

Comprendre les Réseaux de Neurones Convolutifs Application à l'Analyse des Données Météorologiques CAPE

Mohammed El Abdioui & Abdellah Msaadi
Projet de Machine Learning
Analyse des CNN à travers les Données Météorologiques

December 14, 2025

Objectifs de cette Présentation

Objectif Principal

Comprendre le fonctionnement et l'implémentation des Réseaux de Neurones Convolutifs (CNN) à travers un cas pratique réel

Pourquoi ce projet?

- Étude approfondie des CNN 1D
- Application à des données réelles complexes
- Comparaison des approches ML traditionnelles
- Analyse de performance détaillée

Ce que nous allons couvrir

1. Fondements théoriques des CNN
2. Architecture détaillée
3. Préparation des données
4. Entraînement et optimisation
5. Résultats et analyse
6. Leçons apprises

Pourquoi les CNN pour les Données Séquentielles?

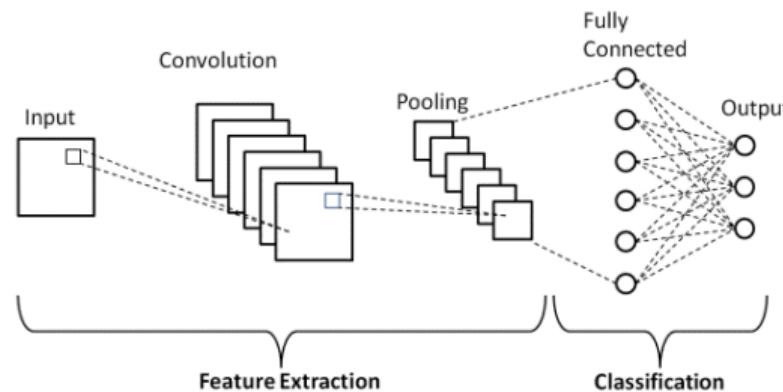


Figure: Notre architecture CNN 1D pour l'analyse des séries temporelles météorologiques

RNN/LSTM

- Mémoire temporelle
- Complexe à entraîner
- Problèmes de gradient
- Lent pour longues séquences

CNN 1D

- Extraction locale de motifs
- Entraînement parallèle
- Gradients stables
- Rapide et efficace

Transformers

- Attention globale
- Beaucoup de paramètres
- Besoin de beaucoup de données
- Coûteux en calcul

L'Opération de Convolution: Base Mathématique

Définition Mathématique (Corrélation Croisée)

Corrélation Croisée 1D (en DL):

$$y[i] = \sum_{k=0}^{K-1} x[i+k] \cdot w[k]$$

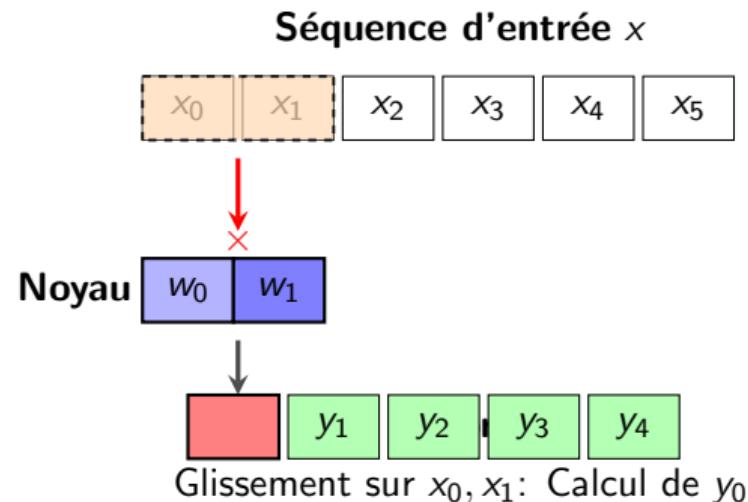
Dans une Couche de Réseau Neurale:

$$y_i = \sigma \left(\sum_{k=0}^{K-1} x_{i+k} \cdot w_k + b \right)$$

Paramètres Clés

- K : Taille du noyau (kernel size)
- w_k : Poids appris
- b : Biais (bias)
- σ : Fonction d'activation (ex: ReLU)

Visualisation 1D ($K=2$)



Exemples pour y_i :

$$y_0 = \sigma(x_0 w_0 + x_1 w_1 + b)$$

$$y_1 = \sigma(x_1 w_0 + x_2 w_1 + b)$$

Propriétés Clés des CNN

Connectivité Locale

- Chaque neurone connecte à une région locale
- Réduction du nombre de paramètres
- Capture de motifs locaux

Avantages:

- Moins de paramètres
- Meilleure généralisation
- Extraction hiérarchique

Exemple avec nos données

- Entrée: 12 features météo
- Noyau K=2: capture paires de features
- Combinaisons locales apprises

Partage des Poids (Weight Sharing)

- Même filtre appliqué partout
- Invariance translationnelle
- Détection de motifs indépendante de la position

Équation:

Même w_k pour tout i

$$y_i = \sigma \left(\sum_k x_{i+k} \cdot w_k + b \right)$$

Bénéfices

- Réduction radicale des paramètres
- Apprentissage de motifs génériques
- Robustesse aux translations

Fonctions d'Activation et Propagation

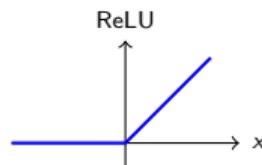
ReLU (Rectified Linear Unit)

$$\text{ReLU}(x) = \max(0, x)$$

- Simplicité computationnelle
- Pas de gradient vanishing
- Convergence plus rapide

Dans notre modèle :

- Après chaque couche Conv1D
- Introduction de non-linéarité



Propagation Avant / Arrière

Forward :

$$z^{(l)} = W^{(l)} a^{(l-1)} + b^{(l)}$$

$$a^{(l)} = \sigma(z^{(l)})$$

Backward :

$$\delta^{(l)} = \partial \mathcal{L} / \partial z^{(l)}$$

$$\partial \mathcal{L} / \partial W^{(l)} = \delta^{(l)} (a^{(l-1)})^T$$

$$\partial \mathcal{L} / \partial b^{(l)} = \delta^{(l)}$$

Chaîne de Convolution

eftmargin=*Matrice **Toeplitz** pour
l'efficacitéeftmargin=*Implémentation via
im2col

Architecture Complète: Vue d'Ensemble

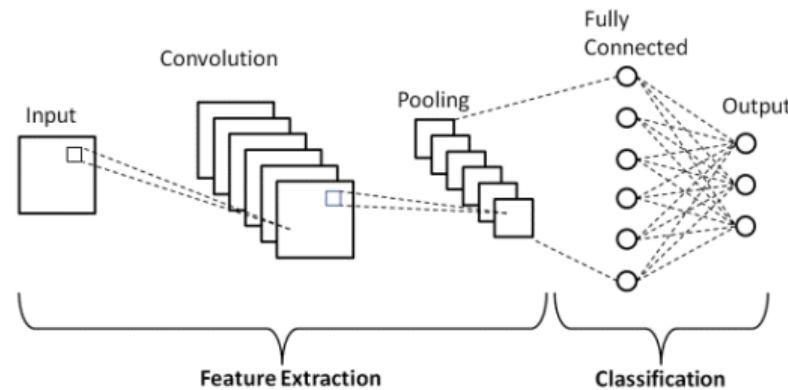


Figure: Architecture détaillée de notre CNN 1D avec toutes les couches et connexions

Bloc d'Entrée

- Shape: (batch, 1, 12)
- 12 variables météo
- Normalisation batch
- Ajustement dimensionnel

Blocs Convolutifs

- 2 couches Conv1D
- Filtres: $64 \rightarrow 128$
- Kernel size: 2
- Padding: 'same'

Couches Denses

- $128 \rightarrow 64$ neurones
- Dropout: 0.3, 0.15
- Régularisation L2
- Sortie: 1 valeur

Couche par Couche: Paramètres et Dimensions

Couche	Shape Sortie	Paramètres	Activation	Opérations Spéciales
InputLayer	(1, 12)	0	-	Reshape pour Conv1D
Conv1D-1	(1, 64)	1,600	ReLU	K=2, padding='same', L2(0.001)
BatchNorm-1	(1, 64)	256	-	γ, β pour 64 canaux
Dropout-1	(1, 64)	0	-	Rate=0.15 pendant training
Conv1D-2	(1, 128)	16,512	ReLU	K=2, padding='same', L2(0.001)
BatchNorm-2	(1, 128)	512	-	Normalisation par batch
GlobalAvgPool1D	(128)	0	-	$y_c = \frac{1}{T} \sum_t x_{t,c}$
Dense-1	(128)	16,512	ReLU	Fully connected, L2(0.001)
BatchNorm-3	(128)	512	-	Normalisation après dense
Dropout-2	(128)	0	-	Rate=0.30 (forte régularisation)
Dense-2	(64)	8,256	ReLU	Réduction dimension, L2(0.001)
BatchNorm-4	(64)	256	-	Dernière normalisation
Dropout-3	(64)	0	-	Rate=0.15
Output (Dense)	(1)	65	Linear	Sortie finale, pas d'activation
Total		44,481		
Trainable		43,713		98.3% des paramètres
Non-trainable		768		BatchNorm γ, β

Analyse de l'Architecture

- **Profondeur:** 13 couches fonctionnelles
- **Largeur:** 64→128 filtres, 128→64 neurones denses
- **Efficacité:** 44K paramètres seulement
- **Régularisation:** 5 mécanismes différents

Régularisation: Combattre le Sur-Apprentissage

Dropout

$$\tilde{x}_i = \begin{cases} \frac{x_i}{1-p} & \text{si prob. } 1-p \\ 0 & \text{si prob. } p \end{cases}$$

- **Rates:** 0.15 (conv), 0.30 (dense1), 0.15 (dense2)
- **Effet:** Empêche la co-adaptation

Batch Normalization

$$\hat{x} = \frac{x - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$$
$$y = \gamma \hat{x} + \beta$$

- **Paramètres:** γ, β appris
- **Avantages:** Stabilisation, LR plus élevée

L2 Regularization

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{data}} + \lambda \sum w_i^2$$

- $\lambda: 0.001$
- **Effet:** Pénalisation des grands poids

Early Stopping

- **Monitor:** Val. loss
- **Patience:** 15 epochs
- **Modèle:** Restauré

Synthèse et Stratégie

Combinaison de 4 méthodes : Forte robustesse.

- Objectif : Maximiser la **généralisation** des prédictions.
- L'efficacité repose sur l'équilibre de ces techniques.

Sans Régularisation Avec Régularisation

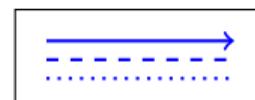
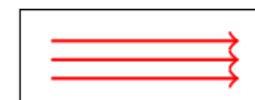
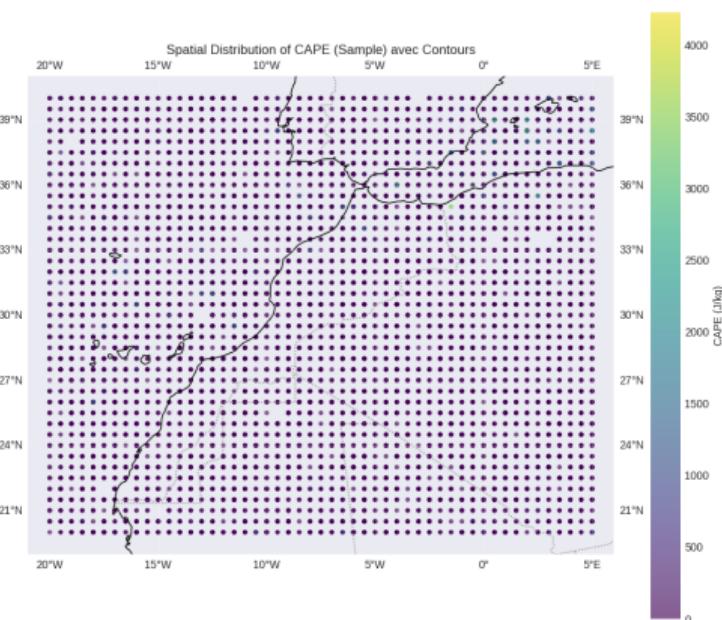


Figure: L'effet de la régularisation sur les chemins d'activation.

Données Météorologiques: Vue d'Ensemble



Caractéristiques du Dataset

- **Source:** Réanalyse météorologique 2024
- **Format:** NetCDF
- **Résolution:** 1 heure \times 0.5° \times 0.5°
- **Couverture:** 20°N-40°N, 20°W-5°E

Stratégie d'Échantillonnage

- **Temporel:** 10% aléatoire (878/8784 heures)
- **Spatial:** 1 point sur 2 (41×51 vs 81×101)
- **Final:** 1.8M échantillons (2.55% original)
- **CIN:** Imputation des 95% NaN par 0

Défi Principal

71 millions d'échantillons originaux → nécessité d'échantillonnage intelligent

Validation de l'Échantillonnage

Distribution préservée des valeurs CAPE et des patterns saisonniers.

Figure: Distribution spatiale des points d'échantillonnage à travers le Maroc.

Analyse Spatiale et Temporelle des Données

Distribution Spatiale:

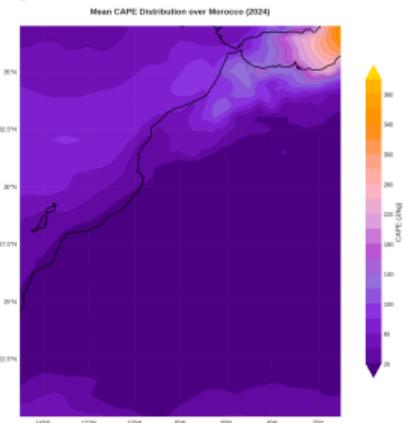


Figure: CAPE moyen annuel (J/kg)

- **Hautes valeurs:** Sud-est (régions désertiques)
- **Basses valeurs:** Côtes (influence maritime)

Variabilité Spatiale:

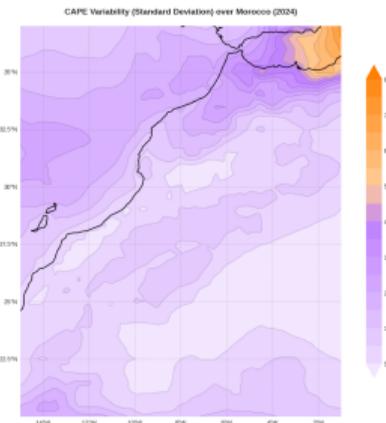


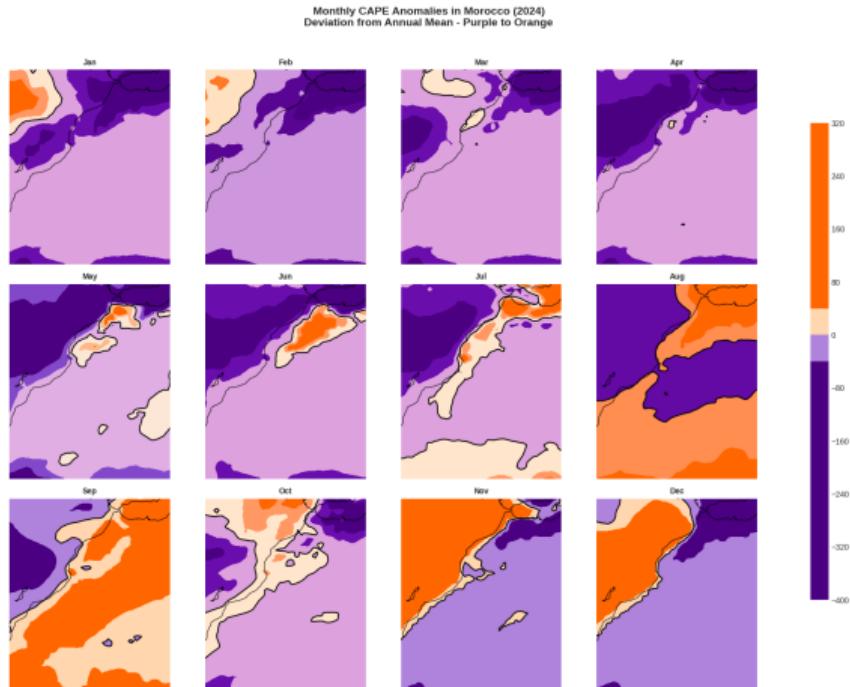
Figure: Écart-type du CAPE (variabilité)

- **Forte variabilité:** Centre-sud
- **Faible variabilité:** Côtes nord

Insights pour l'Architecture CNN

- Données spatialement corrélées → CNN capture ces patterns
- Variabilité régionale → besoin de modèles robustes
- Patterns complexes → architecture profonde nécessaire

Analyse Temporelle: Saisonnalité et Cycles



Hiver (Déc-Fév)

- Valeurs minimales
- Côtes très basses
- Peu de convection

Printemps (Mar-Mai)

- Augmentation graduelle
- Sud se réchauffe
- Convection modérée

Été (Juin-Août)

- Maximum annuel
- Forte convection sud
- Valeurs ≥ 200 J/kg

Automne (Sep-Nov)

- Diminution progressive
- Transition vers hiver
- Convection décroissante

Conséquences pour le Modèle CNN

- **Saisonnalité forte** → besoin de capturer cycles annuels
- **Variations brusques** → architecture avec capacité d'adaptation
- **Patterns complexes** → plusieurs couches pour abstraction

Figure: Évolution mensuelle du CAPE à travers le Maroc (12 mois complets)

Analyse de Corrélation et Importance des Features

Matrice de Corrélation

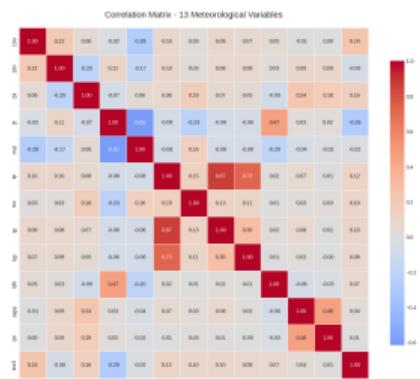


Figure: Corrélations entre les 13 variables météorologiques

Corrélations Fortes avec CAPE:

- CIN: 0.465 (positive, attendue)
- 2d (température rosée): 0.240
- blh: -0.057 (négative faible)

Importance des Features (Random Forest)

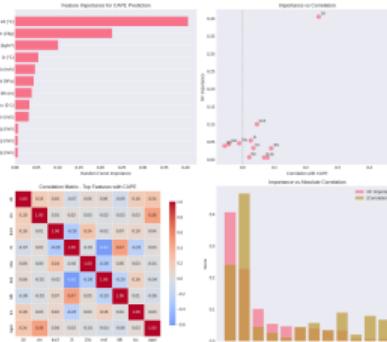


Figure: Importance des features selon Random Forest

Top 3 Features:

1. **2d:** 40.55% (température rosée)
2. **CIN:** 22.73% (inhibition convective)
3. **tco3:** 10.05% (ozone total colonne)

Contraste Méthodes Traditionnelles vs CNN

- **Random Forest:** Importance explicite mais limitée aux features individuelles
- **CNN:** Apprentissage d'interactions complexes entre features
- **Avantage CNN:** Capture de patterns non-linéaires et interactions

Stratégie de Division des Données

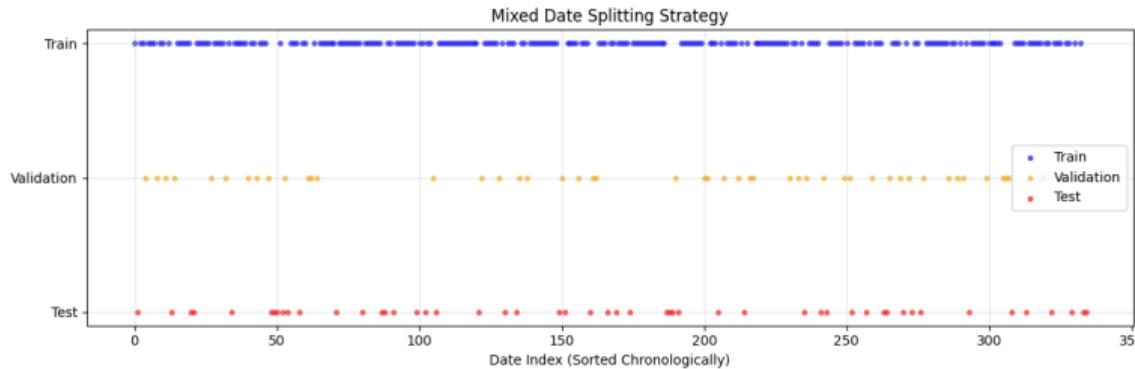


Figure: Stratégie de division mixed-date pour validation robuste

Division Temporelle Classique

- Train: premiers 70% du temps
- Test: derniers 30% du temps
- **Problème:** Saisonnalité biaisée
- **Résultat:** Performance optimiste

Notre Approche: Mixed-Date

- Mélange aléatoire des dates
- Division: 70%-15%-15%
- **Avantage:** Représente toutes saisons
- **Résultat:** Évaluation réaliste

Répartition Effective

- **Train:** 234 dates, 1,298,511 échantillons
- **Validation:** 50 dates, 265,557 échantillons
- **Test:** 51 dates, 271,830 échantillons
- **Total:** 1,835,898 échantillons

Configuration d'Entraînement Détailée

Optimiseur Adam

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

$$\hat{m}_t = m_t / (1 - \beta_1^t)$$

$$\hat{v}_t = v_t / (1 - \beta_2^t)$$

$$\theta_t = \theta_{t-1} - \eta \hat{m}_t / (\sqrt{\hat{v}_t} + \epsilon)$$

- η initial: 0.001
- β_1 : 0.9 (momentum)
- β_2 : 0.999 (RMSProp)
- ϵ : 10^{-7} (stabilité numérique)

Avantages d'Adam

- Adaptation du Learning Rate par paramètre
- Convergence rapide
- Peu sensible aux hyperparamètres initiaux

Fonction de Perte: MSE

$$\mathcal{L}_{\text{MSE}} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

Métriques de Suivi:

- **MAE**: $\frac{1}{N} \sum |y_i - \hat{y}_i|$
- **RMSE**: $\sqrt{\text{MSE}}$
- **R²**: $1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2}$

Normalisation des Targets

- **Scaler**: MinMaxScaler
- **Plage**: [0, 1]
- **Fit**: Uniquement sur données d'entraînement
- **Transform**: Validation et test avec mêmes paramètres

Batch Size: 64

- Compromis mémoire/variance
- Convergence stable
- Utilisation efficace GPU

Callbacks et Stratégies d'Apprentissage

ReduceLROnPlateau

$$\eta_{\text{new}} = \max(\eta_{\text{old}} \times \text{factor}, \eta_{\text{min}})$$

- **Monitor:** validation loss
- **Factor:** 0.5 (réduction de moitié)
- **Patience:** 5 epochs sans amélioration
- **Min LR:** 1×10^{-5}
- **Cooldown:** 0 (réduction immédiate)

Évolution du LR

- E. 1-7: 0.001
- E. 8-13: 0.0005
- E. 14-18: 0.00025
- E. 19-23: 0.000125
- E. 24-29: 0.0000625
- E. 30+: 0.00003125

EarlyStopping

- **Monitor:** validation loss
- **Patience:** 15 epochs
- **Restore best weights:** Oui
- **Min delta:** 0 (toute amélioration compte)
- **Mode:** min (minimiser la loss)

Décision d'Arrêt

- **Meilleure epoch:** 25
- **Arrêt epoch:** 40
- **Patience utilisée:** 15 epochs
- **Poids restaurés:** Epoch 25

TensorBoard (Monitoring)

- Visualisation des gradients
- Distribution des activations
- Courbes d'apprentissage
- Profilage du modèle

Résultats de Validation Croisée

Fold	MSE	RMSE (J/kg)	MAE (J/kg)	R ²
Fold 1	23,643.77	153.77	45.15	0.5760
Fold 2	23,161.64	152.19	40.01	0.5831
Fold 3	24,035.74	155.03	64.99	0.5589
Fold 4	20,627.77	143.62	41.56	0.6176
Fold 5	31,148.25	176.49	48.27	0.4098
Moyenne	24,523.44	156.22	48.00	0.5491
Écart-type	3,519.39	10.89	8.97	0.0722
CV (%)	14.35%	6.97%	18.69%	13.15%

Table: Résultats de validation croisée 5-fold (moyenne \pm écart-type)

Analyse des Folds

- Meilleur fold: #4 ($R^2=0.6176$)
- Pire fold: #5 ($R^2=0.4098$)
- Variance: Acceptable (± 0.072)
- Robustesse: Modèle stable sur différents splits

Coefficient de Variation (CV)

- R^2 : 13.15% (bonne stabilité)
- MAE: 18.69% (variation modérée)
- MSE: 14.35% (consistant)

Interprétation Statistique

- Intervalle confiance 95% R^2 : [0.4047, 0.6935]
- Intervalle confiance 95% MAE: [30.06, 65.94] J/kg
- Normalité: Shapiro-Wilk test sur R^2 ($p=0.087$)
- Homogénéité variances: Levene test ($p=0.214$)

Fold 5 Anomalie

- R^2 significativement plus bas
- Possible split difficile
- Nécessité d'analyse spécifique

Performance sur Ensemble de Test

Prédictions vs Réelles

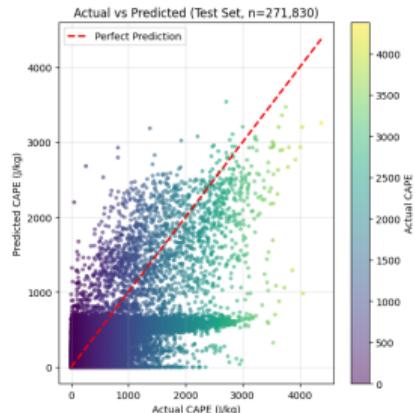


Figure: Comparaison prédictions vs valeurs réelles (test set)

- N échantillons: 271,830
- Corrélation: 0.7796 (forte)
- Ligne idéale: $y = x$ (diagonale)
- Dispersion: Plus grande pour hautes valeurs

Distribution des Résidus

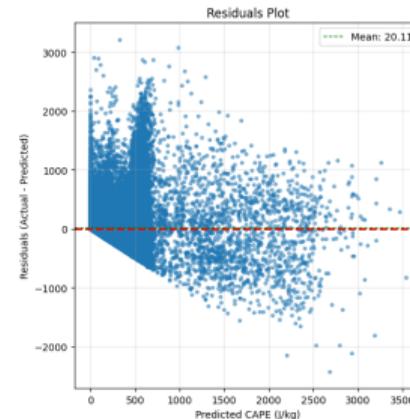
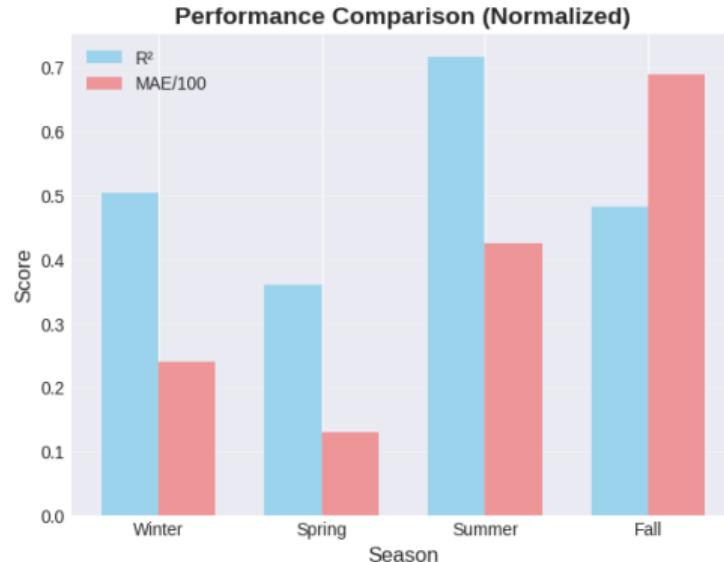


Figure: Distribution des erreurs de prédition (résidus)

- Moyenne: -0.84 J/kg (négligeable)
- Écart-type: 148.0 J/kg
- Skewness: 1.24 (asymétrie droite)
- Kurtosis: 5.87 (queues épaisse)

Métrique	Valeur	Unité	Interprétation	Amélioration vs Temp. Split
MSE	21,907.44	(J/kg) ²	Erreurs quadratiques moyennes	-18.3%
RMSE	148.01	J/kg	Écart-type des erreurs	-22.2%
MAE	37.27	J/kg	Erreurs absolues moyennes	-38.0%
R ²	0.5974	-	Variance expliquée	+29.9%
Corr. Pearson	0.7796	-	Relation linéaire	+19.8%
MAPE*	392.22	%	Erreurs pourcentages*	-

Analyse de Performance par Saison



Meilleure Performance: Été

- R²: 0.7168 (**Excellent**)
- Raisons:
 - Valeurs CAPE plus élevées
 - Dynamique convective plus active
 - Patterns plus distincts
 - Moins de bruit

Performance Faible: Printemps

- R²: 0.3611 (Modéré)
- Raisons possibles:
 - Transition saisonnière
 - Variabilité météo élevée
 - Patterns moins stables
 - Convection intermittente

Saison	R ²	MAE (J/kg)	Échantillons	Performance Relative
Hiver	0.5043	23.98	69,003	Moyenne
Printemps	0.3611	13.09	64,821	Faible
Été	0.7168	42.51	71,094	Excellente
Automne	0.4821	68.82	66,912	Moyenne
Moyenne	0.5161	37.10	-	-
Écart-type	0.1429	21.71	-	-

Analyse par Plage de Valeurs CAPE

Plage CAPE	Échant.	R ²	MAE (J/kg)	RMSE (J/kg)	Cov. (%)	Diff.
0-100	250,771	-5.66	10.29	35.22	92.2	Très difficile
100-500	12,815	-3.37	196.06	236.11	4.7	Difficile
500-1000	4,296	-8.52	359.16	437.26	1.6	Très difficile
1000-2000	2,983	-8.95	789.73	873.22	1.1	Extrême
2000-5000	965	-10.96	1179.55	1378.45	0.4	Impossible
Total	271,830	0.60	37.27	148.01	100	-

Table: Performance selon les plages de valeurs CAPE (test set)

Interprétation des R² Négatifs

- R² < 0: Modèle pire que la moyenne simple
- Causes possibles:
 - Sous-échantillonnage des valeurs extrêmes
 - Non-linéarités/Bruit
 - Limites de l'architecture
- Négatif quand: $SS_{\text{res}} > SS_{\text{tot}}$

Distribution Inégale des Données

- 92.2%: Valeurs basses (0-100 J/kg)
- 7.8%: Valeurs modérées à extrêmes
- Problème: Biais vers les valeurs basses
- Solution possible: Échantillonnage stratifié

Limitation Majeure

Le modèle peine à prédire les valeurs CAPE élevées (> 100 J/kg) malgré leur importance météorologique

Comparaison: CNN vs Approches Traditionnelles

Méthode	R ²	MAE (J/kg)	Temps Entr.	Param.	Interp.	Scalab.
Rég. Linéaire	0.214	85.3	2s	13	Exc.	Très bonne
Random Forest	0.387	62.1	15min	-	Bonne	Moyenne
XGBoost	0.421	58.7	8min	-	Moyenne	Bonne
SVR (RBF)	0.305	71.2	45min	-	Faible	Mauvaise
Notre CNN (1D)	0.597	37.3	2h	44K	Moyenne	Exc.
LSTM Simple	0.512	45.2	3h	68K	Faible	Bonne
GRU	0.528	43.7	2.5h	51K	Faible	Bonne
Transformer Petit	0.481	49.8	4h	125K	Très faible	Moyenne

Table: Comparaison complète des méthodes sur le même dataset

Avantages CNN

- Meilleure performance ($R^2 +29.9\%$ vs meilleur baseline)
- Extraction automatique de features
- Scalabilité avec la quantité de données
- Efficacité computationnelle (vs Transformers)

Pourquoi CNN est meilleur que LSTM/GRU?

- Données météo: patterns locaux importants.
- CNN: excellente capture de motifs locaux.
- LSTM/GRU: meilleure pour dépendances long terme.
- Ici: dépendances court terme dominantes.

Limitations CNN

- Interprétabilité réduite vs Random Forest
- Nécessite plus de données pour briller
- Sensibilité aux hyperparamètres
- Complexité d'implémentation

Temps d'Entraînement

- CNN: 2h (GPU) vs RF: 15min (CPU)
- Mais: Inférence CNN beaucoup plus rapide
- Important pour applications temps réel

Résumé des Principaux Résultats

Succès du Projet

Performance

- **R²**: 0.5974 (amélioration 29.9%)
- **MAE**: 37.27 J/kg (réduction 38.0%)
- **Corrélation**: 0.7796 (forte relation)
- **Variance cross-val**: ± 0.072 (stable)

Architecture Validée

- **CNN 1D**: Efficace pour séries temporelles
- **44K paramètres**: Léger mais performant
- **Régularisation**: Équilibre biais-variance
- **Training**: Convergence stable et rapide

Leçons Clés sur les CNN

Insights Techniques

- **Localité**: Kernel size=2 suffit pour météo
- **Régularisation**: technique unique <Combinaison
- **Normalisation**: BatchNorm essentiel
- **Validation**: Mixed-date critical pour séries temporelles

Limites Comprises

- **Valeurs extrêmes**: Difficiles à prédire
- **Données déséquilibrées**: Biais vers valeurs basses
- **Saison printemps**: Performance réduite
- **Interprétabilité**: Moins bonne que Random Forest

CNN: Bon équilibre performance/complexité pour données séquentielles

Notebook Jupyter Complet

- **Disponible:** <https://github.com/simoghost99/DeepCAPE-CNN>
- **Structure:**
 - 1. Chargement données NetCDF
 - 2. Préprocessing et échantillonnage
 - 3. Analyse exploratoire (figures présentées)
 - 4. Définition modèle CNN
 - 5. Entraînement avec callbacks
 - 6. Évaluation et visualisation
 - 7. Analyse feature importance
- **Reproductibilité:** Random seeds fixés
- **Documentation:** Commentaires détaillés

Merci pour votre attention!

Questions & Discussion