I3T

Modélisation prédictive du niveau d'eau d'un réservoir

Réalisé par : Yesmine Srairi

Encadré par : Mr Sahbi Gsouma

8 août 2025

Introduction

À la suite des étapes de préparation, de nettoyage et de consolidation des données décrites dans le rapport précédent, ce document se concentre sur **la phase de modélisation prédictive** du niveau d'eau dans un réservoir alimenté par deux stations de pompage à distance.

L'objectif est de développer des modèles permettant de prédire en temps réel l'évolution du niveau d'eau, afin d'anticiper les situations critiques, telles que les débordements ou les niveaux trop bas, et de faciliter la prise de décision automatisée (arrêt des pompes, alertes, etc.).

Pour cela, deux approches complémentaires ont été mises en œuvre :
— Un modèle **SARIMAX**, adapté à l'analyse de séries temporelles avec composantes saisonnières et exogènes ;

— Un modèle **LSTM** (Long Short-Term Memory), basé sur l'apprentissage profond, capable de capturer des dynamiques non linéaires complexes dans des séquences temporelles.

Les données utilisées proviennent des vues nettoyées générées lors de la phase BI, en particulier :

— Les débits issus des deux stations de pompage (*TROZA A* et *TROZA B*),
— Les états de fonctionnement des pompes (*marche/arrêt*),
— Le niveau moyen du réservoir,
— Les débits sortants *debit_dn100_avg* et *debit_dn250_avg* ;
— Les indicateurs de niveau critique (*niveau très haut/très bas*).

Ce rapport présente :

— une analyse exploratoire des séries temporelles retenues,
— la construction et l'entraînement des deux modèles,
— une évaluation comparative des performances sur des données réelles,
— et une discussion sur l'efficacité de chaque approche dans un contexte de déploiement sur un système *SCADA* .

Accès aux fichiers du projet

L'ensemble des fichiers sources développés dans le cadre de ce projet est fourni dans le dossier compressé joint à ce rapport. Ce dossier contient :

— **Les scripts Python** utilisés pour le traitement, la modélisation et l'analyse des séries temporelles :

  — `Data.py` : préparation, nettoyage et fusion des données.
  — `TSA_Analysis.py` : analyse exploratoire des séries temporelles et visualisation.
  — `SARIMAX.py` : construction et évaluation du modèle SARIMAX.

— `LSTM.py` : construction et évaluation du modèle LSTM.

— **Les modèles entraînés** enregistrés au format `.pkl`, permettant une réutilisation sans réentraînement :
  — `sarimax_model.pkl`
  — `lstm_model.pkl`

— **Les données nettoyés** au format `.csv`, extraits des vues SQL et utilisés comme données d'entrée pour les modèles.

— **Les figures** générées par les scripts d'analyse et de modélisation, illustrant les résultats et les performances des modèles.

Tous les fichiers sont commentés et organisés de manière modulaire afin de faciliter la lecture, la reproduction des résultats, ou la réutilisation dans d'autres projets.

1. Fichier `Data.py`

Objectif

Le fichier `Data.py` a pour rôle principal de préparer les données brutes issues du système SCADA afin de les rendre exploitables pour les modèles de prévision. Il regroupe les étapes essentielles de :

— Chargement des vues nettoyées depuis les exports SQL au format CSV ;
— Fusion cohérente des différentes sources de données temporelles ;
— Sélection, renommage et nettoyage des variables ;
— Préparation de séries temporelles alignées pour les modèles SARIMAX et LSTM.

Fonctions principales

Le fichier `Data.py` est structuré autour de plusieurs fonctions clés assurant la connexion à la base de données, le chargement et le nettoyage des données ainsi que leur fusion temporelle. Voici les principales fonctions utilisées :

— `get_engine(conn_str)` : établit la connexion à la base de données SQL Server via SQLAlchemy. Cette fonction teste la connexion et retourne un objet `engine` utilisé pour les requêtes SQL.

— `load_or_read_csv(view_name, file_path, engine)` : récupère les données depuis la base SQL (vue `view_name`), puis sauvegarde le fichier CSV localement pour un usage futur.

— `unify_timestamps(df_a, df_b, df_reservoir)` : unifie les horodatages (timestamps) issus des trois sources (`TROZA_A`, `TROZA_B`, `View_Reservoir`) en créant un index temporel maître.

— `clean_troza_columns(df_a, df_b)` : renomme les colonnes spécifiques aux stations TROZA A et B pour uniformiser les noms, facilitant ainsi la fusion et l'analyse ultérieure.

— `main()` : fonction principale orchestrant le processus complet :
  — connexion à la base,
  — chargement ou lecture locale des vues,
  — nettoyage des noms de colonnes,
  — alignement temporel des données,
  — suppression des lignes comportant des valeurs manquantes dans l'une quelconque des sources,
  — sauvegarde des DataFrames nettoyés au format CSV pour utilisation dans les étapes suivantes.

Sortie

— Un DataFrame Pandas contenant les séries temporelles alignées prêtes à être utilisées par les fichiers `SARIMAX.py` et `LSTM.py`.

— Une cohérence temporelle entre les variables d'entrée (`Débit_A`, `Débit_B`, etc.) et la variable cible (`niveau_avg`).

2. TSA_Analysis.py :s

Objectif

Le fichier `TSA_Analysis.py` est dédié à l'analyse préliminaire des séries temporelles. Il permet de mieux comprendre la dynamique des données, d'identifier les composantes saisonnières, les tendances éventuelles, et de déterminer si une différenciation est nécessaire avant la modélisation. : les sorties sont exclusivement des visualisations graphiques et des résultats de tests statistiques.

Fonctions principales :

— **Chargement des données** : les fichiers CSV nettoyés sont importés.

— **Tracé de l'évolution temporelle du niveau d'eau** : permet de visualiser la stabilité ou la variabilité globale du niveau dans le réservoir.

— **Analyse de saisonnalité** : par regroupement des d
onnées par heure ou par jour pour détecter des motifs périodiques.

— **Tests de stationnarité** :
  — Test de **Dickey-Fuller augmenté (ADF)**.

— Test de **KPSS**.

— **Tracés ACF et PACF** : pour analyser l'autocorrélation et identifier d'éventuels ordres de modèles AR ou MA.

Visualisations générées et interprétations :

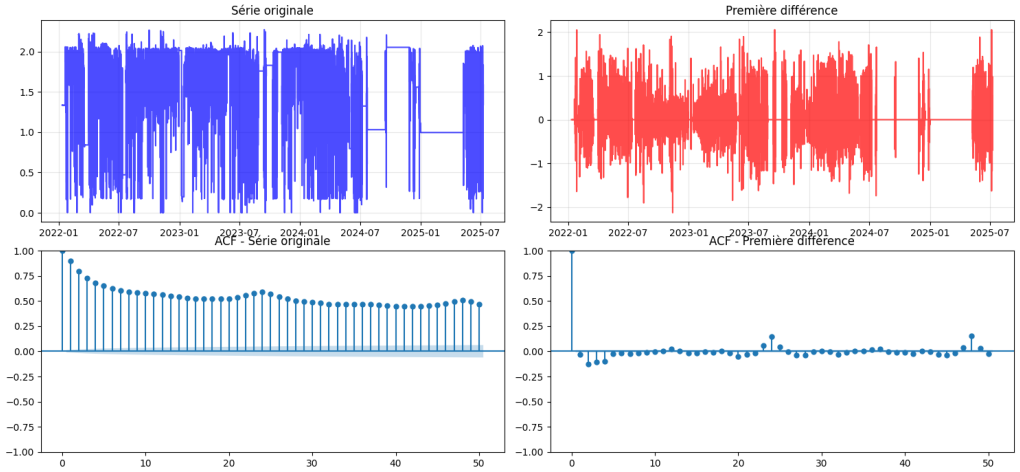

FIGURE 1 – Quatre visualisations du niveau d'eau : série originale, première différence, ACF de la série originale et ACF de la série différenciée

Cette figure comprend quatre graphiques distincts illustrant différents aspects du niveau d'eau dans le réservoir :

— **Série originale** : on observe des fluctuations importantes et persistantes dans une amplitude large (forme rectangulaire), ce qui indique une non-stationnarité de la variance et justifie une transformation pour stabiliser la série. ;

— **Première différence** : la série différenciée paraît beaucoup plus stationnaire, avec des variations centrées autour de zéro, indiquant que la différenciation est pertinente pour stabiliser la moyenne.

— **ACF de la série originale** : l'autocorrélation décroît lentement, signe typique d'une série non stationnaire. On observe également des pics significatifs aux lags multiples de 24, ce qui indique une saisonnalité quotidienne (cycle de 24 heures) dans les données.

— **ACF de la série différenciée** : l'autocorrélation diminue rapidement, avec une valeur significative uniquement au lag 0, ce qui confirme que la différenciation atténue la dépendance temporelle excessive. La saisonnalité à 24 heures reste toutefois visible par des pics à ces intervalles, confirmant un comportement saisonnier persistant même après différenciation.

Ces observations justifient l'usage d'une différenciation dans le modèle SARIMAX,

visant à rendre la série stationnaire avant modélisation.

Interprétation des variables et implications pour la modélisation SARIMAX :

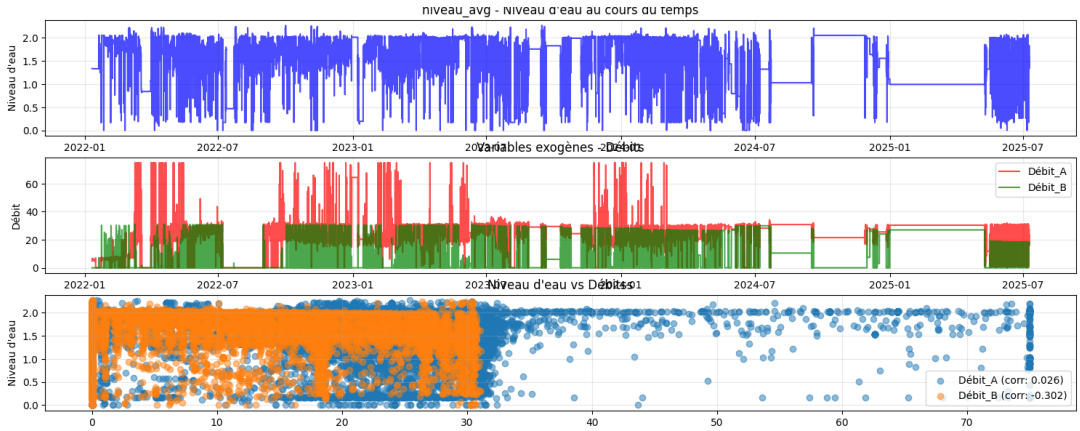

FIGURE 2 –
(a) Évolution temporelle du niveau d'eau dans le réservoir.
(b) Autocorrélation des débits *Débit_A* et *Débit_B* .
(c) Diagrammes de dispersion (scatter plots) entre les débits et le niveau d'eau, .

Variable cible (Niveau d'eau)

La série du niveau d'eau présente une non-stationnarité manifeste, avec des phases de volatilité élevée et des ruptures structurelles observées autour du milieu des années 2024-2025.

Cette caractéristique impose l'utilisation d'une différenciation ainsi qu'un traitement soigneux des valeurs aberrantes pour la modélisation SARIMAX.

Relations entre variables exogènes
Les débits des stations de pompage, *Débit_A* et *Débit_B*, montrent une autocorrélation négative notable : ils ont tendance à évoluer en sens opposé.

Valeur prédictive pour le modèle

Les deux variables exogènes présentent une corrélation modérée avec le niveau d'eau, mais de manière différente :

— *Débit_A* : corrélation faible et positive (environ 0,026)
— *Débit_B* : corrélation modérée et négative (environ -0,302)

Ces corrélations, bien que relativement faibles, justifient l'inclusion des deux débits comme variables exogènes dans le modèle SARIMAX.
De plus, la corrélation négative entre ces deux variables réduit les risques de multicolinéarité, ce qui est bénéfique pour la stabilité du modèle.

Implications pour la modélisation SARIMAX

Il est recommandé d'inclure *Débit_A* et *Débit_B* comme variables exogènes, car leur relation négative fournit une information complémentaire.
Cette configuration devrait permettre une amélioration par rapport à un modèle ARIMA simple, grâce à la prise en compte des relations systémiques détectées.

Analyse des Résultats de la Série Temporelle

Exploration initiale et statistiques descriptives

Les données temporelles couvrent la période du **10 janvier 2022 à 14h00** au **9 juillet 2025 à 10h00**, avec un total de **30 621 observations**. Les variables mesurées comprennent les débits (`Débit_A`, `Débit_B`, `debit_dn100_avg`, `debit_dn250_avg`) et les niveaux d'eau (`niveau_avg`, `niveau_tres_bas`, `niveau_tres_haut`).

Toutes les colonnes sont complètes. Les statistiques descriptives révèlent une forte variabilité dans les débits et les niveaux.

Corrélations avec la variable cible (`niveau_avg`) :
- `Débit_A` : corrélation faible mais positive (0.026)
- `Débit_B` : corrélation modérée et négative (-0.302)

Tests de stationnarité

Deux tests complémentaires ont été utilisés :
- **Test ADF** : p-value = 0.000, statistique ADF = -10.0552 ⇒ série **stationnaire** selon ce test.
- **Test KPSS** : statistique = 2.9675, p-value < 0.01 ⇒ série **non stationnaire** selon ce test.

Conclusion : les résultats contradictoires des deux tests suggèrent une stationnarité faible ou partielle. Une différenciation est donc recommandée après l'analyse des graphes .

Analyse de la saisonnalité

La force saisonnière calculée est de seulement 0.022, ce qui indique une composante saisonnière très faible dans la série temporelle du niveau d'eau
Néanmoins, malgré cette saisonnalité faible, une composante saisonnière a été incluse dans le modèle SARIMAX afin de capturer d'éventuelles variations récurrentes de faible amplitude, notamment à l'échelle quotidienne. Ce choix permet de ne pas négliger des effets saisonniers subtils qui pourraient améliorer les performances prédictives du modèle.

Sélection de modèles SARIMAX

L'identification des ordres du modèle SARIMAX a été guidée par l'observation des graphiques ACF et PACF. Ces graphiques ont révélé des décroissances progressives avec des pics significatifs à des retards multiples de 24 heures, suggérant la présence d'une composante saisonnière quotidienne. Cela a motivé l'inclusion d'un terme saisonnier dans

les modèles candidats.

Quatre modèles candidats ont été testés, en combinant différents ordres saisonniers :

TABLE 1 – Comparaison des modèles SARIMAX par AIC

Modèle	Ordre	Ordre saisonnier	AIC
M1	(1, 1, 1)	(1, 0, 1, 24)	-6285.95
M2	(1, 1, 1)	(1, 1, 1, 24)	-6036.52
M3	(1, 1, 2)	(1, 0, 1, 24)	**-6395.65**
M4	(1, 1, 2)	(1, 1, 1, 24)	-6161.38

Le modèle retenu est :
— **Ordre** : (1, 1, 2)
— **Ordre saisonnier** : (1, 0, 1, 24)
— **Critère AIC** : -6395.65

Impact des variables exogènes

Deux versions du modèle ont été comparées : avec et sans intégration des variables exogènes (`Débit_A`, `Débit_B`).

TABLE 2 – Comparaison des performances avec/sans variables exogènes

Métrique	Sans exogène	Avec exogène	Amélioration
AIC	-2949.46	-2682.58	-266.89
BIC	-2900.83	-2617.73	-283.10
RMSE	0.9634	0.8697	0.0936
MAE	0.8458	0.7481	0.0977

Conclusion : malgré une légère amélioration sur les métriques d'erreur (RMSE, MAE), l'intégration des variables exogènes dégrade fortement les critères AIC/BIC. Le modèle sans variables exogènes est donc retenu.

Sauvegarde des résultats

Les fichiers suivants sont générés automatiquement :
— `tsa_model_selection_results.csv` : performances de tous les modèles testés
— `tsa_best_parameters.txt` : configuration optimale du modèle
— `tsa_analysis_summary.txt` : résumé complet de l'analyse

3. SARIMAX_Model.py

Objectif

Le fichier `SARIMAX_Model.py` est consacré à l'ajustement du modèle SARIMAX pour la prédiction du niveau d'eau moyen `niveau_avg` en intégrant deux variables exogènes : `Débit_A` et `Débit_B`. L'objectif est de modéliser à la fois la dynamique temporelle interne du niveau d'eau et l'effet des débits en entrée et en sortie.

Fonctions principales :

— **Chargement et fusion des données temporelles** : les trois séries temporelles sont fusionnées avec gestion des fréquences et des valeurs manquantes.

— **Spécification du modèle SARIMAX** : le modèle est défini avec les ordres suivants :
  — ARIMA : `(1, 1, 2)`
  — Saisonnière : `(1, 0, 1, 24)`
— **Ajustement et sauvegarde du modèle** : entraînement du modèle complet sur l'ensemble des données disponibles, puis sérialisation avec `joblib`.

— **Évaluation des performances** : génération des prédictions sur l'ensemble historique, puis calcul des métriques (MAE, RMSE, R^2).

— **Analyse des résidus** : inspection des résidus (erreurs du modèle) à travers un tracé temporel, l'ACF des résidus et un test de Ljung-Box pour vérifier l'absence d'autocorrélation.

Visualisations générées et interprétations :

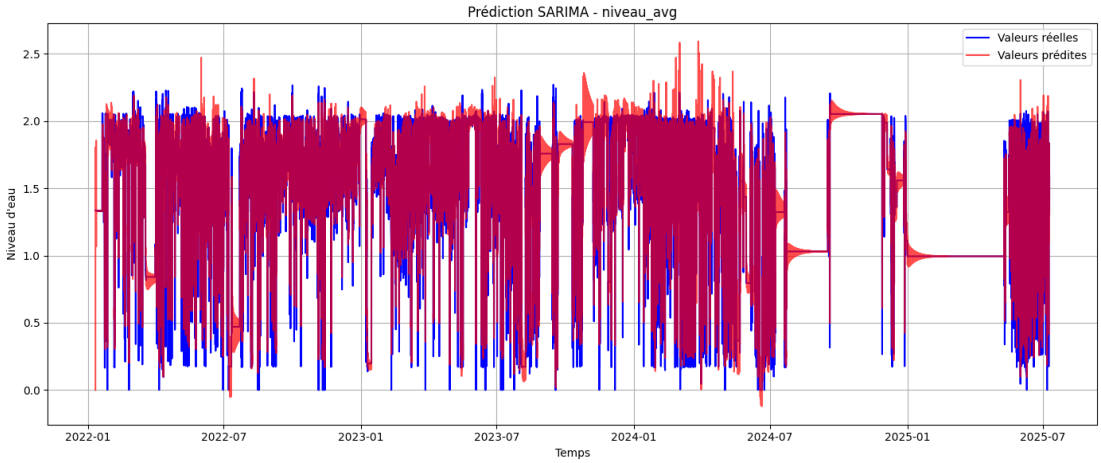

FIGURE 3 – Comparaison entre les valeurs réelles et les prédictions SARIMAX

— Le modèle capture globalement bien la dynamique du niveau d'eau, avec une tendance et des fluctuations proches de la réalité. On observe un bon alignement, notamment dans les zones sans pics extrêmes.

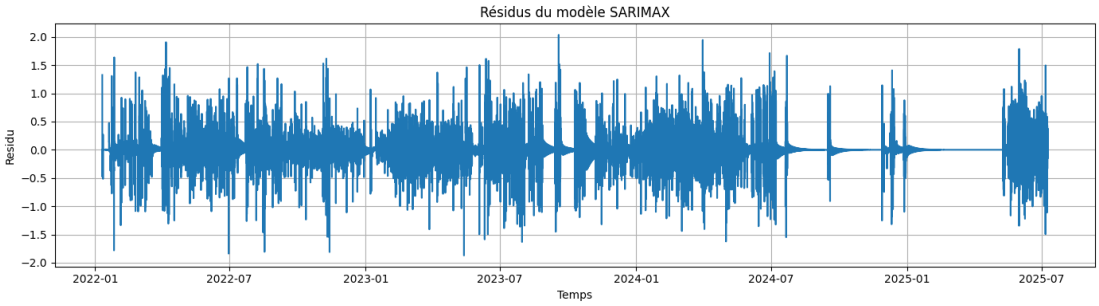

FIGURE 4 – Résidus du modèle SARIMAX

— Les résidus sont centrés autour de zéro, sans tendance marquée, ce qui indique une bonne qualité d'ajustement. L'absence de structure visible suggère que La dynamique globale de la série est correctement modélisée, malgré certaines fluctuations résiduelles..

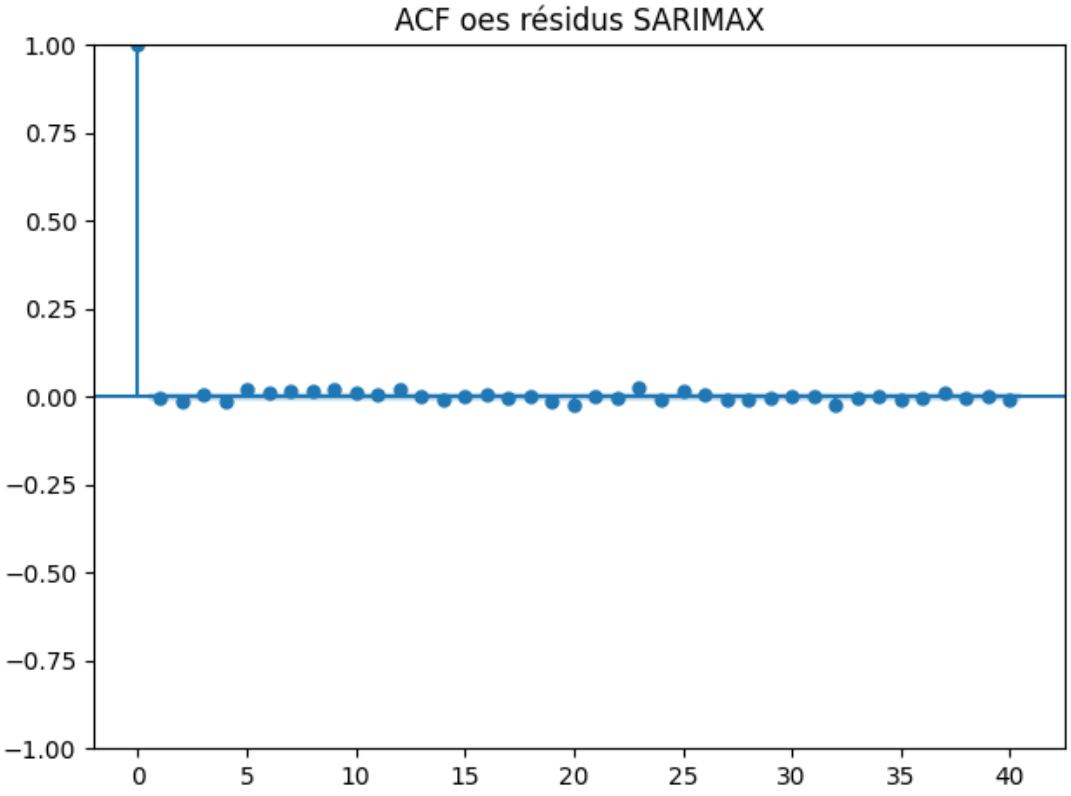

FIGURE 5 – ACF des résidus SARIMAX

— L'ACF des résidus ne montre pas de corrélations significatives au-delà du lag 0, ce qui indique que les erreurs sont assimilables à du bruit blanc. Le modèle a donc correctement capturé les dépendances temporelles restantes.

Métriques de performance du modèle :

— **MAE (Mean Absolute Error)** : 0.1143

— **RMSE (Root Mean Squared Error)** : 0.2178
— **R^2 (coefficient de détermination)** : 0.8279

Interprétation

— **MAE (Mean Absolute Error)** : 0.1143
Cela signifie qu'en moyenne, l'erreur absolue entre les valeurs réelles et les valeurs prédites est d'environ 0.1143. Cela indique une bonne précision des prévisions en termes d'erreurs absolues.

— **RMSE (Root Mean Squared Error)** : 0.2178
Cette valeur mesure la dispersion des erreurs. Un RMSE faible indique que les erreurs de prévision sont relativement faibles. Ici, la valeur de 0.2178 reste modérée, ce qui reflète une performance satisfaisante du modèle.

— **R^2 (coefficient de détermination)** : 0.8279
Le modèle explique environ 82.79% de la variance de la variable cible. Cela suggère que le modèle parvient à capturer la majorité de la dynamique de la série, ce qui confirme sa pertinence pour la prédiction.

Test de Ljung-Box sur les résidus :

TABLE 3 – Résultats du test de Ljung-Box

Lags	Statistique	p-valeur
10	60.28	3.21×10^{-9}
20	99.42	1.60×10^{-12}
30	137.76	9.46×10^{-16}

Interprétation : Les p-valeurs très faibles suggèrent une autocorrélation résiduelle statistiquement significative, bien que visuellement les résidus se comportent comme du bruit blanc. Ce paradoxe peut résulter de légères non-linéarités ou de valeurs extrêmes.

Dans un premier temps, une approche hybride combinant SARIMAX et LSTM a été envisagée, en appliquant un modèle LSTM sur les résidus du modèle SARIMAX afin de capturer d'éventuelles dynamiques non-linéaires non modélisées. Cependant, l'analyse a révélé que les résidus étaient peu structurés et difficilement prédictibles, ce qui limite l'intérêt d'un apprentissage supervisé sur ces derniers.

Décision pour la suite :

Étant donné que les résidus sont peu structurés et difficilement prédictibles, il a été décidé de ne pas utiliser de modèle LSTM sur ces résidus. À la place, un modèle LSTM indépendant a été développé comme solution de prédiction complémentaire, et non comme post-traitement du SARIMAX.

Remarque : Sauvegarde du modèle

Pour permettre une réutilisation ultérieure sans devoir réentraîner le modèle, le modèle SARIMAX final a été sauvegardé localement sous le nom `sarimax_model.pkl`, à l'aide de la bibliothèque `joblib`.

4. LSTM_Model.py

Objectif

Le fichier `LSTM_Model.py` est dédié à la mise en place d'un réseau de neurones récurrent LSTM (*Long Short-Term Memory*) pour la prédiction du niveau d'eau moyen `niveau_avg`. L'objectif est de capturer des relations non-linéaires et des dépendances à long terme que le modèle SARIMAX pourrait ne pas modéliser complètement.

Une approche consistant à appliquer un LSTM sur les résidus du SARIMAX a d'abord été envisagée afin d'exploiter d'éventuelles structures non-linéaires restantes. Cependant, l'analyse des résidus a montré un comportement proche d'un bruit blanc, rendant cette option peu pertinente. Ainsi, un modèle LSTM indépendant a été développé comme solution alternative à SARIMAX.

Fonctions principales :

- **Préparation et fusion des données** : Fusion des trois séries temporelles nettoyées (`View_Reservoir_clean`, `vw_TROZA_A_clean`, `vw_TROZA_B_clean`) sur l'index temporel, normalisation des variables continues avec `MinMaxScaler`, et conservation des variables binaires.

- **Création des séquences d'entrée** : Utilisation d'une fenêtre glissante de 24 heures pour constituer les entrées séquentielles nécessaires à l'entraînement du LSTM.

- **Définition de l'architecture du réseau** :
  - Une couche `LSTM` avec 64 unités et fonction d'activation `tanh`.
  - Une couche dense avec une seule unité en sortie.
- **Entraînement du modèle** : Optimiseur `Adam`, fonction de perte MSE, et arrêt anticipé (`EarlyStopping`) après 5 itérations sans amélioration sur la validation.

- **Évaluation des performances** : Calcul des métriques suivantes après réinversion de la normalisation :
  - **MAE** : 0.0802
  - **RMSE** : 0.1764
  - $\mathbf{R^2}$: 0.8431

  Interprétation :

  - **MAE (Mean Absolute Error) : 0.0802**
    En moyenne, les prédictions s'écartent de la valeur réelle d'environ 0.0802 uni-

tés de niveau d'eau, ce qui indique une précision globalement satisfaisante.

— **RMSE (Root Mean Squared Error) : 0.1764**
L'erreur quadratique moyenne confirme que les écarts importants sont rares et que les prévisions restent proches des valeurs observées.

— **R^2 (coefficient de détermination) : 0.8431**
Le modèle explique environ 84.31% de la variance observée, ce qui traduit une excellente capacité à reproduire la dynamique globale de la série.

Sauvegarde du modèle : Sérialisation du modèle entraîné dans le fichier `lstm_model.h5` pour une utilisation ultérieure.

Visualisations générées :

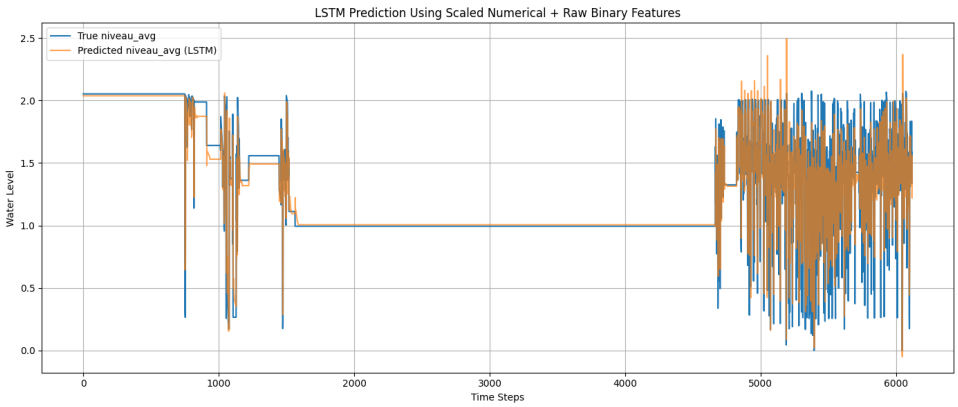

FIGURE 6 – Comparaison entre les prédictions LSTM et les valeurs réelles sur la période de test.

On observe que la partie correspondant à l'ensemble de test présente une apparence inhabituelle, car l'échantillon de 20% retenu se situe sur une période atypique pour le comportement du réservoir. Toutefois, cela ne remet pas en cause la validité de l'évaluation du modèle.

5. Intégration des modèles en temps réel avec données SCADA

Objectif

Cette section présente les étapes nécessaires pour déployer les modèles prédictifs (`SARIMAX` et `LSTM`) sur des données SCADA issues de `PCWin2`, stockées dans un serveur `SQL Server`.
À noter : l'accès aux données en temps réel n'étant pas disponible dans le cadre de ce projet, la mise en œuvre n'a pas pu être réalisée. Néanmoins, les étapes décrites

ci-dessous constituent un guide pour la future intégration. Les règles d'agrégation utilisées sont détaillées dans le rapport BI transmis et implémentées dans le fichier `data.py`.

Flux de données en temps réel

1. **Acquisition des données** : le logiciel `PCWin2` collecte les données brutes du système SCADA et les enregistre dans la base de données SQL Server.
2. **Requête des données** : extraction périodique des nouvelles mesures via `pyodbc` ou `SQLAlchemy`, en interrogeant les mêmes tables utilisées lors de l'entraînement.

Prétraitement et compatibilité des données

Pour garantir que les prédictions soient cohérentes avec le modèle entraîné, les étapes suivantes doivent être strictement reproduites :

— **Agrégation temporelle** : reproduire les mêmes opérations d'agrégation que celles documentées dans le rapport BI et codées dans `data.py`.
— **Renommage et structuration des colonnes** : respecter la même nomenclature de variables que dans les jeux de données nettoyés (ex. `niveau_avg`, `Débit_A`, `Débit_B`).
— **Gestion des valeurs manquantes** : appliquer la même méthode (suppression) que celle utilisée initialement.
— **Alignement temporel** : s'assurer que toutes les variables exogènes et endogènes sont synchronisées sur la même échelle de temps.

Chargement des modèles

— **SARIMAX** : le modèle est chargé depuis le fichier `sarimax_model.pkl` grâce à la bibliothèque `joblib`.
— **LSTM** : le modèle est chargé via `tensorflow.keras.models.load_model()` depuis le fichier `lstm_model.h5`.

Génération des prédictions

1. Passer les données prétraitées en entrée du modèle.
2. Produire la prévision pour l'horizon choisi (par ex. une heure ou un jour).
3. Comparer les résultats des deux modèles si nécessaire.

Utilisation opérationnelle

— Intégrer les prévisions dans une interface graphique (eg :Power BI) pour un affichage en temps réel.
— Déclencher des alertes automatiques lorsque le niveau prévu atteint un seuil critique (`niveau_très_bas` ou `niveau_très_haut`).
— Archiver les données et prévisions pour un suivi des performances.