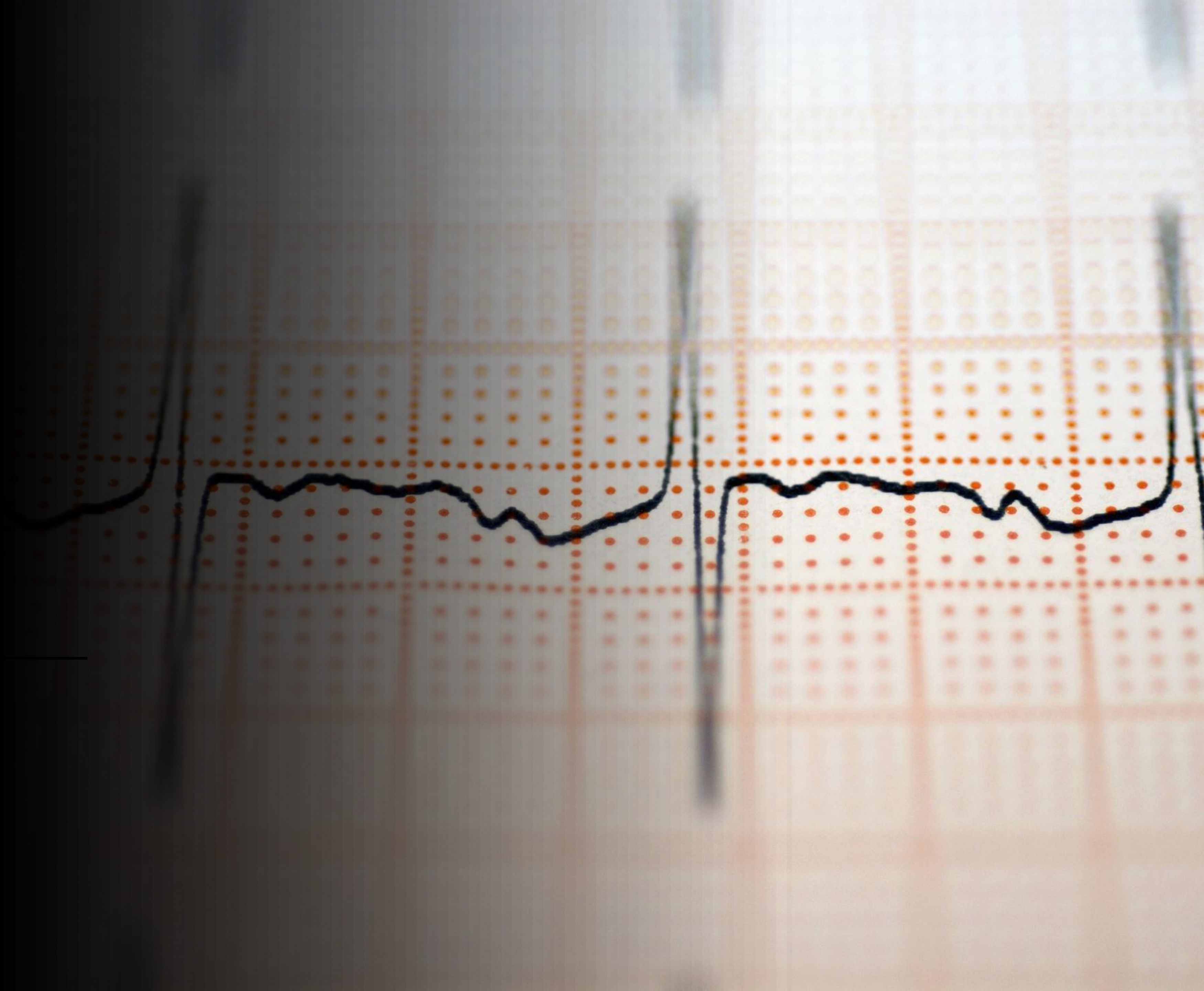
Les séries temporelles

Thomas Wentz



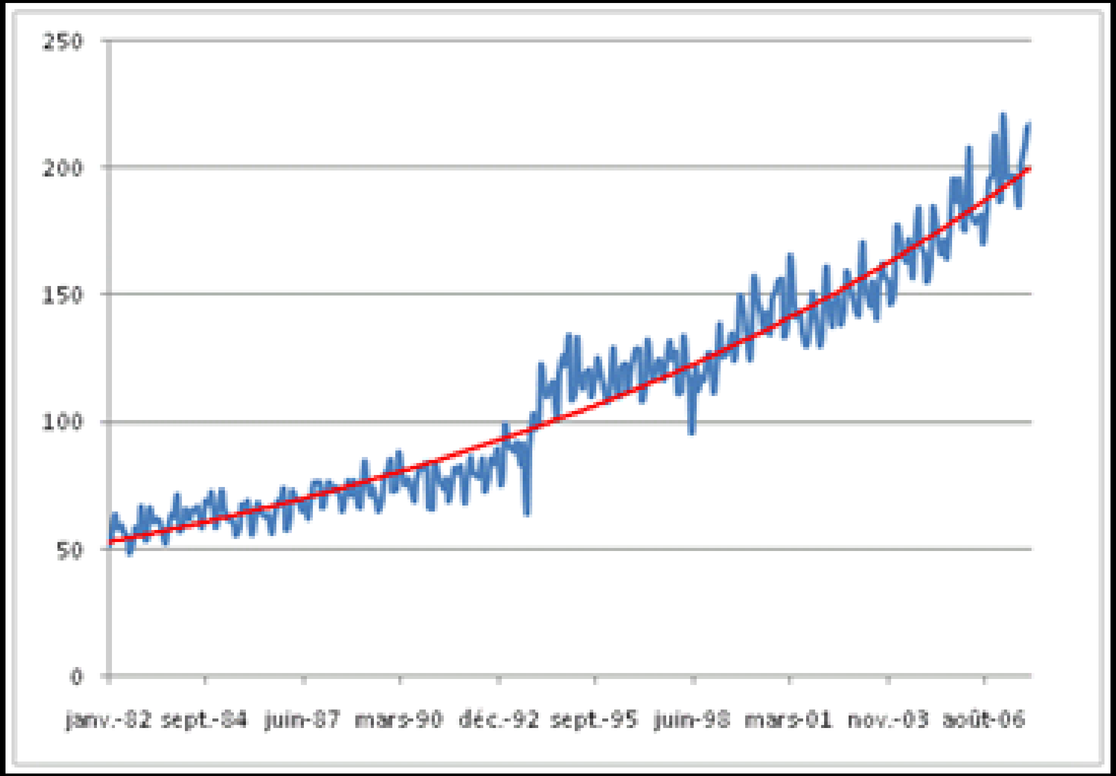
Introduction aux séries temporelles

- Les séries temporelles sont des données qui évoluent au fil du temps
- Elles sont collectées de manière séquentielle.
- Elles sont présentes dans divers domaines pour analyser et prévoir des phénomènes qui changent avec le temps.
- Données de capteurs, phénomène météo, population d’un animal, taux de satisfaction : tout ce qui varie au fil du temps peut être réductible à une série temporelle

Introduction aux séries temporelles

Une série temporelle peut se décomposer en 3 composantes :

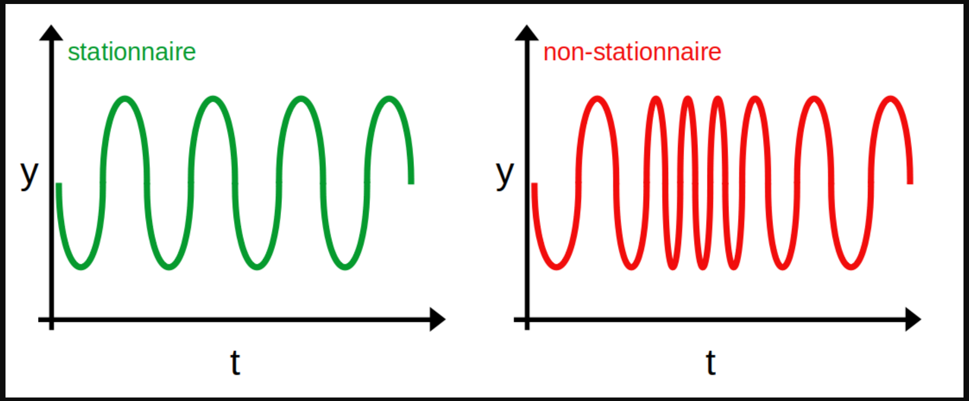
- **Tendance** : orientation générale de la série (vers le haut ou vers le bas)
- **Saisonnalité** : tendances hebdomadaires, mensuelles, trismestrielles ou annuelles
- **Bruit** : ce qui reste après avoir extrait les composants précédents

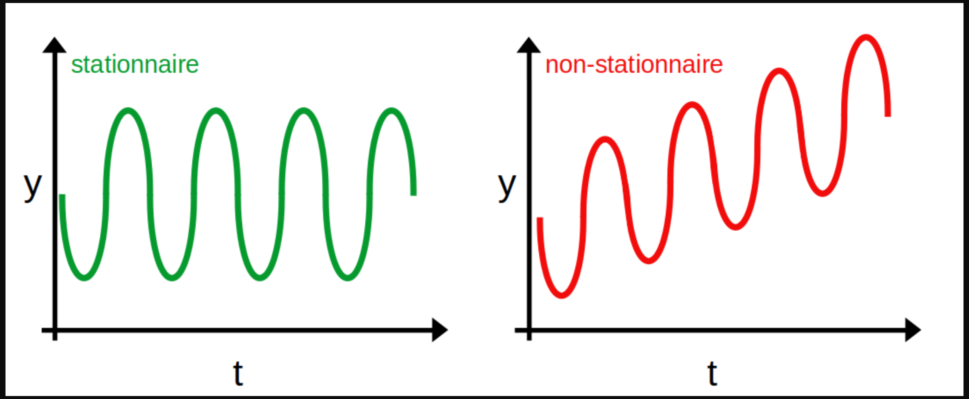


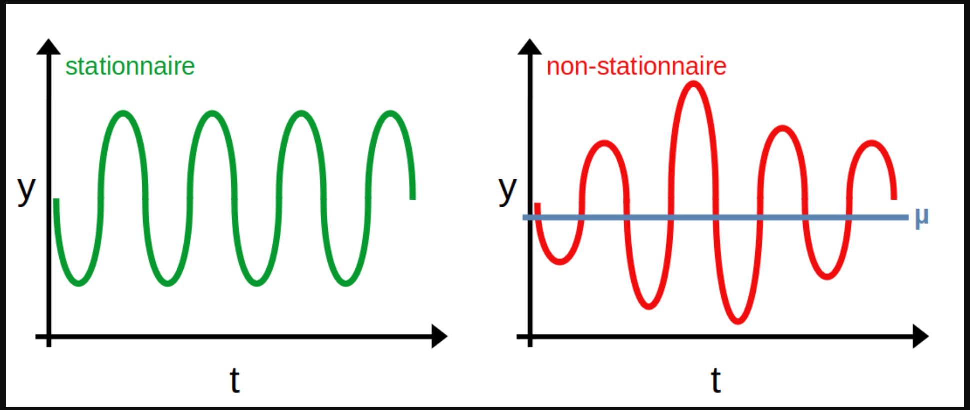
La tendance dans les séries temporelles

La tendance est la direction générale de l'évolution des données sur une longue période. Elle peut être ascendante, descendante ou plate. Elle est souvent représentée par une ligne droite ou une courbe lissée.

La stationnarité des séries temporelles







Une série temporelle est dite **stationnaire** si ses propriétés statistiques ne varient pas avec le temps. La stationnarité est cruciale pour de nombreuses méthodes d'analyse des séries temporelles, car elle permet d'appliquer des modèles et des prévisions plus fiables.

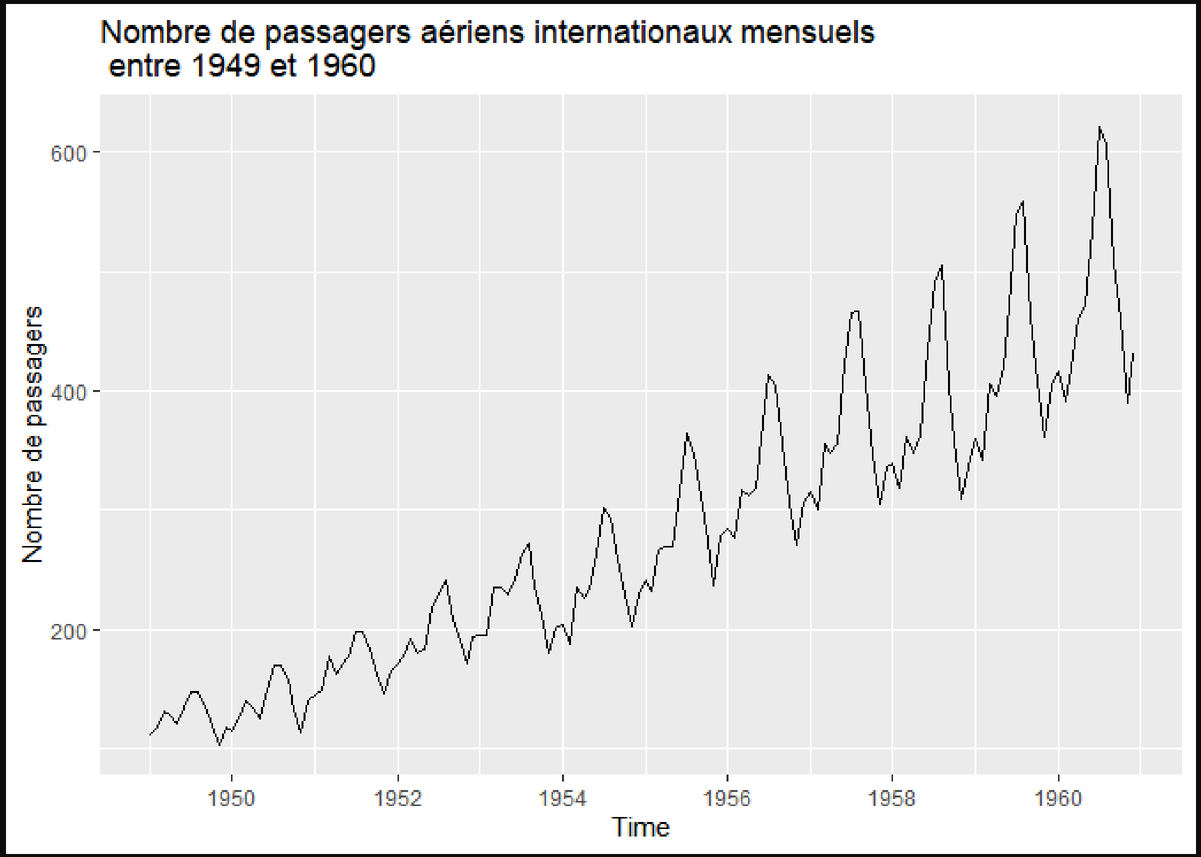
Comment rendre une série temporelle stationnaire?

La principale approche est la differenciation de la série temporelle (une ou plusieurs fois)

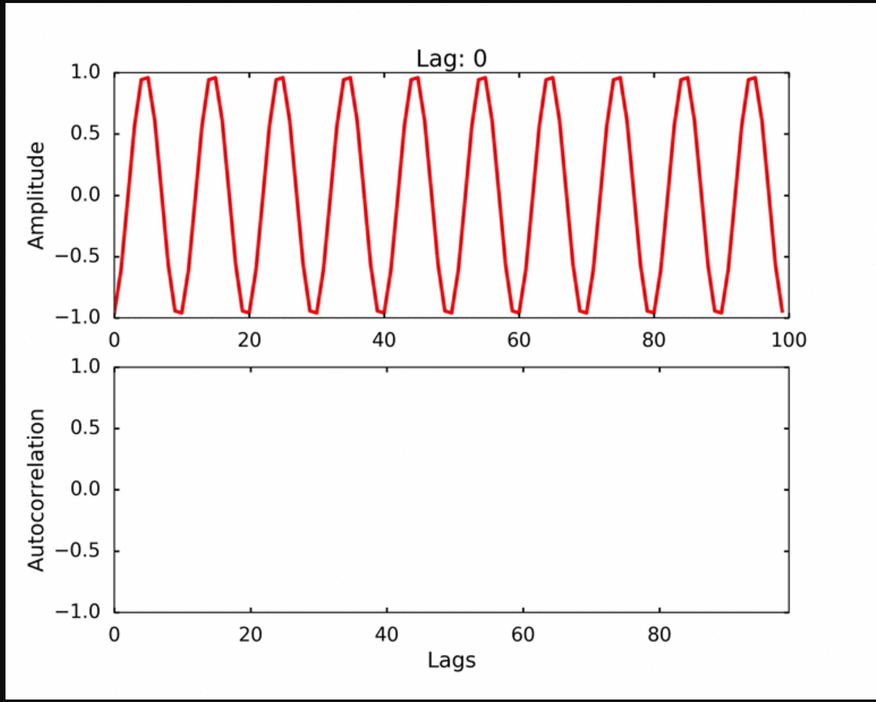
- Si Y(t) est la valeur au temps t, alors la première différence de Y = Y(t) – Y(t-1)
- Exemple : considérons la série suivante :
  - [1, 5, 2, 12, 20]
- La première différence donne :
  - [5-1, 2-5, 12-2, 20-12] = [4, -3, 10, 8]
- La deuxième différenciation donne :
  - [-3-4, -10-3, 8-10] = [-7, -13, -2]

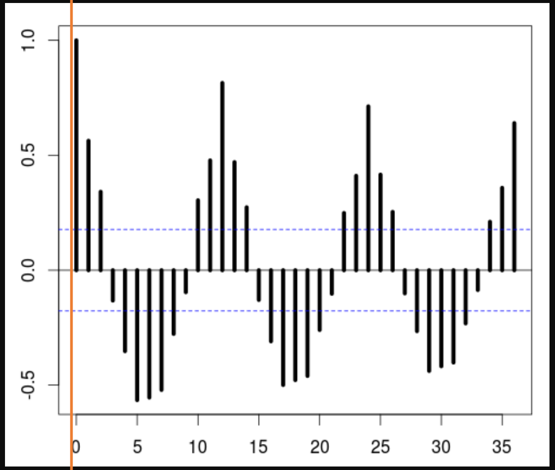
Le test de Dickey-Fuller augmenté est souvent utilisé pour tester la stationnarité. Si la série n'est pas stationnaire, on différencie la série et on teste à nouveau jusqu'à ce qu'elle le devienne.

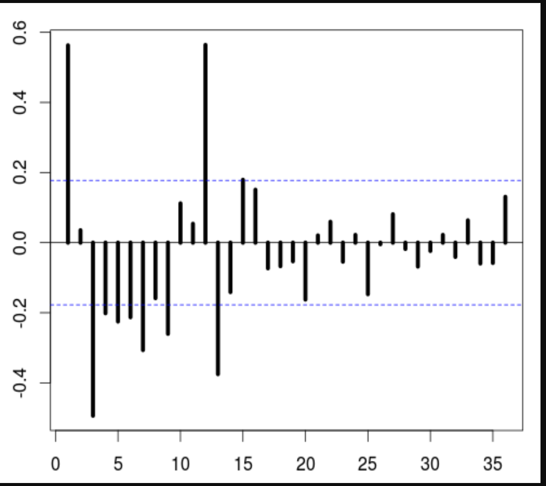
La saisonnalité dans les séries temporelles



La saisonnalité d'une série temporelle correspondant à un phénomène périodique de période identifiée







Détection de la saisonnalité : l’autocorrélation

L'autocorrélation est un indicateur montrant comment les observations d'une série temporelle sont liées entre elles. Il s'agit de la corrélation entre la série et cette même série décalée dans le temps

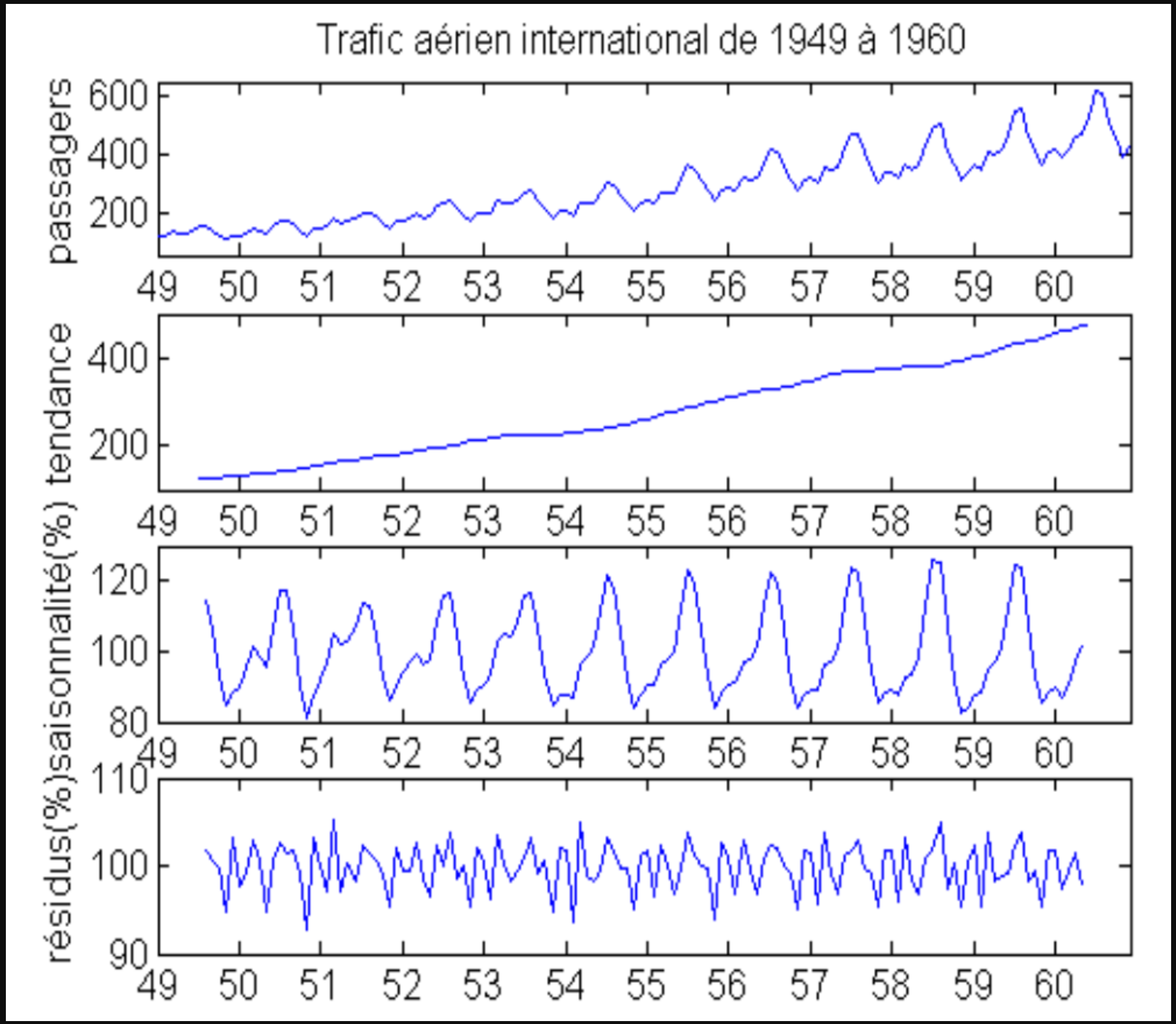
La **fonction d'autocorrélation (ACF)** et la **fonction d'autocorrélation partielle (PACF)** permettent de mesurer l'association entre des valeurs actuelles et passées. Elles indiquent les valeurs passées les plus corrélées aux valeurs suivantes et donc les plus utiles à la prévision de valeurs futures.

⇒ACF au décalage k = corrélation entre les valeurs séparées par k intervalles

⇒=> PACF au décalage k = corrélation entre les valeurs séparées par k intervalles, compte tenu des valeurs des intervalles intermédiaires

Le bruit dans les séries temporelles

- Le bruit est une composante aléatoire des séries temporelles. Il représente les variations imprévisibles qui ne peuvent pas être expliquées par la tendance ou d'autres facteurs systématiques. Le bruit peut masquer des modèles sous-jacents dans les données.



Au final

Les séries temporelles offrent une compréhension approfondie des données qui évoluent dans le temps. La tendance, la stationnarité et le bruit sont des concepts clés pour analyser ces séries.

En comprenant ces concepts, nous pouvons appliquer des techniques avancées d'analyse des séries temporelles pour prendre des décisions éclairées et prévoir les tendances futures.

En conclusion

- Question : comment modéliser une série temporelle pour prédire les futures valeurs qui vont la composer?

Les modèles ARMA

Modèle ARMA
(Autoregressive Moving Average)

- **Le modèle ARMA combine à la fois des termes autorégressifs (AR) et des termes moyens mobiles (MA) pour modéliser une série temporelle.**
- **Modèle autorégressif (AR) :** Un modèle AR utilise les valeurs passées de la série temporelle pour prédire les valeurs futures. Un modèle AR(p):

$$X_{(t)} = c + \phi_1 X_{(t-1)} + \varphi_2 X_{(t-2)} + \ldots + \phi_p * X_{(t-p)} + \varepsilon_{(t)}$$

Où $X_{(t)}$ est la valeur à l'instant t, c est une constante, $\phi_{1,p}$ sont les coefficients autorégressifs et $\varepsilon_{(t)}$ est le terme d'erreur à l'instant t.

- **Modèle moyenne mobile (MA) :** Un modèle MA utilise les erreurs passées du modèle pour prédire les valeurs futures. Un modèle MA(q) :

$$X_{(t)} = \mu + \varepsilon_{(t)} + \theta_1 \varepsilon_{(t-1)} + \vartheta_2 \varepsilon_{(t-2)} + \ldots + \theta_q * \varepsilon_{(t-q)}$$

Où µ est la moyenne de la série temporelle, $\varepsilon_{(t)}$ est le terme d'erreur à l'instant t, et θ_1 à θ_q sont les coefficients de moyenne mobile.

- Le modèle ARMA combine ces deux modèles pour obtenir une prédiction plus précise.
- Un modèle $ARMA_{(p,\ q)}$:

$$X_{(t)} = c + \phi_1 X_{(t-1)} + \varphi_2 X_{(t-2)} + \ldots + \phi_p X_{(t-p)} + \varepsilon_{(t)} + \vartheta_1 \varepsilon_{(t-1)} + \theta_2 \varepsilon_{(t-2)} + \ldots + \vartheta_q \varepsilon_{(t-q)}$$

Modèle ARIMA (Autoregressive Integrated Moving Average)

- Le modèle ARIMA est une extension du modèle ARMA qui prend également en compte la différenciation des données pour rendre la série temporelle stationnaire.
- **Le modèle ARIMA est défini par trois paramètres : p, d et q**.
  - p correspond à l'ordre du modèle autorégressif
  - d correspond au nombre de différenciations
  - q correspond à l'ordre du modèle de moyenne mobile
- Un modèle ARIMA(p, d, q) est défini par l'équation suivante :

$$X_{(t)} = c + \phi_1 X_{(t-1)} + \varphi_2 X_{(t-2)} + \dots + \phi_p X_{(t-p)} + \varepsilon_t + \vartheta_1 \varepsilon_{(t-1)} + \theta_2 \varepsilon_{(t-2)} + \dots + \vartheta_q \varepsilon_{(t-q)}$$

Modèle SARIMA (Seasonal ARIMA)

- Le modèle SARIMA est une extension du modèle ARIMA qui prend également en compte la saisonnalité dans les données
- Le modèle SARIMA est défini par quatre paramètres : p, d, q et P, D, Q, s.
- Les paramètres p, d, et q sont les mêmes que pour le modèle ARIMA
- P, D, Q sont les ordres des termes saisonniers et s est la période saisonnière.

- Un modèle SARIMA(p, d, q)(P, D, Q, s) est défini par l'équation suivante :

$$X_t = c + \phi_1 X_{(t-1)} + \varphi_2 X_{(t-2)} + \dots + \phi_p X_{(t-p)} + \varepsilon_t + \vartheta_1 \varepsilon_{(t-1)} + \theta_2 \varepsilon_{(t-2)} + \dots + \vartheta_q \varepsilon_{(t-q)} + \phi_{1s} X_{(t-s)} + \varphi_{2s} X_{(t-2s)} + \dots + \phi_{Ps} X_{(t-Ps)} + \varepsilon_t + \vartheta_{1s} \varepsilon_{(t-s)} + \theta_{2s} \varepsilon_{(t-2s)} + \dots + \vartheta_{Qs} \varepsilon_{(t-Qs)}$$

Conclusion

Les modèles ARMA, ARIMA et SARIMA sont des outils puissants pour analyser et prédire les séries temporelles.

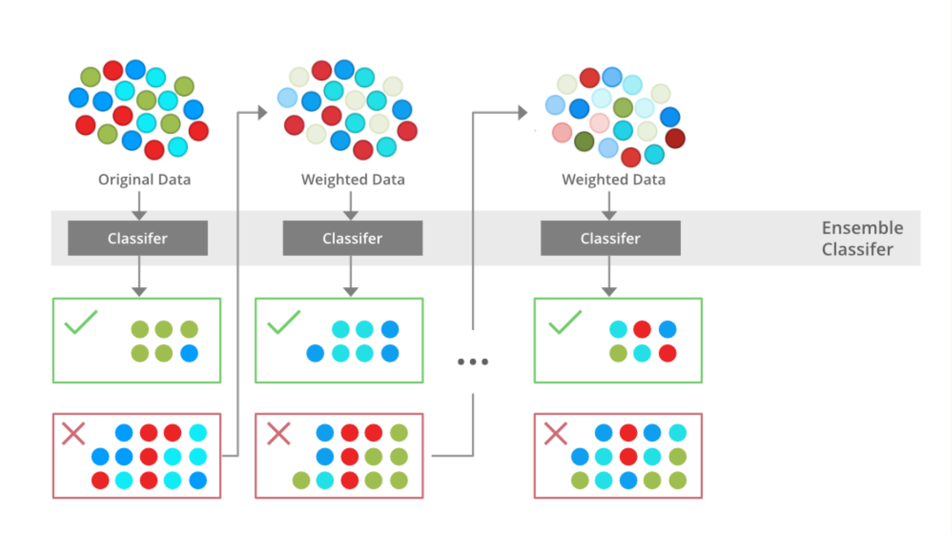
En utilisant ces modèles, vous pouvez capturer les tendances, les saisonnalités et les structures complexes des données pour effectuer des prévisions précises.

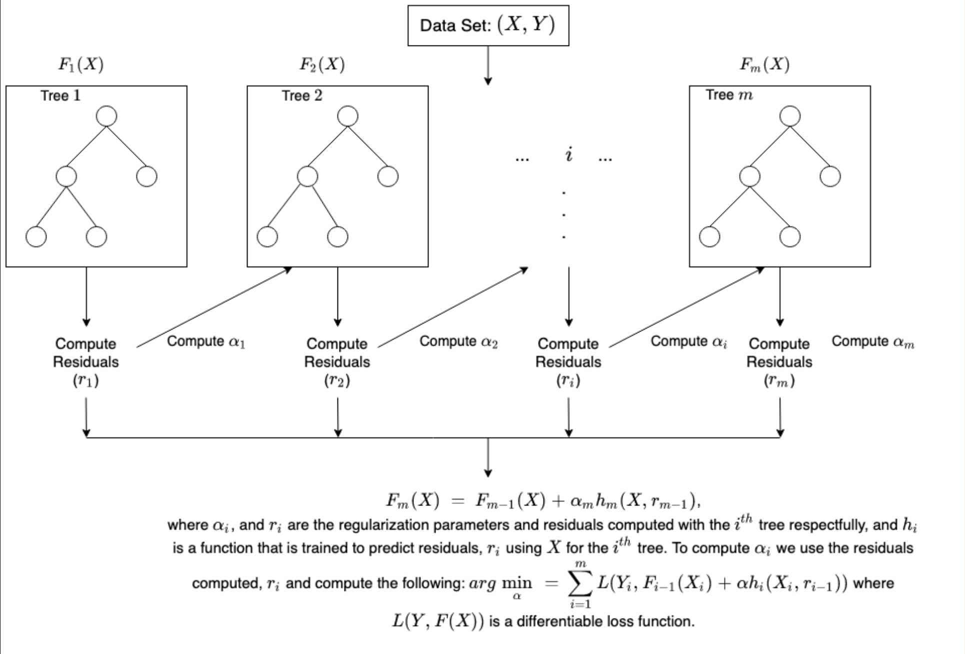
Cependant, il est important de sélectionner les bons ordres et de valider les modèles pour obtenir des résultats fiables.

XGBoost

Comment fonctionne XGBoost pour les séries temporelles?

- **XGBoost** est une méthode avancée basée sur l'algorithme de boosting des arbres de décision.
- **Le boosting est** une technique d'apprentissage automatique qui vise à améliorer les prédictions d'un modèle en ajustant successivement les prédictions en fonction des erreurs des modèles précédents.
- En d'autres termes, il s'agit de construire plusieurs modèles faibles (généralement des arbres de décision) de manière séquentielle, où chaque modèle tente de corriger les erreurs du précédent.

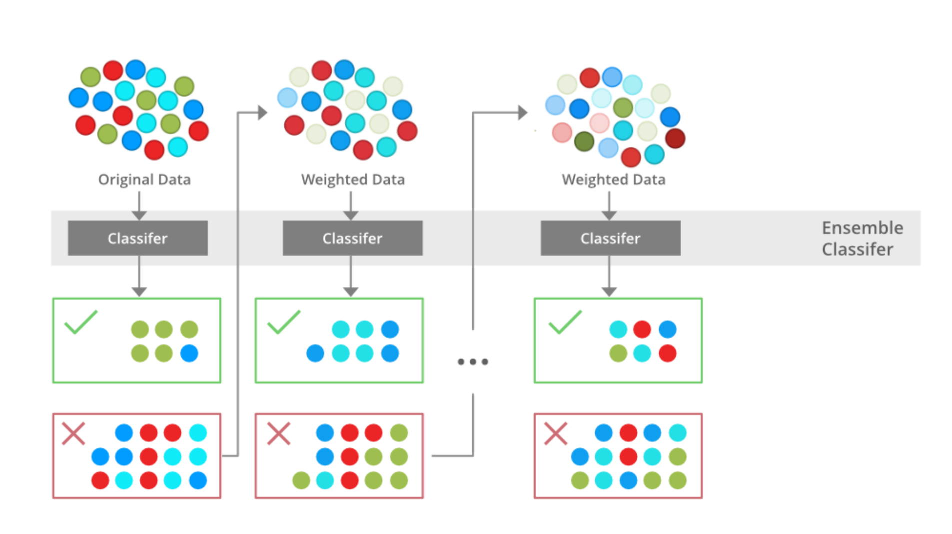


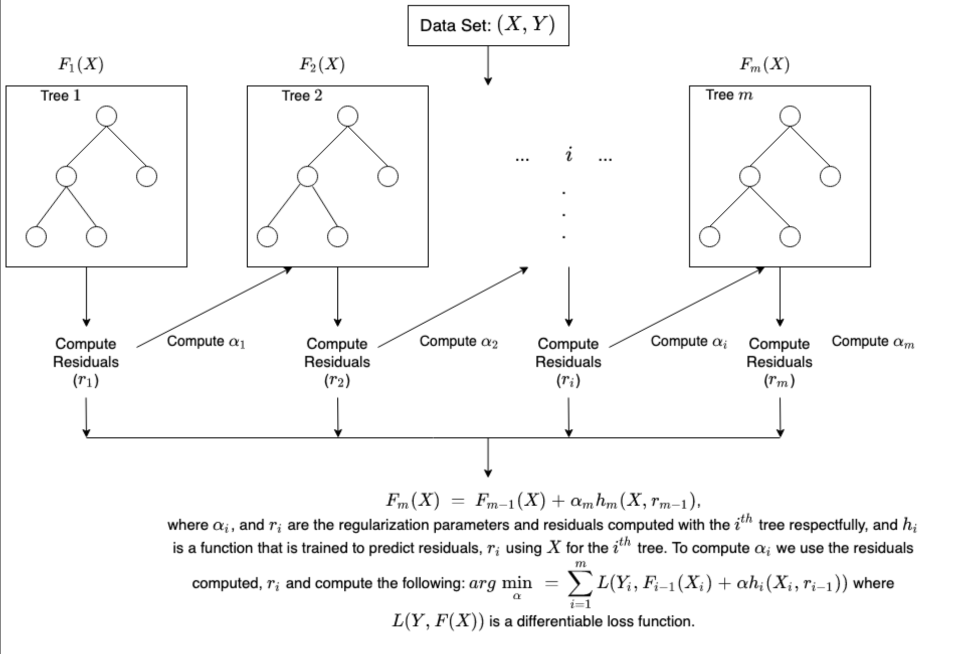


Comment fonctionne XGBoost pour les séries temporelles?

- XGBoost construit un ensemble d'arbres de décision où chaque arbre tente de corriger les erreurs du précédent. Cela permet d'obtenir un modèle final qui est la somme des prédictions de tous les arbres.

- **Optimisation** : L'algorithme utilise la descente de gradient pour minimiser une fonction de perte (comme l'erreur quadratique moyenne) tout en ajoutant de nouveaux arbres





Ingénierie des caractéristiques des séries temporelles avec XGBoost

- **Pour prédire efficacement les séries temporelles avec XGBoost, il est crucial de transformer les données de manière à capturer les tendances, la saisonnalité et d'autres structures inhérentes à la série. Cela peut être réalisé en créant des caractéristiques spécifiques à la série temporelle**

- **Pourquoi ?**
  - **S'adapter à XGBoost** : XGBoost, en tant qu'algorithme basé sur des arbres, ne prend pas intrinsèquement en compte la séquence temporelle des données.
  - **Capturer des tendances** : Les séries temporelles ont souvent des tendances sous-jacentes, qu'elles soient à la hausse, à la baisse ou cycliques. En créant des caractéristiques qui reflètent ces tendances, nous permettons à XGBoost de les identifier et de les utiliser pour améliorer les prédictions.

Ingénierie des caractéristiques des séries temporelles avec XGBoost

Exemples de caractéristiques pour les séries temporelles :

- **Le retard (Lag Features)** : Il s'agit de valeurs décalées d'une série temporelle. Par exemple, si nous voulons prédire la valeur à un moment t, nous pourrions utiliser les valeurs à t-1, t-2, etc. comme caractéristiques. Ces caractéristiques permettent au modèle de "voir" les valeurs précédentes et d'utiliser cette information pour faire une prédiction.

- **La moyenne mobile** : C'est la moyenne des valeurs sur une fenêtre glissante. Elle est utilisée pour lisser les séries temporelles et capturer les tendances à court et à long terme. Par exemple, une moyenne mobile sur 7 jours peut capturer une tendance hebdomadaire.

- **La variance ou écart-type sur une fenêtre** : Ces caractéristiques capturent la volatilité ou la variabilité des données sur une période donnée. Une augmentation soudaine de la variance pourrait indiquer un événement inhabituel dans la série.

Les RNN et les LSTMS

Les limites des CNN

1 : Les problèmes de séquences

Les réseaux neuronaux classiques, tels que les réseaux neuronaux convolutifs (CNN), sont bien adaptés à la modélisation de données structurées, telles que les images, où les relations spatiales locales sont importantes.

Cependant, ils ne sont pas idéalement adaptés à la modélisation de données séquentielles, où les éléments sont liés les uns aux autres dans un ordre précis. Les problèmes de séquences, tels que la traduction automatique, la reconnaissance vocale et l'analyse de texte, nécessitent une compréhension des dépendances temporelles et de la structure à long terme des données.

2 : Défis de la dépendance à long terme

Un défi majeur dans le traitement de séquences avec des réseaux neuronaux classiques est la dépendance à long terme. Les informations pertinentes peuvent être dispersées sur une longue séquence, et les réseaux neuronaux classiques ont du mal à maintenir ces informations sur de longues distances.

Par conséquent, les modèles basés sur les réseaux neuronaux classiques peuvent perdre des informations critiques pour la prédiction future, ce qui limite leurs performances dans des tâches nécessitant une compréhension à long terme.

Pourquoi des réseaux neuronaux récurrents?

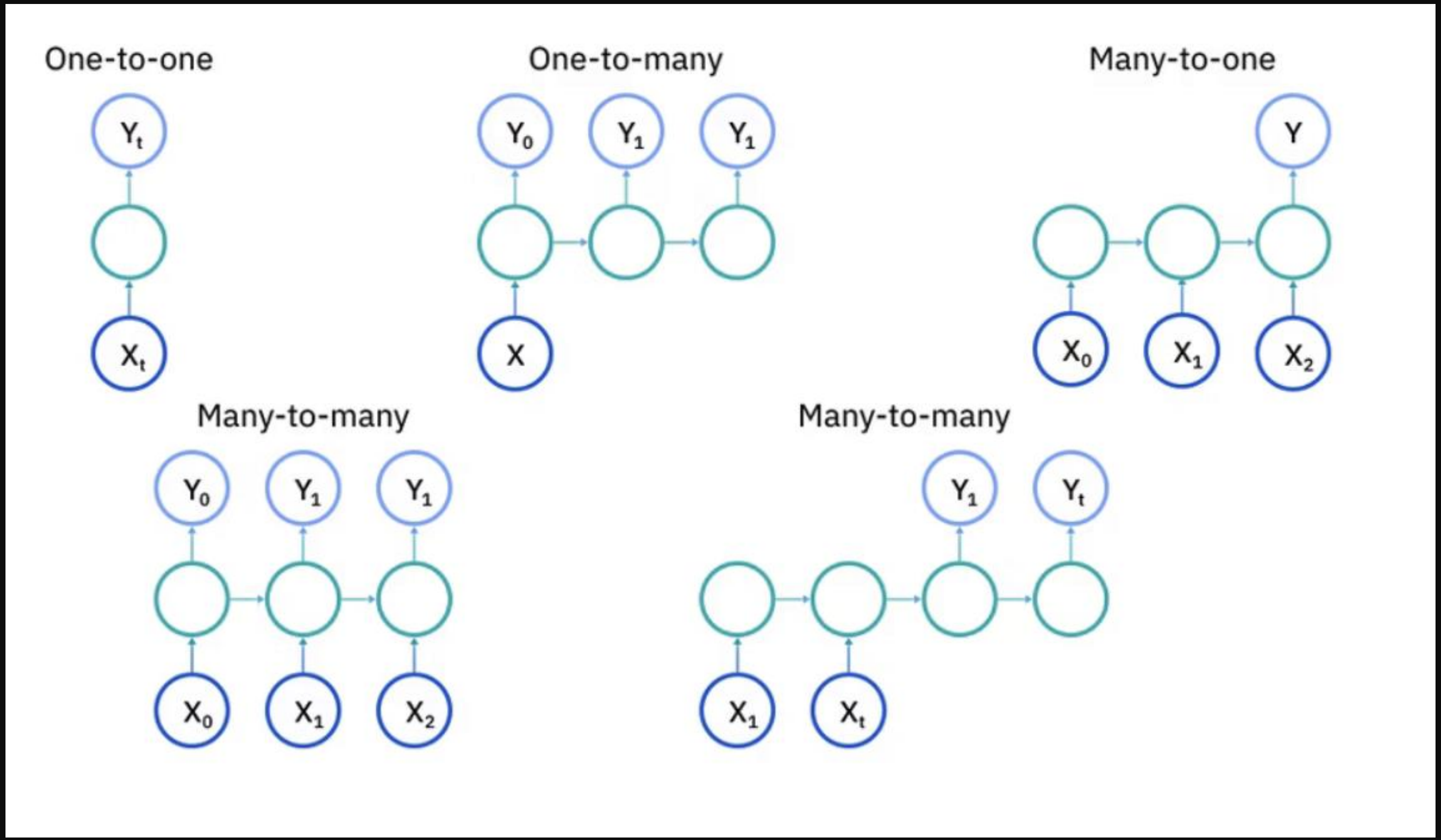
Les RNN sont conçus spécifiquement pour modéliser les séquences en tenant compte de la dépendance à long terme.

Contrairement aux réseaux neuronaux classiques, les RNN ont des **connexions récurrentes** qui leur permettent de mémoriser l'information précédente et de l'utiliser pour influencer les prédictions futures.

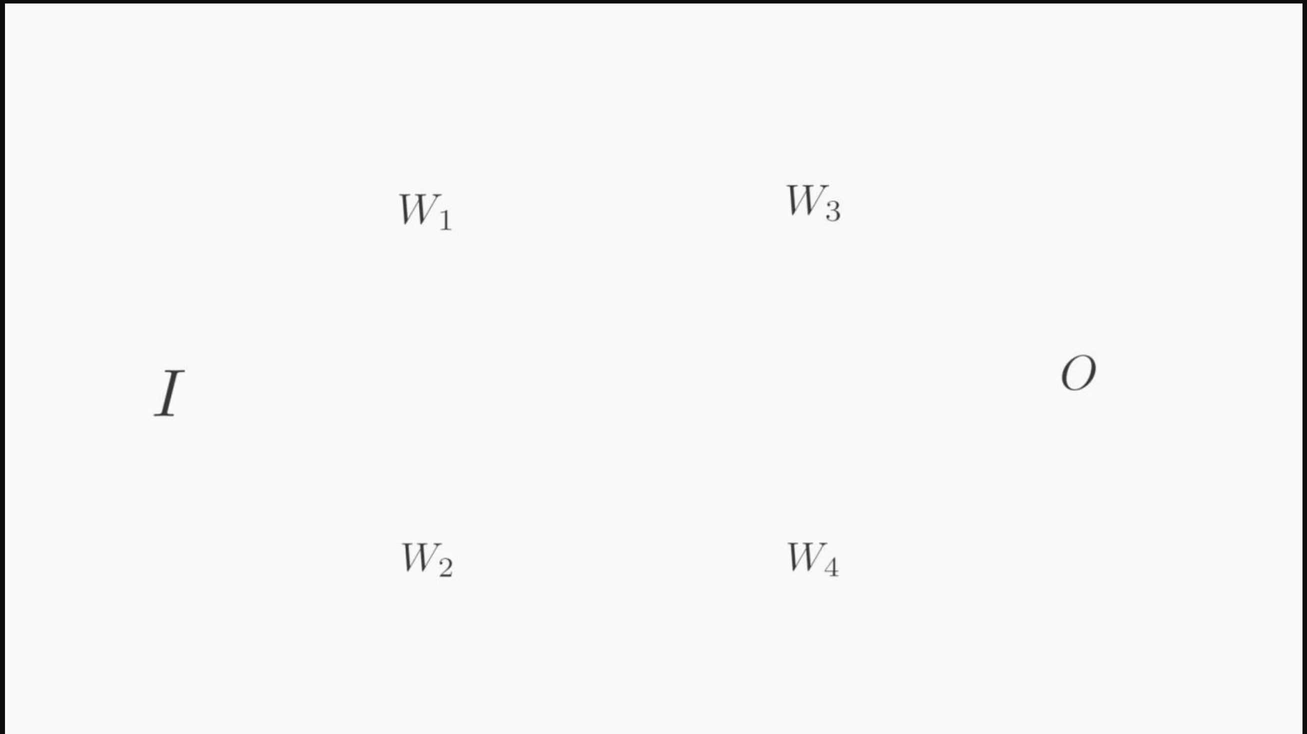
Cette capacité d'apprendre des motifs temporels et de capturer les dépendances à long terme fait des RNN des outils puissants pour traiter des tâches séquentielles complexes.

L'idée fondamentale derrière les RNN est de partager les paramètres entre les différentes étapes de la séquence. Cela signifie que chaque étape utilise les mêmes poids et biais, ce qui permet au réseau de généraliser les connaissances apprises à travers la séquence entière. Cette propriété de partage de poids fait des RNN des modèles très efficaces pour modéliser les motifs temporels et capturer les dépendances à long terme.

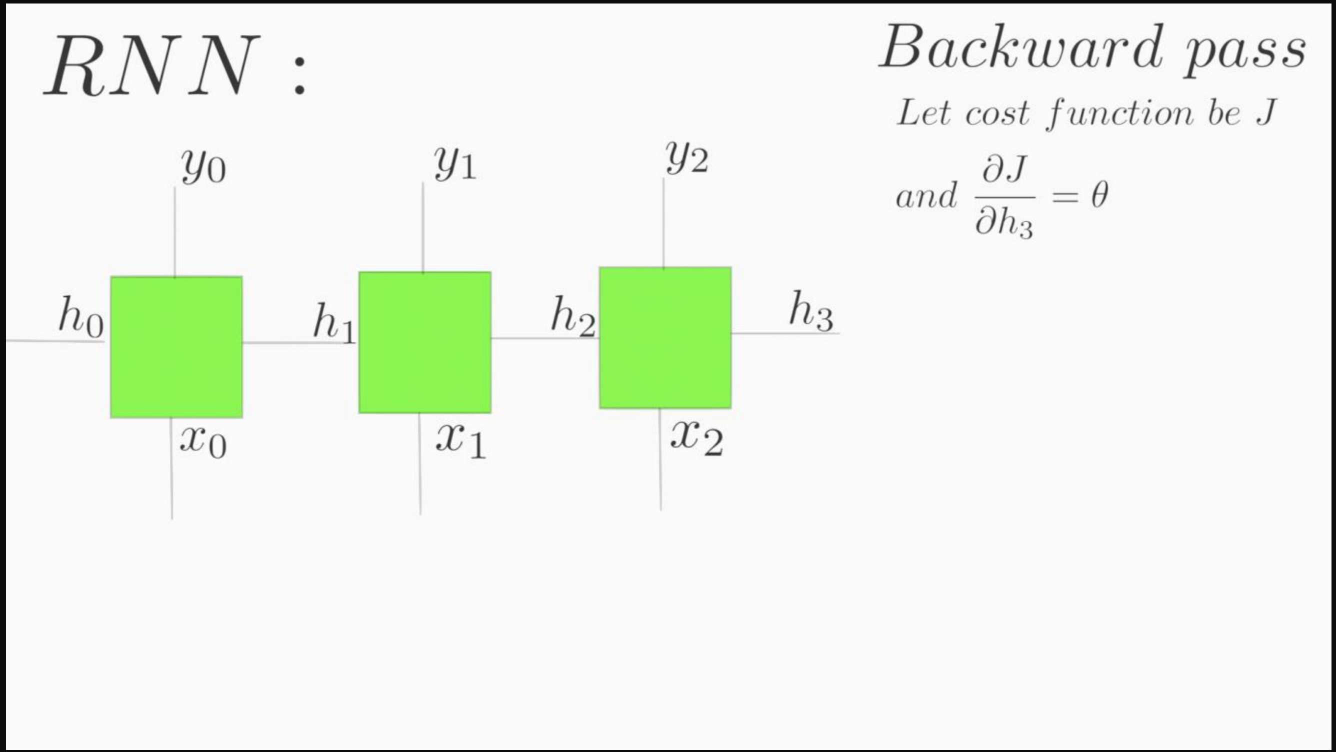
Architecture d'un RNN

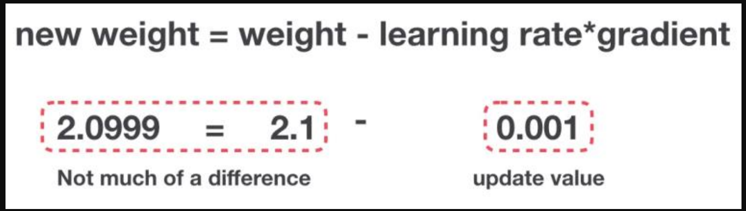


Architecture d’un RNN



Le problème de la disparition du gradient

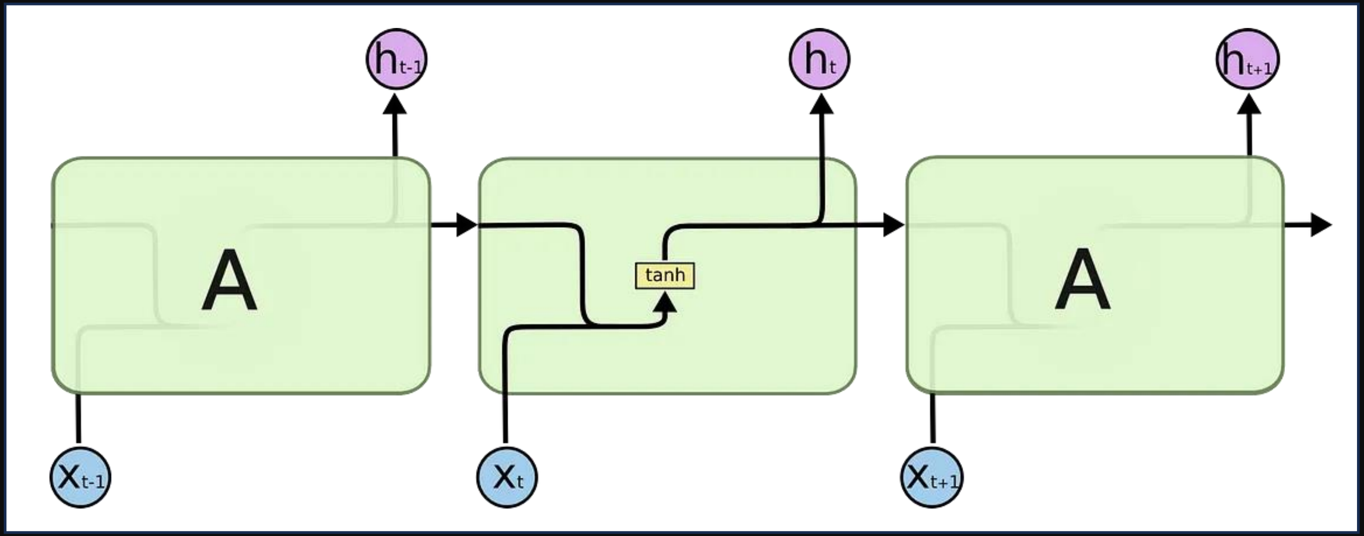




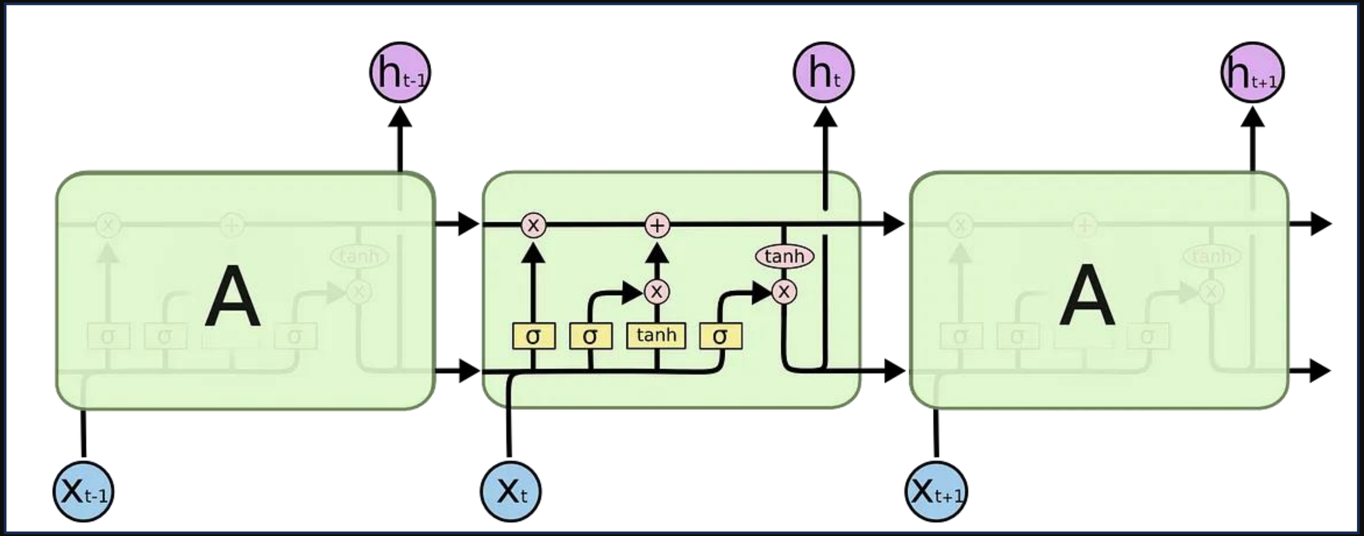
Les LSTM

- Les RNN à mémoire court-terme (LSTM) sont une variante populaire des RNN qui ont été développés pour résoudre les problèmes de dépendance à long terme des RNN à états cachés simples.
- Les LSTM introduisent des **mécanismes de porte qui leur permettent de mémoriser sélectivement l'information et de contrôler les flux d'information à travers les étapes temporelles**.

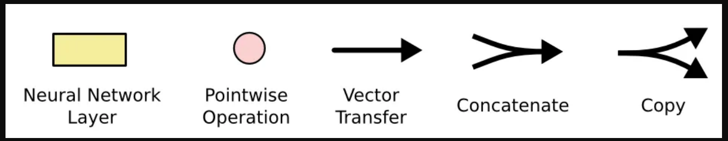
Les LSTM



ht-1
ht
ht+1
A
tanh
A
Xt-1
Xt
Xt+1

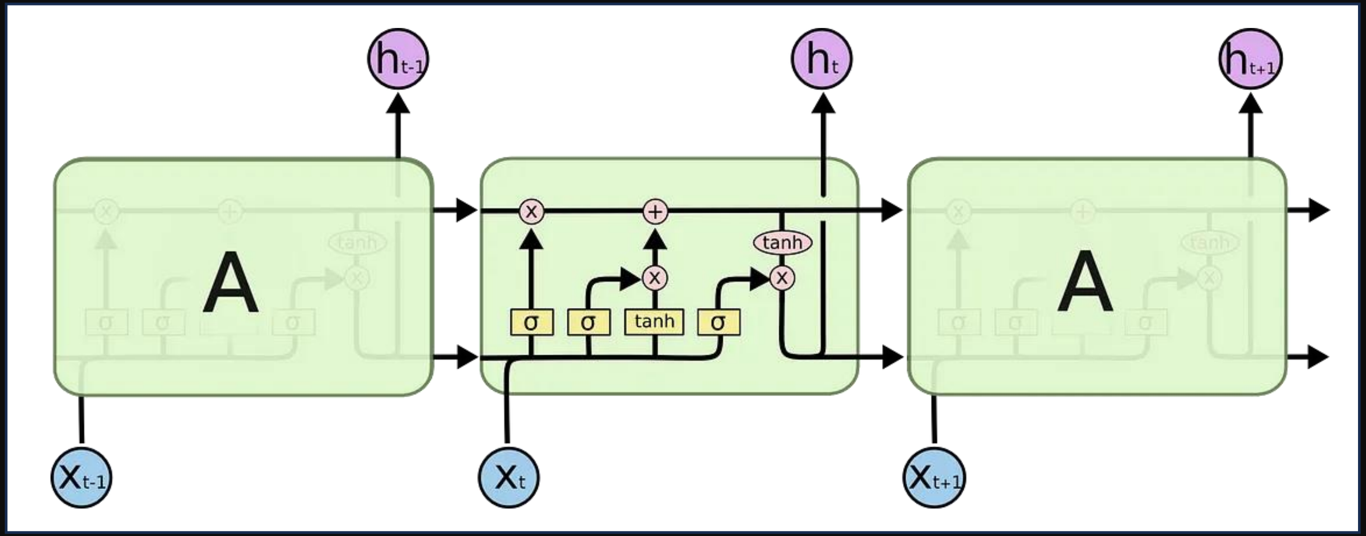


ht-1
ht
ht+1
A
σ
σ
tanh
σ
tanh
A
Xt-1
Xt
Xt+1

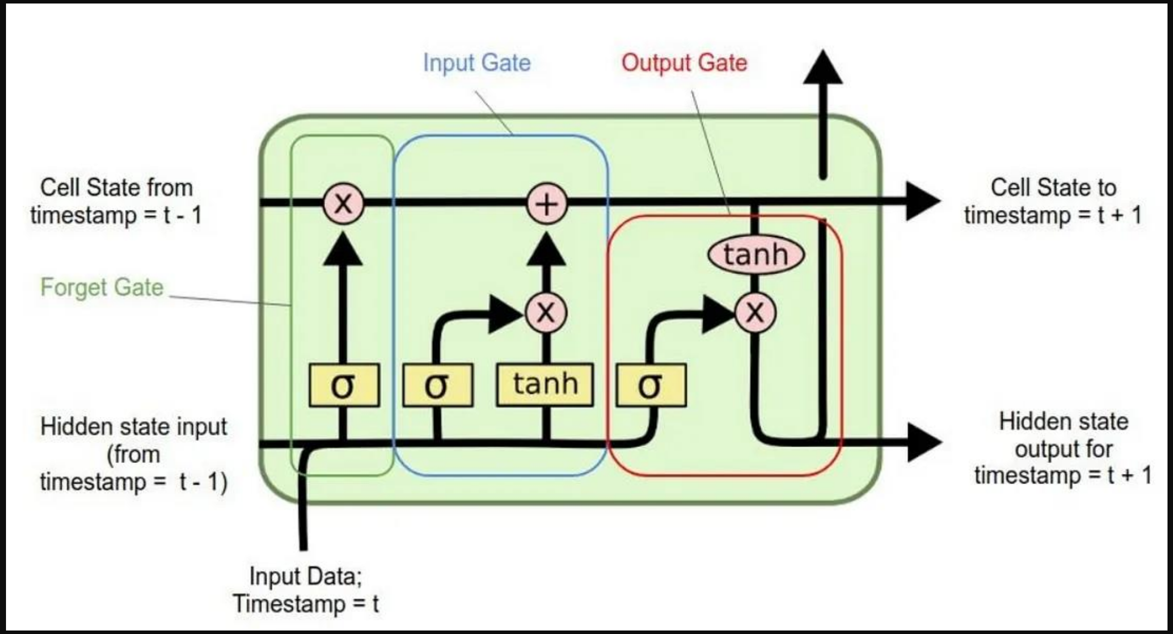


Neural Network Layer
Pointwise Operation
Vector Transfer
Concatenate
Copy

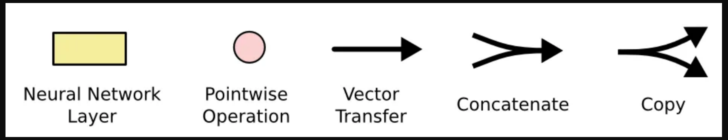
Les LSTM



h_{t-1}
h_t
h_{t+1}
A
A
σ
σ
tanh
σ
tanh
X_{t-1}
X_t
X_{t+1}



Input Gate
Output Gate
Cell State from
timestamp = t - 1
Cell State to
timestamp = t + 1
Forget Gate
tanh
σ
σ
tanh
σ
Hidden state input
(from
timestamp = t - 1)
Hidden state
output for
timestamp = t + 1
Input Data;
Timestamp = t



Neural Network
Layer
Pointwise
Operation
Vector
Transfer
Concatenate
Copy

Les modèles de fondation

Le paradigme :
Les modèles de fondation sont des modèles de grande taille pré-entraînés sur des données massives et hétérogènes, conçus pour être adaptés à de nombreuses tâches en aval par fine-tuning ou par prompting.

L'explosion de la puissance de calcul et l'avènement des transformers a rendu envisageable la création de modèles avec un si grand nombre de paramètres qu'ils capturent les patterns sous-jacents à une énorme variétés de datasets.

NLP, Traitement d'images : déjà indispensables

Données tabulaires, Time series : ne dépassent pas encore les meilleures méthodes, mais on s'y rapproche

La détection d’anomalies

Les types d'anomalies

Les anomalies ponctuelles sont des valeurs isolées aberrantes comme un pic ou une chute brutale.

Les anomalies contextuelles sont des valeurs anormales dans un contexte donné
Exemple : -7° en été

Les anomalies collectives sont des séquences entières anormales, pouvant correspondre à des changements de régime ou une dérive de capteurs.

Applications principales

Industrie / IoT : Détection de pannes, maintenance prédictive.

Finance : Fraude, détection de transactions suspectes.

Énergie : Anomalies de consommation, défauts de capteurs.

Santé : Surveillance de signaux physiologiques (ECG, EEG).

IT / Réseaux : Intrusions, anomalies de trafic, dégradation de performance

Les principales méthodes

1. Méthodes supervisées

- Utiles quand on a des labels d'anomalies, et un cas d'usage délimité
- SVM, Random Forest, RNN/LSTM

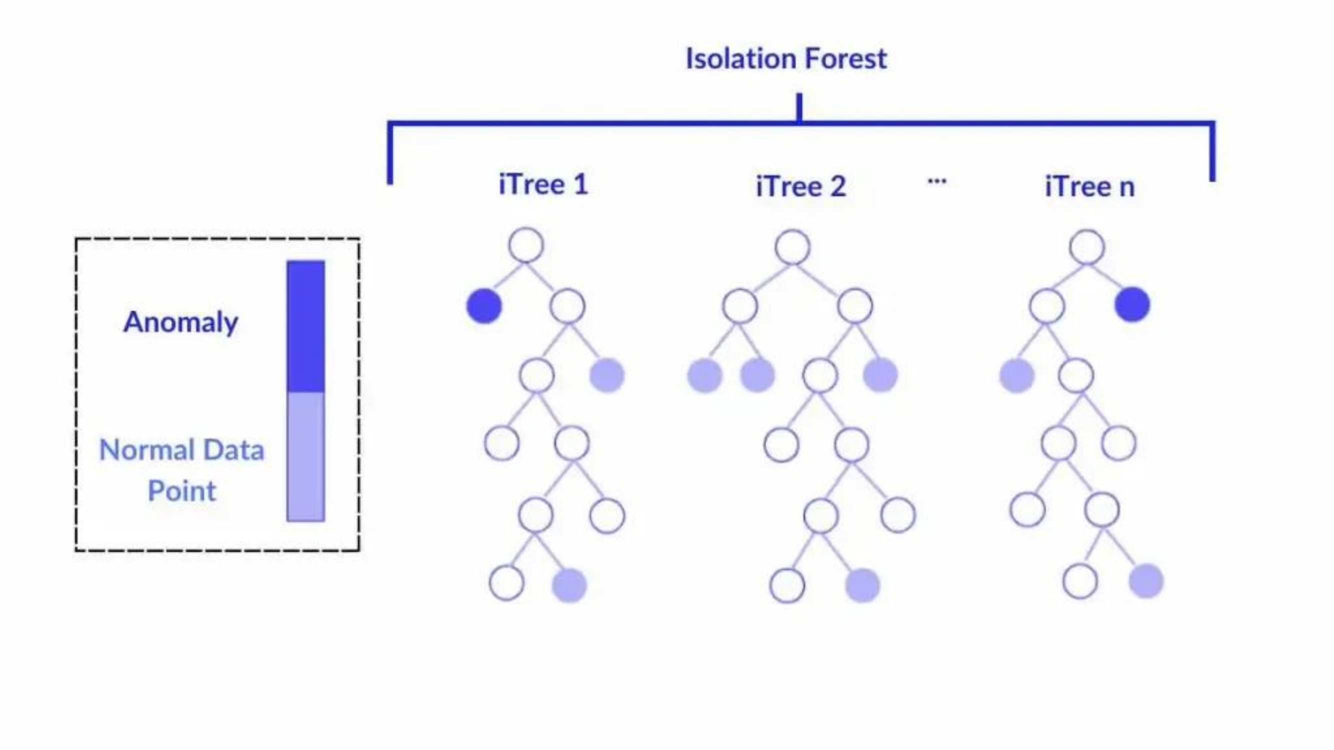
3. Méthodes non supervisées

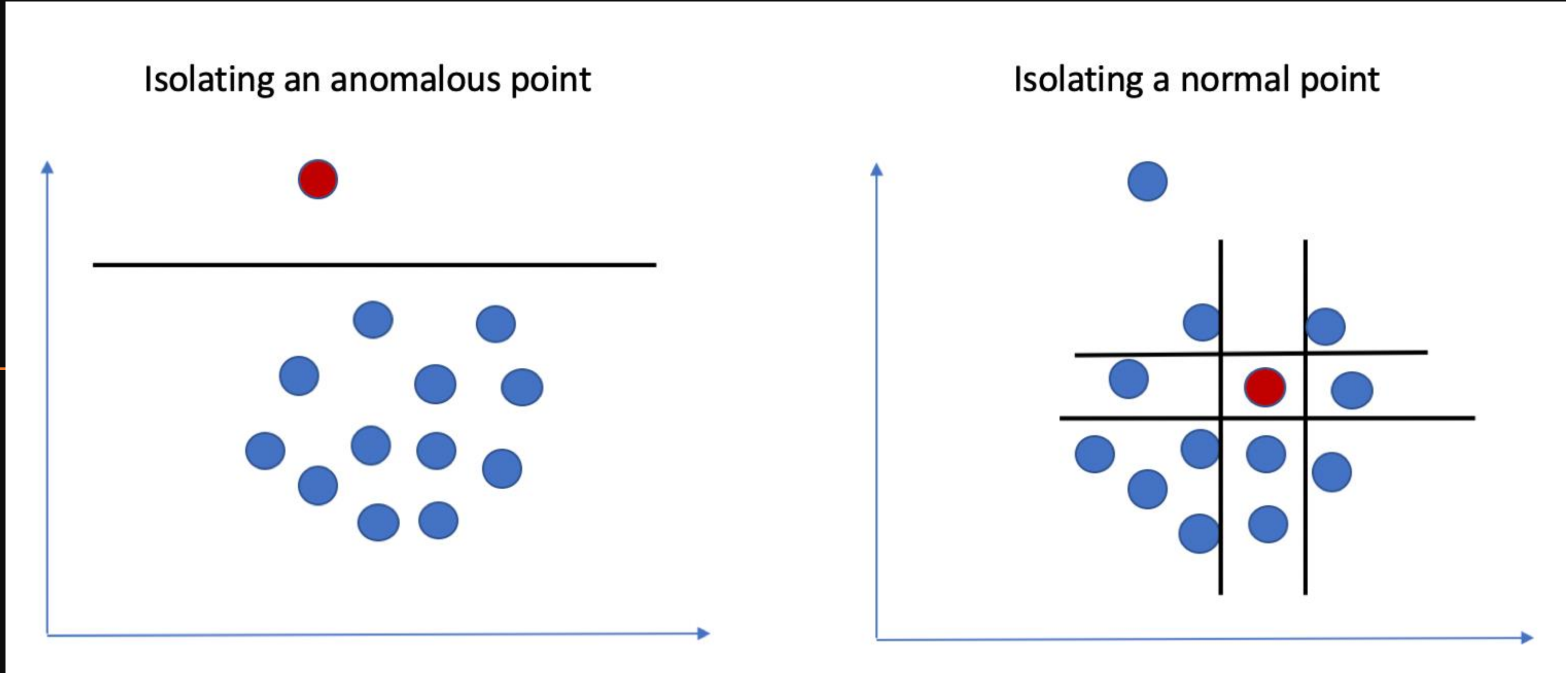
- Utiles quand les anomalies sont rares ou inconnues
- Isolation Forest, Auto-encodeur, SVM

Les Isolation Forest

- Méthode **non supervisée**
- Ne nécessite pas d'exemples d'anomalies
- Efficace en haute dimension
- Hypothèse clé : Les anomalies sont plus faciles à isoler que les observations normales
- Adaptation aux séries temporelles :
  - Extraction de features temporelles (fenêtrage, statistiques locales)
  - Application sur résidus de modèles de prévision
  - Détection d'anomalies ponctuelles ou de segments

Les Isolation Forest





Les Isolation Forest

Isolation Forest repose sur la construction d’arbres binaires aléatoires qui isolent les observations par des coupures successives sur des variables et des seuils choisis au hasard.

Les anomalies sont isolées plus rapidement que les observations normales, car elles se situent dans des régions peu denses de l’espace des données.

Le score d’anomalie est basé sur la profondeur moyenne d’isolement : plus cette profondeur est faible, plus l’observation est considérée comme anormale.

La méthode est rapide, scalable et ne fait aucune hypothèse sur la distribution des données.

En revanche, elle dépend fortement du choix des features temporelles et ne modélise pas explicitement la dynamique des séries temporelles.