

Chapitre 4 : Pré-traitement et Stockage des Données IoT

- 1 Importance du prétraitement des données IoT
- 2 Techniques de nettoyage des données : gestion des valeurs aberrantes et données manquantes
- 3 Normalisation et agrégation des données
- 4 Edge Computing : définition, architecture et avantages
- 5 Comparaison entre Edge Computing et Cloud Computing
- 6 Cas d'utilisation de l'Edge Computing dans l'IoT
- 7 Fog Computing : définition, architecture et cas d'utilisation
- 8 Comparaison entre Fog Computing, Edge Computing et Cloud Computing
- 9 Bases de données NoSQL adaptées à l'IoT : MongoDB, Cassandra
- 10 Bases de données de séries temporelles : InfluxDB, TimescaleDB
- 11 Choix des solutions de stockage selon les besoins des applications IoT

- **Importance du prétraitement des données IoT**
- Techniques de nettoyage des données : gestion des valeurs aberrantes et données manquantes
- Normalisation et agrégation des données
- Edge Computing : définition, architecture et avantages
- Comparaison entre Edge Computing et Cloud Computing
- Cas d'utilisation de l'Edge Computing dans l'IoT

- Fog Computing : définition, architecture et cas d'utilisation
- Comparaison entre Fog Computing, Edge Computing et Cloud Computing
- Bases de données NoSQL adaptées à l'IoT : MongoDB, Cassandra
- Bases de données de séries temporelles : InfluxDB, TimescaleDB
- Choix des solutions de stockage selon les besoins des applications IoT

Pourquoi le prétraitement est-il crucial ?



Impact sur l'IA & ML

- ✓ **Précision des Modèles** : Les données bruitées ou aberrantes faussent l'entraînement, menant à des prédictions erronées.
- ✓ **Garbage In, Garbage Out** : La qualité de l'output d'un algorithme est strictement limitée par la qualité de son input.
- ✓ **Robustesse** : Évite le surapprentissage (overfitting) sur des anomalies non représentatives du phénomène réel.



Optimisation des Ressources

- 🔧 **Économie de Bande Passante** : Filtrer et agréger à la source (Edge) réduit le volume de données transférées vers le Cloud.
- 🔧 **Réduction des Coûts de Stockage** : Ne stocker que l'utile permet une gestion financièrement viable des pétaoctets de données IoT.
- 🔧 **Latence Décisionnelle** : Des données propres et légères accélèrent le temps de traitement et la réactivité du système.

Le saviez-vous ?

Le nettoyage des données représente souvent **80% du temps** d'un projet de Data Science IoT.

Qualité

Efficacité

Fiabilité

- **Importance du prétraitement des données IoT**
- Techniques de nettoyage des données : gestion des valeurs aberrantes et données manquantes
- Normalisation et agrégation des données
- Edge Computing : définition, architecture et avantages
- Comparaison entre Edge Computing et Cloud Computing
- Cas d'utilisation de l'Edge Computing dans l'IoT

- Fog Computing : définition, architecture et cas d'utilisation
- Comparaison entre Fog Computing, Edge Computing et Cloud Computing
- Bases de données NoSQL adaptées à l'IoT : MongoDB, Cassandra
- Bases de données de séries temporelles : InfluxDB, TimescaleDB
- Choix des solutions de stockage selon les besoins des applications IoT

Nature et caractéristiques des données brutes

L'exploration des flux IoT à travers le paradigme des 3V



Volume Massif

Génération continue de données par des milliards d'objets connectés. Nécessite des capacités de stockage et de traitement à très grande échelle.



Vélocité Réelle

Flux de données à haute fréquence exigeant un traitement en temps quasi-réel pour une réactivité immédiate du système.



Variété Formats

Hétérogénéité des sources : données structurées (capteurs), semi-structurées (JSON/XML) ou non-structurées (flux vidéo).



Point de vigilance :

La donnée brute est rarement exploitable immédiatement. Sa nature hétérogène et son débit élevé imposent un prétraitement rigoureux avant toute analyse.

- Importance du prétraitement des données IoT
- **Techniques de nettoyage des données : gestion des valeurs aberrantes et données manquantes**
- Normalisation et agrégation des données
- Edge Computing : définition, architecture et avantages
- Comparaison entre Edge Computing et Cloud Computing
- Cas d'utilisation de l'Edge Computing dans l'IoT

- Fog Computing : définition, architecture et cas d'utilisation
- Comparaison entre Fog Computing, Edge Computing et Cloud Computing
- Bases de données NoSQL adaptées à l'IoT : MongoDB, Cassandra
- Bases de données de séries temporelles : InfluxDB, TimescaleDB
- Choix des solutions de stockage selon les besoins des applications IoT

Cycle de vie de la donnée avant stockage

1



ACQUISITION

Captation des signaux analogiques et conversion numérique par les capteurs. Génération de métadonnées de base.

2



FILTRAGE LOCAL

Validation immédiate au niveau **Edge** : élimination des erreurs de lecture évidentes et vérification d'intégrité.

3



NETTOYAGE

Traitement statistique : **détection des outliers**, imputation des valeurs **manquantes** et **normalisation** des échelles.

4



AGRÉGATION

Réduction stratégique du volume : calcul de moyennes, min/max par fenêtre temporelle avant l'ingestion finale.



Objectif Stratégique

Chaque étape vise à transformer une **donnée brute (bruitée et volumineuse)** en une **donnée exploitable (propre et synthétique)**, minimisant ainsi les coûts de stockage et maximisant la précision des modèles d'IA.

- Importance du prétraitement des données IoT
- **Techniques de nettoyage des données : gestion des valeurs aberrantes et données manquantes**
- Normalisation et agrégation des données
- Edge Computing : définition, architecture et avantages
- Comparaison entre Edge Computing et Cloud Computing
- Cas d'utilisation de l'Edge Computing dans l'IoT

- Fog Computing : définition, architecture et cas d'utilisation
- Comparaison entre Fog Computing, Edge Computing et Cloud Computing
- Bases de données NoSQL adaptées à l'IoT : MongoDB, Cassandra
- Bases de données de séries temporelles : InfluxDB, TimescaleDB
- Choix des solutions de stockage selon les besoins des applications IoT

I. les *Outliers*

+ Qu'est-ce qu'un Outlier ?

Les **valeurs aberrantes** sont des observations très éloignées du comportement général des données.

Distinction Cruciale pour l'Analyste

Il est impératif de distinguer les **"Noise Outliers"** (erreurs à supprimer) des **"Event Outliers"** (anomalies critiques à signaler). Un mauvais nettoyage peut supprimer l'information la plus précieuse du flux IoT.



POINT DE VIGILANCE

"Tous les outliers ne sont pas des erreurs. Certains sont les signaux les plus précieux d'une défaillance imminente ou d'un changement de régime système."

- Importance du prétraitement des données IoT
- **Techniques de nettoyage des données : gestion des valeurs aberrantes et données manquantes**
- Normalisation et agrégation des données
- Edge Computing : définition, architecture et avantages
- Comparaison entre Edge Computing et Cloud Computing
- Cas d'utilisation de l'Edge Computing dans l'IoT

- Fog Computing : définition, architecture et cas d'utilisation
- Comparaison entre Fog Computing, Edge Computing et Cloud Computing
- Bases de données NoSQL adaptées à l'IoT : MongoDB, Cassandra
- Bases de données de séries temporelles : InfluxDB, TimescaleDB
- Choix des solutions de stockage selon les besoins des applications IoT

Origines et causes des Outliers IoT

Pourquoi nos capteurs génèrent-ils des valeurs aberrantes ?



Erreurs de saisie et de mesure

- Dérive des capteurs due à l'usure physique (Aging).
- Chute de tension de la batterie provoquant des lectures erronées.
- Défauts de calibration en usine ou sur site.



Erreurs de Transmission

- Interférences électromagnétiques (EMI) sur le signal.
- Corruption binaire lors de l'encapsulation des paquets.
- Latence réseau provoquant des désynchronisations temporelles.



Événements Réels

- Phénomènes rares mais valides (incendie, fuite, séisme).
- Pannes soudaines d'équipements industriels surveillés.
- Changements contextuels brusques (ouverture d'une porte).

- Importance du prétraitement des données IoT
- **Techniques de nettoyage des données : gestion des valeurs aberrantes et données manquantes**
- Normalisation et agrégation des données
- Edge Computing : définition, architecture et avantages
- Comparaison entre Edge Computing et Cloud Computing
- Cas d'utilisation de l'Edge Computing dans l'IoT

- Fog Computing : définition, architecture et cas d'utilisation
- Comparaison entre Fog Computing, Edge Computing et Cloud Computing
- Bases de données NoSQL adaptées à l'IoT : MongoDB, Cassandra
- Bases de données de séries temporelles : InfluxDB, TimescaleDB
- Choix des solutions de stockage selon les besoins des applications IoT

I.1. Détection des valeurs aberrantes

Boxplot

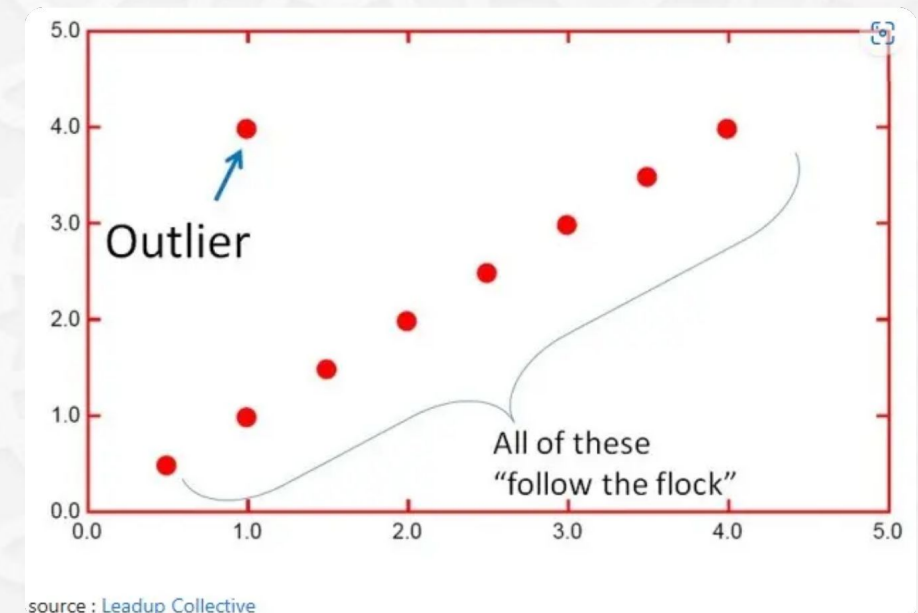
Visualise les quartiles et les points isolés au-delà des moustaches.

Scatter Plots

Révèle les clusters isolés et les relations atypiques entre variables.

Histogrammes

Montre les queues de distribution et les pics anormaux.



ANALYSE EXPLORATOIRE (EDA)

Permet de détecter des anomalies structurelles (clusters isolés, erreurs de saisie).

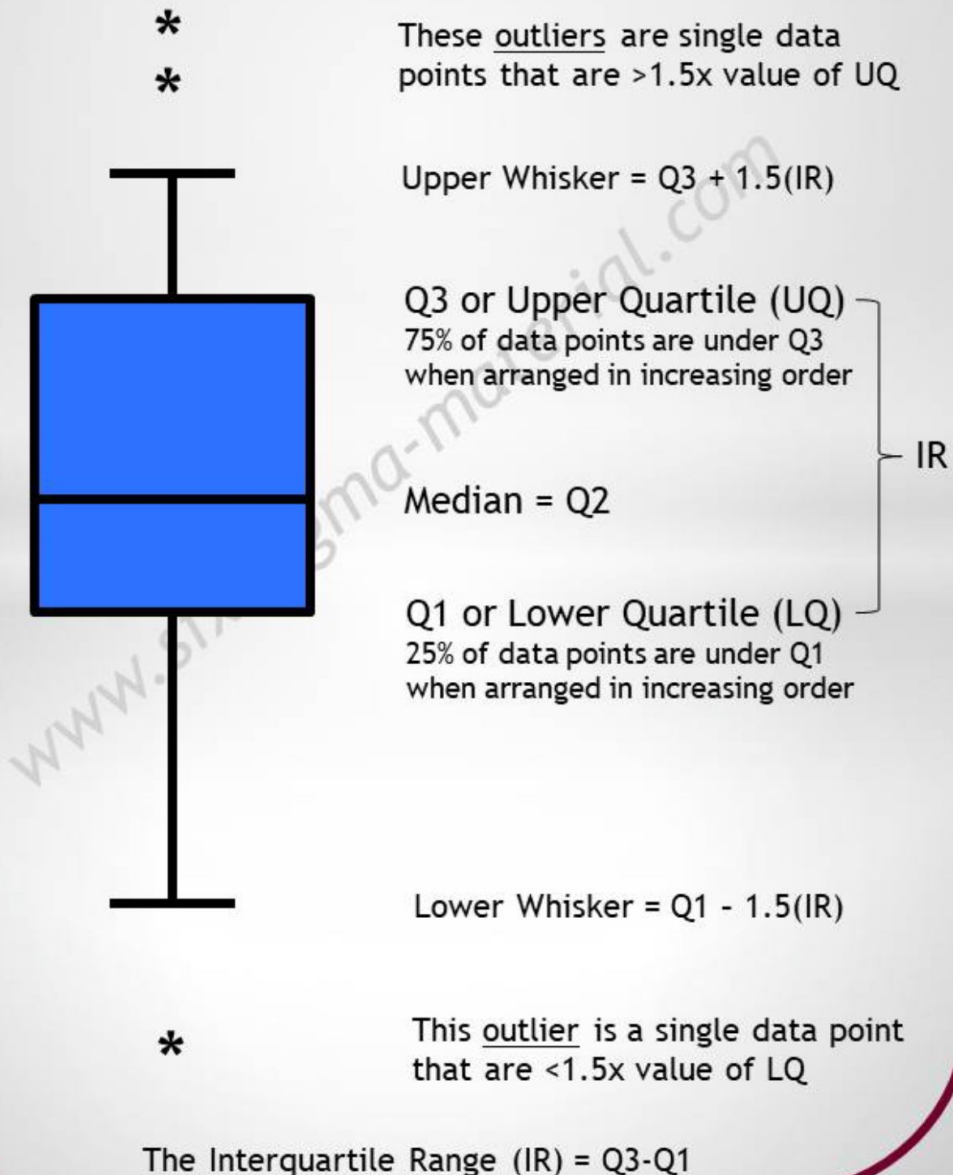
- Importance du prétraitement des données IoT
- Techniques de nettoyage des données : gestion des valeurs aberrantes et données manquantes
- Normalisation et agrégation des données
- Edge Computing : définition, architecture et avantages
- Comparaison entre Edge Computing et Cloud Computing
- Cas d'utilisation de l'Edge Computing dans l'IoT

- Fog Computing : définition, architecture
- Comparaison entre Fog Computing, Edge Computing et Cloud Computing
- Bases de données NoSQL adaptées à l'IoT
- Bases de données de séries temporelles
- Choix des solutions de stockage selon les cas d'usage

1. Détection des valeurs aberrantes:

1.1.1. Méthodes graphiques

- **boxplot:** Les points en dehors des moustaches sont des valeurs aberrantes.



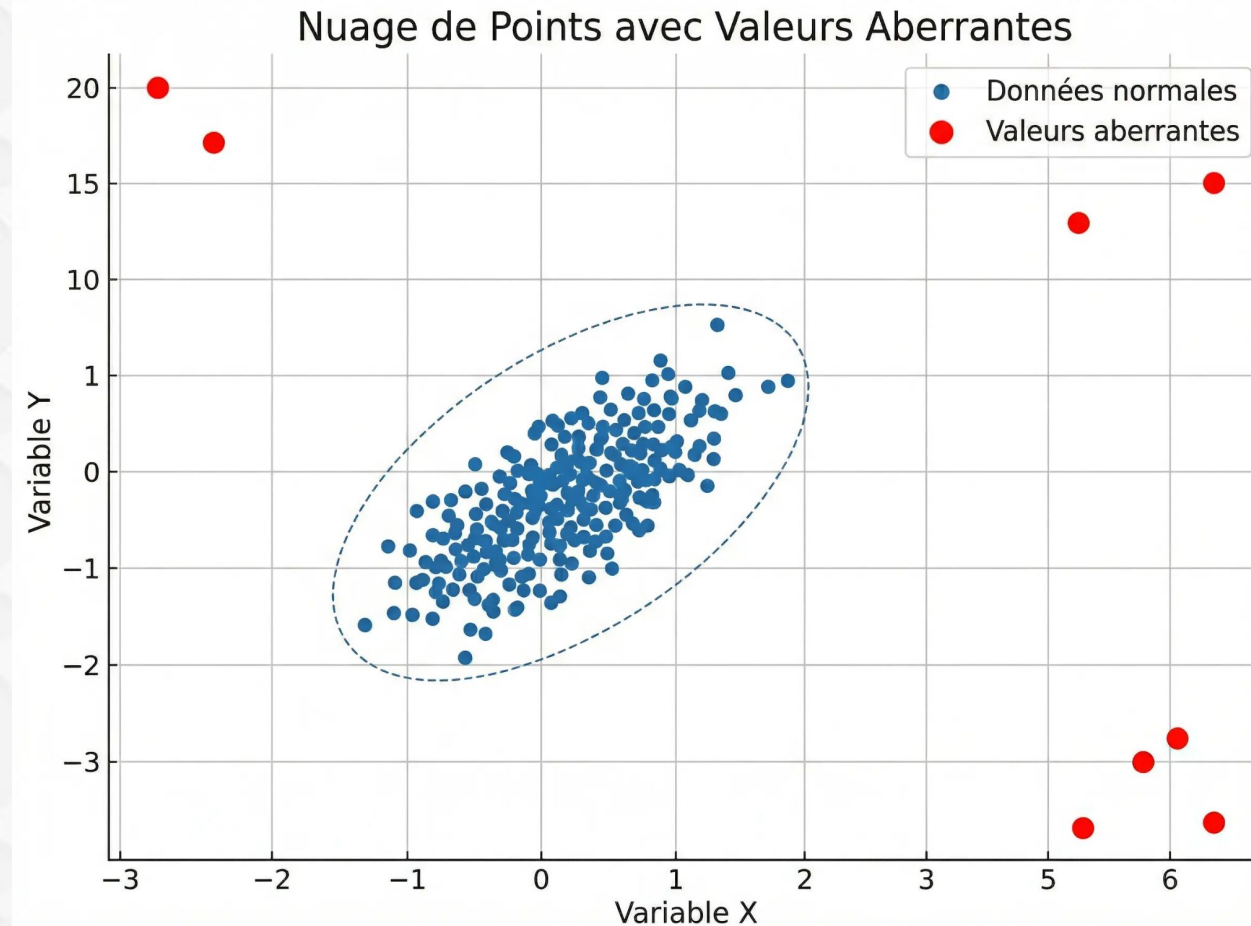
- Importance du prétraitement des données IoT
- **Techniques de nettoyage des données : gestion des valeurs aberrantes et données manquantes**
- Normalisation et agrégation des données
- Edge Computing : définition, architecture et avantages
- Comparaison entre Edge Computing et Cloud Computing
- Cas d'utilisation de l'Edge Computing dans l'IoT

- Fog Computing : définition, architecture et cas d'utilisation
- Comparaison entre Fog Computing, Edge Computing et Cloud Computing
- Bases de données NoSQL adaptées à l'IoT : MongoDB, Cassandra
- Bases de données de séries temporelles : InfluxDB, TimescaleDB
- Choix des solutions de stockage selon les besoins des applications IoT

Détection des valeurs aberrantes:

I.1.1. Méthodes graphiques

- **nuage de points:** Les outliers sont des points éloignés du groupe principal.



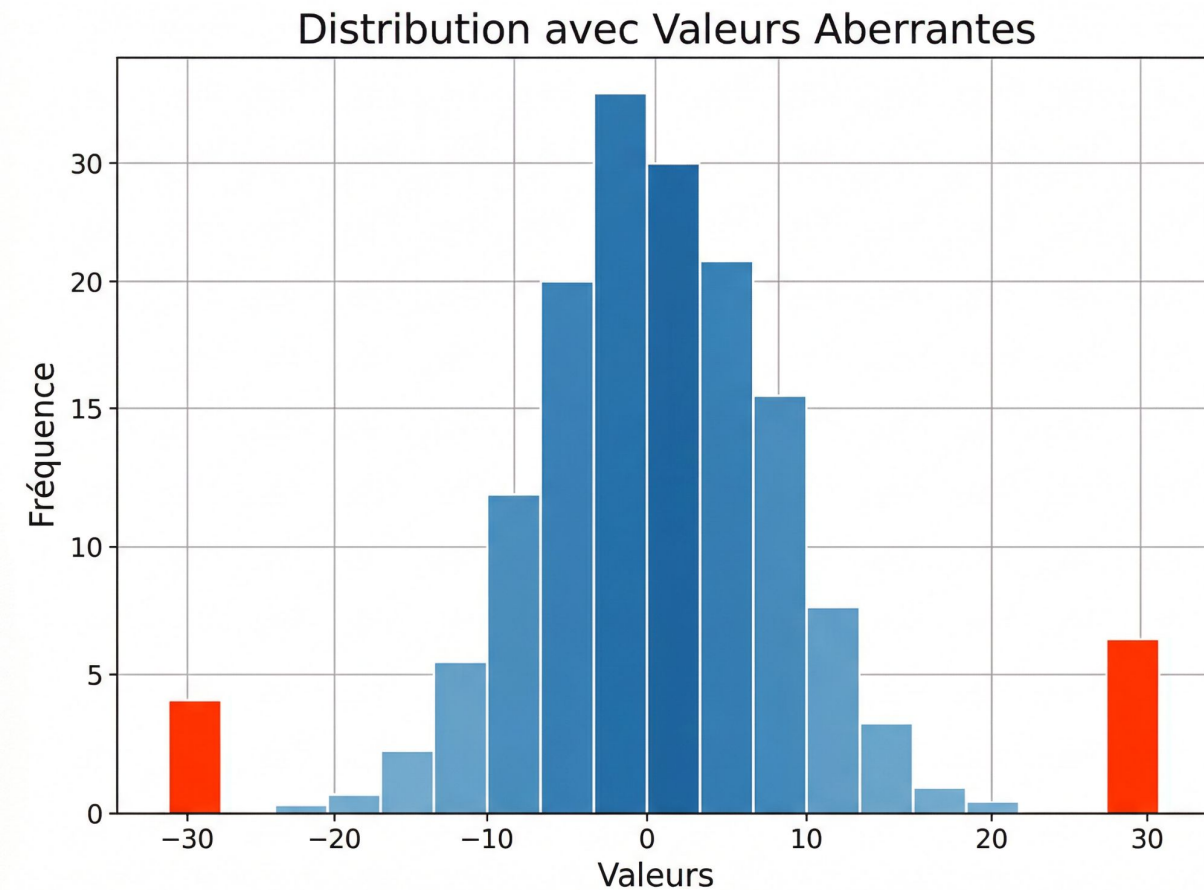
- Importance du prétraitement des données IoT
- **Techniques de nettoyage des données : gestion des valeurs aberrantes et données manquantes**
- Normalisation et agrégation des données
- Edge Computing : définition, architecture et avantages
- Comparaison entre Edge Computing et Cloud Computing
- Cas d'utilisation de l'Edge Computing dans l'IoT

- Fog Computing : définition, architecture et cas d'utilisation
- Comparaison entre Fog Computing, Edge Computing et Cloud Computing
- Bases de données NoSQL adaptées à l'IoT : MongoDB, Cassandra
- Bases de données de séries temporelles : InfluxDB, TimescaleDB
- Choix des solutions de stockage selon les besoins des applications IoT

Détection des valeurs aberrantes:

I.1.1. Méthodes graphiques

- **histogramme**: Les valeurs aberrantes apparaissent comme barres isolées loin du reste.



- Importance du prétraitement des données IoT
- **Techniques de nettoyage des données : gestion des valeurs aberrantes et données manquantes**
- Normalisation et agrégation des données
- Edge Computing : définition, architecture et avantages
- Comparaison entre Edge Computing et Cloud Computing
- Cas d'utilisation de l'Edge Computing dans l'IoT

- Fog Computing : définition, architecture et cas d'utilisation
- Comparaison entre Fog Computing, Edge Computing et Cloud Computing
- Bases de données NoSQL adaptées à l'IoT : MongoDB, Cassandra
- Bases de données de séries temporelles : InfluxDB, TimescaleDB
- Choix des solutions de stockage selon les besoins des applications IoT

Détection des valeurs aberrantes:

I.1.2. Méthodes statistiques: **écart interquartile (IQR)**

√x Règle de Détection

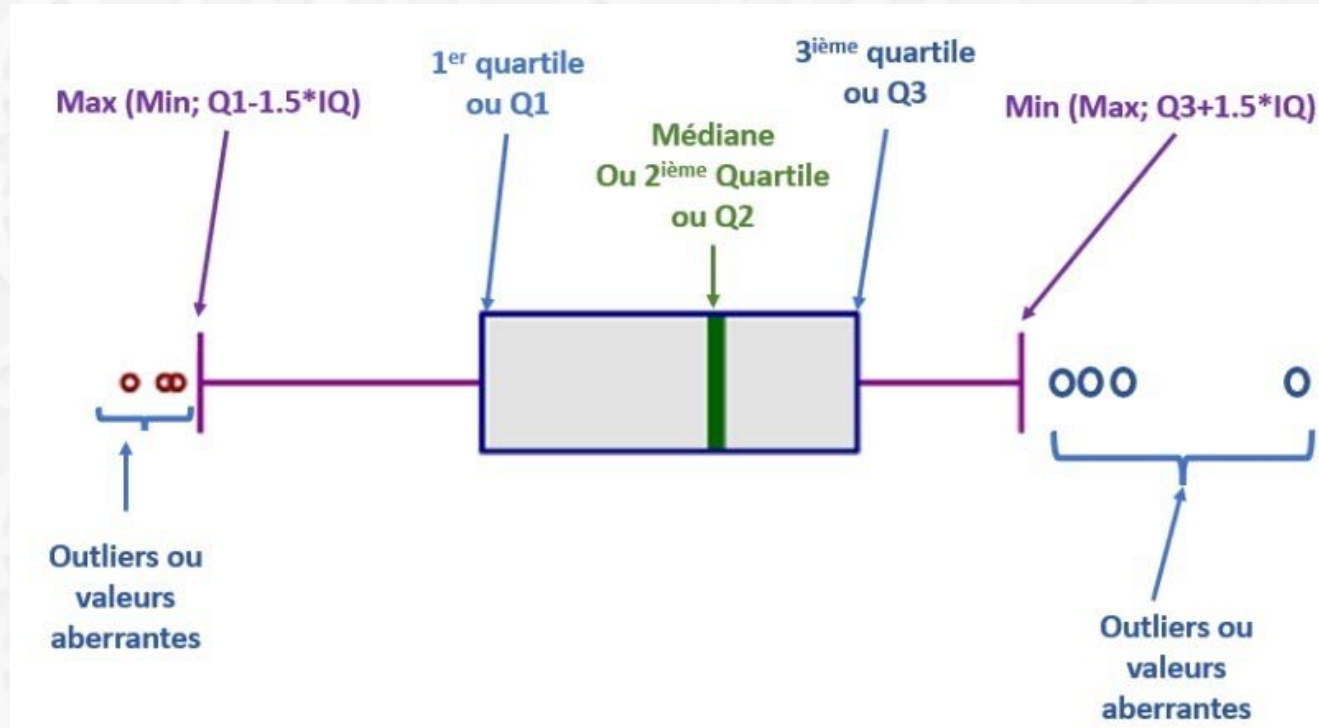
$$\text{IQR} = Q3 - Q1$$

Une valeur est aberrante si :

$$\text{Valeur} < Q1 - 1.5 \times \text{IQR} \text{ OU } \text{Valeur} > Q3 + 1.5 \times \text{IQR}$$

🛡 Pourquoi l'utiliser ?

- ✓ **Insensibilité aux extrêmes** : Contrairement à la moyenne, la médiane et les quartiles ne sont pas "tirés" par les outliers.
- ✓ **Non-paramétrique** : Ne nécessite pas que les données suivent une distribution normale (Loi de Gauss).
- ✓ **Standard de l'industrie** : Base du diagramme en boîtes (Boxplot) pour la visualisation rapide.



- Importance du prétraitement des données IoT
- **Techniques de nettoyage des données** : gestion des valeurs aberrantes et données manquantes
- Normalisation et agrégation des données
- Edge Computing : définition, architecture et avantages
- Comparaison entre Edge Computing et Cloud Computing
- Cas d'utilisation de l'Edge Computing dans l'IoT

- Fog Computing : définition, architecture et cas d'utilisation
- Comparaison entre Fog Computing, Edge Computing et Cloud Computing
- Bases de données NoSQL adaptées à l'IoT : MongoDB, Cassandra
- Bases de données de séries temporelles : InfluxDB, TimescaleDB
- Choix des solutions de stockage selon les besoins des applications IoT

$$Z = \frac{x - \mu}{\sigma}$$

x: Valeur μ : Moyenne σ : Écart-type

Détection des valeurs aberrantes:

I.1.2. Méthodes statistiques

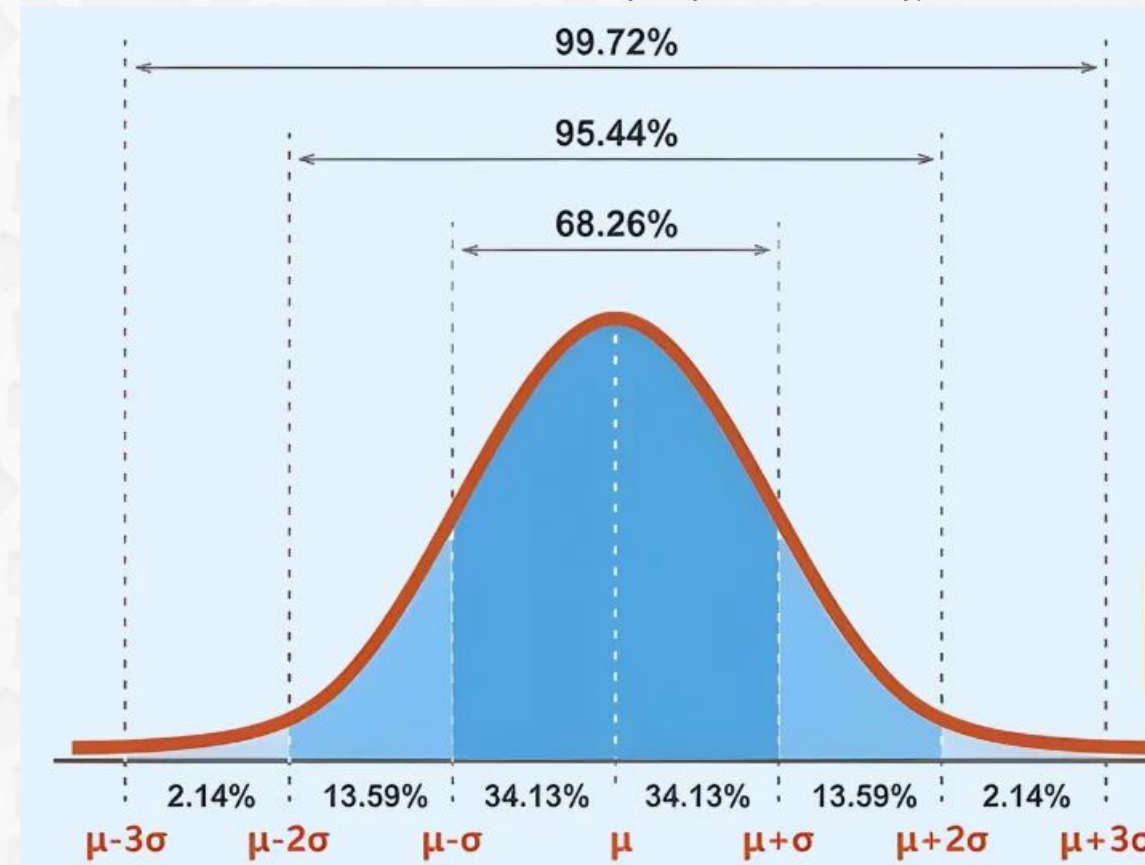
- **Z-Score**: Le mesure z-score combien d'écarts-types une donnée s'éloigne de la moyenne. Il transforme les données brutes en une échelle standardisée.

⚠ Seuil d'Anomalie

Une valeur est généralement considérée comme **aberrante (outlier)** si :

$$|Z| > 3$$

Ceci couvre 99.7% des données dans une distribution normale.



Méthodes statistiques 02/03

- Importance du prétraitement des données IoT
- **Techniques de nettoyage des données : gestion des valeurs aberrantes et données manquantes**
- Normalisation et agrégation des données
- Edge Computing : définition, architecture et avantages
- Comparaison entre Edge Computing et Cloud Computing
- Cas d'utilisation de l'Edge Computing dans l'IoT

- Fog Computing : définition, architecture et cas d'utilisation
- Comparaison entre Fog Computing, Edge Computing et Cloud Computing
- Bases de données NoSQL adaptées à l'IoT : MongoDB, Cassandra
- Bases de données de séries temporelles : InfluxDB, TimescaleDB
- Choix des solutions de stockage selon les besoins des applications IoT

- isolation forest
- DBSCAN pour certains cas multivariés

Détection des valeurs aberrantes:

I.1.2. Méthodes statistiques (**méthodes robustes**):

Score MAD (Median Absolute Deviation): Le score MAD ou l'écart absolu médian est une alternative robuste au Z-Score. Il utilise la médiane au lieu de la moyenne, ce qui le rend insensible aux valeurs extrêmes.

√x Formule du MAD

$$\text{MAD} = \text{median}(|x_i - \text{median}(X)|)$$

On multiplie souvent par un Facteur de correction (ex: 1.4826) pour le rendre comparable à l'écart-type (si les données suivent une loi normale)..

$$Z_{\text{robuste}} = \frac{x_i - \text{median}(x)}{c \cdot \text{MAD}}$$

x: Valeur c: Facteur de correction

⚠ Seuil d'Anomalie

Une valeur est généralement considérée comme **aberrante (outlier)** si :

$$|Z_{\text{robuste}}| > k, \text{ tel que } k=3$$

Outlier si $|x - \text{mediane}| > k \times \text{MAD}$

☞ Conseil d'expert :

Utilisez le MAD lorsque vos données ne sont pas parfaitement gaussiennes ou contiennent de nombreux pics aberrants.

- Importance du prétraitement des données IoT
- **Techniques de nettoyage des données : gestion des valeurs aberrantes et données manquantes**
- Normalisation et agrégation des données
- Edge Computing : définition, architecture et avantages
- Comparaison entre Edge Computing et Cloud Computing
- Cas d'utilisation de l'Edge Computing dans l'IoT

- Fog Computing : définition, architecture et cas d'utilisation
- Comparaison entre Fog Computing, Edge Computing et Cloud Computing
- Bases de données NoSQL adaptées à l'IoT : MongoDB, Cassandra
- Bases de données de séries temporelles : InfluxDB, TimescaleDB
- Choix des solutions de stockage selon les besoins des

Détection des valeurs aberrantes:

I.1.3. Méthodes avancées(Machine Learning)

Algorithmes pour données complexes ou multidimensionnelles.

🌲 Isolation Forest

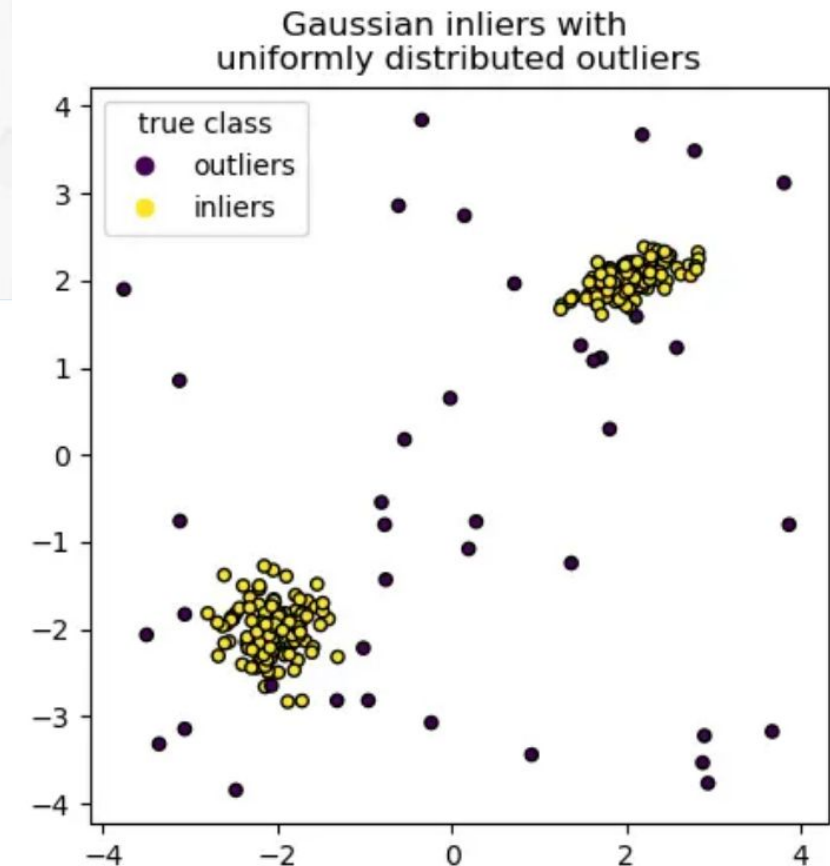
Isole les anomalies en construisant des arbres de décision. Les outliers sont plus faciles à isoler (chemins plus courts).

🗺️ DBSCAN

Algorithme de clustering basé sur la densité. Identifie les points isolés comme du bruit (Noise).

📍 Local Outlier Factor (LOF)

Compare la densité locale d'un point à celle de ses voisins. Détecte les anomalies relatives au contexte local.



PERFORMANCE MULTIVARIÉE

Idéal pour les flux IoT combinant Température, Pression et Humidité.

- Importance du prétraitement des données IoT
- **Techniques de nettoyage des données : gestion des valeurs aberrantes et données manquantes**
- Normalisation et agrégation des données
- Edge Computing : définition, architecture et avantages
- Comparaison entre Edge Computing et Cloud Computing
- Cas d'utilisation de l'Edge Computing dans l'IoT

- Fog Computing : définition, architecture et cas d'utilisation
- Comparaison entre Fog Computing, Edge Computing et Cloud Computing
- Bases de données NoSQL adaptées à l'IoT : MongoDB, Cassandra
- Bases de données de séries temporelles : InfluxDB, TimescaleDB
- Choix des solutions de stockage selon les besoins des applications IoT

Détection des valeurs aberrantes:

I.1.3. Méthodes avancées(Machine Learning): Isolation Forest

Isolation Forest est un algorithme qui isole les données anormales au lieu de mesurer la distance.

"Les anomalies sont peu nombreuses et différentes, ce qui les rend plus faciles à isoler."

MÉCANISME DE DÉTECTION

Points Normaux

PROFONDEUR D'ARBRE ÉLEVÉE
(DIFFICILE À ISOLER)

Anomalies

FAIBLE PROFONDEUR (ISOLÉ
RAPIDEMENT)



Haute Performance

Complexité linéaire $O(n)$, idéal pour les flux de données IoT massifs.



Multidimensionnel

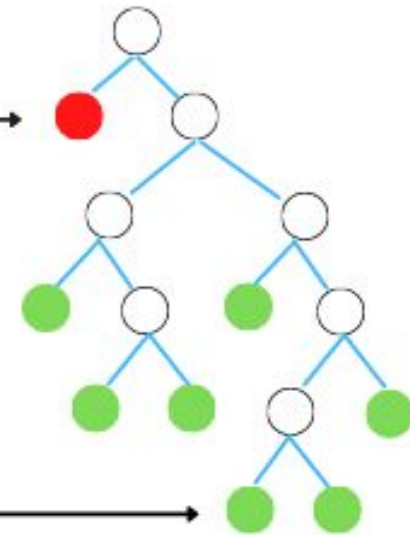
Efficace pour corréler plusieurs capteurs simultanément.

Easy to isolate



Depth

Hard to isolate



L'APPROCHE MACHINE LEARNING POUR LES OUTLIERS

- Importance du prétraitement des données IoT
- **Techniques de nettoyage des données** : gestion des valeurs aberrantes et données manquantes
- Normalisation et agrégation des données
- Edge Computing : définition, architecture et avantages
- Comparaison entre Edge Computing et Cloud Computing
- Cas d'utilisation de l'Edge Computing dans l'IoT

- Fog Computing : définition, architecture et cas d'utilisation
- Comparaison entre Fog Computing, Edge Computing et Cloud Computing
- Bases de données NoSQL adaptées à l'IoT : MongoDB, Cassandra
- Bases de données de séries temporelles : InfluxDB, TimescaleDB
- Choix des solutions de stockage selon les besoins des applications IoT

Détection des valeurs aberrantes:

1.3. Méthodes avancées: DBSCAN, (Density-Based Spatial Clustering of Applications with Noise) **Cas Multivariés**, Il regroupe les points qui sont dans des zones de forte densité et considère les points isolés comme du bruit.

Pourquoi DBSCAN ?

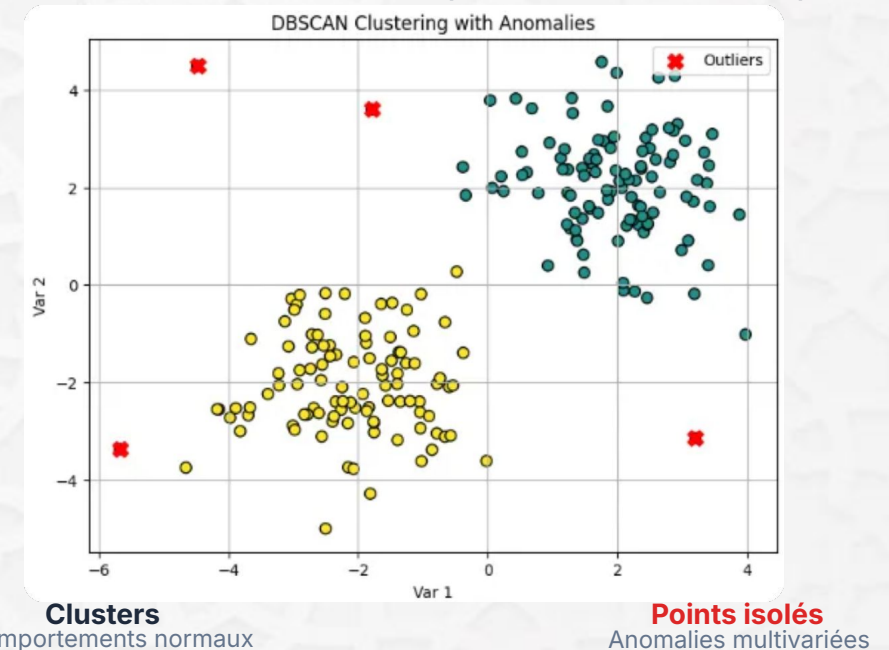
Idéal pour détecter des anomalies qui ne sont pas aberrantes sur une seule variable, mais qui le deviennent par leur **combinaison**.

Exemple : Une température de 30°C est normale, une pression de 1 bar est normale. Mais 30°C avec 1 bar peut être une anomalie critique.

Paramètres Clés

- 🕒 **Epsilon (ϵ)** : Rayon de voisinage.
- 👥 **MinPts** : Nombre minimum de points pour former un cluster.
- 🚫 **Bruit (Noise)** : Points isolés considérés comme outliers.

VISUALISATION MULTIVARIÉE (TEMPÉRATURE VS PRESSION)



DÉTECTION D'ANOMALIES PAR DENSITÉ SPATIALE

- Importance du prétraitement des données IoT
- **Techniques de nettoyage des données : gestion des valeurs aberrantes et données manquantes**
- Normalisation et agrégation des données
- Edge Computing : définition, architecture et avantages
- Comparaison entre Edge Computing et Cloud Computing
- Cas d'utilisation de l'Edge Computing dans l'IoT

- Fog Computing : définition, architecture et cas d'utilisation
- Comparaison entre Fog Computing, Edge Computing et Cloud Computing
- Bases de données NoSQL adaptées à l'IoT : MongoDB, Cassandra
- Bases de données de séries temporelles : InfluxDB, TimescaleDB
- Choix des solutions de stockage selon les besoins des applications IoT

Détection des valeurs aberrantes:

1.3. Méthodes avancées: **Local Outlier Factor (LOF)**

- Importance du prétraitement des données IoT
- **Techniques de nettoyage des données : gestion des valeurs aberrantes et données manquantes**
- Normalisation et agrégation des données
- Edge Computing : définition, architecture et avantages
- Comparaison entre Edge Computing et Cloud Computing
- Cas d'utilisation de l'Edge Computing dans l'IoT

- Fog Computing : définition, architecture et cas d'utilisation
- Comparaison entre Fog Computing, Edge Computing et Cloud Computing
- Bases de données NoSQL adaptées à l'IoT : MongoDB, Cassandra
- Bases de données de séries temporelles : InfluxDB, TimescaleDB
- Choix des solutions de stockage selon les besoins des applications IoT

I.2. Traitement des outliers (suppression, winsorisation et Imputation).



on enlève les outliers

a. Suppression

Retrait définitif des observations identifiées comme aberrantes du jeu de données.

USAGE IDÉAL

Erreurs de mesure, Buit évidente.

- ✓ Simple à appliquer
- ⚠ Risque de perte d'info

on les écrase aux limites

b. Capping / Winsorisation

Remplacer les extrêmes par des quantiles (ex: 1% / 99%) pour limiter l'impact sans supprimer de lignes.

USAGE IDÉAL

Données avec bruit persistant.

- ✓ Garde toutes les observations
- ⚠ Peut déformer légèrement la distribution

```
import numpy as np
df['x'] = np.clip(df['x'],
                 df['x'].quantile(0.01),
                 df['x'].quantile(0.99))
```



Remplacer par une valeur centrale

C. Imputation (Médiane)

Remplacement de l'outlier par la valeur centrale ou l'interpolation des voisins.

USAGE IDÉAL

Séries temporelles continues ou données où la continuité est importante.

- ✓ Préserve la taille du dataset
- ⚠ Perte de l'information extrême

```
median = df['x'].median()
df.loc[(df['x'] < q_low) | (df['x'] >
q_high), 'x'] = median
```


- Importance du prétraitement des données IoT
- **Techniques de nettoyage des données : gestion des valeurs aberrantes et données manquantes**
- Normalisation et agrégation des données
- Edge Computing : définition, architecture et avantages
- Comparaison entre Edge Computing et Cloud Computing
- Cas d'utilisation de l'Edge Computing dans l'IoT

- Fog Computing : définition, architecture et cas d'utilisation
- Comparaison entre Fog Computing, Edge Computing et Cloud Computing
- Bases de données NoSQL adaptées à l'IoT : MongoDB, Cassandra
- Bases de données de séries temporelles : InfluxDB, TimescaleDB
- Choix des solutions de stockage selon les besoins des applications IoT

I.2. Traitement des outliers: Transformations

Réduire l'impact des valeurs extrêmes par changement d'échelle.



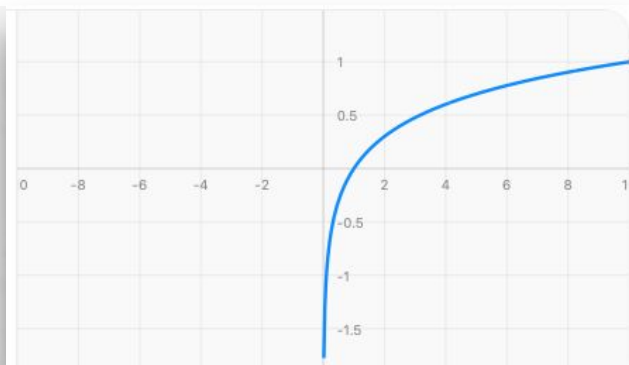
Pourquoi transformer ?

Ces méthodes ne suppriment pas les outliers mais les "rapprochent" du reste de la distribution, rendant les modèles linéaires plus stables et performants.

✧ Logarithmique (T.Forte)

$$y = \log(x)$$

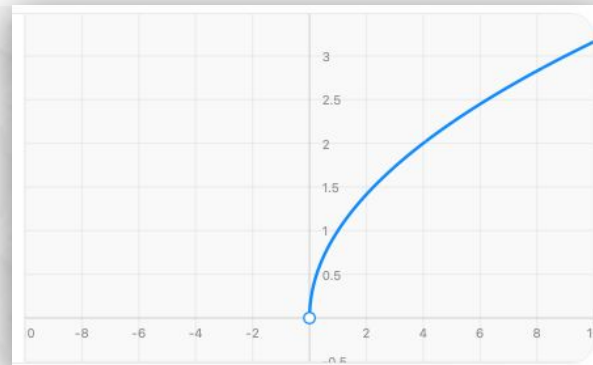
Comprime les grandes valeurs et réduit l'asymétrie (**skewness**) positive.



✧ Racine Carrée (T.Moyenne)

$$y = \sqrt{x}$$

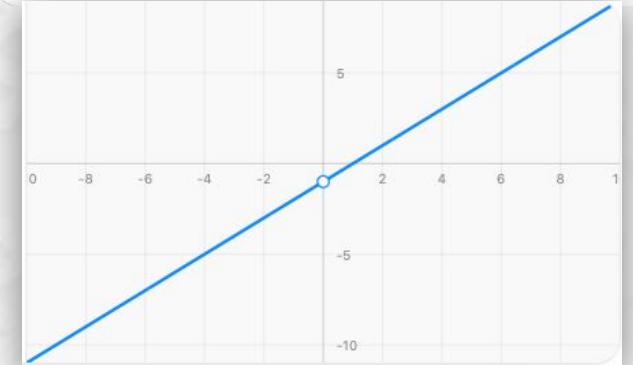
Moins agressive que le log, utile pour les **données de comptage** (nombre d'événements, occurrences).



✎ Box-Cox (T.Adaptative)

$$y = (x^\lambda - 1) / \lambda$$

Transformation paramétrique qui optimise la **normalité** de la distribution.



- Importance du prétraitement des données IoT
- **Techniques de nettoyage des données : gestion des valeurs aberrantes et données manquantes**
- Normalisation et agrégation des données
- Edge Computing : définition, architecture et avantages
- Comparaison entre Edge Computing et Cloud Computing
- Cas d'utilisation de l'Edge Computing dans l'IoT

- Fog Computing : définition, architecture et cas d'utilisation
- Comparaison entre Fog Computing, Edge Computing et Cloud Computing
- Bases de données NoSQL adaptées à l'IoT : MongoDB, Cassandra
- Bases de données de séries temporelles : InfluxDB, TimescaleDB
- Choix des solutions de stockage selon les besoins des applications IoT

I.2. Traitement des outliers: **Modèles Robustes**

Choisir des algorithmes naturellement insensibles aux extrêmes.

Random Forest

Basé sur des **arbres de décision** qui partitionnent l'espace. Les outliers sont isolés dans des feuilles spécifiques sans influencer la structure globale du modèle.

 **Immunité structurelle**

Gradient Boosting

XGBoost, LightGBM ou CatBoost. Bien que sensibles au bruit, ils peuvent être **configurés** avec des fonctions de perte robustes (ex: Huber Loss) pour ignorer les extrêmes.

 **Optimisation paramétrique**



Modèles Linéaires vs Non-Linéaires

La Régression Linéaire est extrêmement sensible aux outliers. Les modèles d'ensemble sont préférables.

CONSEIL D'EXPERT

Privilégiez les modèles basés sur les arbres.

- Importance du prétraitement des données IoT
- **Techniques de nettoyage des données : gestion des valeurs aberrantes et données manquantes**
- Normalisation et agrégation des données
- Edge Computing : définition, architecture et avantages
- Comparaison entre Edge Computing et Cloud Computing
- Cas d'utilisation de l'Edge Computing dans l'IoT

- Fog Computing : définition, architecture et cas d'utilisation
- Comparaison entre Fog Computing, Edge Computing et Cloud Computing
- Bases de données NoSQL adaptées à l'IoT : MongoDB, Cassandra
- Bases de données de séries temporelles : InfluxDB, TimescaleDB
- Choix des solutions de stockage selon les besoins des applications IoT

I.2. Traitement des outliers: **Contextuel**

Valider les anomalies par la connaissance métier (Domain Knowledge).



Validation Métier

L'algorithme détecte un écart statistique, mais seul l'expert métier peut confirmer s'il s'agit d'une erreur ou d'un événement critique.

EXEMPLE 1

Revenu très élevé dans un dataset
bancaire ≠ Erreur (Client VIP).

EXEMPLE 2

Capteur défectueux envoyant 999°C =
Erreur évidente à supprimer.



L'INTELLIGENCE HUMAINE

Les méthodes automatiques :

- *voient des écarts*
- *mais ne comprennent pas le contexte réel*



Il faut combiner : statistiques + modèles +
expertise humaine

- **Transformations** → Réduire l'impact des extrêmes
- **Modèles robustes** → Ignorer les outliers
- **Traitement contextuel** → Comprendre leur signification

- Importance du prétraitement des données IoT
- **Techniques de nettoyage des données : gestion des valeurs aberrantes et données manquantes**
- Normalisation et agrégation des données
- Edge Computing : définition, architecture et avantages
- Comparaison entre Edge Computing et Cloud Computing
- Cas d'utilisation de l'Edge Computing dans l'IoT

- Fog Computing : définition, architecture et cas d'utilisation
- Comparaison entre Fog Computing, Edge Computing et Cloud Computing
- Bases de données NoSQL adaptées à l'IoT : MongoDB, Cassandra
- Bases de données de séries temporelles : InfluxDB, TimescaleDB
- Choix des solutions de stockage selon les besoins des applications IoT

II. les données *manquantes*



Problématique des Manques

Les données manquantes peuvent introduire un **biais** significatif dans les résultats et empêcher l'exécution de nombreux algorithmes de Machine Learning qui ne supportent pas les valeurs nulles.

1) MCAR — Missing Completely At Random

Aucune structure. Le manque est **indépendant de toutes les variables**. Ex: Une panne réseau aléatoire qui empêche la collecte de certaines mesures.

Conséquence :

- Les données restantes sont représentatives.
- Aucune distorsion statistique.

Traitement :

- **Simple (suppression des lignes, imputation simple).**
- **Pas de biais introduit.**

2) MAR — Missing At Random

Le manque **dépend d'autres variables observées**. Ex: Capteurs solaires qui ne produisent pas de données la nuit → le manque dépend de l'heure (variable observée).

Conséquence :

- Peut introduire un biais si mal traité.
- Mais exploitable avec les bonnes méthodes.

Traitement :

- Imputation conditionnelle (régression, k-NN, modèles ML).
- Techniques statistiques avancées (EM, multiple imputation).

3) MNAR — Missing Not At Random

Le manque **dépend de la valeur elle-même**. Ex: Un capteur de température qui s'éteint au-delà de 100°C → les valeurs élevées disparaissent précisément parce qu'elles sont élevées.

Conséquence :

- Introduit un biais fort et non observable directement.
- Les données restantes sont trompeuses.

Traitement :

- **Très difficile.**
- **Nécessite des hypothèses fortes ou des modèles spécifiques (modèles de sélection, modèles à variables latentes).**

- Importance du prétraitement des données IoT
- Techniques de nettoyage des données : gestion des valeurs aberrantes et données manquantes
- Normalisation et agrégation des données
- Edge Computing : définition, architecture et avantages
- Comparaison entre Edge Computing et Cloud Computing
- Cas d'utilisation de l'Edge Computing dans l'IoT

II.1. Détection des Manques

Identifier où et pourquoi les données sont absentes.

> Analyse Numérique(Le "Combien ?")

Compter les valeurs **manquant** (NaN, None, null) dans chaque colonne.

```
df.isnull().sum()
```

👁 Analyse Visuelle

Utiliser des **Heatmaps** pour repérer les patterns de manques (ex: blocs de données manquantes).

```
sns.heatmap(df.isnull())
```



HEATMAP DE
NULLITÉ

Les zones jaunes indiquent les données manquantes. Permet de voir si les manques sont corrélés entre variables.

Approche

Outil

Ce qu'on apprend

Action suivante

Numérique

sum()

Le volume de pertes par capteur.

Décider s'il faut supprimer la colonne (trop de manques) ou l'imputer.

Visuelle

heatmap()

La nature de la panne (ponctuelle, globale, ou intermittente).

Choisir la technique d'imputation (ex: interpolation linéaire si le trou est court, ou suppression si le trou est trop long).

- Importance du prétraitement des données IoT
- **Techniques de nettoyage des données : gestion des valeurs aberrantes et données manquantes**
- Normalisation et agrégation des données
- Edge Computing : définition, architecture et avantages
- Comparaison entre Edge Computing et Cloud Computing
- Cas d'utilisation de l'Edge Computing dans l'IoT

- Fog Computing : définition, architecture et cas d'utilisation
- Comparaison entre Fog Computing, Edge Computing et Cloud Computing
- Bases de données NoSQL adaptées à l'IoT : MongoDB, Cassandra
- Bases de données de séries temporelles : InfluxDB, TimescaleDB
- Choix des solutions de stockage selon les besoins des applications IoT

II.2. Traitement des **Manques** : Suppression

Éliminer les manques au risque de perdre de l'information.

Suppression de Lignes

Supprime toute observation contenant au moins une valeur manquante.

```
df.dropna()
```

À éviter si le **dataset est petit** ou si les **manques sont nombreux**.

Suppression de Colonnes

Supprime une variable entière si elle contient trop de manques (**ex: > 50%**).

```
df.dropna(axis=1)
```

À éviter si la variable est **critique**(MNAR) pour le modèle.



Point de Vigilance

La suppression (Listwise Deletion) peut introduire un biais massif si les données ne sont pas MCAR(indépendant). Elle réduit également la puissance statistique de vos analyses.

- Importance du prétraitement des données IoT
- **Techniques de nettoyage des données : gestion des valeurs aberrantes et données manquantes**
- Normalisation et agrégation des données
- Edge Computing : définition, architecture et avantages
- Comparaison entre Edge Computing et Cloud Computing
- Cas d'utilisation de l'Edge Computing dans l'IoT

- Fog Computing : définition, architecture et cas d'utilisation
- Comparaison entre Fog Computing, Edge Computing et Cloud Computing
- Bases de données NoSQL adaptées à l'IoT : MongoDB, Cassandra
- Bases de données de séries temporelles : InfluxDB, TimescaleDB
- Choix des solutions de stockage selon les besoins des applications IoT

II.2. Traitement des **Manques** : Imputation Simple

Remplacer les manques par des statistiques centrales.

Moyenne / Médiane

Utilisé pour les **variables numériques**. La médiane est préférable si des outliers sont présents.

```
df['age'].fillna(df['age'].median())
```

Rapide mais réduit la variance artificielle du dataset.

Mode (Catégories)

Utilisé pour les **variables catégorielles**. Remplace par la valeur la plus fréquente.

```
df['city'].fillna(df['city'].mode()[0])
```

Simple mais peut renforcer les classes majoritaires.



Conseil d'Expert

L'imputation simple est une solution de secours. Elle peut introduire un biais important si le pourcentage de manques est élevé (> 5-10%).

- Importance du prétraitement des données IoT
- **Techniques de nettoyage des données** : gestion des valeurs aberrantes et données manquantes
- Normalisation et agrégation des données
- Edge Computing : définition, architecture et avantages
- Comparaison entre Edge Computing et Cloud Computing
- Cas d'utilisation de l'Edge Computing dans l'IoT

- Fog Computing : définition, architecture et cas d'utilisation
- Comparaison entre Fog Computing, Edge Computing et Cloud Computing
- Bases de données NoSQL adaptées à l'IoT : MongoDB, Cassandra
- Bases de données de séries temporelles : InfluxDB, TimescaleDB
- Choix des solutions de stockage selon les besoins des applications IoT

II.2. Traitement des **Manques** : Imputation Avancée

Utiliser des modèles prédictifs pour estimer les manques.

KNN Imputer

Remplace la valeur manquante par la moyenne des **K plus proches voisins**. Basé sur la similarité contextuelle des autres variables.

✓ **Précision contextuelle**

MICE (Multiple Imputation)

Multiple Imputation by Chained Equations. Modélise chaque variable avec manques en fonction des autres variables de manière itérative.

✓ **Robustesse statistique**

Complexité vs Précision

Plus précis que l'imputation simple, mais beaucoup plus coûteux en temps de calcul.

USAGE IOT

Idéal pour les séries temporelles corrélées.

- Importance du prétraitement des données IoT
- **Techniques de nettoyage des données : gestion des valeurs aberrantes et données manquantes**
- Normalisation et agrégation des données
- Edge Computing : définition, architecture et avantages
- Comparaison entre Edge Computing et Cloud Computing
- Cas d'utilisation de l'Edge Computing dans l'IoT

- Fog Computing : définition, architecture et cas d'utilisation
- Comparaison entre Fog Computing, Edge Computing et Cloud Computing
- Bases de données NoSQL adaptées à l'IoT : MongoDB, Cassandra
- Bases de données de séries temporelles : InfluxDB, TimescaleDB
- Choix des solutions de stockage selon les besoins des applications IoT

II.2. Traitement des **Manques** : Valeurs Spéciales & Feature Engineering

Transformer le manque en information exploitable par le modèle.



Valeur Spéciale

Ajouter une **catégorie explicite** pour les manques. Utile pour les variables catégorielles.

```
df['city'].fillna('Unknown')
```

Préserve l'information que la donnée était absente.



Variable Indicatrice

Créer une nouvelle colonne binaire (0/1) indiquant si la donnée était manquante.

```
df['age_missing'] = df['age'].isnull().astype(int)
```

Permet au modèle d'exploiter le pattern du manque.



Pourquoi le Feature Engineering ?

Le fait qu'une donnée soit manquante est souvent une information en soi (ex: un client qui ne remplit pas son revenu peut être un indicateur de risque).

- Importance du prétraitement des données IoT
- **Techniques de nettoyage des données : gestion des valeurs aberrantes et données manquantes**
- Normalisation et agrégation des données
- Edge Computing : définition, architecture et avantages
- Comparaison entre Edge Computing et Cloud Computing
- Cas d'utilisation de l'Edge Computing dans l'IoT

- Fog Computing : définition, architecture et cas d'utilisation
- Comparaison entre Fog Computing, Edge Computing et Cloud Computing
- Bases de données NoSQL adaptées à l'IoT : MongoDB, Cassandra
- Bases de données de séries temporelles : InfluxDB, TimescaleDB
- Choix des solutions de stockage selon les besoins des applications IoT

Bonnes Pratiques de Nettoyage

Garantir la robustesse et l'intégrité du pipeline de données.

Analyse de Cause

Toujours analyser **pourquoi** les données sont manquantes ou aberrantes avant de choisir une méthode de traitement.

Éviter le Data Leakage

Calculer les statistiques d'imputation (moyenne, médiane) sur le **Train Set** uniquement, puis les appliquer au Test Set.

Tests Multiples

Tester plusieurs méthodes d'imputation et comparer leur impact sur la performance finale du modèle.

Ordre des Opérations

Appliquer les transformations et le nettoyage **après** le split Train/Test pour éviter toute fuite d'information.

- Importance du prétraitement des données IoT
- **Techniques de nettoyage des données : gestion des valeurs aberrantes et données manquantes**
- Normalisation et agrégation des données
- Edge Computing : définition, architecture et avantages
- Comparaison entre Edge Computing et Cloud Computing
- Cas d'utilisation de l'Edge Computing dans l'IoT

- Fog Computing : définition, architecture et cas d'utilisation
- Comparaison entre Fog Computing, Edge Computing et Cloud Computing
- Bases de données NoSQL adaptées à l'IoT : MongoDB, Cassandra
- Bases de données de séries temporelles : InfluxDB, TimescaleDB
- Choix des solutions de stockage selon les besoins des applications IoT

Pipeline de Nettoyage Recommandé

Un flux structuré pour une préparation de données optimale.

1



EDA

Analyse exploratoire
initiale.

2



Détection

Outliers &
Manques.

3



Traitement

Gestion des
Outliers.

4



Analyse

Impact des
manques.

5



Imputation

Remplissage /
Suppression.

6



Validation

Test sur modèle
final.

Synthèse du Pipeline



Ce pipeline garantit que chaque décision de nettoyage est justifiée par une analyse préalable et validée par une évaluation finale. Il minimise les risques de biais et de perte d'information.

- Importance du prétraitement des données IoT
- Techniques de nettoyage des données : gestion des valeurs aberrantes et données manquantes
- **Normalisation et agrégation des données**
- Edge Computing : définition, architecture et avantages
- Comparaison entre Edge Computing et Cloud Computing
- Cas d'utilisation de l'Edge Computing dans l'IoT

- Fog Computing : définition, architecture et cas d'utilisation
- Comparaison entre Fog Computing, Edge Computing et Cloud Computing
- Bases de données NoSQL adaptées à l'IoT : MongoDB, Cassandra
- Bases de données de séries temporelles : InfluxDB, TimescaleDB
- Choix des solutions de stockage selon les besoins des applications IoT

III.1. Normalisation des données

Définition

La normalisation consiste à **transformer les données** pour les rendre **homogènes, comparables** et **exploitables**. Elle permet d'éviter les incohérences et les erreurs d'analyse.

"La qualité de l'analyse dépend directement de la propreté et de l'uniformité des données d'entrée."

Objectifs clés



Uniformiser(Standardiser)

Formats (dates, textes, unités)



Réduire

Doublons et incohérences



Faciliter

Traitements statistiques et analytiques

- Importance du prétraitement des données IoT
- Techniques de nettoyage des données : gestion des valeurs aberrantes et données manquantes
- **Normalisation et agrégation des données**
- Edge Computing : définition, architecture et avantages
- Comparaison entre Edge Computing et Cloud Computing
- Cas d'utilisation de l'Edge Computing dans l'IoT

- Fog Computing : définition, architecture et cas d'utilisation
- Comparaison entre Fog Computing, Edge Computing et Cloud Computing
- Bases de données NoSQL adaptées à l'IoT : MongoDB, Cassandra
- Bases de données de séries temporelles : InfluxDB, TimescaleDB
- Choix des solutions de stockage selon les besoins des applications IoT

III.1. Normalisation des données

Techniques de normalisation

Standardisation

Uniformisation des formats pour assurer la cohérence **textuelle** et temporelle.

Dates, nombres, textes

- **Unités :**
Température : capteur 1 → 25 °C, capteur 2 → 77 °F
Convertir tout en °C (25 °C, 77 °F → 25 °C)
- **Dates / heures**
Format 1 : JJ/MM/AAAA, format 2 : AAAA-MM-JJ
Uniformiser en ISO 8601 (2023-04-09)
- **Texte**
États : "ON", "On", "1"
Normaliser en "ON"

Conversion d'unités

Transformation des mesures vers un système d'unités commun.

kg → g, MAD → USD

Statistiques : impossible de calculer une moyenne si la moitié des enregistrements sont en mètres et l'autre moitié en centimètres

→ on convertit tout en la même unité (ex. cm).

Mise à l'échelle

Ramener les valeurs numériques dans un intervalle standardisé (Scaling).

Valeurs entre 0 et 1

Machine-learning : les modèles sont sensibles aux échelles.

Exemple : humidité $\in [0, 1]$ vs pression $\in [0, 10000]$

→ on applique un **Min-Max scaling** ($x' = (x - x_{\min}) / (x_{\max} - x_{\min})$) afin que chaque variable soit contenue dans **[0, 1]**.

- Importance du prétraitement des données IoT
- Techniques de nettoyage des données : gestion des valeurs aberrantes et données manquantes
- **Normalisation et agrégation des données**
- Edge Computing : définition, architecture et avantages
- Comparaison entre Edge Computing et Cloud Computing
- Cas d’utilisation de l’Edge Computing dans l’IoT

- Fog Computing : définition, architecture et cas d’utilisation
- Comparaison entre Fog Computing, Edge Computing et Cloud Computing
- Bases de données NoSQL adaptées à l’IoT : MongoDB, Cassandra
- Bases de données de séries temporelles : InfluxDB, TimescaleDB
- Choix des solutions de stockage selon les besoins des applications IoT

III.1.Techniques de normalisation

Exemple pratique 1 : Nettoyage de données

Données brutes

Hétérogène

VILLE	DATE	VENTE
casa	01/02/2024	1000
Casablanca	2024-02-01	1500
CASA	02-01-2024	1200

⚠ Problèmes : Casse incohérente, formats de date multiples.

Après normalisation

Homogène

VILLE	DATE	VENTE
Casablanca	2024-02-01	1000
Casablanca	2024-02-01	1500
Casablanca	2024-02-01	1200

✓ Solution : Uniformisation du nom de ville et du format ISO 8601.

- Importance du prétraitement des données IoT
- Techniques de nettoyage des données : gestion des valeurs aberrantes et données manquantes
- **Normalisation et agrégation des données**
- Edge Computing : définition, architecture et avantages
- Comparaison entre Edge Computing et Cloud Computing
- Cas d'utilisation de l'Edge Computing dans l'IoT

- Fog Computing : définition, architecture et cas d'utilisation
- Comparaison entre Fog Computing, Edge Computing et Cloud Computing
- Bases de données NoSQL adaptées à l'IoT : MongoDB, Cassandra
- Bases de données de séries temporelles : InfluxDB, TimescaleDB
- Choix des solutions de stockage selon les besoins des applications IoT

III.1. Techniques de normalisation

Exemple pratique 2 : Mise à l'échelle (Scaling)

√x Formule Min-Max

$$X_{\text{norm}} = (X - X_{\text{min}}) / (X_{\text{max}} - X_{\text{min}})$$

Cette technique ramène toutes les valeurs dans l'intervalle **[0, 1]**, préservant les relations proportionnelles entre les données.

Application pas à pas

DONNÉES :

10

20

30

Min = 10 | Max = 30

1

Valeur 10 : $(10 - 10) / (30 - 10)$

0

2

Valeur 20 : $(20 - 10) / (30 - 10)$

0.5

3

Valeur 30 : $(30 - 10) / (30 - 10)$

1

RÉSULTAT NORMALISÉ :

0, 0.5, 1

- Importance du prétraitement des données IoT
- Techniques de nettoyage des données : gestion des valeurs aberrantes et données manquantes
- **Normalisation et agrégation des données**
- Edge Computing : définition, architecture et avantages
- Comparaison entre Edge Computing et Cloud Computing
- Cas d'utilisation de l'Edge Computing dans l'IoT

- Fog Computing : définition, architecture et cas d'utilisation
- Comparaison entre Fog Computing, Edge Computing et Cloud Computing
- Bases de données NoSQL adaptées à l'IoT : MongoDB, Cassandra
- Bases de données de séries temporelles : InfluxDB, TimescaleDB
- Choix des solutions de stockage selon les besoins des applications IoT

III.2. Agrégation des données

Définition

L'agrégation consiste à **regrouper plusieurs données** pour produire une information synthétique. Elle permet de passer du détail à la vue d'ensemble(ex. : calculer le chiffre d'affaires total, la note moyenne, le nombre de clients, etc.)..

Objectifs principaux

- ✓ Résumer les données massives
- ✓ Extraire des indicateurs clés KPIs(Key Performance Indicator)
- ✓ Faciliter la prise de décision stratégique

Les Quatre fonctions SQL/analytiques fondamentales

SUM
SOMME

AVG
MOYENNE

COUNT
Nombre (comptage)

MIN / MAX
Extrêmes (valeur minimale
et maximale)

! Ces fonctions sont essentielles pour le monitoring en temps réel.

- Importance du prétraitement des données IoT
- Techniques de nettoyage des données : gestion des valeurs aberrantes et données manquantes
- Normalisation et agrégation des données
- **Edge Computing : définition, architecture et avantages**
- Comparaison entre Edge Computing et Cloud Computing
- Cas d'utilisation de l'Edge Computing dans l'IoT

- Fog Computing : définition, architecture et cas d'utilisation
- Comparaison entre Fog Computing, Edge Computing et Cloud Computing
- Bases de données NoSQL adaptées à l'IoT : MongoDB, Cassandra
- Bases de données de séries temporelles : InfluxDB, TimescaleDB
- Choix des solutions de stockage selon les besoins des applications IoT

IV. Edge Computing : définition, architecture et avantage

IV.1. Définition et concepts fondamentaux

L'**Edge Computing** consiste à **traiter les données là où elles sont générées (capteurs, objets IoT, passerelles)**, au bord du réseau, afin de **réduire la latence, économiser la bande passante** et permettre une **prise de décision locale et autonome**, même **sans connexion permanente à Internet**.

Edge Computing s'appuie sur une intelligence décentralisée : chaque nœud de périphérie dispose de sa propre capacité de traitement, d'analyse et parfois même d'inférence d'intelligence artificielle.

- Importance du prétraitement des données IoT
- Techniques de nettoyage des données : gestion des valeurs aberrantes et données manquantes
- Normalisation et agrégation des données
- **Edge Computing : définition, architecture et avantages**
- Comparaison entre Edge Computing et Cloud Computing
- Cas d'utilisation de l'Edge Computing dans l'IoT

- Fog Computing : définition, architecture et cas d'utilisation
- Comparaison entre Fog Computing, Edge Computing et Cloud Computing
- Bases de données NoSQL adaptées à l'IoT : MongoDB, Cassandra
- Bases de données de séries temporelles : InfluxDB, TimescaleDB
- Choix des solutions de stockage selon les besoins des applications IoT

IV. Edge Computing : définition, architecture et avantage

IV.2. Architecture de référence Edge IoT

Hierarchie fonctionnelle du traitement de la donnée



COUCHE 01

Niveau Edge (périphérie)

- ✓ Capteurs & Actuateurs
- ✓ Microcontrôleurs, dispositifs embarqués
- ✓ Contraintes énergétiques fortes
- ✓ Certains embarquent déjà des puces TinyML pour un prétraitement minimal.

Génération & Action locale



COUCHE 02

Niveau Fog (Gateway)

- ⚡ Filtrage & Nettoyage (Prétraitement)
- ⚡ Agrégation temporelle des flux
- ⚡ Analyse en temps réel (Latence < 10ms)
- ⚡ Hébergent les algorithmes d'inférence IA

Intelligence de proximité



COUCHE 03

Niveau Cloud (centralisé)

- ☁ Stockage historique massif (Big Data)
- 🧠 Entraînement de modèles IA avancé
- 💻 Dashboards & Business Intelligence

Vision globale & Analyse froide

- Importance du prétraitement des données IoT
- Techniques de nettoyage des données : gestion des valeurs aberrantes et données manquantes
- Normalisation et agrégation des données
- **Edge Computing : définition, architecture et avantages**
- Comparaison entre Edge Computing et Cloud Computing
- Cas d'utilisation de l'Edge Computing dans l'IoT

- Fog Computing : définition, architecture et cas d'utilisation
- Comparaison entre Fog Computing, Edge Computing et Cloud Computing
- Bases de données NoSQL adaptées à l'IoT : MongoDB, Cassandra
- Bases de données de séries temporelles : InfluxDB, TimescaleDB
- Choix des solutions de stockage selon les besoins des applications IoT

IV. Edge Computing : définition, architecture et avantage

IV.3. Avantages de l'Edge Computing

1. Réduction de la latence

Décisions quasi instantanées
(milli-secondes).

Critique pour systèmes temps réel
(ex : véhicules autonomes)

2. Amélioration de la sécurité et confidentialité

Données sensibles traitées
localement

Moins d'exposition aux attaques
réseau

3. Fiabilité accrue

Fonctionnement même sans
connexion cloud

Résilience des systèmes distribués

6. Réduction du trafic réseau

Moins de données envoyées au
cloud

Optimisation de la bande passante

5. Scalabilité

Distribution de la charge de calcul

Moins de dépendance à une
infrastructure centrale

< 10ms

CIBLE DE LATENCE

Focus : Industrie 4.0

Dans une chaîne d'assemblage robotisée, la synchronisation entre capteurs et actionneurs ne tolère pas les allers-retours vers le cloud. L'Edge Computing permet de décentraliser l'intelligence pour une réactivité optimale.



- Importance du prétraitement des données IoT
- Techniques de nettoyage des données : gestion des valeurs aberrantes et données manquantes
- Normalisation et agrégation des données
- Edge Computing : définition, architecture et avantages
- Comparaison entre Edge Computing et Cloud Computing
- Cas d'utilisation de l'Edge Computing dans l'IoT

- Fog Computing : définition, architecture et cas d'utilisation
- Comparaison entre Fog Computing, Edge Computing et Cloud Computing
- Bases de données NoSQL adaptées à l'IoT : MongoDB, Cassandra
- Bases de données de séries temporelles : InfluxDB, TimescaleDB
- Choix des solutions de stockage selon les besoins des applications IoT

V. Edge Computing vs Cloud Computing

Critère	Edge Computing	Cloud Computing
Localisation	Proche de la source	Centres de données distants
Latence	Très faible	Plus élevée
Bande passante	Faible utilisation	Forte utilisation
Traitement	Décentralisé	Centralisé
Scalabilité	Distribuée	Très élevée (élasticité)
Coût	Moins de transmission	Coût lié au stockage et transfert
Sécurité	Meilleure confidentialité locale	Dépend du fournisseur cloud
Maintenance	Complexe (plus de nœuds)	Centralisée

- Importance du prétraitement des données IoT
- Techniques de nettoyage des données : gestion des valeurs aberrantes et données manquantes
- Normalisation et agrégation des données
- Edge Computing : définition, architecture et avantages
- Comparaison entre Edge Computing et Cloud Computing
- Cas d'utilisation de l'Edge Computing dans l'IoT

- Fog Computing : définition, architecture et cas d'utilisation
- Comparaison entre Fog Computing, Edge Computing et Cloud Computing
- Bases de données NoSQL adaptées à l'IoT : MongoDB, Cassandra
- Bases de données de séries temporelles : InfluxDB, TimescaleDB
- Choix des solutions de stockage selon les besoins des applications IoT

VI. Cas d'utilisation de l'Edge Computing dans l'IoT

5.1 Industrie 4.0 (Smart Manufacturing)

- Maintenance prédictive
- Détection d'anomalies en temps réel
- Contrôle qualité automatisé

5.2 Smart Cities

- Gestion intelligente du trafic
- Vidéosurveillance avec reconnaissance en temps réel
- Gestion de l'énergie

5.3 Santé (Healthcare IoT)

- Monitoring patient en temps réel
- Dispositifs médicaux connectés
- Alertes critiques instantanées

5.4 Véhicules autonomes

- Traitement local des capteurs (LIDAR, caméra)
- Prise de décision immédiate

5.5 Agriculture intelligente

- Capteurs de sol et météo
- Irrigation intelligente
- Détection de maladies des plantes

5.6 Retail (magasins intelligents)

- Analyse comportement client
- Gestion des stocks en temps réel

- Importance du prétraitement des données IoT
- Techniques de nettoyage des données : gestion des valeurs aberrantes et données manquantes
- Normalisation et agrégation des données
- Edge Computing : définition, architecture et avantages
- Comparaison entre Edge Computing et Cloud Computing
- Cas d'utilisation de l'Edge Computing dans l'IoT

- **Fog Computing : définition, architecture et cas d'utilisation**
- Comparaison entre Fog Computing, Edge Computing et Cloud Computing
- Bases de données NoSQL adaptées à l'IoT : MongoDB, Cassandra
- Bases de données de séries temporelles : InfluxDB, TimescaleDB
- Choix des solutions de stockage selon les besoins des applications IoT

VII. Fog Computing

IVII.1. définition

Qu'est-ce que le Fog ?

Le **Fog Computing** (informatique en brouillard) est une extension du cloud qui introduit une **couche intermédiaire** entre le cloud et les objets connectés.

👉 Il permet de traiter les données plus près des utilisateurs, mais avec plus de capacité que l'Edge.

Positionnement Hiérarchique



Edge

Très proche des capteurs



Fog

Niveau intermédiaire (LAN, Gateways)



Cloud

Distant (Data Centers)

- Importance du prétraitement des données IoT
- Techniques de nettoyage des données : gestion des valeurs aberrantes et données manquantes
- Normalisation et agrégation des données
- Edge Computing : définition, architecture et avantages
- Comparaison entre Edge Computing et Cloud Computing
- Cas d'utilisation de l'Edge Computing dans l'IoT

- **Fog Computing : définition, architecture et cas d'utilisation**
- Comparaison entre Fog Computing, Edge Computing et Cloud Computing
- Bases de données NoSQL adaptées à l'IoT : MongoDB, Cassandra
- Bases de données de séries temporelles : InfluxDB, TimescaleDB
- Choix des solutions de stockage selon les besoins des applications IoT

VII. Fog Computing

IVII.2. architecture



COUCHE 01

Niveau Edge (périphérie)

- ✓ Capteurs & Actuateurs
- ✓ Microcontrôleurs, dispositifs embarqués
- ✓ Contraintes énergétiques fortes
- ✓ Certains embarquent déjà des puces TinyML pour un prétraitement minimal.

Génération & Action locale



COUCHE 02

Niveau Fog (Gateway)

- ⚡ Filtrage & Nettoyage (Prétraitement)
- ⚡ Agrégation temporelle des flux
- ⚡ Analyse en temps réel (Latence < 10ms)
- ⚡ Hébergent les algorithmes d'inférence IA

Intelligence de proximité



COUCHE 03

Niveau Cloud (centralisé)

- ☁ Stockage historique massif (Big Data)
- 🧠 Entraînement de modèles IA avancé
- 💻 Dashboards & Business Intelligence

Vision globale & Analyse froide

- Importance du prétraitement des données IoT
- Techniques de nettoyage des données : gestion des valeurs aberrantes et données manquantes
- Normalisation et agrégation des données
- Edge Computing : définition, architecture et avantages
- Comparaison entre Edge Computing et Cloud Computing
- Cas d'utilisation de l'Edge Computing dans l'IoT

- **Fog Computing : définition, architecture et cas d'utilisation**
- Comparaison entre Fog Computing, Edge Computing et Cloud Computing
- Bases de données NoSQL adaptées à l'IoT : MongoDB, Cassandra
- Bases de données de séries temporelles : InfluxDB, TimescaleDB
- Choix des solutions de stockage selon les besoins des applications IoT

VII. Fog Computing

IVII.3. cas d'utilisation



Smart Grid

Analyse locale de la consommation électrique et réaction ultra-rapide aux anomalies du réseau pour éviter les surcharges.



Vidéosurveillance

Traitement local des flux vidéo pour la détection d'événements critiques sans saturer la bande passante vers le cloud.



Transport Intelligent

Gestion du trafic en temps réel et coordination entre véhicules et infrastructures (V2I) pour la sécurité routière.



Industrie 4.0

Supervision des systèmes industriels et optimisation des processus en temps réel sur la ligne de production.

- Importance du prétraitement des données IoT
 - Techniques de nettoyage des données : gestion des valeurs aberrantes et données manquantes
 - Normalisation et agrégation des données
 - Edge Computing : définition, architecture et avantages
 - Comparaison entre Edge Computing et Cloud Computing
 - Cas d'utilisation de l'Edge Computing dans l'IoT
- Fog Computing : définition, architecture et cas d'utilisation
 - **Comparaison entre Fog Computing, Edge Computing et Cloud Computing**
 - Bases de données NoSQL adaptées à l'IoT : MongoDB, Cassandra
 - Bases de données de séries temporelles : InfluxDB, TimescaleDB
 - Choix des solutions de stockage selon les besoins des applications IoT

VIII. Comparaison entre Fog Computing, Edge Computing et Cloud Computing

CRITÈRE	EDGE COMPUTING	FOG COMPUTING	CLOUD COMPUTING
Position	Très proche (device)	Intermédiaire (LAN)	Distant (Data Center)
Latence	Très faible	Faible	Plus élevée
Capacité	Limitée	Moyenne	Très élevée
Traitement	Local immédiat	Pré-traitement / Coordination	Analyse complète / IA
Scalabilité	Faible	Moyenne	Très forte

Edge
Réaction immédiate

Fog
Coordination intelligente

Cloud
Puissance globale

Hiérarchie de calcul : Edge vs Fog vs Cloud

Distribution stratégique de l'intelligence et des ressources de stockage.



i L'architecture moderne privilégie un modèle **hybride** exploitant les forces de chaque niveau.

IoT INFRASTRUCTURE ARCHITECTURE

- Importance du prétraitement des données IoT
- Techniques de nettoyage des données : gestion des valeurs aberrantes et données manquantes
- Normalisation et agrégation des données
- Edge Computing : définition, architecture et avantages
- Comparaison entre Edge Computing et Cloud Computing
- Cas d'utilisation de l'Edge Computing dans l'IoT

- Fog Computing : définition, architecture et cas d'utilisation
- Comparaison entre Fog Computing, Edge Computing et Cloud Computing
- Bases de données NoSQL adaptées à l'IoT : MongoDB, Cassandra
- Bases de données de séries temporelles : InfluxDB, TimescaleDB
- Choix des solutions de stockage selon les besoins des applications IoT

IX. Bases de données NoSQL adaptées à l'IoT

IX.1. MongoDB

Orientée Documents

MongoDB est une **base de données NoSQL orientée documents** qui stocke les données sous forme de **documents JSON/BSON** plutôt que sous forme de tables relationnelles.



Modèle JSON / BSON

Stockage flexible sous forme de documents, idéal pour les données hétérogènes.



Schéma Dynamique

Pas de structure rigide imposée (ajouter facilement de nouveaux champs), permettant une évolution rapide des capteurs.



Facilité d'utilisation

Écosystème riche et requêtes intuitives pour les développeurs.

💡 Cas d'usage IoT:

- ✓ Données de capteurs **hétérogènes** (formats variés)
- ✓ Applications web IoT et Dashboards
- ✓ Prototypage rapide de solutions connectées

EXEMPLE DE DOCUMENT

```
1 {
2   "sensor_id": "TEMP_01",
3   "timestamp": "2025-03-10T10:00:00Z",
4   "value": 22.5,
5   "unit": "Celsius",
6   "metadata": {
7     "room": "A101"
8   }
9 }
```

- Importance du prétraitement des données IoT
- Techniques de nettoyage des données : gestion des valeurs aberrantes et données manquantes
- Normalisation et agrégation des données
- Edge Computing : définition, architecture et avantages
- Comparaison entre Edge Computing et Cloud Computing
- Cas d'utilisation de l'Edge Computing dans l'IoT

- Fog Computing : définition, architecture et cas d'utilisation
- Comparaison entre Fog Computing, Edge Computing et Cloud Computing
- Bases de données NoSQL adaptées à l'IoT : MongoDB, Cassandra
- Bases de données de séries temporelles : InfluxDB, TimescaleDB
- Choix des solutions de stockage selon les besoins des applications IoT

IX. Bases de données NoSQL adaptées à l'IoT

IX.2. Apache Cassandra

Distribuée & Scalable

Apache Cassandra est une base de données NoSQL **distribuée** conçue pour :

- gérer de **très grands volumes de données**
- fonctionner sur plusieurs serveurs (cluster)
- offrir une **haute disponibilité sans point unique de panne**



Architecture Peer-to-Peer

Pas de point de défaillance unique (No Single Point of Failure). Haute disponibilité garantie.



Performance en Écriture

Optimisée pour l'**ingestion massive de flux de données en temps réel**.



Scalabilité Horizontale

Ajout de nœuds sans interruption de service pour gérer des pétaoctets de données.

💡 Cas d'usage IoT

- ✓ Données massives en temps réel (millions de capteurs)
- ✓ Systèmes critiques (Télécoms, Industrie lourde)
- ✓ Applications nécessitant une disponibilité 24/7

100%

Disponibilité & Scalabilité
"Write-heavy workloads are Cassandra's specialty"

- Importance du prétraitement des données IoT
- Techniques de nettoyage des données : gestion des valeurs aberrantes et données manquantes
- Normalisation et agrégation des données
- Edge Computing : définition, architecture et avantages
- Comparaison entre Edge Computing et Cloud Computing
- Cas d'utilisation de l'Edge Computing dans l'IoT

- Fog Computing : définition, architecture et cas d'utilisation
- Comparaison entre Fog Computing, Edge Computing et Cloud Computing
- Bases de données NoSQL adaptées à l'IoT : MongoDB, Cassandra
- **Bases de données de séries temporelles : InfluxDB, TimescaleDB**
- Choix des solutions de stockage selon les besoins des applications IoT

X. Bases de données de séries temporelles

X.1. InfluxDB

Native Time-Series

InfluxDB est une base de données NoSQL **spécialisée dans les séries temporelles (time-series)**, conçue pour stocker et analyser des données horodatées :

- métriques système
- données IoT
- monitoring temps réel



Spécialisée Séries Temporelles

Optimisée pour les données horodatées (time-stamped) à haute fréquence.



Langage Flux

Langage de requête puissant pour l'analyse temporelle et les agrégations complexes.



Performance Temps Réel

Requêtes ultra-rapides pour le monitoring et les mesures instantanées.

💡 Cas d'usage IoT

- ✓ Mesures de température toutes les secondes
- ✓ Monitoring d'infrastructures IT et réseaux
- ✓ Analyse de métriques de performance industrielle

EXEMPLE DE REQUÊTE FLUX

```
from(bucket: "iot_data")
  |> range(start: -1h)
  |> filter(fn: (r) => r._measurement == "temp")
  |> mean()
```

- Importance du prétraitement des données IoT
- Techniques de nettoyage des données : gestion des valeurs aberrantes et données manquantes
- Normalisation et agrégation des données
- Edge Computing : définition, architecture et avantages
- Comparaison entre Edge Computing et Cloud Computing
- Cas d'utilisation de l'Edge Computing dans l'IoT

- Fog Computing : définition, architecture et cas d'utilisation
- Comparaison entre Fog Computing, Edge Computing et Cloud Computing
- Bases de données NoSQL adaptées à l'IoT : MongoDB, Cassandra
- **Bases de données de séries temporelles : InfluxDB, TimescaleDB**
- Choix des solutions de stockage selon les besoins des applications IoT

X. Bases de données de séries temporelles

X.2. TimescaleDB

PostgreSQL Extension

TimescaleDB est une base de données **time-series** construite comme une extension de **PostgreSQL**.



SQL Natif

Combine la puissance du **SQL relationnel** avec les **performances des séries temporelles**.



Hypertables

Partitionnement automatique des données par temps pour des performances constantes.



Flexibilité Totale

Plus flexible qu'InfluxDB pour les jointures complexes avec des données métier.

💡 Cas d'usage IoT

- ✓ Analyse complexe nécessitant des requêtes SQL
- ✓ Jointures entre données capteurs et données clients
- ✓ Migration facile depuis une base PostgreSQL existante

EXEMPLE DE REQUÊTE SQL

```
SELECT time_bucket('1 hour', time) AS hour,  
       avg(temperature)  
FROM sensor_data  
GROUP BY hour  
ORDER BY hour DESC;
```


- Importance du prétraitement des données IoT
- Techniques de nettoyage des données : gestion des valeurs aberrantes et données manquantes
- Normalisation et agrégation des données
- Edge Computing : définition, architecture et avantages
- Comparaison entre Edge Computing et Cloud Computing
- Cas d'utilisation de l'Edge Computing dans l'IoT

- Fog Computing : définition, architecture et cas d'utilisation
- Comparaison entre Fog Computing, Edge Computing et Cloud Computing
- Bases de données NoSQL adaptées à l'IoT : MongoDB, Cassandra
- Bases de données de séries temporelles : InfluxDB, TimescaleDB
- **Choix des solutions de stockage selon les besoins des applications IoT**

XI. Choix des solutions de stockage



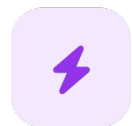
Volume de données

Très grand volume → **Cassandra**
Volume moyen → **MongoDB**



Type de données

Structurées → **TimescaleDB**
Non structurées → **MongoDB**



Temps Réel

Critique → **InfluxDB**
Analyse différée → **Cloud + SQL**



Scalabilité

Forte scalabilité → **Cassandra**
Modérée → **MongoDB**



Comment choisir ?

Le choix dépend de l'équilibre entre la vitesse d'ingestion, la complexité des requêtes et le volume total à stocker.

- Importance du prétraitement des données IoT
- Techniques de nettoyage des données : gestion des valeurs aberrantes et données manquantes
- Normalisation et agrégation des données
- Edge Computing : définition, architecture et avantages
- Comparaison entre Edge Computing et Cloud Computing
- Cas d'utilisation de l'Edge Computing dans l'IoT

- Fog Computing : définition, architecture et cas d'utilisation
- Comparaison entre Fog Computing, Edge Computing et Cloud Computing
- Bases de données NoSQL adaptées à l'IoT : MongoDB, Cassandra
- Bases de données de séries temporelles : InfluxDB, TimescaleDB
- **Choix des solutions de stockage selon les besoins des applications IoT**

XI. Choix des solutions de stockage

Tableau de synthèse des besoins

BESOIN SPÉCIFIQUE	SOLUTION RECOMMANDÉE
Données IoT massives (Millions de capteurs)	Apache Cassandra
Données flexibles et hétérogènes (JSON)	MongoDB
Données temporelles à haute fréquence	InfluxDB
Analyse SQL complexe + Séries temporelles	TimescaleDB

“

*"L'architecture hybride
Edge-Fog-Cloud est le socle de l'IoT
moderne et de l'IA distribuée."*



Note : Dans une architecture réelle, plusieurs solutions peuvent coexister (ex: InfluxDB pour le monitoring temps réel et Cassandra pour l'archivage massif).