

Rapport sur

# Smart Traffic Control Hub

Réalisé par : SIGINOV

Encadré par : Geomatic & GeoIT

Adresse de l'entreprise

## ➤ OBJECTIF DU RAPPORT :

Décrire la conception et la mise en œuvre d'un prototype du Smart Traffic Control Hub visant à améliorer la gestion du trafic dans une ville intelligente.

## ➤ CONCEPTION INITIALE :

- **Data setup:**

Simulated Data Environment: we'll incorporate relevant data from **Esri's Living Atlas** of the World, such as real-time traffic conditions, road closures, and live weather data, to enrich the traffic simulation.

- **Gis mapping:**

Use **Esri's ArcGIS** to create a detailed map of the city, including roads, intersections, and traffic flow.

- **Traffic simulation:**

Utilize **Esri's Network Analyst** extension to run traffic simulations, identifying congestion points and suggesting optimal traffic signal timings. Live data streaming and processing using **Esri's GeoEvent Server**.

- **Scaled Deployment Strategy:**

Plan a phased city-wide rollout starting with the most critical intersections.

- **Analytics Blueprint:**

Propose a basic analytics dashboard using **Esri's Operations Dashboard** for ArcGIS.

- **Backup Data:**

To integrate a historical traffic data backup feature into our Smart Traffic Management System, we will establish an **enterprise geodatabase on ArcGIS**.

## ➤ REALISATIONS :

### PARTIE ROUTE

#### 1. Source des Routes

Utilisation d'une source officielle des routes de la France pour assurer la précision des données.

#### 2. Structure de la Couche des Routes :

La couche des routes est structurée avec une table attributaire contenant des informations essentielles telles que l'ID de la route, la longueur de la route, ainsi que la localisation des capteurs d'entrée et de sortie.

#### 3. Calcul du Temps de Trajet :

Une fonction en arcpy a été développée pour calculer le temps de trajet à travers les données du réseau routier. La formule utilisée prend en compte la longueur de la route et suppose une vitesse moyenne de 40 km/h dans les zones urbaines.

### PARTIE CAPTEURS

#### 1. Utilisation d'une API pour Générer des Données Aléatoires :

Plutôt que de dépendre des données d'Esri's Living Atlas, une API a été sélectionnée pour générer des données aléatoires jouant le rôle des capteurs d'entrée et de sortie de chaque route. Cela permet une simulation plus dynamique du trafic, prenant en compte les variations aléatoires.

#### 2. Traitement des Données par une Fonction :

Une fonction a été mise en œuvre pour ajuster le champ du temps dans la couche des routes en fonction du retard dû à l'embouteillage. Cette approche cible spécifiquement les routes présentant un taux d'embouteillage élevé, optimisant ainsi le traitement des itinéraires entre l'origine et la destination.

## PARTIE LOCALISATION

### 1. Utilisation d'une API pour la Localisation de l'Utilisateur :

Intégration d'une API pour localiser l'utilisateur en se basant sur la localisation de son PC portable, contribuant à une expérience utilisateur plus personnalisée.

## TRAITEMENT DES DONNEES

### 1. Agrégation des Données provenant des API :

Une fois les données agrégées, la deuxième étape du traitement des données implique l'utilisation de l'extension **Network Analyst d'ArcGIS** pour calculer le chemin le plus court en termes de temps.

Cette extension offre des fonctionnalités puissantes pour l'analyse du réseau routier. En spécifiant le point de départ et d'arrivée, ainsi que les données agrégées sur les routes et les conditions de trafic, l'algorithme de Network Analyst génère un itinéraire optimal en considérant les temps de trajet actuels.

### 3. Fonction pour Générer le Résultat :

La dernière étape du traitement des données consiste en l'élaboration d'une fonction spécialisée qui prend en paramètre le réseau routier, le point de départ et d'arrivée pour générer le shapefile du chemin optimal. Ce résultat de l'itinéraire optimal est ensuite converti en un fichier JSON. Ce format de données structuré offre une flexibilité considérable et facilite l'intégration avec d'autres systèmes ou la visualisation à l'aide d'interfaces web.

En combinant ces trois étapes, le traitement des données vise à fournir des recommandations de trajet optimales en fonction des conditions en temps réel, intégrant des informations dynamiques sur le trafic pour améliorer la précision des résultats.

## PARTIE GESTION DES FLUX DE DONNEES EN TEMPS REEL :

Il s'agit d'une simulation et de traitement de données de trafic routier, implémentant Apache Kafka et Python pour gérer des flux de données en temps réel. L'objectif principal est de créer un système capable de simuler,

produire, consommer et traiter des données de trafic en vue d'analyses plus approfondies.

## **1. Configuration de Kafka**

Producteur Kafka : Envoie des données sérialisées en JSON à un topic spécifique.

Consommateur Kafka : Récupère et déséréalise les données du même topic.

## **2. Simulation des Données de Trafic**

Génération des Données : Utilisation d'appels API pour simuler les taux de trafic.

Calcul de la Congestion : Basé sur les taux de trafic et d'autres paramètres.

## **3. Traitement des Données**

Gestion des Messages : Utilisation d'une file d'attente thread-safe pour les messages Kafka.

Traitement des Messages : Utilisation de threads pour traiter les données en file d'attente.

## **4. Multithreading**

Threads pour Producteur et Consommateur : Assurent une exécution simultanée de la production et de la consommation des données.

Threads de Traitement : Permettent un traitement parallèle des données.

## **5. Dépendances et Conception**

Utilisation d'API Externes : Pour obtenir des données de trafic réalistes.

Robustesse et Extensibilité : Conception visant à rendre le système adaptable et fiable.

## **PARTIE PREVENTION AVEC MODELE LSTM :**

Cette section du projet détaille l'application d'un modèle LSTM (Long Short-Term Memory) pour la prévention des embouteillages, en se concentrant sur le traitement des données de trafic générées par un système de streaming Kafka. L'approche adoptée vise à prédire les ratios de congestion du trafic, un indicateur clé pour la gestion proactive du trafic routier. Les étapes clés du processus sont les suivantes :

## 1 Intégration des Données Kafka

**Récupération des Données** : Les données de trafic sont extraites d'un flux Kafka, où chaque message dans le flux représente un enregistrement de données de trafic, stocké sous forme d'objet JSON.

**Fichier JSON** : Ces données de flux sont ensuite consolidées dans un fichier JSON (traffic\_data.json), facilitant leur traitement ultérieur.

## 2 Prétraitement des Données

**Extraction des Données** : Les ratios de congestion (traffic\_jam\_ratio) sont extraits du fichier JSON et convertis en un tableau numpy.

**Normalisation** : Les données sont normalisées via un MinMaxScaler pour optimiser l'apprentissage du modèle LSTM.

## 3 Préparation des Séquences pour le LSTM

**Création de Séquences** : Des séquences de données sont générées, incorporant un nombre défini (look\_back) de mesures temporelles antérieures pour prédire le ratio de congestion suivant.

**Redimensionnement** : Les séquences sont redimensionnées pour correspondre à l'input du modèle LSTM.

## 4 Division en Ensembles d'Apprentissage et de Test

Les données sont divisées en ensembles d'apprentissage et de test, permettant une évaluation objective de la performance du modèle.

## 5 Construction et Entraînement du Modèle

**Architecture du Modèle** : Le modèle se compose de couches LSTM, adaptées au traitement de séquences temporelles, suivies d'une couche Dense pour la sortie prédictive.

**Processus d'Entraînement** : Le modèle est entraîné sur l'ensemble d'apprentissage avec une fonction de perte spécifique et un optimiseur.

## 6 Évaluation et Performance

Le modèle est évalué sur les ensembles d'apprentissage et de test pour mesurer sa précision dans la prédiction des ratios de congestion du trafic.

## CONCLUSION DU RAPPORT

Ce rapport présente un prototype innovant du Smart Traffic Control Hub, une initiative ambitieuse pour transformer la gestion du trafic dans une ville intelligente. Le projet, encadré par Geomatic & GeoIT et réalisé par SIGINOV, a été conçu pour intégrer des technologies avancées et des méthodologies de pointe en matière de traitement et d'analyse des données de trafic.