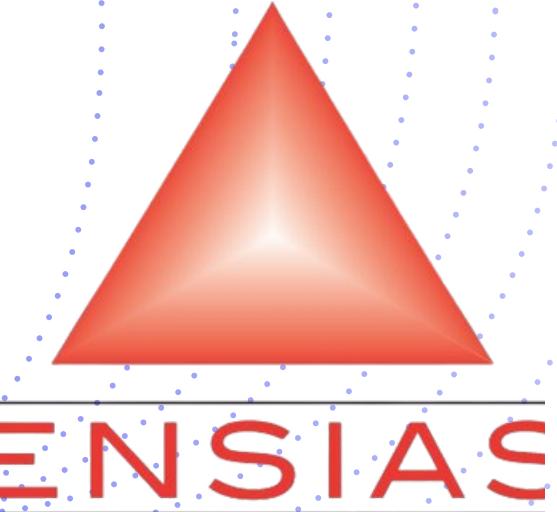




جامعة محمد الخامس بالرباط
Université Mohammed V de Rabat



PROJET NOSQL

DÉTECTION ET PRÉVENTION DES MENACES INTERNES
DANS LES SI À L'AIDE DE L'APPRENTISSAGE
AUTOMATIQUE

Réalisé par:

Houda Ait Mouch

Redouane Ghatrif

Anas Mahmoudi

Ranya Serraj Andaloussi

Soutenu le 11/04/2025 devant le Jury:

Pr.Moumane Karima

2024-2025

SOMMAIRE

- 1 Introduction
- 2 Méthodologie et Conception
- 3 Implémentation et Expérimentations
- 4 Résultats et Comparaison
- 5 Conclusion



1 INTRODUCTION

Les incidents liés aux menaces internes ont augmenté de 44 % au cours des deux dernières années, et le coût d'un incident dépasse désormais 15,3 millions de dollars, selon le rapport 2022 Cost of Insider Threats de l'institut Ponemon.



LES MENACES INTERNES

Une menace interne est une menace à la sécurité de l'information qui provient d'un individu ayant un accès légitime aux systèmes et aux ressources de l'organisation, et qui en abuse intentionnellement ou non.



TYPE DES MENACES INTERNES

Les menaces internes se classent en trois catégories principales : les actions malveillantes commises délibérément par des individus de l'organisation, les comportements négligents ou erreurs humaines non intentionnelles, et les comptes légitimes piratés par des acteurs externes, permettant une intrusion déguisée en accès autorisé.



IMPACT DU ML ET NOSQL

L'essor des techniques de Machine Learning (ML) et des bases de données NoSQL offre de nouvelles perspectives prometteuses pour la détection des menaces internes.

La synergie entre bases NoSQL et Machine Learning crée un paradigme particulièrement puissant pour la détection des menaces interne



PROBLEMATIQUE



Comment détecter les menaces internes au sein d'un SI?



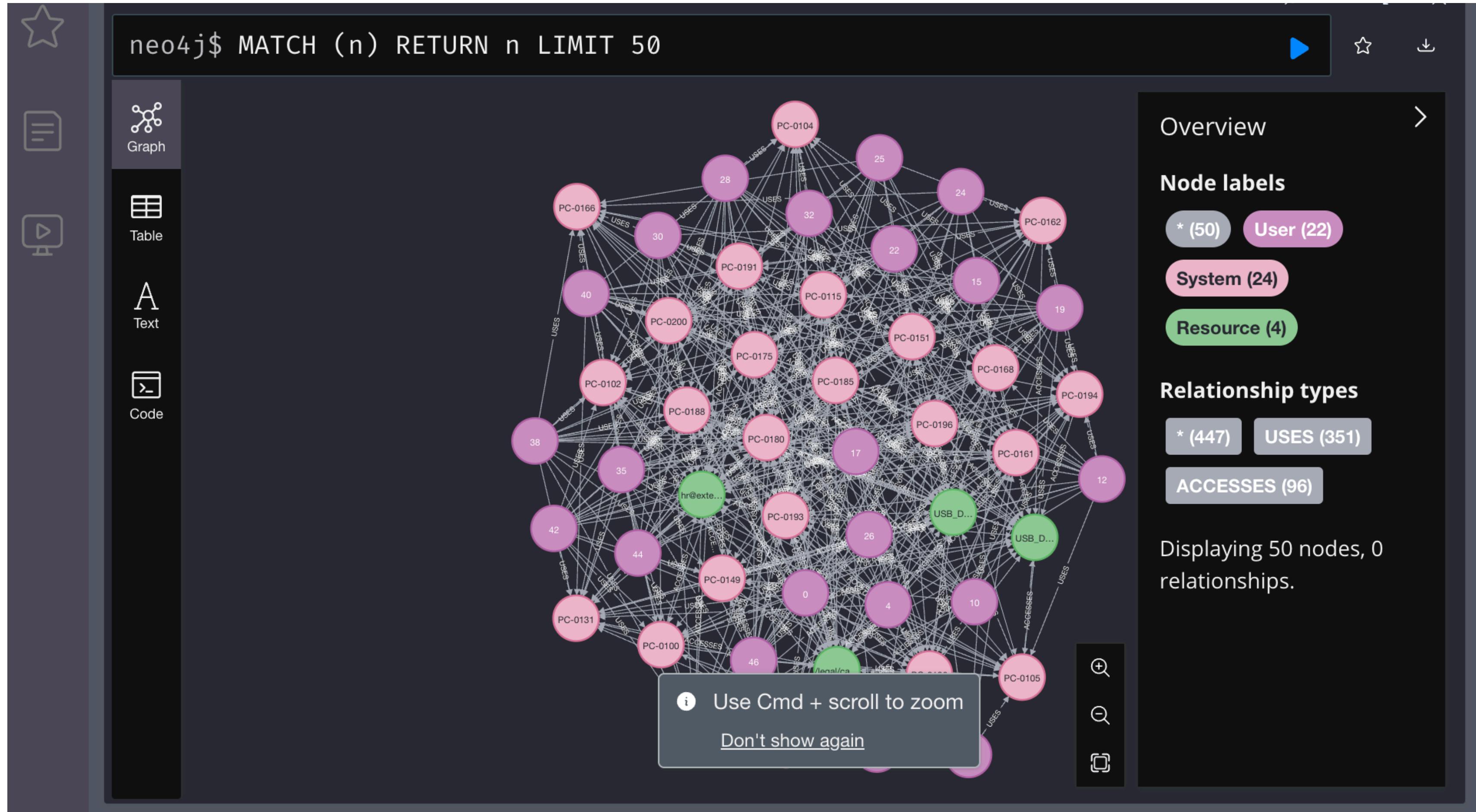
Méthodologie et Conception

JUSTIFICATION de Neo4j

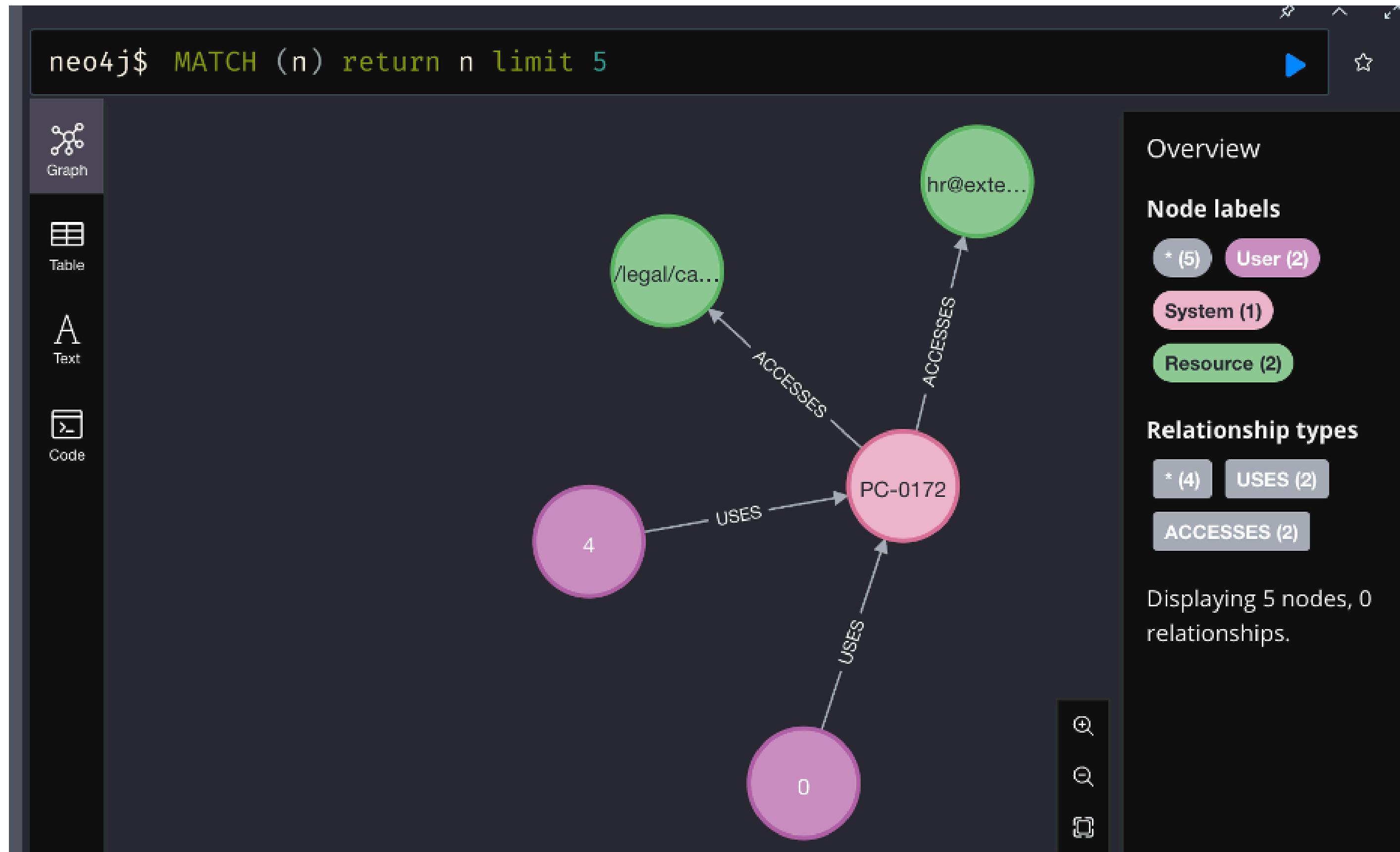


- 1 Modélisation native des relations complexes
- 2 Langage de requête Cypher
- 3 Algorithmes natifs d'analyse de graphes
- 4 Visualisation intégrée
- 5 Evolutivité

Modélisation des interactions internes



Modélisation des interactions internes



Activités uniques (comportements rares ou isolés)

```
MATCH (u:User)-[r:PERFORMS]->(a:Activity)
WITH u, a, count(r) AS freq, min(r.timestamp) AS first_time
WHERE freq = 1
RETURN u.name AS User, a.name AS Activity, first_time AS FirstOccurrence
ORDER BY first_time DESC
```



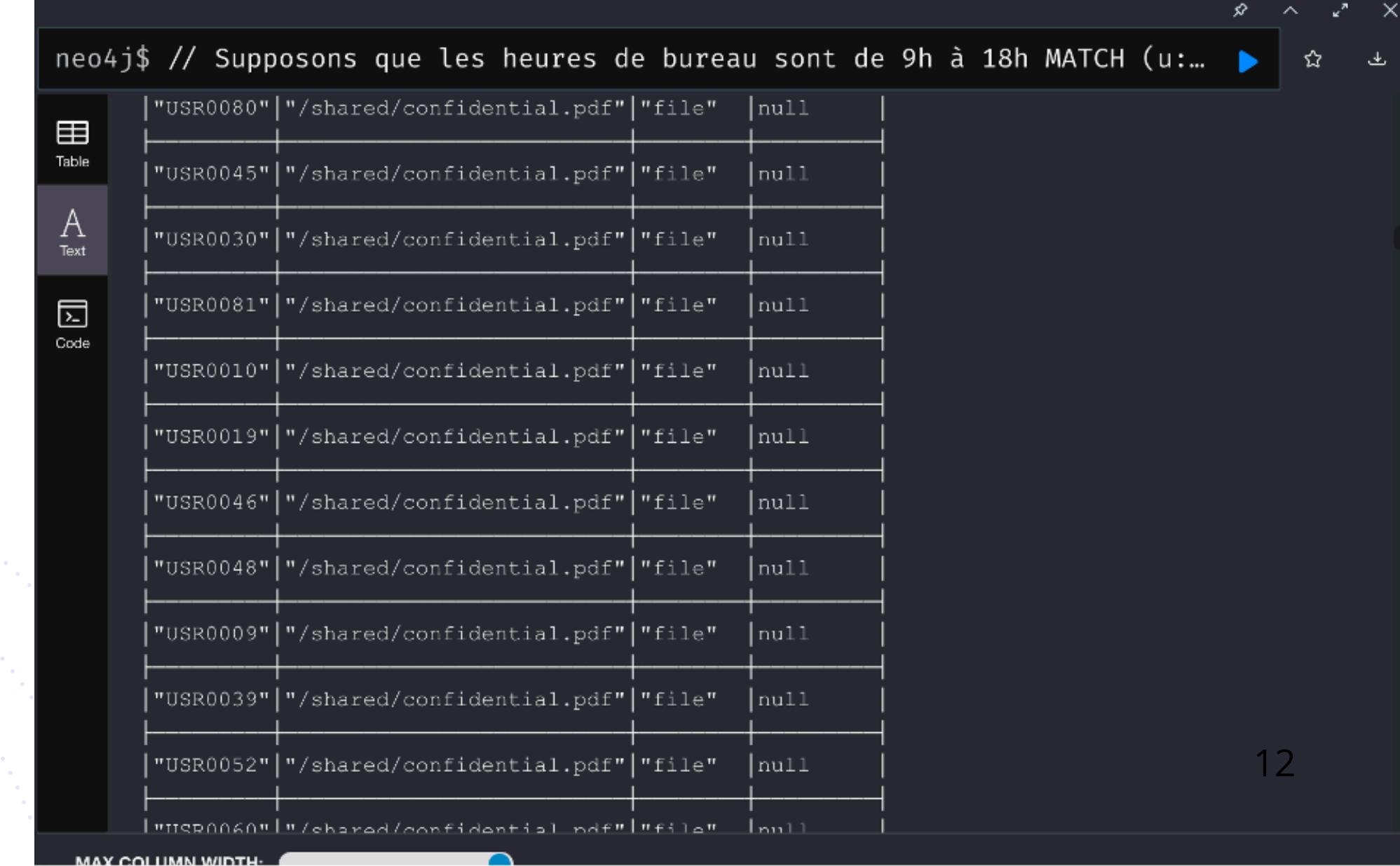
The screenshot shows the Neo4j browser interface with a query results table. The table has columns: User, Activity, and FirstOccurrence. The data shows six rows of results for user "USR0036".

	User	Activity	FirstOccurrence
1	"USR0042"	"logon"	"2025-03-29T07:59:00"
2	"USR0036"	"http"	"2025-03-29T07:57:00"
3	"USR0036"	"http"	"2025-03-29T07:57:00"
4	"USR0036"	"http"	"2025-03-29T07:57:00"
5	"USR0036"	"http"	"2025-03-29T07:57:00"
6	"USR0036"	"http"	"2025-03-29T07:57:00"
7			

Started streaming 3486 records in less than 1 ms and completed after 13 ms, displaying first 1000 rows.

Accès à des ressources sensibles en dehors des heures de bureau

```
// Supposons que les heures de bureau sont de 9h à 18h
MATCH (u:User)-[:PERFORMS]->(a:Activity)-[:INVOLVES]->(r:Resource)
WHERE
    r.name CONTAINS "confidential" OR r.name CONTAINS "sensitive" // Remplacez par vos propres critères
    AND (datetime(r.timestamp).hour < 9 OR datetime(r.timestamp).hour > 18)
RETURN u.name AS User, r.name AS SensitiveResource, a.name AS Activity, r.t
```



The screenshot shows the Neo4j browser interface with a query results table. The table has four columns: User, SensitiveResource, Activity, and timestamp. The timestamp column is truncated at the end. The data consists of 12 rows, each representing a user interacting with a sensitive resource during non-business hours.

User	SensitiveResource	Activity	timestamp
"USR0080"	"/shared/confidential.pdf"	"file"	null
"USR0045"	"/shared/confidential.pdf"	"file"	null
"USR0030"	"/shared/confidential.pdf"	"file"	null
"USR0081"	"/shared/confidential.pdf"	"file"	null
"USR0010"	"/shared/confidential.pdf"	"file"	null
"USR0019"	"/shared/confidential.pdf"	"file"	null
"USR0046"	"/shared/confidential.pdf"	"file"	null
"USR0048"	"/shared/confidential.pdf"	"file"	null
"USR0009"	"/shared/confidential.pdf"	"file"	null
"USR0039"	"/shared/confidential.pdf"	"file"	null
"USR0052"	"/shared/confidential.pdf"	"file"	null
"USR0060"	"/shared/confidential.pdf"	"file"	null

Détection de connexions rapides à plusieurs systèmes

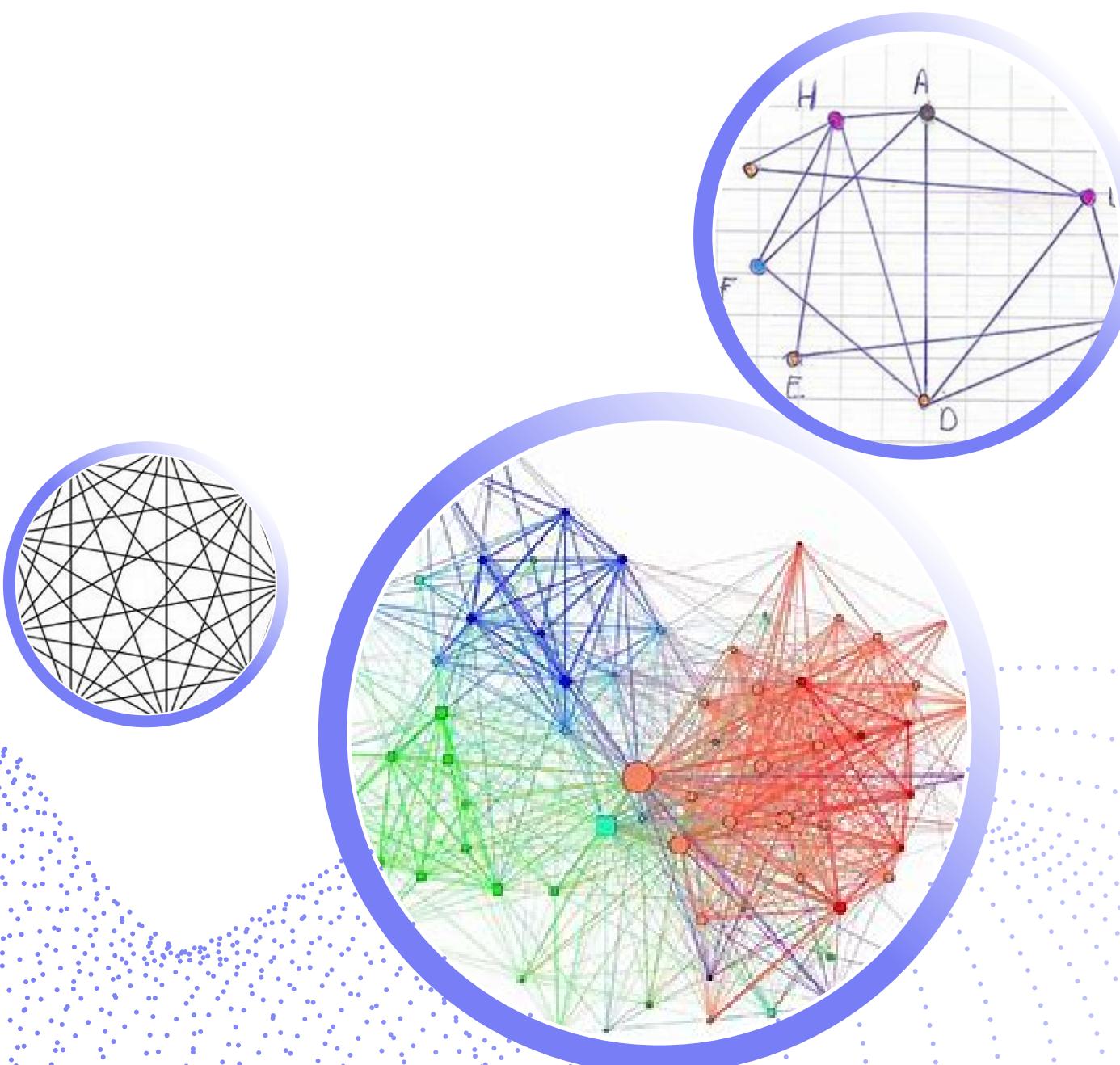
The screenshot shows the Neo4j browser interface. At the top, there is a code editor window with the following Cypher query:

```
neo4j$  
1 // Trouver les utilisateurs qui utilisent plusieurs systèmes  
2 différents dans un court laps de temps  
3 MATCH (u:User)-[r1:USES]→(s1:System)  
4 MATCH (u)-[r2:USES]→(s2:System)  
5 WHERE s1 ≠ s2  
6 AND datetime(r1.timestamp) < datetime(r2.timestamp) <  
    datetime(r1.timestamp) + duration('PT1H') // Dans une fenêtre d'une  
    heure  
7 RETURN u.name AS User, s1.name AS FirstSystem, s2.name AS  
    SecondSystem,  
    r1.timestamp AS FirstAccess, r2.timestamp AS SecondAccess  
8
```

Below the code editor is a table view showing the results of the query. The table has columns: User, FirstSystem, SecondSystem, FirstAccess, and SecondAccess. The data is as follows:

	User	FirstSystem	SecondSystem	FirstAccess	SecondAccess
1	"USR0089"	"PC-0111"	"PC-0172"	"2025-03-28T08:22:00"	"2025-03-28T08:52:00"
2	"USR0089"	"PC-0109"	"PC-0172"	"2025-03-28T08:42:00"	"2025-03-28T08:52:00"
3	"USR0089"	"PC-0118"	"PC-0172"	"2025-03-28T08:30:00"	"2025-03-28T08:52:00"
4	"USR0089"	"PC-0167"	"PC-0161"	"2025-03-28T16:37:00"	"2025-03-28T17:18:00"
5	"USR0089"	"PC-0162"	"PC-0106"	"2025-03-29T00:55:00"	"2025-03-29T01:49:00"
6	"USR0089"	"PC-0117"	"PC-0106"	"2025-03-29T01:42:00"	"2025-03-29T01:49:00"

NOTION DE CENTRALITE



Les métriques de centralités mesurent l'importance ou l'influence d'un nœud (ou d'un utilisateur dans ce cas) dans un réseau. Des anomalies peuvent se manifester lorsque certains nœuds présentent des valeurs de centralité très élevées ou très faibles par rapport à la normale, ce qui peut indiquer des comportements suspect.

IMPLEMENTATION DE CENTRALITE

Top 5 des utilisateurs les plus actifs

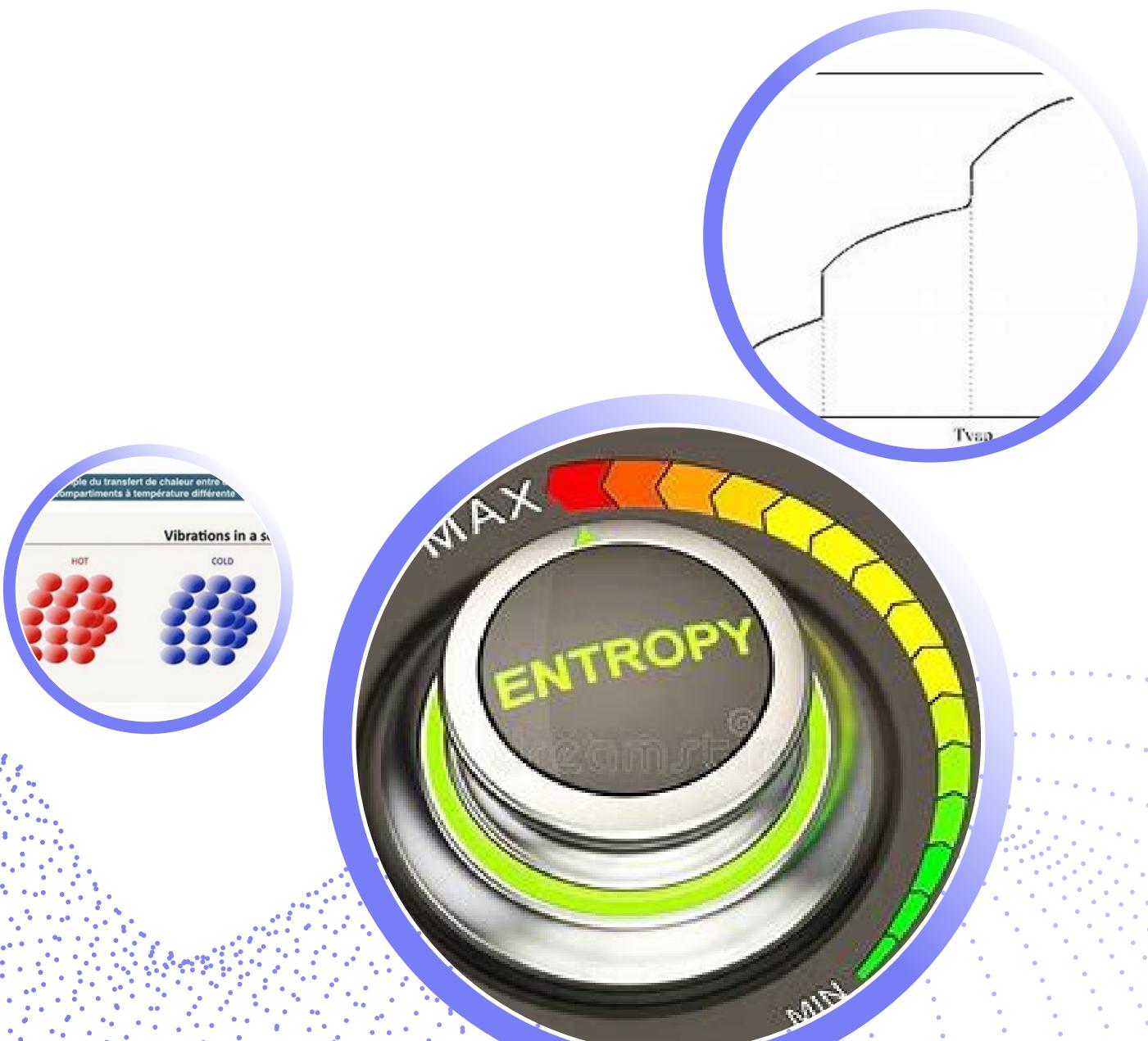
```
MATCH (u:User)-[:PERFORMS]->(a:Activity)  
RETURN u.name, COUNT(a) AS degree  
ORDER BY degree DESC  
LIMIT 5;
```

The screenshot shows the Neo4j browser interface with a query results table. The table has two columns: 'u.name' and 'degree'. The results are as follows:

	u.name	degree
1	"USR0040"	35
2	"USR0056"	35
3	"USR0060"	35
4	"USR0097"	35
5	"USR0018"	35
6	"USR0033"	35
7		15

At the bottom of the table, a message reads: "Started streaming 50 records after 4 ms and completed after 7 ms."

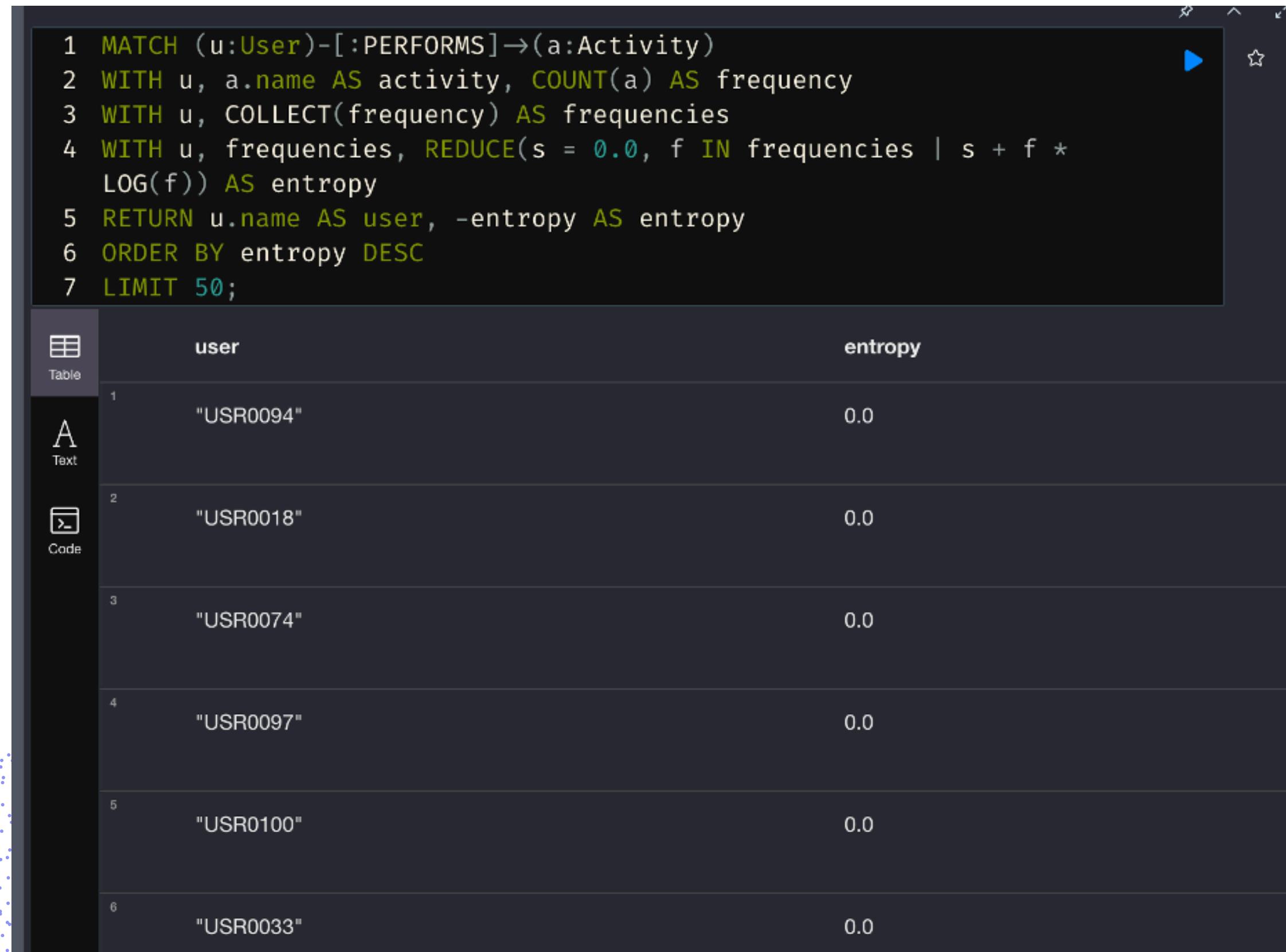
NOTION D'ENTROPIE



L'entropie mesure le niveau de désordre ou d'incertitude dans un système. Dans le contexte des graphes, l'entropie peut être utilisée pour mesurer l'imprévisibilité des interactions d'un utilisateur.

IMPLEMENTATION D'ENTROPIE

Calcul de l'entropie des activités



The screenshot shows a Neo4j browser window with a query editor and a results table.

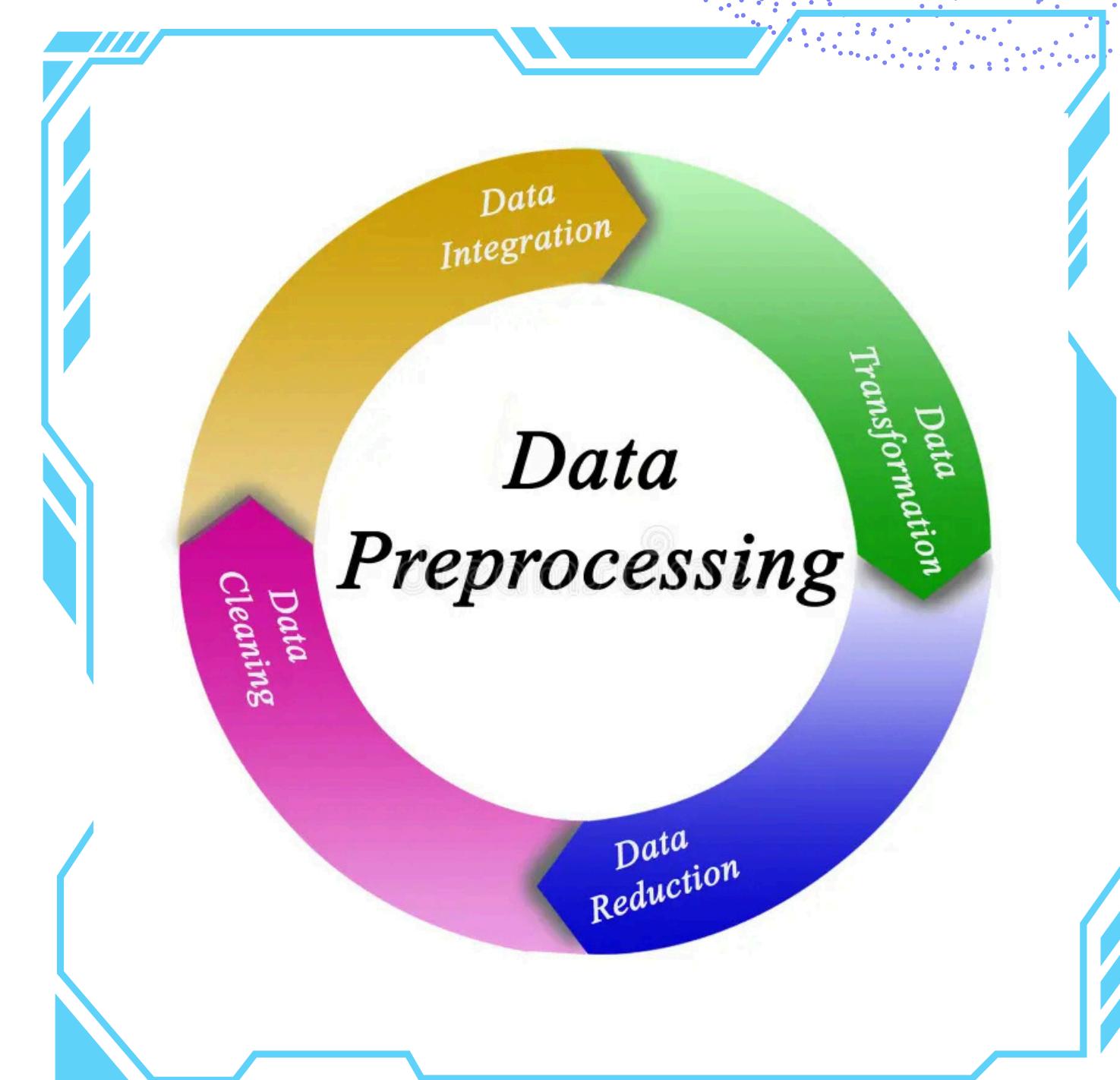
```
1 MATCH (u:User)-[:PERFORMS]→(a:Activity)
2 WITH u, a.name AS activity, COUNT(a) AS frequency
3 WITH u, COLLECT(frequency) AS frequencies
4 WITH u, frequencies, REDUCE(s = 0.0, f IN frequencies | s + f * LOG(f)) AS entropy
5 RETURN u.name AS user, -entropy AS entropy
6 ORDER BY entropy DESC
7 LIMIT 50;
```

The results table has two columns: "user" and "entropy". The data is as follows:

user	entropy
"USR0094"	0.0
"USR0018"	0.0
"USR0074"	0.0
"USR0097"	0.0
"USR0100"	0.0
"USR0033"	0.0

IMPLÉMENTATION ET EXPERIMENTATIONS

- 1 Prétraitement et extraction de caractéristiques
- 2 Exploration comportementale
- 3 Modèles supervisés
- 4 Modèles non supervisés
- 5 Comparaison finale des modèles



PRÉTRAITEMENT ET EXTRACTION DE CARACTÉRISTIQUES

Nettoyage & Normalisation :

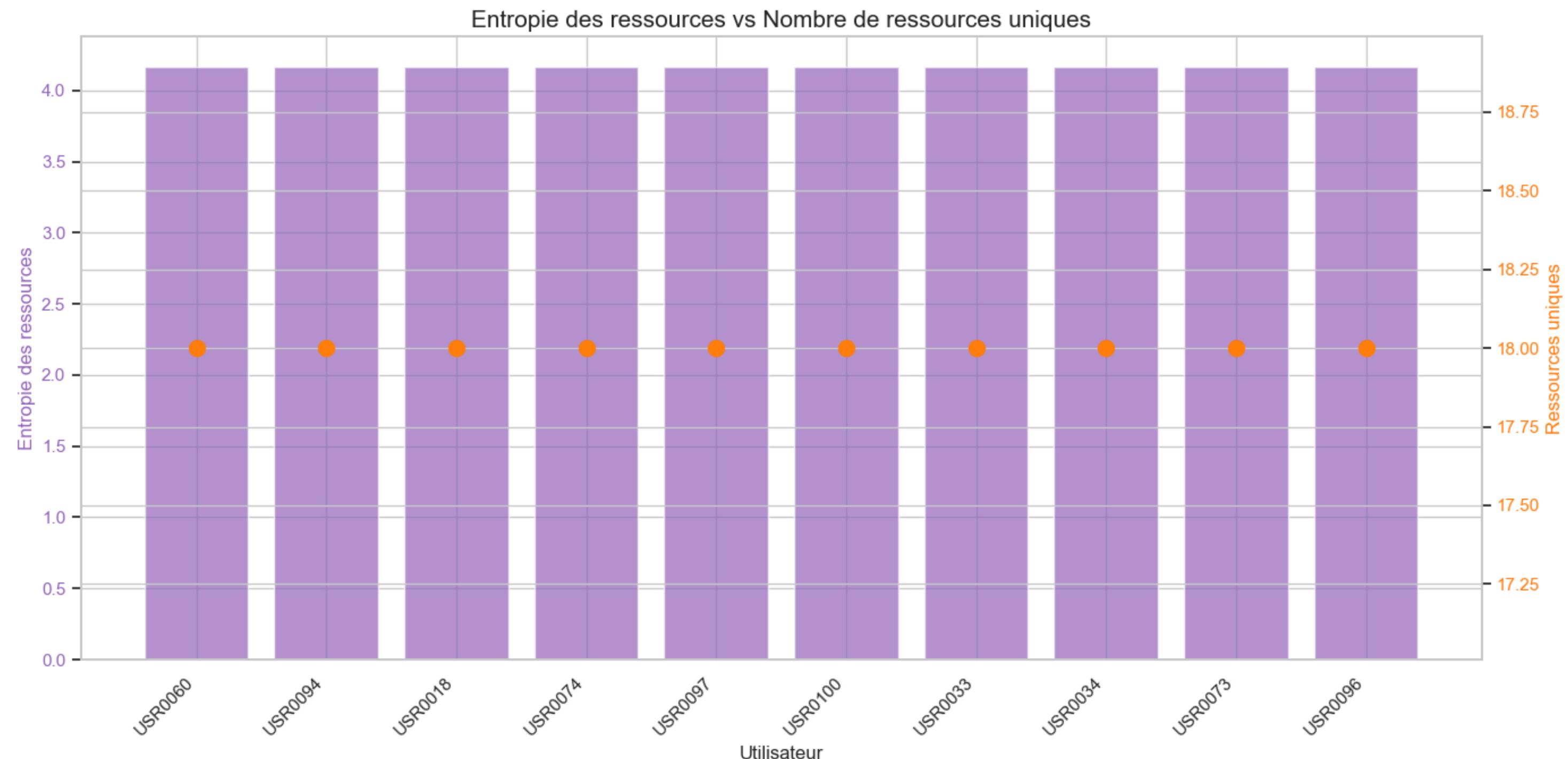
- Suppression des doublons
- Uniformisation des données

Enrichissement des profils utilisateurs avec des variables comportementales :

- Entropie : mesure de la diversité des comportements
- Ratio hors horaires : proportion d'activités en dehors des plages horaires classiques
- Variété de ressources/systèmes utilisés

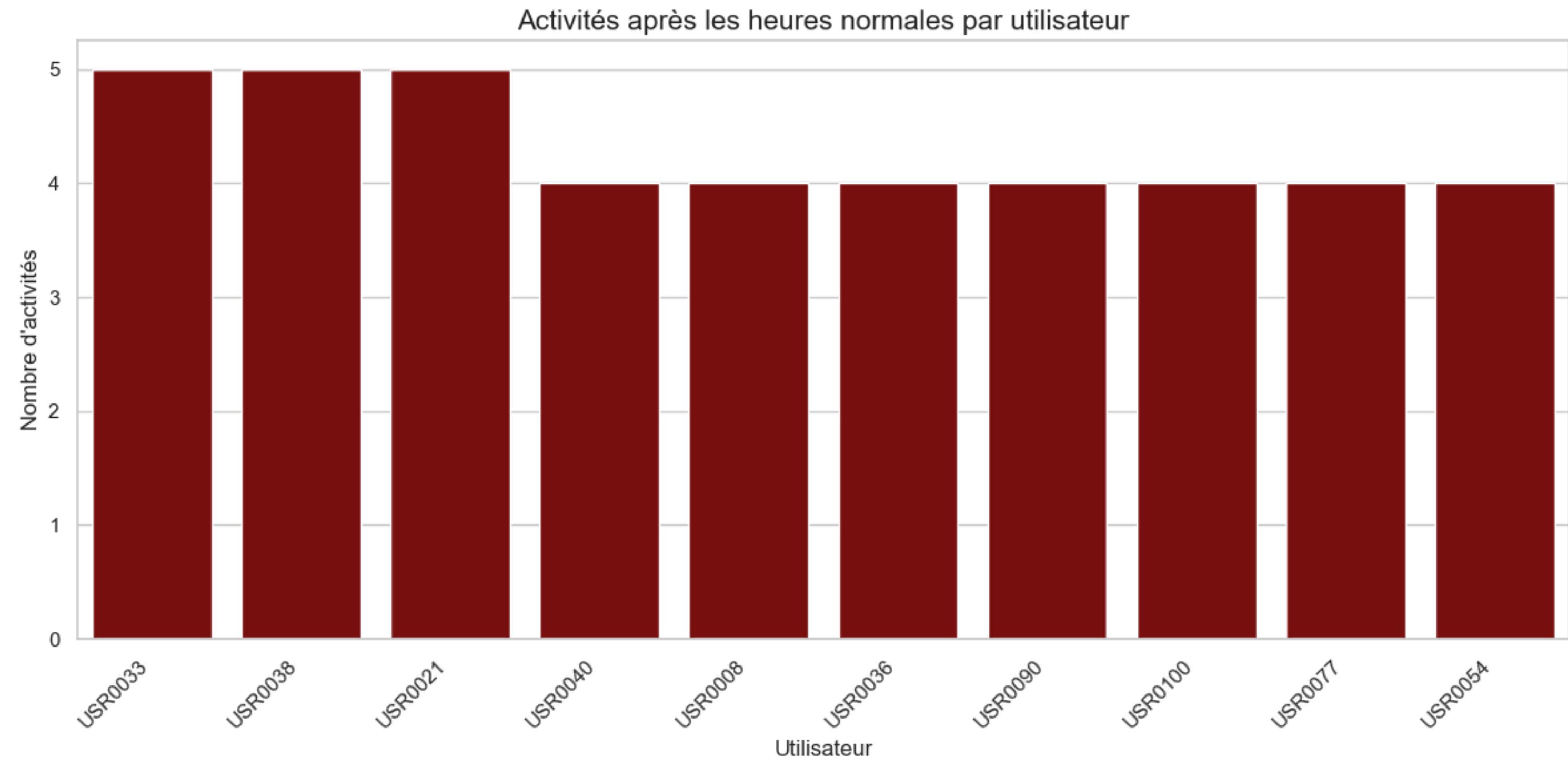
EXPLORATION COMPORTEMENTALE

Entropie des ressources vs nombre de ressources uniques



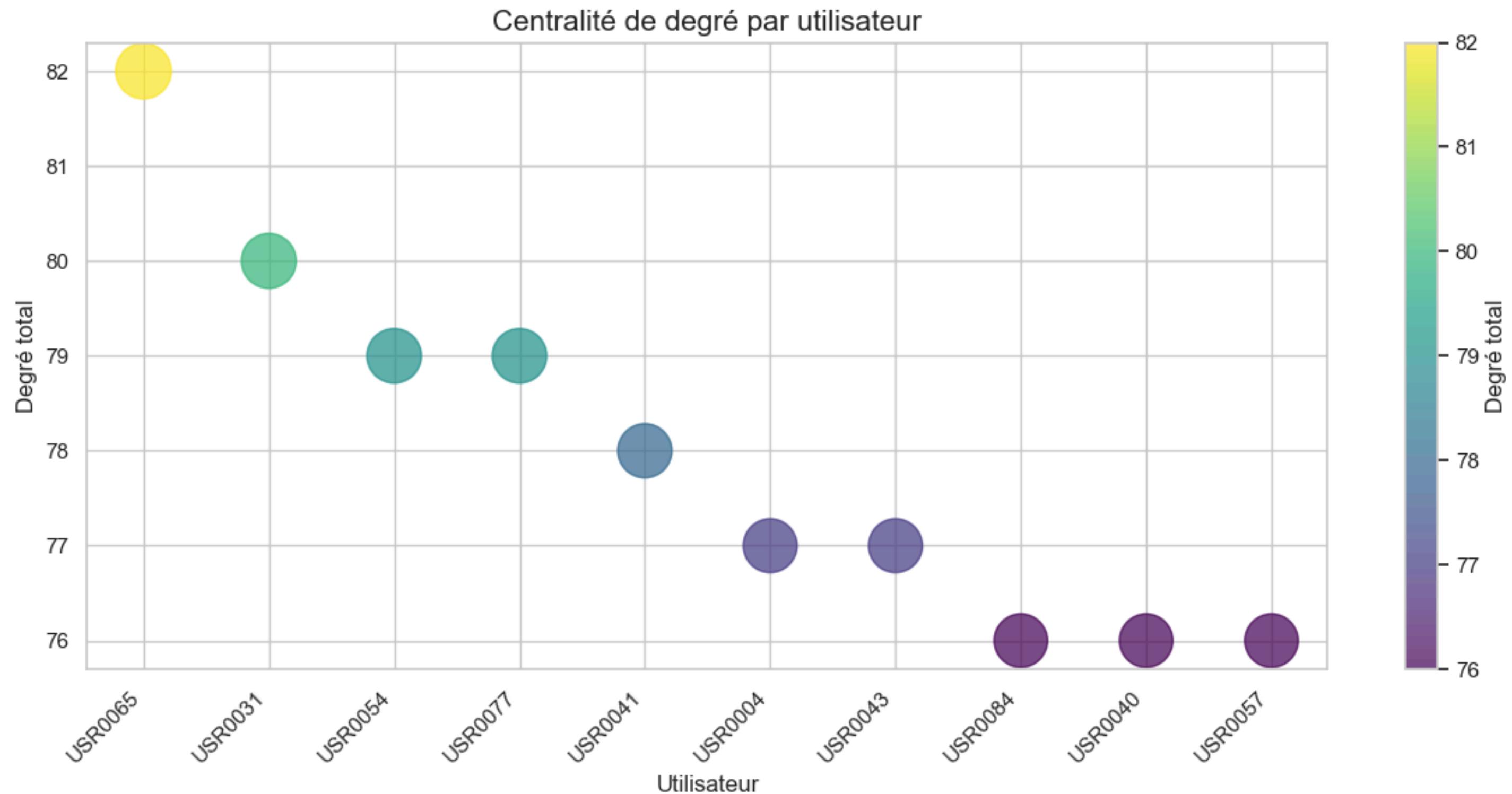
EXPLORATION COMPORTEMENTALE

Activité hors heures de bureau



EXPLORATION COMPORTEMENTALE

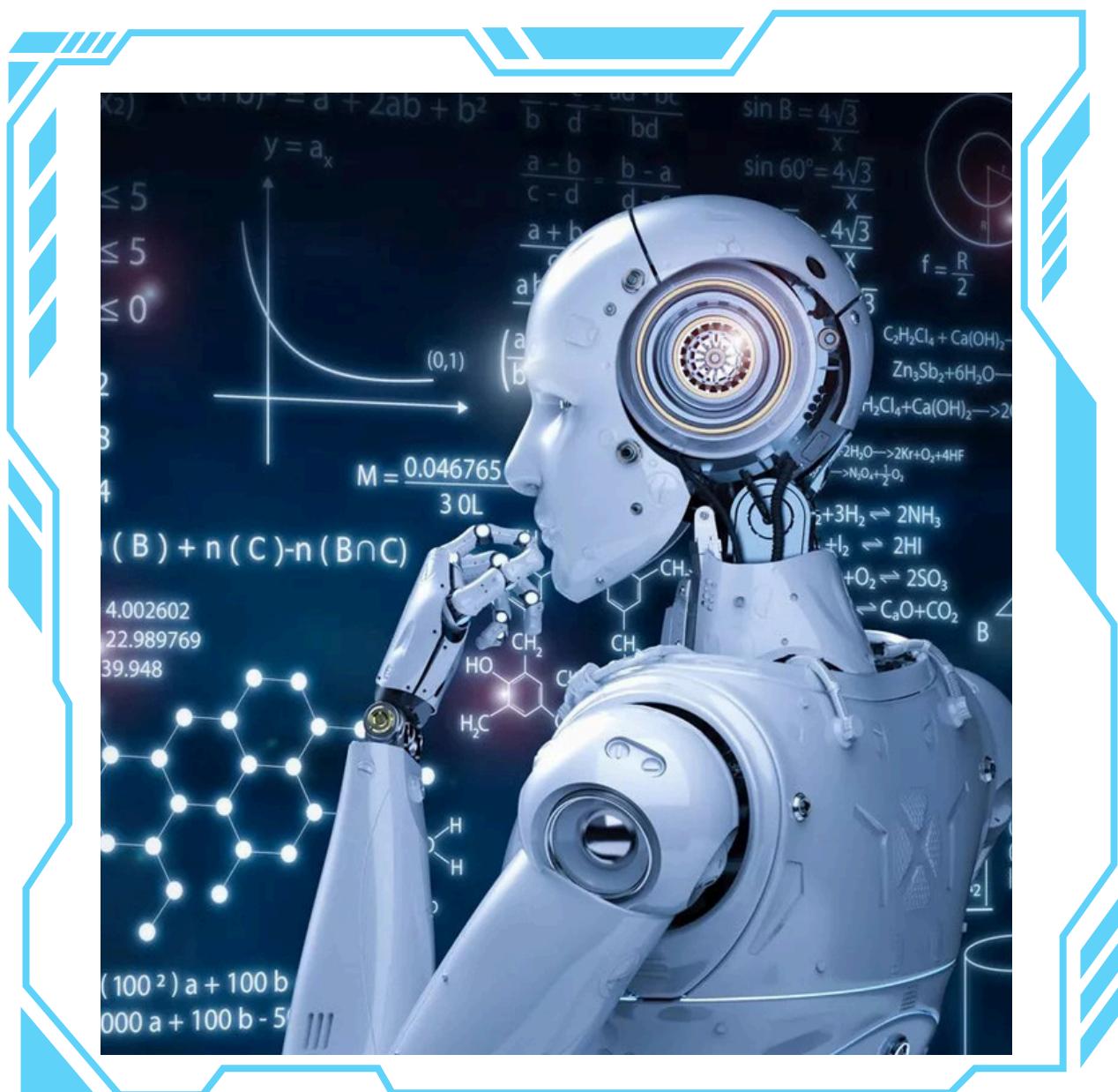
Centralité des utilisateurs



MODÈLES SUPERVISÉS

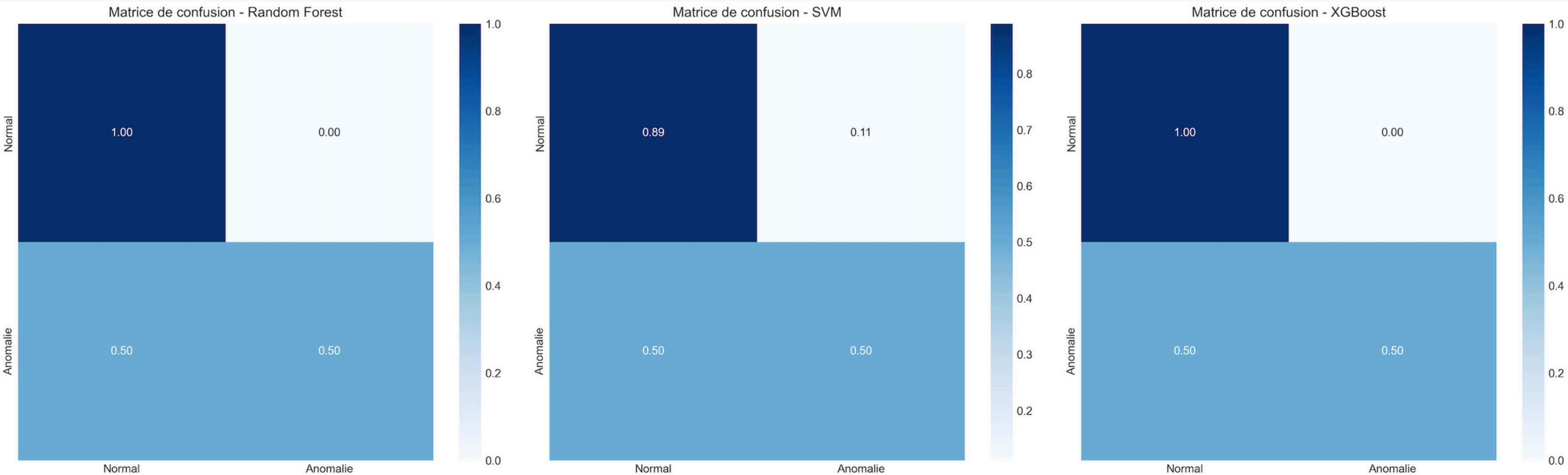
Algorithmes de classification supervisée :

- **Random Forest**: construit plusieurs arbres de décision sur des sous-ensembles aléatoires et vote sur la classification finale. Il détecte les anomalies par apprentissage sur des exemples étiquetés.
- **XGBoost (Extreme Gradient Boosting)**: algorithme de boosting qui optimise les erreurs successives en mettant plus d'accent sur les exemples mal classés. Il s'adapte bien aux structures de données complexes.
- **SVM (Support Vector Machine)**: sépare les comportements normaux des malveillants à l'aide d'un hyperplan optimal basé sur des exemples labellisés.



MODÈLES SUPERVISÉS

Matrices de confusion



Random Forest

- 100% précision (comportements normaux)
- 50% précision (anomalies)

SVM

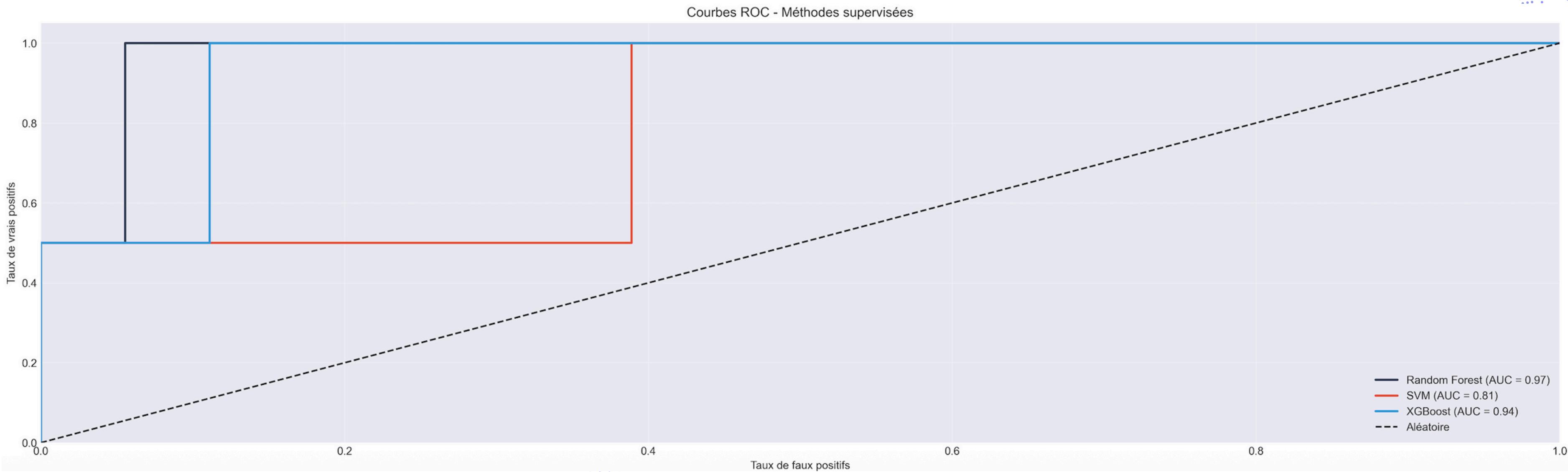
- 89% précision (comportements normaux) / 11% faux positifs
- 50% précision (anomalies)

XGBoost

- 100% précision (comportements normaux)
- 50% précision (anomalies)

MODÈLES SUPERVISÉS

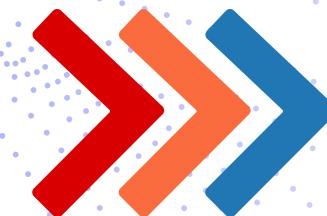
Les courbes ROC (Receiver Operating Characteristic) et les valeurs d'AUC (Area Under the Curve)



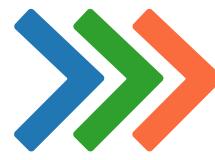
Random Forest : AUC = 0.97

XGBoost : AUC = 0.94

SVM : AUC = 0.81



Plus l'AUC est proche de 1, meilleur est le modèle.

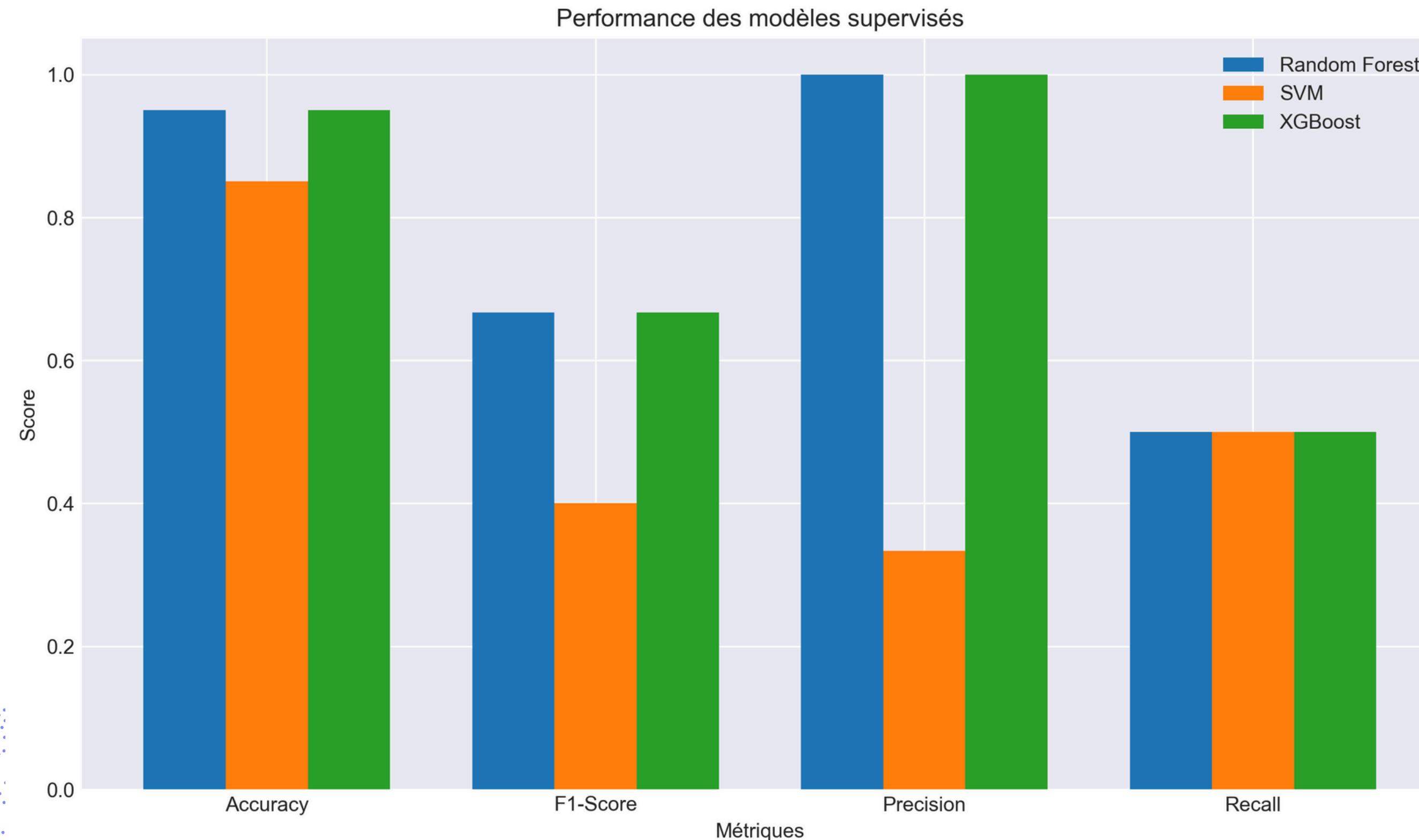


-Random Forest et XGBoost:

- Précision parfaite (1.0), pas de faux positifs.
- Rappel de 50%, ils ne détectent que la moitié des anomalies.

-SVM:

- Précision faible (0.33) et F1-Score faible (0.40).
- Produit beaucoup de faux positifs et est moins fiable que les autres modèles.



MODEÈLES NON SUPERVISÉS

Algorithmes de détection d'anomalies:

- **Isolation Forest :**

Isole les observations en construisant des arbres aléatoires. Les points rapidement isolés sont considérés comme anormaux, permettant d'identifier des comportements rares.

- **One-Class SVM :**

Apprend la frontière du comportement normal. Les observations en dehors de cette frontière sont classées comme anomalies.

- **Autoencodeur :**

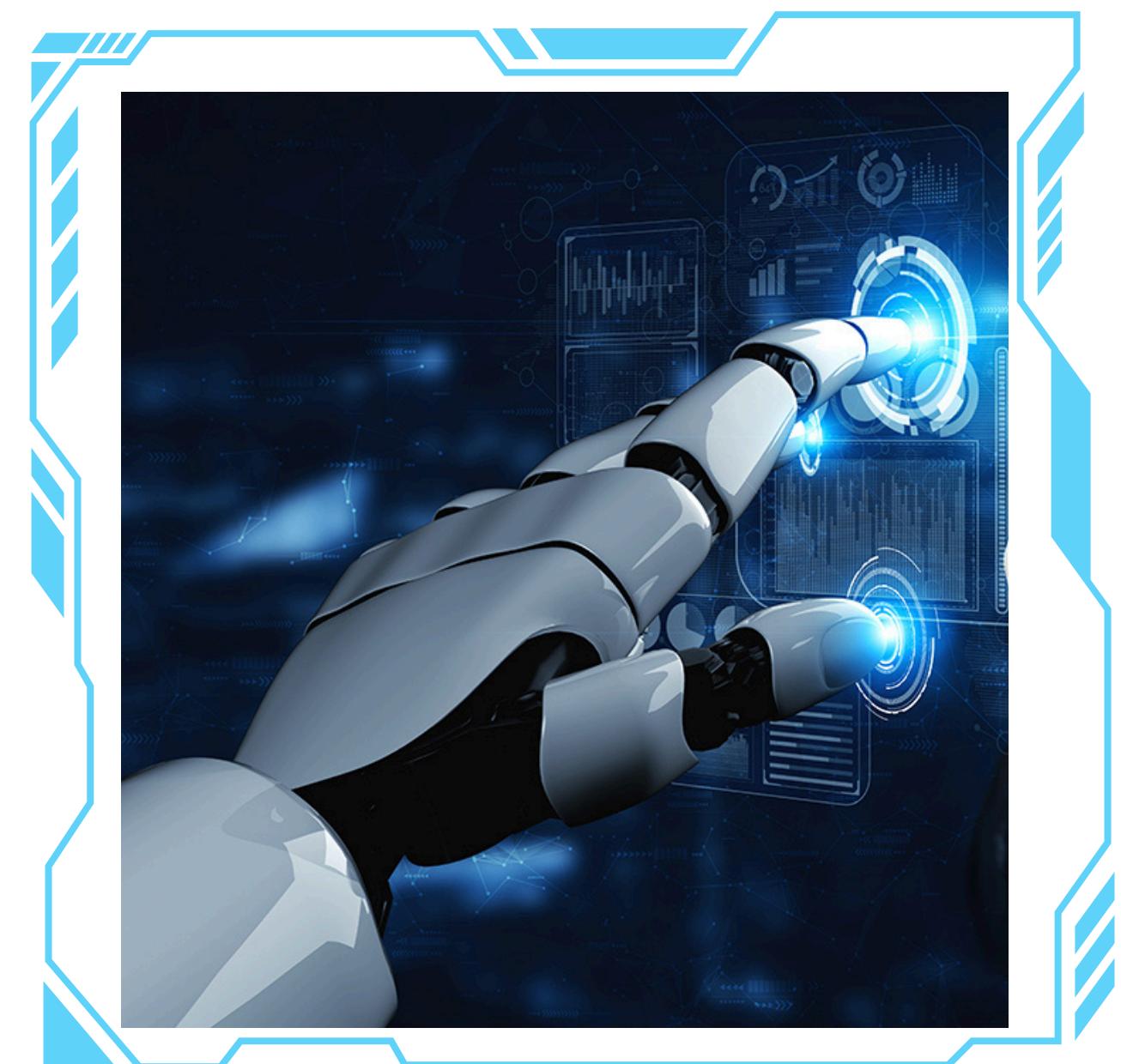
Réseau de neurones entraîné à reproduire les données normales. Une erreur de reconstruction élevée indique un comportement inhabituel.

- **K-Means :**

Regroupe les utilisateurs en clusters. Ceux éloignés du centre de leur cluster sont considérés comme anomalies.

- **DBSCAN : (Density-Based Spatial Clustering of Applications with Noise)**

Déetecte les regroupements denses. Les points trop éloignés des groupes sont considérés comme anomalies.



MODÈLES NON SUPERVISÉS

Matrices de confusion

Isolation Forest :

100% de précision et de rappel (aucune erreur détectée).

OCSVM :

Précision et rappel modérés à 0.4, avec quelques faux positifs et faux négatifs.

K-means :

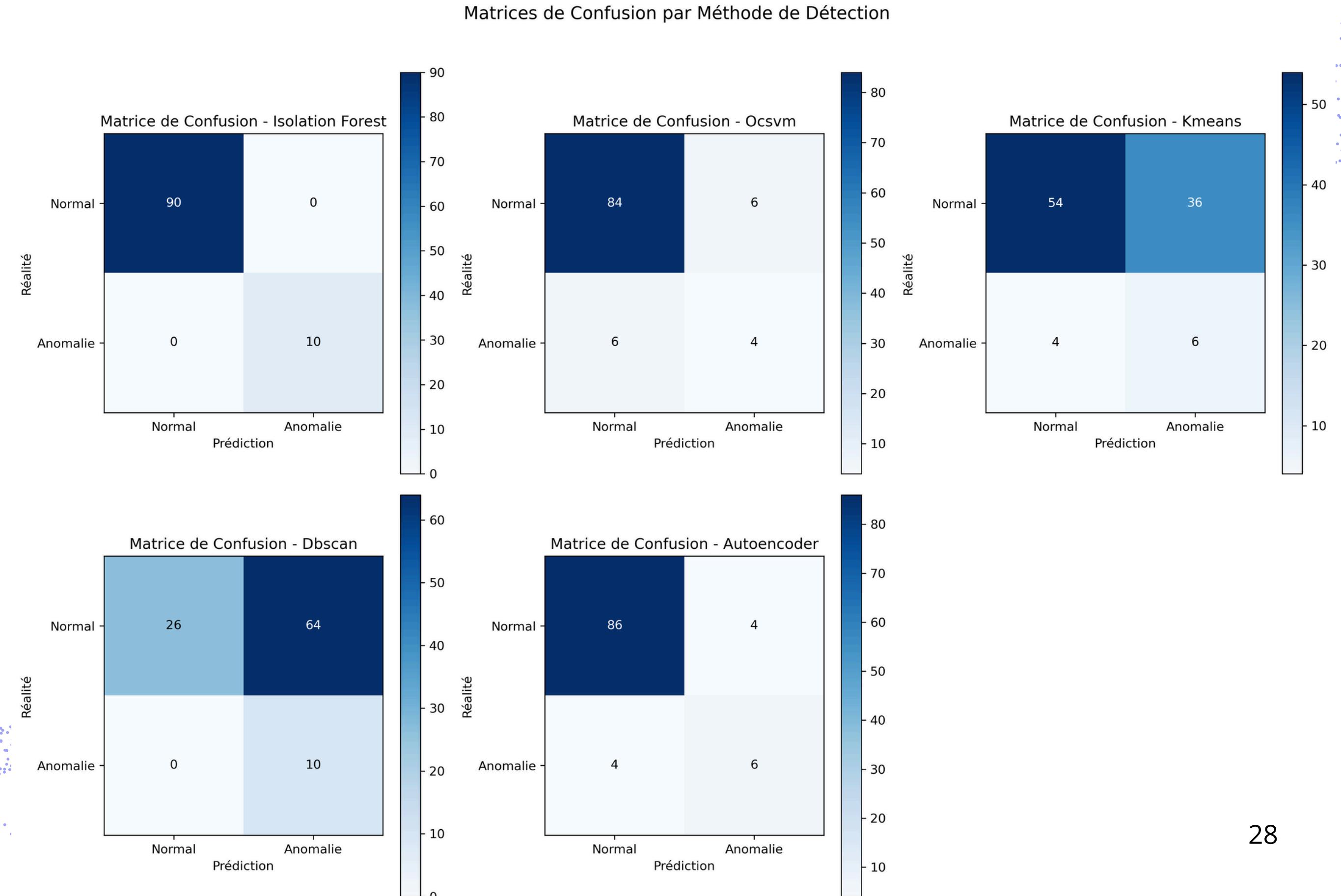
Faible précision (0.14) due à un grand nombre de faux positifs, et un rappel modéré (0.6).

DBSCAN :

Déetecte toutes les anomalies avec un rappel parfait (1.0), mais souffre d'un grand nombre de faux positifs.

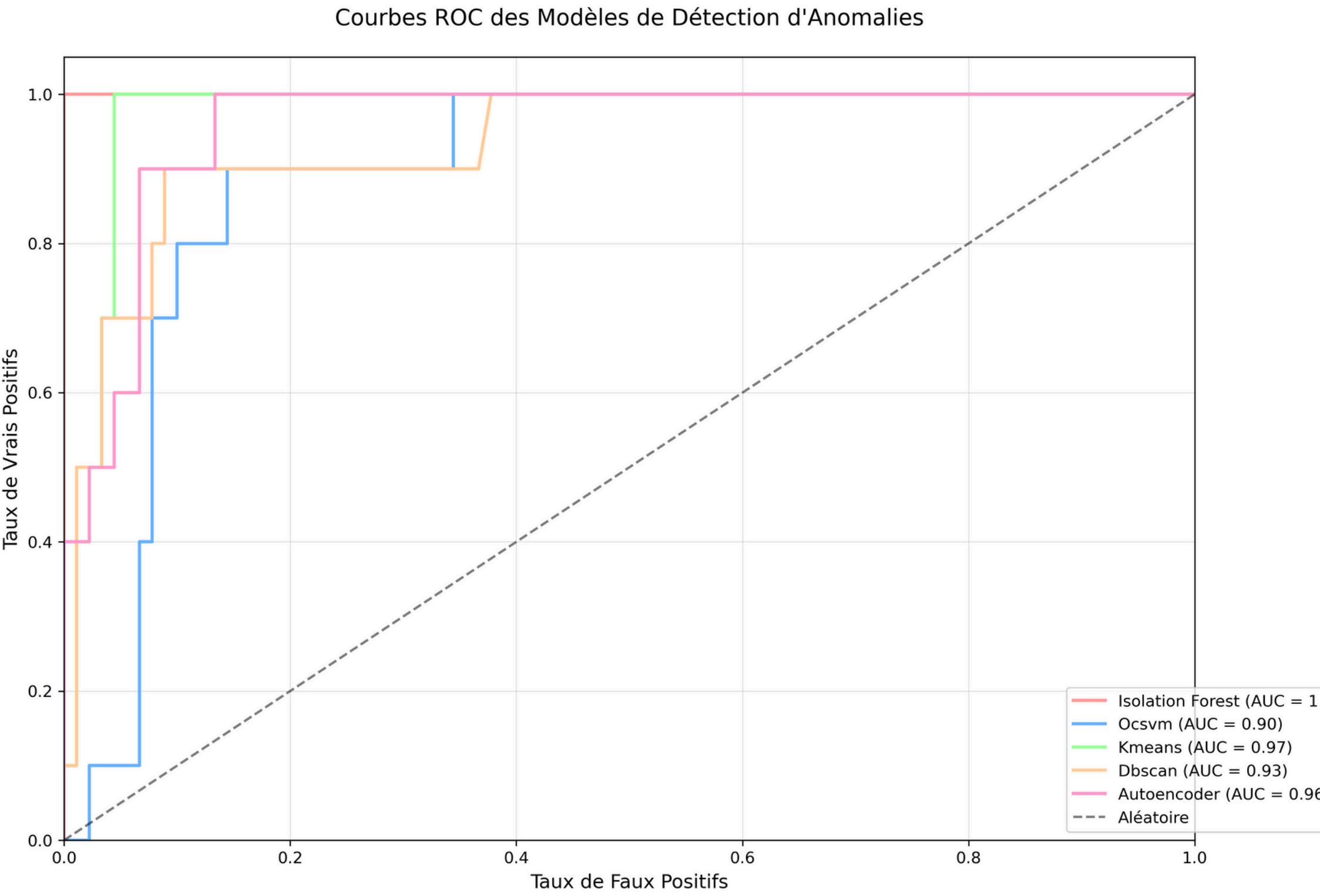
Autoencoder :

Bon équilibre entre précision et rappel (0.7 chacun).



MODÈLES NON SUPERVISÉS

Les courbes ROC (Receiver Operating Characteristic) et les valeurs d'AUC (Area Under the Curve)



Isolation Forest : AUC = 1.00 (parfait).

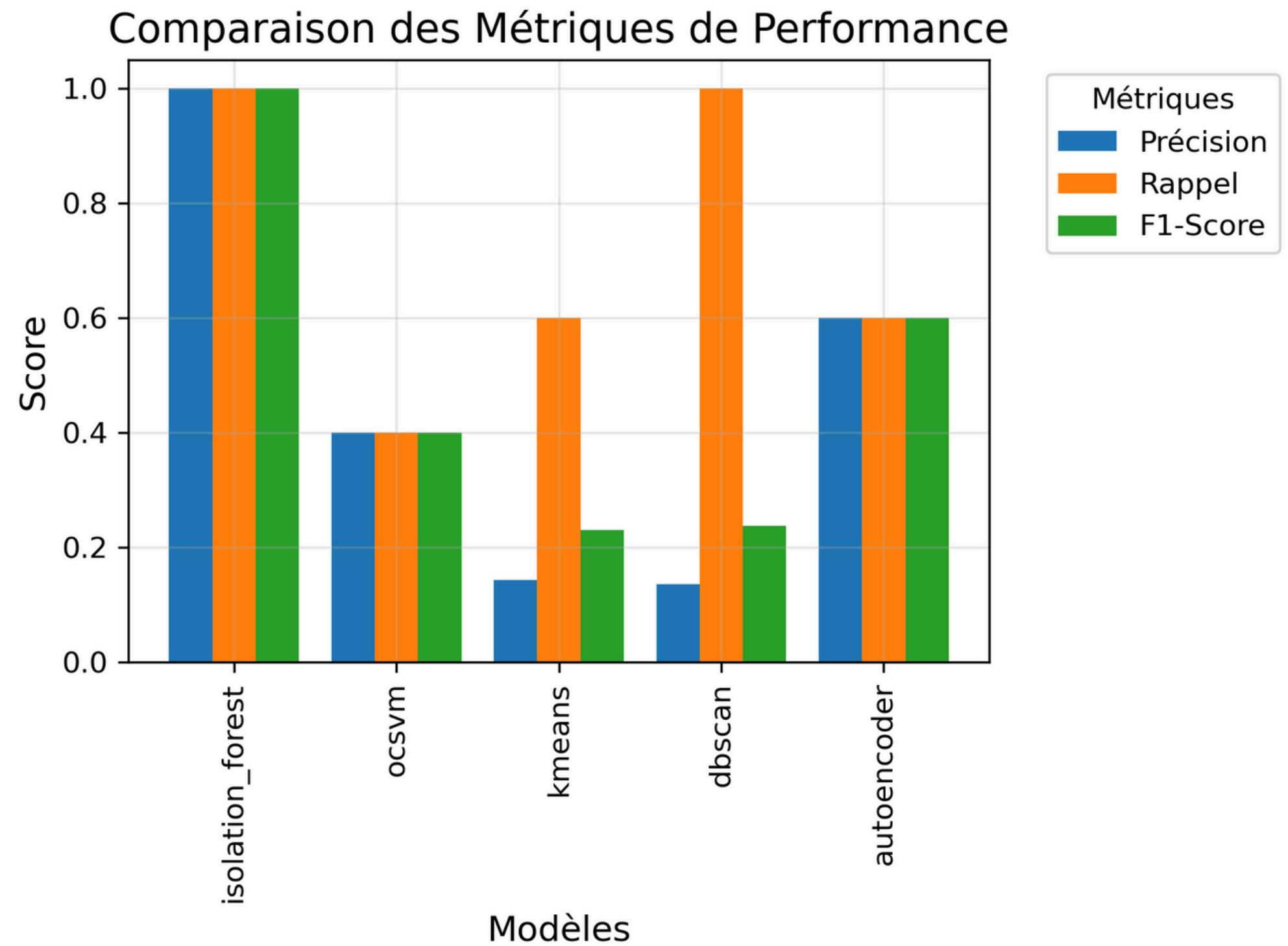
- Autoencoder : AUC = 0.98.
- K-means : AUC = 0.97.
- DBSCAN : AUC = 0.93.
- OCSVM : AUC = 0.90.



Plus l'AUC est proche de 1, meilleur est le modèle.

➤ • **Isolation Forest**: Précision, rappel et F1-Score parfaits (1.0) -> Aucune erreur, très efficace.

- **OCSVM**: Scores équilibrés mais faibles (0.4) -> Fait autant de faux positifs que de faux négatifs.
- **K-Means**: Rappel modéré (0.6) mais très faible précision (0.14) -> Trop de fausses alertes.
- **DBSCAN**: Rappel parfait (1.0) mais faible précision (0.14) -> Déetecte toutes les anomalies, mais beaucoup de faux positifs.
- **Autoencoder**: Bon équilibre (0.7 partout) -> Fiable et stable pour déetecter les anomalies.



RÉSULTATS ET DISCUSSION

- 1 **Analyse des Résultats**
- 2 **Limitations des Modèles performants**
- 3 **Comparaison avec les Solutions Traditionnelles
(SIEM, IDS)**



ANALYSE DES RÉSULTATS



Modèles les plus performants

- **Isolation Forest** : AUC = 1.00, F1-Score = 1.00 → Précision et rappel parfaits.
- **Random Forest** : AUC = 0.97 → Très bonne détection des comportements normaux.
- **XGBoost** : AUC = 0.94 → Solide, proche de Random Forest.

Modèles prometteurs

- **Autoencoder** : F1-Score = 0.70 → Bon équilibre entre précision et rappel, performant dans des contextes complexes.

Modèles à performance déséquilibrée

- **DBSCAN & K-Means** : Bon rappel, mais faible précision → Trop de faux positifs.

Modèle moyen

- **SVM** : AUC = 0.81 → Résultats corrects mais moins fiables que les meilleurs modèles.

LIMITATIONS DES MODÈLES PERFORMANTS

Isolation Forest

- Sensibilité aux paramètres d'entrée et au bruit
- Performances réduites si les anomalies sont proches des données normales

LIMITATIONS DES MODÈLES PERFORMANTS



Random Forest

1

- Déséquilibre des classes : Faible capacité à identifier les anomalies dans des données déséquilibrées
- Complexité et sur-apprentissage : Sensible au bruit et à la complexité
- Interprétabilité limitée : Difficulté à expliquer les décisions.

XGBoost

2

- Dépendance aux hyperparamètres : Sensibilité aux choix des paramètres
- Sur-ajustement : Risque de sur-apprentissage si les données sont bruyantes
- Complexité computationnelle : Nécessite beaucoup de ressources pour l'entraînement

COMPARAISON AVEC LES SOLUTIONS TRADITIONNELLES (SIEM, IDS)

Systèmes IDS(Systèmes de Détection d'Intrusion)

- 1 • Basé sur des signatures : Limité aux attaques identifiées, vulnérable aux nouvelles attaques
- Basé sur des anomalies : Susceptible de générer de nombreux faux positifs

Systèmes SIEM(Systèmes de Gestion des Événements de Sécurité)

- 2 • Centralisation des événements : Bonne vision d'ensemble mais limité par les règles préétablies
- Faux positifs : En raison de règles rigides et d'une adaptation lente



COMPARAISON AVEC NOTRE APPROCHE

- 1 Adaptabilité et détection proactive :
Identifie des anomalies même inconnues
- 2 Moins de faux positifs :
Précision accrue grâce aux modèles ML
- 3 Analyse comportementale :
Déetecte des menaces internes difficiles à capturer par des systèmes traditionnels



APPLICATIONS ET AMÉLIORATIONS FUTURES



Applications possibles

- Secteurs : Finance, infrastructures critiques, entreprises technologiques
- Objectif : Déetecter la fraude interne, les fuites de données, et surveiller les comportements suspects en temps réel

Améliorations futures

- Enrichissement des données : Intégrer plus de sources pour une détection plus précise
- Optimisation des modèles : Améliorer la gestion du déséquilibre des classes
- Adaptation dynamique : Réévaluation continue des seuils pour s'adapter aux nouveaux comportements

CONCLUSION

- Notre approche montre une excellente capacité à détecter les comportements normaux et à discriminer certaines menaces.
- Des améliorations sont nécessaires pour mieux gérer les anomalies rares et évoluer avec les menaces internes.
- Innovation : Une solution plus flexible et efficace pour la détection des menaces internes par rapport aux systèmes traditionnels.



**MERCI POUR
VOTRE ATTENTION**