Carrefour odza
- ❖ Carrefour mvog-ada
- ❖ Carrefour awae

III.4. Modèles et algorithmes

Pour entraîner et tester notre jeu de données, nous avons utilisé les modèles et algorithmes de Machine Learning suivants :

- **Gradient Boosting Classifier** : pour ce modèle nous avons eu une performance de 57.48%.
- **Régression logistique** : ici nous avons eu une performance de 57.49%.
- **K-NearestNeighbor (KNN) classifier** : ici nous avons eu une performance de 52.81%.
- **Support Vector Machine (SVM) Classifier** : ici nous avons eu une performance de 57.49%.
- **Random Forest Classifier** : ici nous avons eu une performance de 57.08%.
- **Decision Tree Classifier** : ici nous avons eu une performance de 57.05%.

Ci-dessous nous avons les captures d'écran des informations détaillées sur les performances des différents modèles qui assez performant pour les tests de prédictions :

```

Support Vector Machine (SVM) Classifier:
Accuracy: 57.49%
Classification Report:
      precision    recall  f1-score   support

     0       0.00      0.00      0.00      72660
     1       0.57      1.00      0.73      98261

 accuracy          0.57      170921
 macro avg         0.29      0.50      0.37      170921
 weighted avg      0.33      0.57      0.42      170921

k-Nearest Neighbors (KNN) Classifier:
Accuracy: 52.81%
Classification Report:
      precision    recall  f1-score   support

     0       0.44      0.39      0.41      72660
     1       0.58      0.63      0.60      98261

 accuracy          0.53      170921
 macro avg         0.51      0.51      0.51      170921
 weighted avg      0.52      0.53      0.52      170921

```

Figure 25 : classification report du modèle SVM et KNN

```

Logistic Regression Classifier:
Accuracy: 57.49%
Classification Report:
      precision    recall  f1-score   support

     0       0.00      0.00      0.00      72660
     1       0.57      1.00      0.73      98261

 accuracy          0.57      170921
 macro avg         0.29      0.50      0.37      170921
 weighted avg      0.33      0.57      0.42      170921

Gradient Boosting Classifier:
Accuracy: 57.48%
Classification Report:
      precision    recall  f1-score   support

     0       0.48      0.00      0.01      72660
     1       0.58      1.00      0.73      98261

 accuracy          0.57      170921
 macro avg         0.53      0.50      0.37      170921
 weighted avg      0.53      0.57      0.42      170921

```

Figure 26 : classification report du modèle logistic regression et gradient Boosting classifieur

```

Decision Tree Classifier:
Accuracy: 57.05%
Classification Report:
      precision    recall  f1-score   support

     0       0.46      0.06      0.11      72660
     1       0.58      0.95      0.72      98261

 accuracy      0.57      0.57      0.57      170921
  macro avg       0.52      0.50      0.41      170921
 weighted avg       0.53      0.57      0.46      170921

Random Forest Classifier:
Accuracy: 57.08%
Classification Report:
      precision    recall  f1-score   support

     0       0.46      0.06      0.11      72660
     1       0.58      0.95      0.72      98261

 accuracy      0.57      0.57      0.57      170921
  macro avg       0.52      0.50      0.41      170921
 weighted avg       0.53      0.57      0.46      170921

```

Figure 27 : classification report du modèle Decision Tree Classifier et Random Forest Classifier

➤ SYNTHESE

	Decision Tree Classifier	Random Forest	Support Vector Machine (SVM)	k-Nearest Neighbors (KNN)	Logistic Regression	Gradient Boosting
Rapidité d'exécution	Rapide à entraîner, surtout si l'arbre est peu profond.	L'entraînement peut être lent, car il implique la construction de multiples modèles.	Peut-être lent à entraîner, surtout avec un grand nombre de données et le choix d'un noyau non linéaire.	Rapide à l'entraînement car il ne requiert pas de phase d'apprentissage, mais lent lors de la prédiction, surtout avec de grandes quantités de données.	Généralement très rapides à entraîner et à prédire.	L'entraînement peut être lent, car il implique la construction de multiples modèles.
Performance	57.05%	57.08%	57.49%	52.81%	57.49%	57.48%
Utilisation	Bien pour les ensembles de données de taille moyenne.	Peuvent être parallélisés pour améliorer la vitesse.	Convient bien aux grands ensembles de données.	La prédiction est plus coûteuse car elle exige de calculer la distance entre les exemples de test et tous les exemples d'entraînement.	Convient bien aux grands ensembles de données.	Peuvent être parallélisés pour améliorer la vitesse.

Aux vues des tests de performances effectuées sur notre jeu de données d'environ 569 000 données, il en ressort que les modèles et les algorithmes les plus appropriés pour notre classification sont les algorithmes de régression logistique et de gradient boosting car ils ont eu de meilleures performances et de plus, leur temps d'exécution était assez raisonnable (moins de 4 minutes).

III. CHALLENGES ET PERSPECTIVES

IV.1. Challenges

Nous avons fait face à de nombreux défis durant la réalisation de ce projet. Notamment au niveau de la collecte des données et la génération aléatoire de données. En effet, collecter les données respectives au problème qui nous a été posé avec le temps imparti qui était extrêmement court, était quasiment impossible pour nous. Nous avons donc songé à générer des données en se basant sur des données déjà existantes qui avaient fait l'objet d'une étude quelques années plus tôt. Cependant, c'était vraiment fastidieux de générer ces données et de suivre une certaine logique au niveau des données pour éviter que celles-ci soient trop parfaites et s'écartent fortement de la réalité.

Au tout début, nous avons pensé à certaines librairies python notamment la librairie Fake, mais celle-ci génère des données de façon aléatoire ce qui devait provoquer une incohérence vu la structure des données que nous voulions. C'est pourquoi nous nous sommes penchés sur Excel. Nous avons généré les données de la colonne « vehicles » (nombres de véhicules ayant traversé un carrefour par heure) sur Excel en utilisant la fonction **AleaEntreBorne (min, max)** en définissant après une certaine période une nouvelle fonction **AleaEntreBorne** ayant un max ou un min différent afin de varier au mieux les données. Ensuite, pour générer les données de la colonne « Datetime », nous avons essayé de remplir quelques lignes de la colonne afin de déduire le reste par propagation mais pour notre cas, la propagation sur Excel s'est avérée incohérente, c'est pourquoi après plusieurs tentatives sans succès nous sommes passé à Ubuntu pour utiliser **Libre Office de Open Calc** ce qui a été paradoxalement plus satisfaisant qu'Excel pour la propagation. Enfin, pour générer les données de la colonne « Junction » nous avons simplement utilisé la propagation identique suivant la période du 01/01/2019 au 31/12/2023. Au bout de l'effort, nous avons pu générer exactement 569 736 données.

IV.2. Perspectives

Pour mener à bien ce projet dans le futur, il faudrait faire une étude sur une longue période, afin d'avoir un maximum d'informations sur le réseau routier, sur la densité du trafic à un carrefour, sur le point de ramassage des passagers etc.

On pourrait donc utiliser des capteurs sensorielles cités dans la partie **III.2**, afin d'avoir une meilleure précision dans nos données.

Il faudrait aussi réaliser un maximum d'interviews afin de connaître l'avis des différents chauffeurs de taxis et prendre cela en compte lors de la réalisation de ce projet.

Cependant nous pouvons déjà commencer ce projet en lisant ces différents documents qui portent sur :

- Le plan de mobilité urbaine dans la ville Yaoundé
- Dysfonctionnement de l'offre de transports en Afrique : cas de Yaoundé
- Etc.

CONCLUSION

Après avoir abordé de manière approfondie le problème de modélisation d'un parcours opportuniste d'itinéraire, notre projet nous a conduit à une question fondamentale : comment maximiser les gains tout en évitant les zones fortement encombrées, les embouteillages, et en réduisant au minimum les trajets à vide ? De plus, nous nous sommes efforcés de répondre à ce défi en tenant compte de la possibilité de transporter le maximum de passagers dans la même direction, tout en privilégiant les destinations à forte demande. En utilisant une approche combinant l'analyse des données en temps réel, les algorithmes d'optimisation et les modèles prédictifs, nous avons développé un système efficace et adaptable pour résoudre ce problème complexe. Une fois l'implémentation terminée, notre projet pourra offrir ainsi de nouvelles perspectives pour améliorer l'efficacité des itinéraires de transport en maximisant les gains tout en réduisant les temps de trajet et les coûts.

BIBLIOGRAPHIE

- [1] : <https://www.techno-science.net/glossaire-definition/Algorithme-de-Dijkstra.html>
- [2] : <https://devmath.fr/informatique/algorithme-de-bellman-ford/>
- [3] : <https://theses.hal.science/tel-01104895/document>
- [4] : <https://theses.hal.science/tel-02917965v1/file/2016GregoireSCANO.pdf>
- [5] : <https://theses.hal.science/tel-00735879/document>
- [6] : https://theses.hal.science/tel-00553335/file/these_graebener.pdf
- [7] : https://www.larousse.fr/encyclopedie/divers/intelligence_artificielle/187257
- [8] : https://fr.wikipedia.org/wiki/Apprentissage_non_supervis%C3%A9
- [9] : [Qu'est-ce que l'apprentissage supervisé ? | IBM](#)
- [10] : [L'apprentissage supervisé : définition et exemples | Blent.ai](#)
- [11] : Évaluation et optimisation de systèmes de taxis collectifs en simulation Eugénie
- [12] : Dysfonctionnements de l'offre de transport en Afrique : cas de Yaoundé Documents de travail Jean Patrick Mfoulou Olugu
- [13] : Plan de Mobilité Urbaine Soutenable pour la Communauté Urbaine de Yaoundé Rapport de synthèse - version finale