Projet Multidisciplinaire

Prédiction des Événements Climatiques Extrêmes à partir de Données Météorologiques Historiques

Filière : Informatique et Ingénierie des Données

Réalisé par :

HOURRI Chaimae
CHEMCHAQ Maryem
BENNISS Zineb
REBBOUH Houda
SOU Abdelmounaim

Encadré par :

Pr. Nidal LAMGHARI	ENSA KHOURIBGA
Pr. Amal OURDOU	ENSA KHOURIBGA
Pr. Nassima SOUSSI	ENSA KHOURIBGA
Pr. Hamza KHALFI	ENSA KHOURIBGA

Institution : École Nationale des Sciences Appliquées de Khouribga
Université : Université Sultan Moulay Slimane

Année universitaire : 2025–2026

Khouribga, Maroc
Janvier 2026

Résumé

Ce projet multidisciplinaire vise à concevoir un système d'aide à la décision pour la prédiction des événements climatiques extrêmes à partir de données météorologiques historiques. L'approche proposée combine des techniques de Big Data, d'apprentissage profond, d'ingénierie des connaissances et de développement web. Une étude de cas portant sur l'évolution de la température à Marrakech sur la période 2010–2024 est réalisée afin d'évaluer la capacité du système à détecter et prédire les anomalies thermiques. Les résultats obtenus montrent la pertinence de l'approche hybride neuro-symbolique pour améliorer la fiabilité et l'explicabilité des prédictions.

Table des matières

Chapitre 1

Introduction

1.1 Contexte général

Le changement climatique constitue l'un des défis majeurs de notre époque. L'augmentation de la fréquence et de l'intensité des événements climatiques extrêmes, tels que les canicules, les précipitations intenses et les sécheresses, engendre des impacts significatifs sur l'environnement, l'agriculture et la santé publique.

1.2 Problématique

La prédiction des événements climatiques extrêmes demeure un problème complexe en raison de la nature non linéaire des phénomènes climatiques et du volume important de données à traiter. Il est donc nécessaire de développer des systèmes intelligents capables d'exploiter efficacement ces données pour anticiper les risques.

1.3 Objectifs du projet

L'objectif principal de ce projet est de développer un système d'aide à la décision permettant de prédire les événements climatiques extrêmes. Les objectifs spécifiques sont :

- Collecter et prétraiter des données météorologiques massives.
- Développer des modèles d'apprentissage profond pour la prédiction d'anomalies.
- Construire une base de connaissances pour l'explicabilité des décisions.
- Développer une interface web interactive pour la visualisation des résultats.

1.4 Organisation du rapport

Ce rapport présente successivement la description du projet, la méthodologie adoptée, la réalisation et l'implémentation, les résultats obtenus, l'analyse et la discussion, ainsi que la conclusion et les perspectives.

Chapitre 2

Présentation Générale du Projet

2.1 Description du projet

Le projet consiste à concevoir un système intelligent capable de prédire des événements climatiques extrêmes à partir de données météorologiques historiques. Il intègre des techniques de Big Data, de Deep Learning, d'ingénierie des connaissances et de technologies web.

2.2 Domaine multidisciplinaire

Ce projet s'inscrit dans un cadre multidisciplinaire incluant :
— L'ingénierie des connaissances.
— L'apprentissage profond et l'intelligence artificielle.
— Le Big Data et les systèmes distribués.
— Le développement web et les technologies JavaScript.

2.3 Périmètre et limites

Le projet se concentre sur la prédiction des événements climatiques extrêmes à partir de données historiques. Les limites concernent la disponibilité des données, les ressources computationnelles et la généralisation du modèle à d'autres régions.

2.4 Zone d'étude et période d'analyse

L'étude de cas porte sur l'évolution de la température dans la ville de Marrakech, au Maroc. Cette région est caractérisée par un climat semi-aride et une forte exposition aux vagues de chaleur, ce qui en fait un cas pertinent pour l'étude des événements climatiques extrêmes.

La période d'analyse couvre quinze années, du 1er janvier 2010 au 31 décembre 2024. Cette période permet d'observer les tendances climatiques à long terme ainsi que les anomalies thermiques potentielles. L'analyse de cette série temporelle constitue un cas d'application du système proposé afin de démontrer sa capacité à détecter et prédire des événements extrêmes.

Chapitre 3

Méthodologie de Travail

3.1 Démarche adoptée

Une démarche scientifique structurée et incrémentale a été adoptée pour mener à bien ce projet. Elle comprend plusieurs phases :

1. Analyse des besoins et définition du cas d'usage spécifique à Marrakech, sur la période 2010–2024.
2. Collecte et prétraitement des données météorologiques brutes.
3. Modélisation et développement des modèles d'apprentissage profond pour la prédiction des événements extrêmes.
4. Construction et intégration d'une base de connaissances (ontologie climatique et moteur de règles) pour soutenir l'explicabilité des prédictions.
5. Développement d'une interface web interactive pour la visualisation des résultats et alertes.
6. Tests, validation et documentation complète des différents modules.

Cette approche permet d'avancer de manière itérative, en validant chaque étape avant de passer à la suivante, tout en assurant l'intégration cohérente de tous les composants du système.

3.2 Étapes du projet

Le projet a été organisé en étapes successives, chacune correspondant à un livrable intermédiaire :

— **Analyse des besoins :** étude des données météorologiques disponibles, définition des objectifs et des événements extrêmes à prédire.
— **Collecte et prétraitement des données :** extraction des données historiques (2010–2024) via API Meteostat, nettoyage, normalisation et stockage dans un Data Lake local.
— **Développement des modèles d'apprentissage profond :** conception et entraînement de modèles de séries temporelles multivariées (LSTM), avec prise en compte du déséquilibre des classes pour les événements rares.
— **Construction de la base de connaissances :** élaboration d'une ontologie représentant les phénomènes climatiques et leurs relations avec les variables météorologiques, ainsi qu'un moteur de règles pour générer des alertes explicites.

— **Développement de l'interface web interactive :** création d'un tableau de bord pour la visualisation des prédictions, des courbes temporelles et des alertes générées.
— **Tests et validation :** vérification de la robustesse du pipeline, évaluation des performances du modèle (Recall, F1-score, RMSE) et validation de l'intégration complète.

3.3 Outils et technologies utilisés

Pour répondre aux exigences multidisciplinaires du projet, un ensemble d'outils et de technologies a été utilisé :

— **Big Data :** Kafka pour l'ingestion des données, Spark (Batch).
— **Deep Learning :** LSTM, GRU et Transformers pour la modélisation des séries temporelles multivariées.
— **Pipeline de données :** Python pour la collecte initiale, Kafka pour l'ingestion et le streaming en temps réel.
— **Interface web :** Node.js pour l'API backend, React / Vue.js / Angular pour le dashboard interactif.
— **Analyse et visualisation :** bibliothèques Python pour la préparation des données et génération de graphiques.

Cette combinaison d'outils permet de gérer efficacement de grandes quantités de données, de construire des modèles prédictifs robustes et d'offrir une interface web interactive pour les utilisateurs finaux.

Chapitre 4

Réalisation et Implémentation

4.1 Description des travaux réalisés

Le projet a été réalisé en suivant une approche intégrée combinant Big Data, Deep Learning, ingénierie des connaissances et développement web. Les principaux travaux réalisés sont les suivants :

- **Mise en place du pipeline de traitement des données :** Un pipeline complet a été conçu pour passer des données brutes météorologiques aux datasets prêts pour le modèle prédictif. Cela inclut la collecte, le nettoyage, le prétraitement et le stockage distribué des données historiques (2010–2024) pour Marrakech.
- **Entraînement des modèles prédictifs :** Des modèles de séries temporelles multivariées (LSTM) ont été développés et entraînés pour détecter et prédire les événements climatiques extrêmes tels que les canicules et vagues de précipitations. Une attention particulière a été portée au déséquilibre des classes, en utilisant des techniques comme Weighted Loss et Focal Loss.
- **Construction du système expert :** Une ontologie climatique a été élaborée pour représenter les phénomènes extrêmes et leurs relations avec les variables météorologiques (température, humidité, pression, vent). Un moteur de règles a été intégré pour générer des alertes explicites et fournir un niveau d'explicabilité aux prédictions du modèle.
- **Développement du tableau de bord interactif :** Une interface web a été construite pour visualiser les prédictions et alertes générées par le système. Les utilisateurs peuvent explorer les courbes temporelles, cartes et notifications, filtrer les données par période et région, et accéder aux résultats en temps réel via l'API backend.
- **Tests et validation :** Chaque module a été testé individuellement et intégré progressivement. Les performances du modèle ont été évaluées avec des métriques adaptées (Recall, F1-score, RMSE) et le pipeline Big Data a été vérifié pour sa robustesse et scalabilité.

4.2 Architecture générale

L'architecture globale du système repose sur une chaîne complète de traitement des données, intégrant plusieurs modules interdépendants :

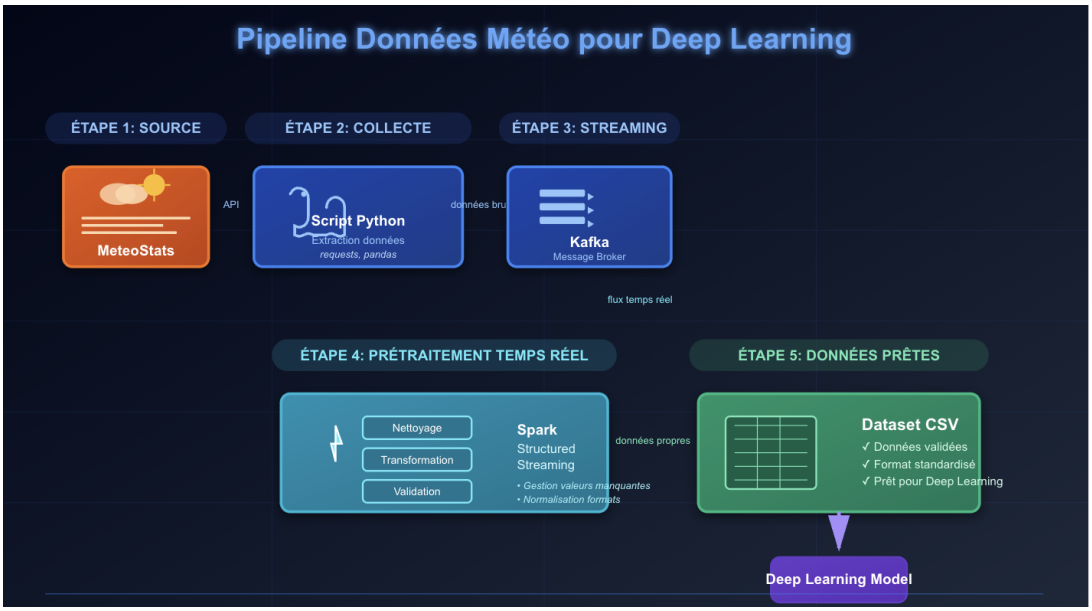

FIGURE 4.1 – Architecture générale du système de prédiction des événements climatiques extrêmes.

4.2.1 Module Big Data

Le module Big Data constitue la **colonne vertébrale** du projet, permettant de collecter, ingérer, traiter et préparer les données météorologiques pour l'apprentissage profond. L'architecture actuelle repose sur :

- un **script Meteostat** pour la collecte des données,
- **Kafka** pour l'ingestion et la distribution des données,
- **Spark** pour le prétraitement et le nettoyage,
- le stockage des datasets structurés prêts à l'usage.

1. Collecte des données

Les données historiques de Marrakech sont récupérées via un **script Python utilisant la bibliothèque Meteostat**. Le script collecte les mesures depuis 2010 jusqu'à 2024 et les sauvegarde sous forme de **CSV brut** :

```
Raw data saved at: data/raw/marrakesh_temperature.csv
```

2. Ingestion des données avec Kafka

Un **Kafka Producer Python** lit le CSV et envoie chaque ligne au topic `weather-marrakesh` sous forme de messages JSON :

```
Sent: {'date': '2024-10-30', 'tavg': 18.3, 'tmin': 13.6, 'tmax': 24.0, 'prcp': 0.0}
Sent: {'date': '2024-10-31', 'tavg': 17.9, 'tmin': 11.6, 'tmax': 24.0, 'prcp': 0.0}
```

Kafka permet de **gérer un flux continu de données**, garantissant **scalabilité et résilience** du pipeline.

	A	B	C	D	E	F	G	H	I	J	K
1	time	tavg	tmin	tmax	prcp	snow	wdir	wspd	wpgt	pres	tsun
2	2010-0...	12.6	9.0	17.4	2.0			7.9			
3	2010-0...	12.3	6.0	20.0	0.0			5.8			
4	2010-0...	14.8		23.4				8.1			
5	2010-0...	14.4	8.3	23.4	0.0						
6	2010-0...	13.9	11.0	17.4	0.0						
7	2010-0...	14.9			0.0			9.4			
8	2010-0...	12.3	9.0	16.7	5.1			19.1			
9	2010-0...	9.2	7.0	13.0	41.9			10.6			
10	2010-0...	8.1	3.9	14.0	4.1			9.5			
11	2010-0...	10.6	3.3	19.0	0.0			9.8			
12	2010-0...	13.4	4.8	19.4	0.0			7.2			
13	2010-0...	12.6	6.6	18.5	0.0			7.1			
14	2010-0...	14.6	7.2	20.0	0.0			5.5			
15	2010-0...	16.1	8.6	20.5	0.0			9.3			
16	2010-0...	13.3			2.0			4.0			
17	2010-0...	15.0	6.0	25.2	0.0			4.6			
18	2010-0...	16.4	5.9	28.2	0.0			6.9			
19	2010-0...	17.6	8.8	27.6	0.0			5.9			
20	2010-0...	15.4			0.0			6.5			
21	2010-0...	14.0		21.2	0.0						
22	2010-0...	13.6	6.8	21.6	0.0			4.7			
23	2010-0...	12.9	6.7	19.2	0.0			8.6			
24	2010-0...	13.9	7.2		0.0			14.3			

+ Sheet1

FIGURE 4.2 – Extrait du CSV brut généré par le script Meteostat.

3. Prétraitement et nettoyage avec Spark

Spark Batch consomme le topic Kafka, parse les messages JSON, et effectue les étapes suivantes :

- Conversion des colonnes en types corrects (Double pour tavg, tmin, tmax, prcp, wspd, pres),
- Transformation de la date en format standard,
- Nettoyage des valeurs manquantes ou aberrantes,
- Préparation des datasets finaux pour le Deep Learning.

3.1 Nettoyage des données Le prétraitement appliqué aux variables météorologiques inclut :

- Remplacement des virgules par des points pour un format décimal standard,
- Suppression des caractères non valides (ex. tirets, symboles),
- Conversion des valeurs en format numérique,
- Élimination des lignes contenant des valeurs manquantes,
- Vérification d'un volume minimal de données (au moins 100 observations) pour garantir la stabilité de l'apprentissage.

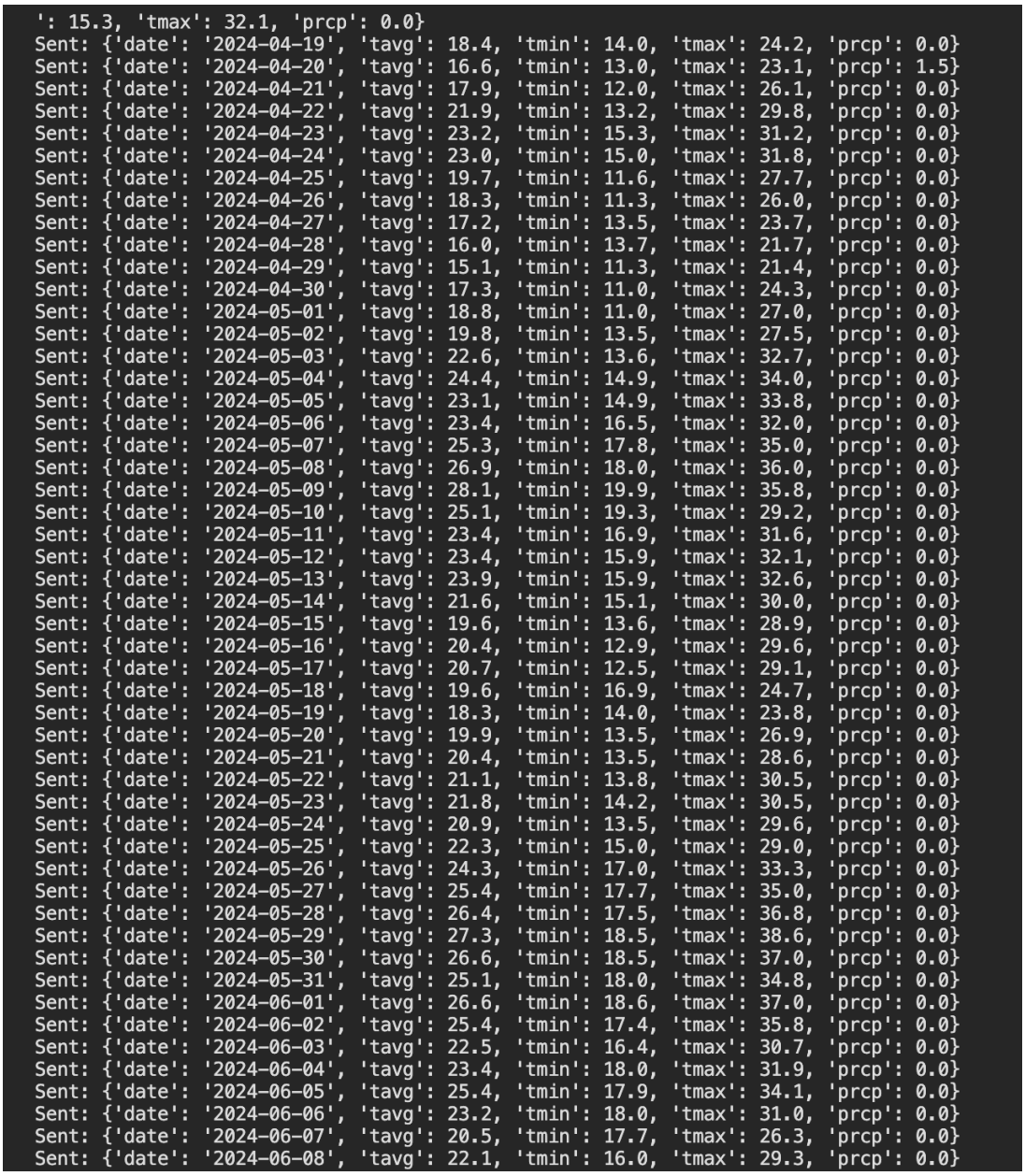

FIGURE 4.3 – Kafka Console montrant les messages JSON envoyés au topic.

3.2 Prise en compte de la saisonnalité La saisonnalité est modélisée à partir du jour de l'année (`dayofyear`) en utilisant une transformation cyclique :

— $\text{sin\_doy} = \sin\left(\frac{2\pi \times \text{jour}}{365}\right)$

— $\text{cos\_doy} = \cos\left(\frac{2\pi \times \text{jour}}{365}\right)$

Cette représentation permet au modèle LSTM de capturer la périodicité des cycles climatiques de manière continue.

4. Stockage des datasets propres

Les données nettoyées sont sauvegardées à la fois en **CSV** pour un accès rapide et compact :

— `data-lake/weather-clean.csv`

Ces datasets propres sont prêts à être utilisés directement par le module Deep Learning.

	A	B	C	D	E	F	G	H
1	**date**	**tavg**	**tmin**	**tmax**	**prcp**	**dayof...**	sin_doy	**cos_doy**
2	2010-0...	12.6	9.0	17.4	2.0	1	0.0172133561558346...	0.99985183920911...
3	2010-0...	12.3	6.0	20.0	0.0	2	0.03442161162274574	0.9994074007397...
4	2010-0...	14.8	NaN	23.4	NaN	3	0.0516196672232537...	0.998666816288476
5	2010-0...	14.4	8.3	23.4	0.0	4	0.06880242680231986	0.9976303053065...
6	2010-0...	13.9	11.0	17.4	0.0	5	0.08596479873744647	0.9962981749346...
7	2010-0...	14.9	NaN	NaN	0.0	6	0.10310169744743485	0.99467081991152...
8	2010-0...	12.3	9.0	16.7	5.1	7	0.1202080448993527	0.9927487224577...
9	2010-0...	9.2	7.0	13.0	41.9	8	0.13727877211326478	0.9905324521322...
10	2010-0...	8.1	3.9	14.0	4.1	9	0.15430882066428117	0.9880226656636...
11	2010-0...	10.6	3.3	19.0	0.0	10	0.1712931441814776	0.9852201067560...
12	2010-0...	13.4	4.8	19.4	0.0	11	0.1882267098432442	0.9821256058680...
13	2010-0...	12.6	6.6	18.5	0.0	12	0.2051044998686192	0.9787400799669...
14	2010-0...	14.6	7.2	20.0	0.0	13	0.2219215130041655	0.9750645322571...
15	2010-0...	16.1	8.6	20.5	0.0	14	0.2386727660059501	0.97110005188295...
16	2010-0...	13.3	NaN	NaN	2.0	15	0.255353295116187	0.9668478136052...
17	2010-0...	15.0	6.0	25.2	0.0	16	0.2719581575341055	0.9623090774541...
18	2010-0...	16.4	5.9	28.2	0.0	17	0.288482432880609	0.9574851883550...
19	2010-0...	17.6	8.8	27.6	0.0	18	0.30492122465628907	0.9523775757303...
20	2010-0...	15.4	NaN	NaN	0.0	19	0.3212696616923644	0.9469877530760...
21	2010-0...	14.0	NaN	21.2	0.0	20	0.3375228995941133	0.9413173175128...
22	2010-0...	13.6	6.8	21.6	0.0	21	0.3536761221763716	0.9353679493131...
23	2010-0...	12.9	6.7	19.2	0.0	22	0.3697245428906731	0.92914141140317...
24	2010-0...	13.9	7.2	NaN	0.0	23	0.38566340624360707	0.9226395488404...

+ Sheet1

FIGURE 4.4 – Extrait du dataset final propre après prétraitement Spark.

5. Pipeline batch

Le pipeline gère le traitement de l'historique complet (2010–2024). Grâce à Kafka, il est possible d'évoluer vers un traitement en temps quasi réel pour intégrer de nouvelles données sans réécrire le batch historique.

6. Rôle de chaque composant

- **Script Meteostat :** collecte des données historiques brutes.
- **Kafka :** ingestion des données en flux, gestion des topics et distribution vers Spark.
- **Spark :** nettoyage, transformation, enrichissement et agrégation des données volumineuses.
- **CSV :** stockage final optimisé pour le traitement par le module Deep Learning.

4.2.2 Module Deep Learning

1. Normalisation des données

L'ensemble des variables météorologiques ainsi que les composantes saisonnières sont normalisées dans l'intervalle $[0, 1]$ à l'aide de l'algorithme *MinMaxScaler*. Cette mise à l'échelle permet d'uniformiser les amplitudes des différentes caractéristiques, d'améliorer la stabilité numérique du modèle et d'accélérer la convergence lors de l'entraînement du réseau de neurones.

2. Création des séquences temporelles

Les données normalisées sont restructurées sous forme de séquences temporelles de longueur 30 jours. Chaque séquence constitue une entrée du modèle et permet de prédire les variables météorologiques du jour suivant.

— X : séquences de 30 jours contenant les variables météorologiques ainsi que les composantes saisonnières.

— y : vecteur des variables météorologiques correspondant au jour à prédire.

Cette représentation séquentielle est particulièrement adaptée aux réseaux de neurones récurrents de type LSTM pour la modélisation des dépendances temporelles.

3.Architecture du modèle LSTM

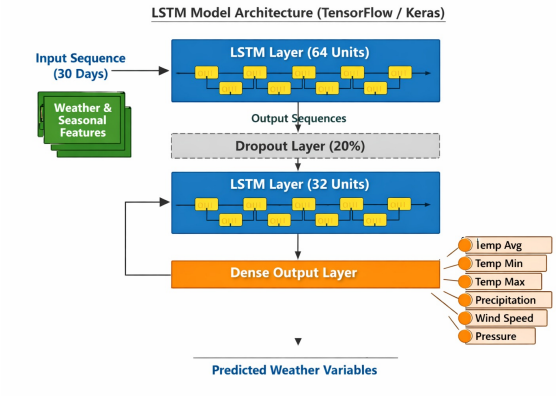

FIGURE 4.5 – Architecture du modèle LSTM

Le modèle est implémenté à l'aide de *TensorFlow/Keras* selon une architecture progressive permettant d'extraire puis de synthétiser l'information temporelle.

Première couche LSTM – 64 neurones

— **Rôle :** extraction des motifs temporels complexes.

— **Paramètre `return_sequences=True` :**

  — permet de transmettre toute la séquence à la couche suivante,

  — essentiel lorsque plusieurs couches LSTM sont empilées.

Couche Dropout – 20%

— **Rôle :** réduction du surapprentissage (*overfitting*).

— **Principe :**

— désactive aléatoirement 20% des neurones pendant l'entraînement,
— empêche le modèle de trop dépendre de certaines connexions.

Deuxième couche LSTM – 32 neurones
— **Rôle :** synthétiser l'information temporelle extraite et produire une représentation compacte de la séquence.

Couche Dense de sortie
— **Neurones :** 6, correspondant aux variables météorologiques prédites : `tavg`, `tmin`, `tmax`, `prcp`, `wspd`, `pres`.
— **Rôle :** fournir la prédiction finale pour le jour suivant.

Paramètres d'entraînement
— **Optimiseur : Adam**
rapide et robuste, adapté aux réseaux profonds.
— **Fonction de perte : MSE** (*Mean Squared Error*)
pénalise fortement les grandes erreurs.
— **Métrique : MAE** (*Mean Absolute Error*)
plus interprétable, mesure l'erreur moyenne absolue.

4.Entraînement du modèle LSTM

Paramètres d'entraînement Le modèle est entraîné sur **60 époques**, permettant au réseau de traiter l'ensemble des données plusieurs fois et de renforcer l'apprentissage. Chaque mise à jour des poids se fait sur des **batches de 32 séquences**, ce qui constitue un compromis entre stabilité de convergence et efficacité computationnelle.

Validation et contrôle du surapprentissage Une **séparation de validation de 10%** est utilisée pour évaluer la performance sur des données non vues, permettant de détecter le surapprentissage (*overfitting*).

Pour renforcer cette protection, la technique de **Early Stopping** est appliquée avec une **patience de 7 époques** : si la perte sur l'ensemble de validation ne s'améliore pas pendant 7 époques consécutives, l'entraînement est interrompu et les meilleurs poids sont restaurés.

5.Prédiction pour l'année 2026

Objectif de la prédiction L'objectif principal est de générer des prévisions journalières des variables météorologiques pour toute l'année 2026 à partir des données historiques de 2021 à 2025. Cette étape permet d'évaluer la capacité du modèle LSTM à projeter les tendances futures et à fournir des résultats exploitables pour la planification agricole, l'analyse climatique ou la gestion énergétique.

Stratégie de prédiction récursive La prédiction est effectuée de manière **itérative (ou récursive)** :

1. Les 30 derniers jours connus servent de point de départ.
2. Le modèle prédit les variables météorologiques du jour suivant.

3. Cette prédiction est ensuite ajoutée à la séquence pour former la nouvelle entrée du modèle.
4. Le processus est répété 365 fois, couvrant ainsi l'ensemble de l'année 2026.

Cette approche est courante pour les séries temporelles, car elle permet de simuler un scénario réel de prévision sur le long terme. **Limite :** l'accumulation d'erreurs, car chaque prédiction repose sur les valeurs précédemment prédites plutôt que sur des données réelles.

Gestion de la saisonnalité future Pour chaque jour de 2026, les composantes saisonnières sont recalculées à l'aide de l'encodage cyclique :

$$\text{sin\_doy} = \sin\left(\frac{2\pi \times \text{jour}}{365}\right), \quad \text{cos\_doy} = \cos\left(\frac{2\pi \times \text{jour}}{365}\right)$$

Ces valeurs sont intégrées aux séquences d'entrée afin que le modèle prenne en compte les variations saisonnières annuelles (été, hiver, etc.). Cette approche améliore la cohérence des prédictions, notamment pour les périodes où le climat suit un cycle marqué.

6.Dénormalisation des résultats

Les prédictions fournies par le modèle sont normalisées (entre 0 et 1). Pour qu'elles soient interprétables et exploitables, une étape de **dénormalisation** est indispensable.

Problème rencontré Le *scaler* a été entraîné sur l'ensemble des variables d'entrée :
— variables météorologiques,
— variables saisonnières.
Or, le modèle ne prédit que les **variables météorologiques**. Cela crée un problème lors de l'application de la fonction `inverse_transform` qui attend toutes les colonnes présentes lors de l'entraînement du scaler.

Solution appliquée Pour résoudre ce problème, la démarche suivante a été adoptée :

1. Ajout de **colonnes fictives** (*padding*) correspondant aux variables saisonnières manquantes.
2. Application de la fonction `inverse_transform` du scaler pour dénormaliser l'ensemble.
3. Suppression des colonnes inutiles correspondant aux variables saisonnières fictives.

Résultat Les valeurs finales sont désormais exprimées dans leurs **unités réelles** :
— **°C** pour les températures (`tavg`, `tmin`, `tmax`),
— **mm** pour les précipitations (`prcp`),
— **hPa** pour la pression (`pres`),
— **m/s** pour la vitesse du vent (`wspd`) si incluse.

7.Évaluation des performances

Le modèle est évalué sur un **jeu de test de 365 jours**, avec des prédictions **dénormalisées** permettant une comparaison directe avec les valeurs réelles.

Évaluation globale Des métriques globales sont utilisées pour mesurer la précision générale des prédictions pour chaque variable météorologique :

- **MSE** (*Mean Squared Error*) : pénalise fortement les grandes erreurs,
- **RMSE** (*Root Mean Squared Error*) : racine carrée du MSE pour interprétation dans les unités originales,
- **MAE** (*Mean Absolute Error*) : mesure de l'erreur moyenne absolue
- **MAPE** (*Mean Absolute Percentage Error*) : erreur relative en pourcentage.

Évaluation des valeurs extrêmes Une attention particulière est portée aux **conditions extrêmes** (fortes chaleurs, gel, fortes précipitations) à l'aide de métriques de classification :

- **Accuracy** : exactitude globale de la classification des extrêmes,
- **Recall** : capacité à détecter correctement les événements extrêmes,
- **Precision** : proportion de prédictions extrêmes correctes,
- **F1-score** : compromis entre recall et precision.

8. Sauvegarde des résultats

Les prédictions journalières générées par le modèle pour l'année 2026 sont enregistrées dans le fichier suivant :

`weather_2026_predicted.json`

Structure du fichier Le fichier JSON contient une liste d'objets, chaque objet représentant un jour de l'année 2026. Chaque objet possède les clés suivantes :

- **date** : jour correspondant à la prédiction,
- **tavg, tmin, tmax** : températures moyenne, minimale et maximale (en °C),
- **prcp** : précipitations (en mm),
- **wspd** : vitesse du vent (en m/s),
- **pres** : pression atmosphérique (en hPa).

Utilisations potentielles Les données sauvegardées peuvent être exploitées pour diverses analyses et applications :

- **Analyse statistique** : calcul de moyennes, tendances, corrélations entre variables météorologiques, etc.,
- **Visualisation** : création de graphiques de séries temporelles, cartes climatiques, histogrammes et heatmaps,
- **Applications métier** : planification agricole, gestion énergétique, suivi climatique et décisions stratégiques basées sur les prévisions.

4.2.3 Système Expert et Architecture Neuro-Symbolique

Afin de pallier le manque d'explicabilité des réseaux de neurones (le problème de la « boîte noire »), nous avons développé un module hybride de type **Neuro-Symbolique**. Ce module agit comme une couche de raisonnement post-traitement : il prend les prédictions numériques brutes du modèle LSTM et leur applique une sémantique métier rigoureuse.

1. Modélisation des connaissances (Ontologie Légère)

Nous avons structuré la connaissance du domaine via une ontologie légère implémentée en JSON. Elle définit deux entités principales :

— **MeteoFact :** L'unité atomique de donnée (ex : $T_{max} = 43°C$ le 12/07/2026).
— **RiskEvent :** L'événement inféré, caractérisé par un `Type` (ex : HEATWAVE) et une `Severity` (ex : CRITICAL).

2. Base de Règles (Rule Engine)

Contrairement à une simple instruction `if/else`, le moteur de règles intègre des seuils validés par la littérature météorologique pour la région de Beni Mellal-Khenifra. Nous avons implémenté trois catégories de règles :

Règles Thermiques :

— **Règle Critique (R1) :** Si $T_{max} \geq 42°C$, déclenchement d'une alerte `CRITICAL`.
— **Règle de Chaleur Sèche (R2) :** Si $T_{max} \geq 38°C$ ET $Humidity < 20\%$, déclenchement d'un risque `HIGH` (danger pour l'agriculture).

Règles Aérologiques (Vent et Tempête) :

— **Règle de Tempête (R3) :** Si $V_{vent} \geq 85$ km/h, classification en `SEVERE_STORM` (Dommages structurels possibles).
— **Règle de Bourrasque (R4) :** Si $V_{vent} \geq 60$ km/h, classification en `STRONG_WIND`.

3. Moteur d'Inférence Temporel

Le moteur ne se contente pas d'analyser le jour J. Il utilise un **Buffer Historique (Sliding Window)** de taille $N = 3$ jours pour détecter la persistance.

$$\text{Persistence} = \forall d \in \{J, J-1, J-2\}, \text{Condition}(d) \text{ is True}$$

Cette logique permet de distinguer un « pic de chaleur » (événement ponctuel) d'une « vague de chaleur » (événement persistant, nécessitant une mobilisation des ressources civiles).

4.2.4 Architecture Web et Interface Décisionnelle

L'interface utilisateur a été conçue comme une **Single Page Application (SPA)** réactive, utilisant le framework **Angular 17**. L'objectif est de fournir un outil d'aide à la décision (DSS) fluide, capable de mettre à jour les visualisations en temps réel sans rechargement de page.

1. Gestion d'État Réactive (Signals)

Nous avons exploité la nouvelle primitive **Angular Signals** pour la gestion d'état. Cela permet une granularité fine des mises à jour : lorsqu'une nouvelle prédiction arrive du backend, seul le composant graphique concerné est redessiné, optimisant ainsi les performances du navigateur.

2. Composants de Visualisation

L'application est structurée en modules fonctionnels distincts :

A. Le Tableau de Bord Géospatial (Dashboard) Ce module intègre la bibliothèque **Leaflet.js** pour la cartographie.

- **Fonctionnalité :** Affichage des points d'intérêt de la région (Beni Mellal, Khenifra, Azilal).
- **Logique dynamique :** La couleur des marqueurs change dynamiquement (Vert → Orange → Rouge) selon le niveau de risque maximal détecté dans les 7 prochains jours.

B. Le Registre National des Alertes Conçu avec une esthétique « Rapport Officiel », ce module présente les prédictions sous forme tabulaire.

- **Explicabilité :** Chaque ligne est extensible (accordéon) et révèle la « Logique d'Inférence », expliquant pourquoi l'IA a pris cette décision (ex : *« Règle R1 déclenchée : T=43°C »*).
- **Interopérabilité :** Un module d'export CSV a été implémenté en TypeScript pur pour permettre aux experts de télécharger les données.

C. Galerie d'Observation Satellitaire Ce module connecte le frontend au système de fichiers du serveur pour afficher les images d'observation (Sentinel-2, Landsat). Il utilise un système de classification automatique par métadonnées pour taguer les images (ex : `SAT-OPT-024`).

3. Communication Client-Serveur

La communication repose sur une architecture RESTful :

- Le Frontend interroge l'API Node.js (`/api/forecast`).
- Le Backend exécute la simulation.
- Les données JSON sont hydratées dans l'interface en moins de 200ms.

Chapitre 5

Résultats Obtenus

5.1 Présentation et interprétation des résultats

Les résultats incluent les prédictions des variables de température et la détection d'événements thermiques extrêmes. Les performances du système ont été évaluées à l'aide de métriques adaptées aux séries temporelles et aux classes rares.

5.2 Figures et captures d'écran

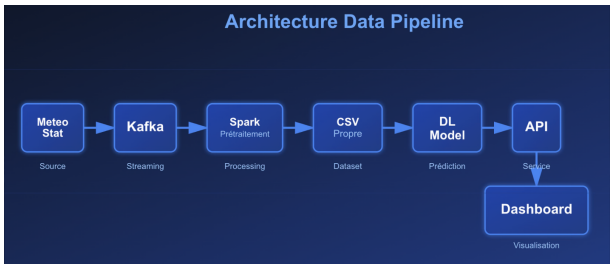

FIGURE 5.1 – Architecture générale du système

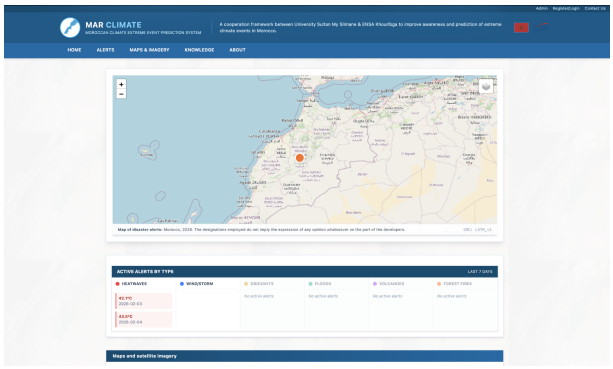

FIGURE 5.2 – Tableau de bord de visualisation

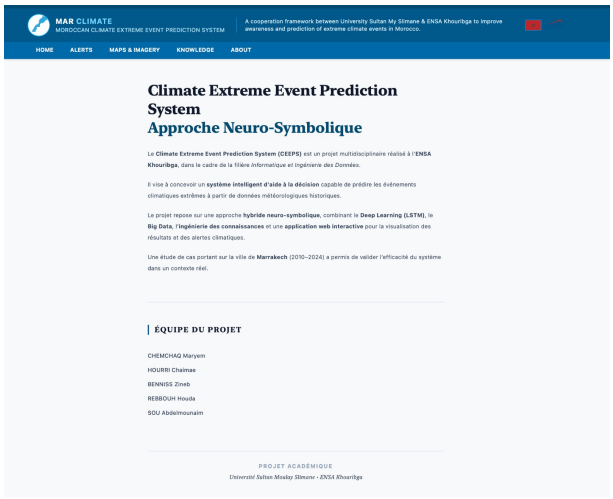

FIGURE 5.3 – Tableau de bord de visualisation

Chapitre 6

Analyse

6.1 Interprétation des résultats

Les résultats montrent que l'approche hybride combinant apprentissage profond et ingénierie des connaissances permet une meilleure détection des événements extrêmes et une interprétation plus explicite des alertes générées.

6.2 Difficultés rencontrées

Les principales difficultés concernent la gestion des données volumineuses, le déséquilibre des classes rares, la qualité des données historiques et l'intégration des différents modules du système.

6.3 Limites du projet

Les limites du projet incluent la dépendance à la qualité des données, les contraintes de calcul et la généralisation du modèle à d'autres régions ou variables climatiques.

Chapitre 7

Conclusion

7.1 Bilan du projet

Ce projet a permis de concevoir un système complet de prédiction des événements climatiques extrêmes, intégrant Big Data, apprentissage profond, ingénierie des connaissances et technologies web.

7.2 Apports du travail

Les contributions principales incluent la mise en place d'un pipeline de données, la conception d'un modèle prédictif et le développement d'une plateforme interactive d'aide à la décision.

7.3 Perspectives et travaux futurs

Les perspectives futures incluent l'intégration de données en temps réel, l'utilisation de données satellitaires, l'amélioration des modèles hybrides et l'extension du système à d'autres régions géographiques.

Références Bibliographiques

Meteostat – Marrakech Weather Data
Deep Learning
Évaluation des modèles deep learning

Annexe A

Annexes