



LA SIMILITUDE ENTRE L'IOT ET L'INTELLIGENCE ARTIFICIELLE

smart city:
prédiction de trafic

Sous la supervision de:
Mme RAFALIA Najat

Réalisé par :
RAJI Fatima Zahraa
ZIYAD Mouhssine

REMERCIEMENT

Tout d'abord nous tenons à remercier dans un premier temps, toute l'équipe pédagogique de notre université et les intervenants professionnels responsables de la formation technique de ses étudiants, et spécialement du master Big Data et Cloud Computing responsable de notre formation intéressante. Cela étant, Nous présentons nos remerciements les plus considérables à notre professeur : madame **RAFALIA Najat** pour sa motivation, ses conseils, ses qualités techniques et professionnelles et pour tous les efforts fournis favorisant l'aboutissement du de ce mini-projet que nous menons.

TABLE DES FIGURES

Figure 1:schema d'IA.....	8
Figure 2 : ANACONDA	14
Figure 3 : Python	14
Figure 4 : Visual Studio Code	15
Figure 5 : Bibliotheque importée	15
Figure 6:KNN	18
Figure 7:Regression Lineaire	19
Figure 8:SVM	19
Figure 9:LSTM état de warmup.....	20
Figure 10:entrainement du LSTM	21
Figure 11:importation du dataset.....	23
Figure 12:information sur dataset	24
Figure 13:suppression des colones non utile	25
Figure 14:inspection du data	25
Figure 15:fonction du changement du l interval du temps	26
Figure 16:graphe périodique	26
Figure 17:fonction de normalisation	26
Figure 18:screen de division du data	27
Figure 19:Algorithmes utilisés	27
Figure 20:accuracy calculé	27
Figure 21:réalisation du modèle LSTM	28
Figure 22:accuracy du LSTM	28
Figure 23:test du modèle LSTM	29

Figure 24:tableau comparatif entre les vrai données et prediction.....	30
Figure 25:partie du code de la simulation	30
Figure 26:fonction exportInsert	31
Figure 27:screen de la simulation	31

TABLE OF CONTENTS

Introduction	6
Chapitre 1 : Contexte de mini projet	7
1. Intelligence artificielle :	7
1.1. Type d'intelligence ARTIFICIELLE :	8
1.2. Forme d'intelligence artificielle	8
2. Internet des Objets :	9
2.1. Fonctionnement de l'IOT	10
2.2. la rencontre de L'IOT et l'IA	10
3. Application :	11
4. L'IA et l'IOT dans la prédiction des données :	12
Chapitre 2 : OUTILS et algorithmes	14
1. Outils de travail :	14
2. Bibliothèque utilisée :	15
3. Algorithmes utilisés	18
Chapitre 3 : IMPLEMENTATION	22
1. Objectif :	22
2. Dataset :	22
3. Préparation de jeux de données :	24
4. Réalisation du modèle :	27
5. Test et validation :	29
6. Simulation :	30
Conclusion :	32
Reference :	33

INTRODUCTION

L'Internet des objets (IoT) et l'intelligence artificielle (IA) sont deux sujets les plus brûlants de la technologie sont les principaux moteurs de la transformation numérique, offrant au monde une meilleure façon de vivre et de gérer les entreprises.

L'Internet des objets (IoT) est une technologie qui différencie et nous aide à reconstruire notre quotidien, mais l'intelligence artificielle (IA) est devenue le véritable moteur du plein potentiel de l'Internet des objets. Leur combinaison offre un potentiel énorme, l'IA apporte ses capacités (ML/Deep Learning) aux objets, et l'IoT apporte à l'IA la connectivité, la signalisation et l'échange de données. Il existe pas mal de segments majeurs dans lesquels l'AIoT aura de plus en plus un impact significatif prenons le cas des villes intelligentes qui souffrent de l'embouteillage, mais grâce à ces deux technologies l'utilisateur peut avoir une prédiction préalable sur le Trafic et qui serait le cas de notre mini projet. En résumé, les constructions de ce rapport se composent de trois chapitres principales, le premier chapitre consiste sur le contexte du mini projet qui explique l'IOT et l'IA et puis le résultat de leur fusion, ensuite passons au 2-ème chapitre qui représente les outils, les bibliothèques et les algorithmes utilisés dans notre mini projet, et finalement le dernier chapitre qui entame la partie pratique ou bien implémentation du travail.

CHAPITRE 1 : CONTEXTE DE MINI PROJET

Ce premier chapitre aura comme contenu une présentation du mini-projet, « la similitude entre L'IOT et l'IA » notre sujet contient comme notion deux sujets les plus brûlants de la technologie.

1. INTELLIGENCE ARTIFICIELLE :

On peut définir l'intelligence artificielle comme un système ou un ensemble de machines qui accomplissent les tâches qui doivent être accomplies en imitant l'intelligence humaine, et qui peuvent continuellement s'améliorer à la lumière des informations qu'ils collectent.

Qu'il s'agisse d'une tâche simple ou complexe, l'IA vise à apprendre, raisonner et percevoir les données. Il peut facilement imiter la tâche et l'exécuter. Il comprend également des concepts tels que Deep Learning et Machine Learning pour s'adapter facilement à de nouveaux scénarios.

L'intelligence artificielle s'utilise en différents domaines citons : l'Automobile, banque-finance, logistique, énergie, industrie... Aucun secteur d'activité n'est épargné par la montée en puissance de l'intelligence artificielle.

La révolution actuelle de l'intelligence artificielle et de la science qui en découle est rendue possible par « une combinaison de 3 facteurs ».

Selon Harry Shum : « **une vaste quantité de data une puissance informatique extraordinaire, notamment grâce au cloud et des algorithmes révolutionnaires, basés sur le deep-learning** »

Exemples d'IA : Voitures autonomes, filtres anti-spam, Robo advisors, Siri, Alexa, les recommandations de Netflix, les bots conversationnels, les assistants intelligents, etc.

1.1. TYPE D'INTELLIGENCE ARTIFICIELLE :

Il existe trois types d'IA : l'intelligence artificielle étroite (ANI), l'intelligence artificielle générale (AGI) et la super intelligence artificielle (ASI).

L'intelligence artificielle étroite : l'ANI est classée dans la catégorie d'intelligence artificielle faible car elle n'est spécialisée que dans une gamme étroite de paramètres ou de situations, comme la reconnaissance vocale ou les voitures sans conducteur, par exemple.

L'intelligence artificielle générale : l'AGI est considérée comme une intelligence artificielle forte, car elle travaille à un niveau supérieur, qui correspond à l'intelligence humaine.

La super intelligence artificielle : bien que ce type d'IA ne soit actuellement pas développé, l'ASI signifie qu'une machine dispose d'une super intelligence ou qu'elle est plus intelligente qu'un humain.

1.2. FORME D'INTELLIGENCE ARTIFICIELLE

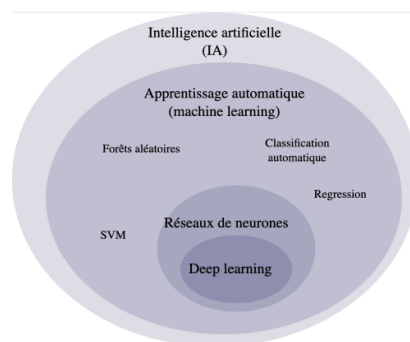


Figure 1:schema d'IA

Il existe trois grandes formes d'IA utilisées aujourd'hui, qui sont les suivantes :

L'IA simple ou basée sur des règles est un logiciel qui a des règles ou des politiques qui relient les événements déclencheurs aux actions. Ces règles sont programmées, de sorte que certaines personnes pourraient ne pas reconnaître cela comme une forme d'IA. Cependant, de nombreuses plateformes d'IA s'appuient sur cette stratégie.

L'apprentissage automatique (ML) est une forme d'IA où l'application apprend le comportement plutôt que de le programmer. L'apprentissage peut prendre la forme de la

surveillance d'un système en direct et de la mise en relation des réponses humaines aux événements, puis de leur répétition lorsque les mêmes conditions se produisent, en soit en analysant les comportements passés, soit en demandant à un expert de fournir les données .

L'inférence ou les réseaux de neurones utilisent l'IA pour construire un "moteur" conçu pour imiter un cerveau biologique simple et faire des déductions qui génèrent des réponses aux déclencheurs en fonction de ce que le moteur "infère" des conditions. Aujourd'hui, cette technologie est appliquée le plus souvent à l'analyse d'images et aux analyses complexes.

2. INTERNET DES OBJETS :

L'IoT, l'Internet of Things ou Internet des objets, désigne le processus de connexion d'objets physiques à Internet, des objets du quotidien tels que les ampoules, aux dispositifs médicaux, appareils portables, appareils intelligents ou encore feux de circulation routière dans les villes intelligentes. « l'IoT » consiste à utiliser les données issues « d'Objets Connectés » pour le piloter ou l'aider à se piloter. On appelle « Objet Connecté » quelque chose ou quelqu'un sur lequel sont placés des capteurs qui remontent des données vers une « Application » qui les utilise.

Cette notion d'internet des objets peut être expliquée grâce au concept de « Service oriented Architecture » décomposé en quatre couches différentes appelées layer. Nous vous l'expliquons à l'aide du tableau ci-dessous :

SENSING	NETWORKING	SERVICE	INTERFACE
Les "hardwares" physiques comme les capteurs intelligents	Connecte les équipements entre eux	Les technologies du "middleware" qui permet de faire communiquer entre eux "hardware" et "software"	Les plateformes qui présentent les applications aux utilisateurs finaux

2.1. FONCTIONNEMENT DE L'IOT

La capacité d'envoyer et/ou de recevoir des informations rend les objets intelligents. Ils n'ont pas besoin d'un stockage important ni d'un superordinateur interne mais d'une connexion à l'une de ces deux capacités. On peut classer ces objets connectés en en trois catégories :

- **Les capteurs qui collectent des informations :**

Les capteurs qui collectent des informations sur l'environnement (température, mouvement, lumière, etc.) et les partagent. Ainsi, dans les exploitations agricoles, ces informations permettent par exemple d'arroser de manière équilibrée, et ainsi d'optimiser les récoltes et d'éviter tout gaspillage.

- **Les capteurs qui collectent et aussi, qui reçoivent les informations :**

Dans le cas de l'agriculture, le système d'irrigation peut alors être lancé automatiquement en fonction des besoins évalués, selon les informations transmises par les capteurs (le terrain est assez humide, ou la météo indique qu'il pleuvra demain, etc.). Ces données peuvent aussi être envoyées à de superordinateurs qui exécutent des algorithmes capables de créer des analyses visuelles sur la façon d'optimiser les pratiques agricoles.

- **Les objets qui reçoivent des informations et agissent en conséquence :**

Telles qu'une imprimante qui reçoit l'ordre d'imprimer ou encore une voiture qui s'ouvre grâce au signal des clefs.

2.2. LA RENCONTRE DE L'IOT ET L'IA

La fusion AI+IoT offrira un potentiel énorme, car elle continue de repousser les limites du traitement des données et de l'apprentissage intelligent pour les années à venir. L'innovation AIoT ne fait que s'accélérer et promet de nous conduire vers un avenir plus connecté et cela générera plus d'innovation et de valeur à nos organisations.

L'intelligence artificielle (AI) et l'internet des objets (IoT) sont deux des composantes stratégiques pour la transition vers L'industrie 4.0. En effet l'internet des objets permet

d'accumuler des données de différents actifs qu'une organisation gère (équipements manufacturiers, bâtiments, produits connectés, tâches humaines, etc.) et même d'interagir avec ces actifs alors que l'intelligence artificielle permet d'apprendre de ces données et de prendre des décisions plus rapides.

L'intelligence artificielle des objets ou IAoT (Artificial Intelligence of Things) combine les technologies de l'intelligence artificielle (IA) et l'infrastructure de l'Internet des objets (IoT, Internet of Things) : le fonctionnement des objets connectés est plus efficace, les interactions homme-machine s'améliorent, et la gestion des données et l'analytique sont plus poussées.

3. APPLICATION :

Il existe quatre segments majeurs dans lesquels l'IAoT aura de plus en plus un impact significatif :

APPAREILS PORTABLES

Les appareils portables tels que les montres intelligentes surveillent et suivent en permanence les préférences, les habitudes, la localisation et les comportements des utilisateurs. Cela a non seulement conduit à des applications percutantes dans le secteur des technologies de la santé, mais fonctionne également bien pour le sport, la santé, la sécurité des travailleurs, etc.

BATIMENTS INTELLIGENTS

Les maisons qui répondent à chacune de vos demandes ne sont plus de la science-fiction. Les maisons intelligentes peuvent tirer parti des appareils électroménagers, de l'éclairage, des appareils électroniques et plus encore, en apprenant les habitudes des propriétaires et en développant une « assistance » automatisée. Il en est de même avec les bâtiments (tours à bureaux, bâtiments industriels ou gouvernementaux). Cet accès transparent apporte également des avantages supplémentaires d'une efficacité énergétique améliorée.

VILLE INTELLIGENTE

Alors que de plus en plus de personnes affluent des zones rurales vers les zones urbaines, les villes évoluent vers des lieux de vie plus sécuritaire et plus pratiques. Les innovations des villes

intelligentes suivent le rythme, avec des investissements destinés à améliorer la sécurité publique, les transports et l'efficacité énergétique. Les applications pratiques de l'IA dans le contrôle du trafic deviennent déjà claires. À New Delhi, où se trouvent certaines des routes les plus encombrées au monde, un système de gestion intelligente des transports (ITMS) est utilisé pour prendre des « décisions dynamiques » en temps réel sur les flux de trafic.

INDUSTRIE INTELLIGENTE

Enfin, les industries, de l'exploitation minière à la fabrication et assemblage, en passant par la transformation alimentaire, s'appuient sur l'AIoT pour devenir plus efficaces et réduire les erreurs humaines.

De l'analyse de données en temps réel aux capteurs de la chaîne d'approvisionnement, les appareils intelligents aident à prévenir les erreurs coûteuses dans l'industrie.

4. L'IA ET L'IOT DANS LA PREDICTION DES DONNEES :

La prévision des données est vitale pour chaque secteur. En prédisant les données, les applications d'IA peuvent permettre à l'IoT d'éviter les temps d'arrêt, d'améliorer l'efficacité opérationnelle, d'augmenter la gestion des risques et de créer de nouveaux services.

La prévision des données, en revanche, signifie hiérarchiser les risques. Mettre l'IA et l'IoT dans un seul conteneur identifie efficacement la menace, vous permettant de repérer les logiciels malveillants sur n'importe quel réseau, de détecter les intrusions et de guider la réponse. De cette façon, vous pouvez prendre de meilleures décisions commerciales en temps réel sans compromettre la sécurité.

CONCLUSION :

L'AIoT s'appuie sur 2 technologies émergentes clés :

L'INTERNET DES OBJETS (IOT): grâce à une technologie embarquée sur des équipements ou appareils, (capteurs, actionneurs, puces RFID...), l'IoT permet d'identifier et faire communiquer entre eux, tous les maillons des chaînes de valeur (machines, produits en cours

de fabrication, finis et en cours d'utilisation, collaborateurs, fournisseurs, clients, infrastructures...), que l'on peut désigner comme des « objets ».

L'INTELLIGENCE ARTIFICIELLE (IA): Fonctions et systèmes programmables qui permettent aux appareils d'apprendre, de raisonner et de traiter des informations comme les humains.

Les appareils interconnectés transforment la façon dont nous interagissons avec nos appareils à la maison et au travail, créant ainsi l'AIoT («Artificial Intelligence of Things»).

CHAPITRE 2 : OUTILS ET ALGORITHMES

En cette phase nous indiquons les différents outils, algorithmes et bibliothèques utilisés pour la réalisation de ce mini-projet :

1. OUTILS DE TRAVAIL :

- ANACONDA



Figure 2 : ANACONDA

Navigateur ANACONDA Anaconda est une distribution libre et open source des langages de programmation Python et R appliqué au développement d'applications dédiées à la science des données et à l'apprentissage automatique (traitement de données à grande échelle, analyse prédictive, calcul scientifique), qui vise à simplifier la gestion des paquets et de déploiement. Les versions de paquetages sont gérées par le système de gestion de paquets conda.

- PYTHON:



Figure 3 : Python

Python est le langage de programmation open source le plus employé par les informaticiens. Ce langage s'est propulsé en tête de la gestion d'infrastructure, d'analyse de données ou dans

le domaine du développement de logiciels. Ainsi, développer du code avec Python est plus rapide qu'avec d'autres langages.

- **VISUAL STUDIO CODE:**



Figure 4 : Visual Studio Code

Visual Studio Code est un éditeur de code source léger mais puissant qui s'exécute sur votre bureau. Il est livré avec un support intégré pour JavaScript, TypeScript et Node.js et dispose d'un riche écosystème d'extensions pour d'autres langages (Python, PHP, Java...) et moteurs d'exécution (tels que .NET et Unity).

2. BIBLIOTHEQUE UTILISEE :

```
import os
from datetime import datetime
import IPython
import IPython.display
#matplotlib seaborn
import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns
#ray modin.pandas
import numpy as np
import ray
import modin.pandas as pd
ray.init(runtime_env={'env_vars': {'__MODIN_AUTOIMPORT_PANDAS__': '1'}})
#sklearn
from sklearn.linear_model import LinearRegression
from sklearn.svm import SVR
from sklearn.neighbors import KNeighborsRegressor
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
#tensorflow
import tensorflow as tf
```

Figure 5 : Bibliotheque importée

- **PANDAS :**

La bibliothèque logicielle open-source Pandas est spécifiquement conçue pour la manipulation et l'analyse de données en langage Python. Elle est à la fois performante,

flexible et simple d'utilisation. Grâce à Pandas, le langage Python permet enfin de charger, d'aligner, de manipuler ou encore de fusionner des données.

Le fonctionnement de Pandas repose sur les « DataFrames », elle permet d'importer et d'exporter les données dans différents formats comme CSV ou JSON. Par ailleurs, Pandas offre aussi des fonctionnalités de Data Cleaning.

- **MODIN.PANDAS :**

Modin est un remplacement direct pour les pandas il utilise Ray , Dask ou Unidist pour fournir un moyen simple d'accélérer les cahiers, scripts et bibliothèques pandas. Contrairement aux autres bibliothèques DataFrame distribuées, Modin offre une intégration et une compatibilité transparentes avec le code pandas existant, Modin fonctionne particulièrement bien sur des ensembles de données plus volumineux, où les pandas deviennent douloureusement lents ou manquent de mémoire .

- **SCIKIT-LEARN :**

Encore appelé sklearn, est la bibliothèque la plus puissante et la plus robuste pour la machine learning en Python. Elle fournit une sélection d'outils efficaces pour l'apprentissage automatique et la modélisation statistique, notamment la classification, la régression et le clustering via une interface cohérente en Python. Cette bibliothèque, qui est en grande partie écrite en Python, s'appuie sur NumPy, SciPy et Matplotlib.

- **NUMPY :**

Le terme NumPy est en fait l'abréviation de " Numerical Python ". Il s'agit d'une bibliothèque Open Source en langage Python. On utilise cet outil pour la programmation scientifique en Python, et notamment pour la programmation en Data Science, pour l'ingénierie, les mathématiques ou la science. Cette bibliothèque est très utile pour effectuer des opérations mathématiques et statistiques en Python. Elle fonctionne à merveille pour la multiplication de matrices ou de tableaux multidimensionnels. L'intégration avec C/C++ et Fortran est très facile.

- **RAY :**

Ray est un framework unifié pour la mise à l'échelle des applications IA et Python Ray se compose d'un environnement d'exécution distribué principal et d'une boîte à outils de bibliothèques (Ray AIR) pour simplifier le calcul ML .

Avec Ray, on peut facilement faire évoluer le même code d'un ordinateur portable vers un cluster. Ray est conçu pour être polyvalent, ce qui signifie qu'il peut exécuter de manière performante tout type de charge de travail. Si l'application est écrite en Python, on peut la mettre à l'échelle avec Ray, aucune autre infrastructure n'est requise.

- **TENSORFLOW :**

TensorFlow est une bibliothèque logicielle open source pour le calcul numérique haute performance. Son architecture flexible permet un déploiement facile du calcul sur une variété de plates-formes (CPU, GPU, TPU) et des ordinateurs de bureau aux clusters de serveurs en passant par les appareils mobiles et périphériques.

Développé à l'origine par des chercheurs et des ingénieurs de l'équipe Google Brain au sein de l'organisation IA de Google, il s'accompagne d'un solide support pour l'apprentissage automatique et l'apprentissage en profondeur, et le noyau de calcul numérique flexible est utilisé dans de nombreux autres domaines scientifiques. TensorFlow est sous licence Apache 2.0.

- **SEABORN :**

Seaborn est une bibliothèque qui offre la possibilité de résumer et de visualiser des données. Elle permet de créer de jolis graphiques statistiques en Python. Cette bibliothèque apporte des fonctionnalités inédites qui favorisent l'exploration et la compréhension des données. Son interface utilise de fonctions intuitives qui assurent notamment la cartographie sémantique et aident à la conversion des données en graphiques statistiques à visualiser. Il convient de voir Seaborn comme un complément de la bibliothèque principale de visualisation de données en Python et non comme un remplaçant.

- **MATPLOTLIB :**

Matplotlib est une bibliothèque Python capable de produire des graphes de qualité. matplotlib peut être utilisé dans des scripts Python, le shell Python et IPython, le notebook Jupyter, des serveurs d'application web et dans quatre outils d'interface graphique. 27 Matplotlib essay de rendre les tâches simples "simples" et de rendre possible les choses compliquées. Vous pouvez générer des graphes, histogrammes, des spectres de puissance (lié à la transformée de Fourier), des graphiques à barres, des graphiques d'erreur, des nuages de dispersion, etc... en quelques lignes de code. Pour des graphiques simples, le module matplotlib.pyplot fournit une interface comme MATLAB spécialement adaptée avec IPython.

- **PYGAME :**

Pygame est un logiciel python multiplateforme spécialement conçu pour la conception de jeux vidéo. Il comprend également des graphiques, des visuels et des sons qui peuvent être utilisés pour améliorer le jeu en cours de conception. Il contient diverses bibliothèques qui fonctionnent avec des images et des sons et peuvent créer des graphiques pour les jeux. Il simplifie l'ensemble du processus de conception de jeux et facilite la tâche des débutants qui souhaitent développer des jeux.

3. ALGORITHMES UTILISES

- **KNN:**

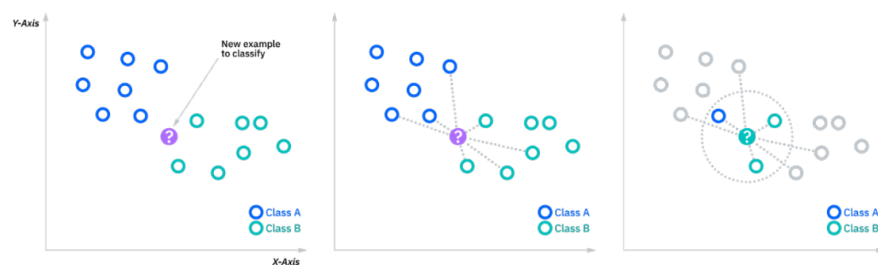


Figure 6:KNN

L'algorithme des k plus proches voisins, également connu sous le nom de KNN ou k-NN, est un discriminant d'apprentissage supervisé non paramétrique, qui utilise la proximité pour effectuer des classifications ou des prédictions sur le regroupement d'un point de données individuel.

Cet algorithme se base sur le paramètre k qui joue un rôle important dans la prédiction et classification et les distance calculées (euclidienne, Manhattan ..).

- **RÉGRESSION LINÉAIRE :**

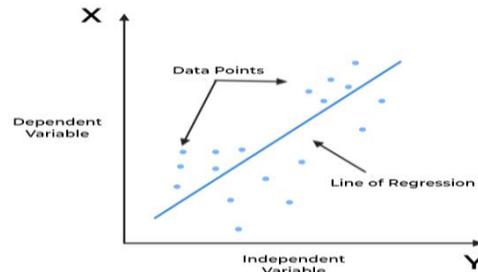


Figure 7:Regression Lineaire

La Régression Linéaire est utilisée pour prédire la valeur d'une variable dépendante basé sur la valeur d'une variable indépendante. Il s'agirait par exemple de prédire les ventes annuelles d'un commercial en fonction de son niveau d'études ou de son expérience. On utilise ces estimations de régression pour expliquer les relations entre variable dépendante et une ou plusieurs variables indépendantes. La forme la plus simple de l'équation de régression avec une variable dépendante et une variable indépendante est définie par la formule $y = c + b * x$, avec y = variable dépendante estimé, c = constante, b = coefficient de régression et x = variable indépendante. On parle ici de Régression linéaire simple. Pour la regression linéaire multiple on écrira $y = c + b * x_1 + \dots + n * x_n$ avec x_1 jusqu'à x_n les variables indépendantes et b jusqu'à n les coefficient de regression respectifs des variables.

- **SVM:**

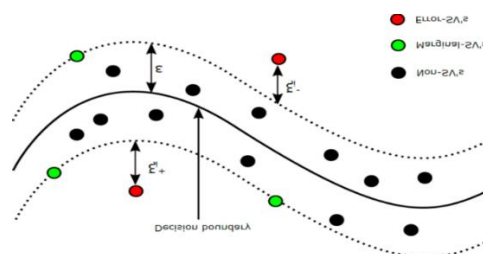


Figure 8:SVM

La machine à vecteurs de support (SVM) est un modèle d'apprentissage supervisé bien connu pour la classification. La régression vectorielle de support (SVR) est une technique de résolution de problèmes de régression analogue à la SVM. SVR est une technique efficace

pour les prédictions d'ensembles de données à valeur réelle, malgré son manque de popularité par rapport à SVM. Le but du SVM est de trouver un hyperplan dans un espace à N dimensions où N est le nombre de caractéristiques qui classent les points de données. Le SVM est une forme étendue du classificateur à marge maximale qui permet une meilleure classification des données dans un espace de grande dimension.

- **RNN :**

Un réseau neuronal récurrent (RNN) est un type de réseau neuronal bien adapté aux données de séries chronologiques. Les RNN traitent une série chronologique étape par étape, en maintenant un état interne d'un pas de temps à l'autre. Dans notre cas, nous avons utilisé une couche RNN appelée Long Short-Term Memory. Un argument de constructeur important pour toutes les couches Keras RNN, telles que `tf.keras.layers.LSTM`, est l'argument `return_sequences`. Ce paramètre peut configurer la couche de l'une des deux manières suivantes :

- Si `FAUX`, la valeur par défaut, la couche ne renvoie que la sortie du dernier pas de temps, ce qui donne au modèle le temps de réchauffer son état interne avant de faire une seule prédiction :

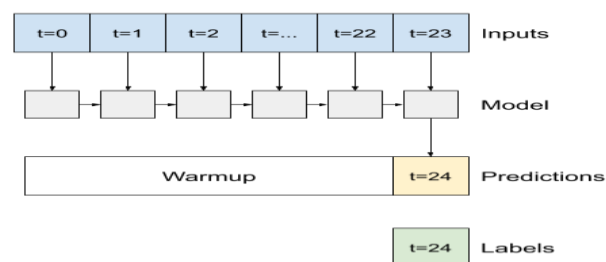


Figure 9: LSTM état de warmup

- Si VRAI, la couche renvoie une sortie pour chaque entrée. Ceci est utile pour:
Empiler les couches RNN.

Entraîner un modèle sur plusieurs pas de temps simultanément.

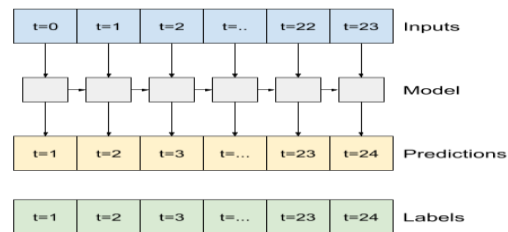


Figure 10:entrainement du LSTM

CHAPITRE 3 : IMPLEMENTATION

Ce chapitre aura comme contenu les différentes étapes suivies pour la réalisation du projet et commençons par l'objectif du mini projet et passons à la démarche suivie :

1. OBJECTIF :

Chaque pays adopte désormais le concept de ville intelligente dans le but d'intégrer les technologies de pointe dans tous les aspects de la vie urbaine. Maintenir le trafic dans les villes est l'une des tâches les plus difficiles et une variété de approches sont utilisées pour y parvenir, de nombreux utilisateurs ont besoin d'avoir une décision préalable sur la route voulu. Par exemple, pendant les jours ouvrables, nous avons besoin d'avoir des informations sur un trafic quotidien ou parfois sur le trafic horaire, mais malgré cela nous tombons dans l'embouteillage.

Pour résoudre ce problème la solution était de combiner l'IOT et l'IA ou cette dernière est basée sur les objets (IoT) est proposé pour estimer la circulation des véhicules dans une zone particulière de la ville pour gérer efficacement le trafic. Les données de trafic ont été collectées à l'aide de capteurs de périphérie IoT.

Les données obtenues ont été traitées et analysées. Pour créer des modèles capables de prédire le Trafic de 6h avant ou l'utilisateur pourrait avoir une idée préalable de la route voulu.

2. DATASET :

Le choix de dataset est toujours important pour un bon résultat du côté qualité et quantité des données ou ses derniers ont un impact direct sur l'efficacité du modèle résultant. Pour développer leur capacité à accumuler des connaissances et à prendre des décisions de façon autonome, les machines ont en effet besoin de consommer une grande quantité d'informations, plus celles-ci sont nombreuses et fiables, plus le résultat obtenu sera précis et adapté aux besoins.

```
df=pd.read_csv(r"C:\Users\ZIYAD\Desktop\IOT project\TRRR\trafficData02112014.csv")
df
```

✓ 29.3s

	status	avgMeasuredTime	avgSpeed	extID	medianMeasuredTime	TIMESTAMP	vehicleCount	_id	REPORT_ID
0	OK	66	56	668	66	2014-02-13T11:30:00	7	190000	158324
1	OK	69	53	668	69	2014-02-13T11:35:00	5	190449	158324
2	OK	69	53	668	69	2014-02-13T11:40:00	6	190898	158324
3	OK	70	52	668	70	2014-02-13T11:45:00	3	191347	158324
4	OK	64	57	668	64	2014-02-13T11:50:00	6	191796	158324
...
25097088	OK	1798	2	623	1798	2014-11-13T10:20:00	0	32507360	210199
25097089	OK	1798	2	623	1798	2014-11-13T10:30:00	0	32507801	210199
25097090	OK	1798	2	623	1798	2014-11-13T10:35:00	0	32508244	210199
25097091	OK	1798	2	623	1798	2014-11-13T10:40:00	0	32508648	210199
25097092	OK	1798	2	623	1798	2014-11-13T10:45:00	0	32509519	210199

25097093 rows x 9 columns

Figure 11:importation du dataset

AVGMEASURED TIME : le temps écoulé lors du passage des véhicules selon leur vitesse calculée par moyenne.

AVGSPEED : la Vitesse Moyenne des véhicules passée.

EXTID : id de chaque boulevard.

MEDIANMEASURED TIME : le temps écoulé lors du passage des véhicules selon leur vitesse calculée par la médiane.

TIMESTAMP : la date ainsi que l'heure de chaque enregistrement incrémenté par 5 min

VEHICLECOUNT : le nombre de véhicules passés

_ID : clé primaire de notre data

REPORT_ID : id du capteur (camera) trouvé dans chaque boulevard

Comme est cité dans l'image ci-dessus, notre dataset se constitue de 9 colonnes (8 features et 1 Target), ou nous avons considéré que « avgSpeed » autant que Target vu que les autres colonnes ont une influence sur cette colonne et va aider à prédire si il y aura de l'embouteillage ou non.

3. PREPARATION DE JEUX DE DONNEES :

Cette phase se considère la plus importante pour obtenir un bon résultat, Un modèle d'apprentissage réussi passe avant tout par des données de qualité , il est donc nécessaire de prétraiter les données recueillies afin d'en extraire tout le potentiel. Données mal annotées, data non disponibles, doublons, informations incohérentes ou superflues... L'intégration des données peut engendrer un certain nombre de complications au sein de votre entrepôt de données. Cette étape vise donc à nettoyer et à normaliser (rendre comparables) les données brutes , voir a les améliorer et Rendre ce type de données cohérentes et exploitables par les algorithmes.

- **INSPECTER ET NETTOYER**

```
df.info()
✓ 0.1s

<class 'modin.pandas.dataframe.DataFrame'>
RangeIndex: 32075 entries, 0 to 32074
Data columns (total 8 columns):
#   Column                Non-Null Count  Dtype
---  ---                ---
0   avgMeasuredTime        32075 non-null  int64
1   avgSpeed               32075 non-null  int64
2   extID                 32075 non-null  int64
3   medianMeasuredTime     32075 non-null  int64
4   TIMESTAMP             32075 non-null  object
5   vehicleCount           32075 non-null  int64
6   _id                   32075 non-null  int64
7   REPORT_ID             32075 non-null  int64
dtypes: int64(7), object(1)
memory usage: 2.0 MB

UserWarning: Distributing <class 'int'> object. This may take some time.
```

Figure 12:information sur dataset


```
#df=pd.read_csv("183063.csv")
#date_time = pd.to_datetime(df.pop('TIMESTAMP'), format='%Y-%m-%dT%H:%M:%S')
df.drop(columns=['extID','REPORT_ID','_id','Unnamed: 0','Min','status','medianMeasuredTime'],inplace=True)
df.head()
```

	avgMeasuredTime	avgSpeed	TIMESTAMP	vehicleCount	M	D	H
0	86	48	2014-02-13T11:30:00	4	2	13	11
1	91	45	2014-02-13T11:35:00	3	2	13	11
2	105	39	2014-02-13T11:40:00	7	2	13	11
3	112	36	2014-02-13T11:45:00	10	2	13	11
4	97	42	2014-02-13T11:50:00	8	2	13	11

Figure 13:suppression des colonnes non utile

En cette phase nous avons supprimer les colonnes non utiles pour notre modele et qui n auront aucune influence sur le reultats.

```
df.describe().transpose()
```

✓ 0.4s

	count	mean	std	min	25%	50%	75%	max
avgMeasuredTime	32075.0	6.284221e+01	1.731865e+01	0.0	55.0	61.0	67.0	275.0
avgSpeed	32075.0	6.110376e+01	1.144311e+01	0.0	55.0	60.0	67.0	132.0
extID	32075.0	6.680000e+02	0.000000e+00	668.0	668.0	668.0	668.0	668.0
medianMeasuredTime	32075.0	6.284221e+01	1.731865e+01	0.0	55.0	61.0	67.0	275.0
vehicleCount	32075.0	4.303944e+00	5.162599e+00	0.0	0.0	2.0	7.0	36.0
_id	32075.0	7.290175e+06	4.101691e+06	190000.0	3737176.5	7286405.0	10846055.5	14353801.0
REPORT_ID	32075.0	1.583240e+05	0.000000e+00	158324.0	158324.0	158324.0	158324.0	158324.0

Figure 14:inspection du data

Les figures ci-dessus nous permet d'avoir une première vu de notre jeux de données ou ce derniers se constitues de 25 M lignes, tout les types sont des entiers sauf TIMESTAMP qu'il s'agit d'un object ,pour le deuxième screens nous permet de savoir que les valeurs de notre data est logique.

• MANIPULATION DE DONNÉES:

En cette étape, nous avons pris la main à convertir le type de colonnes TIMESTAMP à datetime ainsi sous-échantillonner les données d'intervalles de 5 minutes à des intervalles d'une heure est cela grâce à la fonction apparu ci-dessous et cela dans le but de diminuer la quantité de data qui s'est rétréci de 25M à 2M et aider l'entrainement de nos algorithmes ainsi augmenter leur performance .

```

df3 = pd.DataFrame(columns=df.columns)
#from 5min to 1hour
for m in df['M'].unique():
    for d in df['D'].unique():
        for h in df['H'].unique():
            DDD=df[(df['H'] == h)&(df['D'] == d)&(df['M'] == m)]
            if DDD['avgSpeed'].mean() > 0:
                #print([DDD['avgMeasuredTime'].mean(),DDD['avgSpeed'].mean(),DDD['vehicleCount'].sum(),DDD['M'].mean(),DDD['D'].mean(),DDD['H'].mean()])
                df4=pd.DataFrame([DDD['avgMeasuredTime'].mean(),DDD['avgSpeed'].mean(),DDD['TIMESTAMP'].values[0],DDD['vehicleCount'].sum(),DDD['M'].mean(),DDD['D'].mean(),DDD['H'].mean()],
                                columns=df.columns.tolist())
                df3=pd.concat([df3,df4],ignore_index=True)

#extract datetime from timestamp
date_time = pd.to_datetime(df3.pop('TIMESTAMP'), format='%Y-%m-%dT%H:%M:%S')

```

✓ 19.5s

Figure 15:fonction du changement du l interval du temps

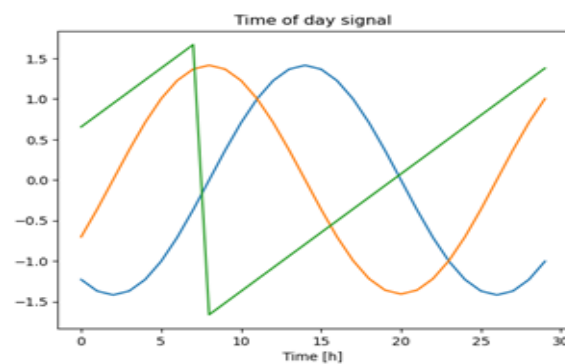


Figure 16:graphe périodique

- NORMALISATION ET DIVISION DU DATA :**

```

#Normalize the data
mean = df.mean()
std = df.std()
df = (df - mean) / std
df.head()

```

✓ 0.0s

	avgMeasuredTime	avgSpeed	vehicleCount	Day sin	Day cos	Year sin	Year cos
0	-0.438201	0.164485	-0.208904	0.183409	-1.395105	0.742746	1.777563
1	0.153914	-0.389065	1.353849	-0.001508	-1.407180	0.743052	1.777070
2	0.528130	-0.665840	2.286459	-0.368176	-1.359085	0.743665	1.776081
3	1.503935	-1.321900	2.639339	-0.709857	-1.218078	0.744277	1.775092
4	0.296021	-0.440320	2.009197	-1.003265	-0.993768	0.744888	1.774102

Figure 17:fonction de normalisation

Il est important de mettre à l'échelle les fonctionnalités avant de former un modèle. La normalisation est une façon courante d'effectuer cette mise à l'échelle : soustrayez la moyenne et divisez par l'écart type de chaque caractéristique. Ces valeurs apparus dans la

figure ci-dessus aide les algorithmes de machine Learning et spécialement les réseaux neurones d'avoir une bonne performance.

```
#Split the data
column_indices = {name: i for i, name in enumerate(df.columns)}

n = len(df)
train_df = df[0:int(n*0.75)]      #75% train
val_df = df[int(n*0.75):int(n*0.9)] #15% validation
test_df = df[int(n*0.9):]         #10% test

num_features = df.shape[1]

✓ 0.0s
```

Figure 18:screen de division du data

La figure ci-dessus montre la répartition de data(75%, 15%, 10%) pour les ensembles d'entraînement, de validation et de test.

4. REALISATION DU MODELE :

```
scoring = 'accuracy'
models= []
models.append(('LR', LinearRegression()))
models.append(('SVM', SVR()))
models.append(('KNN', KNeighborsRegressor(n_neighbors=48)))
results= names= tmodels= []
for name, model in models:
    names.append(name)
    msg = "For %s Model" % (name)
    print(msg)
    clf = model
    clf.fit(x_train, y_train)
    accuracy = clf.score(x_test, y_test)
    print("Test Accuracy:",accuracy)
    tmodels.append(clf)
    print("*****")
✓ 1.1s
```

Figure 19:Algorithmes utilisés

```
For LR Model
Test Accuracy: 0.41672933108977694
*****
For SVM Model
Test Accuracy: 0.4655619170729991
*****
For KNN Model
Test Accuracy: 0.4399653241230149
*****
```

Figure 20:accuracy calculé

Dans notre mini projet nous avons utilisé des algorithmes de machine Learning dédiée a la prédiction (Régression Linéaire ,KNN, SVR).Après leurs entrainement nous avons constaté d'après le screen ci dessus que leur accuracy n'était pas assez satisfaisante (Régression linéaire : 41% ,SVR: 46%, KNN :43%) ,ce qui nous a poussé à utiliser des algorithmes de Deep Learning(réseaux neurones) spécialement le LSTM.

```
lstm_model = tf.keras.models.Sequential([
    tf.keras.layers.LSTM(32, return_sequences=True),
    tf.keras.layers.Dense(units=6)])
✓ 0.0s

history = compile_and_fit(lstm_model, wide_window)
val_performance['LSTM'] = lstm_model.evaluate(wide_window.val)
performance['LSTM'] = lstm_model.evaluate(wide_window.test, verbose=0)
✓ 2m 5.1s

Output exceeds the size limit. Open the full output data in a text editor
Epoch 1/100
117/117 [=====] - 3s 14ms/step - loss: 0.2872 - m
```

Figure 21:réalisation du modèle LSTM

```
Epoch 100/100
117/117 [=====] - 1s 11ms/step - loss: 0.2157 - m
23/23 [=====] - 0s 6ms/step - loss: 1.0663 - mean

performance['LSTM'][1]*100
✓ 0.0s

91.42711162567139
```

Figure 22:accuracy du LSTM

Après la réalisation du modèle LSTM et son entrainement nous remarquons qu'il s'agit du meilleur algorithme pour notre cas (performance de 91%).

5. TEST ET VALIDATION :

Cette dernière étape tend à confronter le modèle à la réalité du terrain. Dans cette phase de test, on se sert de l'autre partie des données, soit la dataset de test. Ce sous-ensemble d'informations affine le modèle grâce aux scénarios ou données que l'ordinateur n'a pas encore expérimentés en phase d'entraînement.

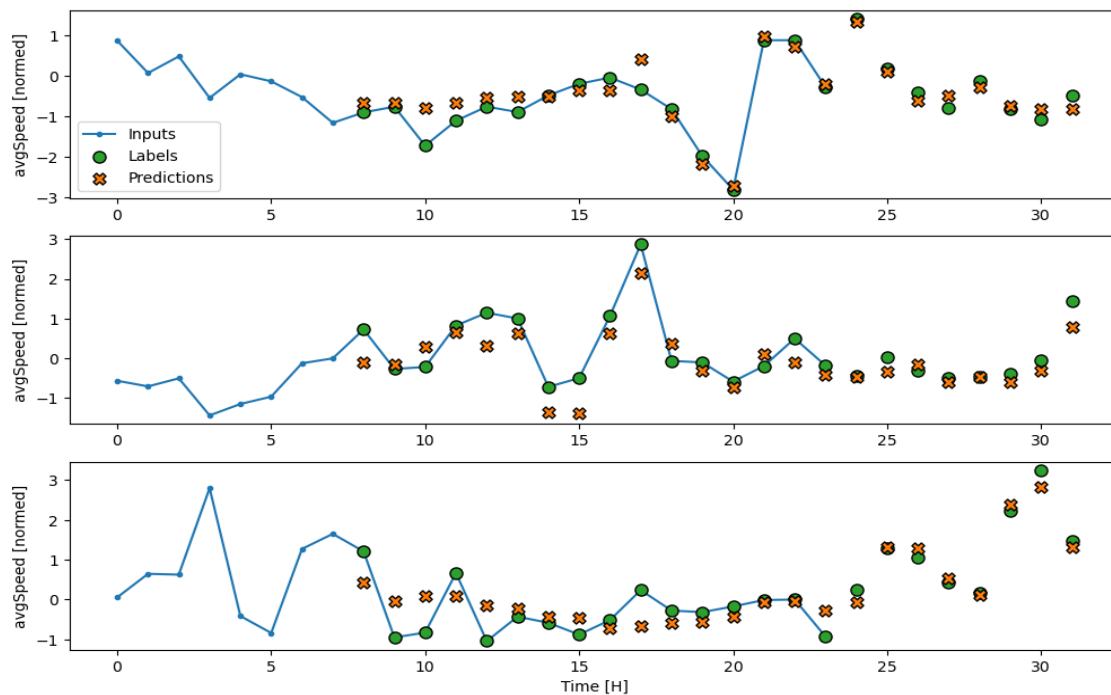


Figure 23: test du modèle LSTM

Cette figure nous montre le parcours du test du modèle LSTM comme c'est apparu ci-dessus le modèle LSTM a besoin est une méthode de warmup pour initialiser son état interne en fonction des entrées. Une fois formé, cet état capturera les parties pertinentes de l'historique des entrées. Et puis prédire ce qui peut arriver avec un minimum taux d'erreur.

6. SIMULATION :

```
print("prediction de",len(pren2),"heure")
pren2.head()
```

✓ 0.0s

prediction de 24 heure

	label	predictions
0	46.166664	48.844486
1	56.750000	46.764557
2	48.666664	44.283428
3	53.250000	45.644321
4	45.333332	44.060692

Figure 24:tableau comparatif entre les vrai données et prediction

Après avoir extraire les données de la phase du test nous l'avons utiliser pour faire une prédiction de 24h Ce qui le montre la figure ci-dessus . c'est un tableau comparatif entre les données du test et la prédiction.

```
from traffic_signal import *
from road import *
from curve import *
from simulation import *
from vehicle import *
from vehicle_generator import *
from window import *
from simulation import Simulation
# Create simulation + data
sim = Simulation(pren2)
# Add multiple roads
sim.create_roads([
    ((-10, 108), (290, 108)),
    ((-10, 104), (290, 104)),

    ((290, 100), (-10, 100)),
    ((290, 96), (80, 96)),
    ((80, 96), (-10, 96)),
])
sim.create_gen({
    'vehicle_rate': 60,
    'vehicles': [
        [3, {"path": [0]}],
        [6, {"path": [1]}],

        [3, {"path": [3, 4]}],
        [6, {"path": [2]}],
    ]
})
# Start simulation
win = Window(sim)
win.offset = (-145, -95)
win.zoom = 8
win.run(steps_per_update=5)
```

Figure 25:partie du code de la simulation

Le screen ci-dessus nous montre la partie de la fonction de la simulation qui permet de créer et configurer la simulation en utilisant les voies créées les générateurs les véhicules.

```
def exportInsert(self):
    if self.i==24:print("end")
    threading.Timer(12.0, self.exportInsert).start()
    self.vr=int((self.df.label.max()-self.df.label[self.i])*2)
    self.prediction=int(self.df.predictions[self.i])
    print("self.vehicle_rate=",self.vr)
    self.i+=1
```

Figure 26:fonction exportInsert

Cette fonction permet de prendre les données déjà traitées de notre modèle chaque 12 seconde qui est l'équivalent de 1h et l'appliquer dans la simulation apparaît dans le figure ci-dessous. Cette fonction est réursive et s'arrête après 24 h vu que notre modèle traite juste la durée d'un jour.

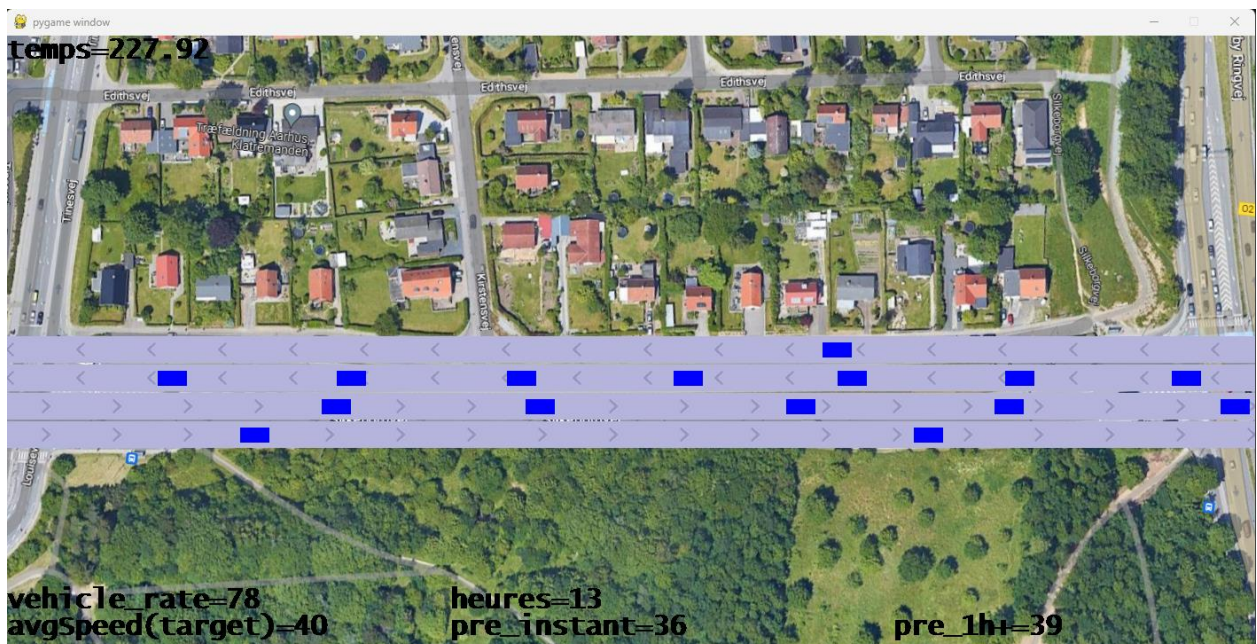


Figure 27:screen de la simulation

Cette figure montre la simulation d'une seule route et affiche l'information tels que le nombre de véhicule passées chaque 1h ainsi leur vitesse ou cette dernière s'incrémente donc y a peu de véhicule qui sont passées et nous trouvons aussi la prédiction de notre modèle a l'instant et aussi d'une heure après.

CONCLUSION :

L'IoT traite des appareils qui interagissent via Internet, l'IA permet aux appareils d'apprendre de leurs données et de leur expérience. Dans les villes intelligentes spécialement dans l'ingénierie des transports, la combinaison de ces 2 technologies offre un des potentiels importants dans la prévision du trafic. De plus, la prédiction basée sur l'heure et sur l'ensemble de donnée des années précédente offert par des capteurs de l'IOT qui sont après traitées et analysées par un système intelligent en utilisant l'algorithme LSTM après avoir tester chacun de régression Linéaire, KNN et le SVR ou leur performance n'est pas satisfaisante. Ce modèle peut être utilisés dans un contrôleur de feux tricolores intelligent, alimenté par des capteurs de trafic qui comptent le nombre de véhicules passant par une voie et leur vitesse à chaque période donnée ; avec ces lectures, une base de données similaire à celle utilisée dans notre projet. Une fois la base de données générée, le système intelligent peut être entraîné pour chaque intersection. Ensuite, le flux de trafic pour la période suivante peut être prédit en utilisant un nombre donné de lectures passées.

Une fois la prédiction faite, elle permettra aux utilisateur d accéder et jeter un coup d'œil sur la route qui veulent utiliser.

REFERENCE :

- <http://iot.ee.surrey.ac.uk:8080/datasets.html#traffic>
- https://link.springer.com/chapter/10.1007/978-981-16-9605-3_68
- <https://ak-tyagi.com/static/pdf/43.pdf>
- <https://uohjs.com/hufatat/2023/01/UOHJS-222-15-Hamdan-Alshehri.pdf>
- <https://www.mdpi.com/2220-9964/11/2/85>
- *Intelligent Traffic Management System Based on the Internet of Vehicles (IoV)*
(hindawi.com)
- <https://actualiteinformatique.fr/internet-of-things-iot/quest-ce-que-iot-internet-of-things-internet-des-objets>
- <https://www.synox.io/actualites-sectorielle/4-choses-a-savoir-sur-linternet-des-objets/#:~:text=Comment%20fonctionne%20l'IoT%20%3F,fil%20sur%20des%20plateformes%20IoT.>
- https://link.springer.com/chapter/10.1007/978-981-16-9605-3_68
- <https://link.springer.com/article/10.1007/s12243-021-00891-7>