Validation ontologique des explications contrefactuelles pour les séries temporelles : Application aux batteries lithium-ion

Slimane Arbaoui*, Ali Ayadi*, Ahmed Samet*, Tedjani Mesbahi*, Romuald Boné*

*Laboratoire ICube, INSA Strasbourg, Université de Strasbourg, Strasbourg, France
prénom.nom@insa-strasbourg.fr,

Résumé. Les modèles d'apprentissage profond sont essentiels dans des secteurs critiques tels que la finance et la santé, grâce à leur capacité à prédire des phénomènes complexes , notamment avec des séries temporelles. Cependant, leur nature opaque, souvent qualifiée de "boîte noire", soulève des préoccupations liées à leur explicabilité. Nous proposons deux méthodes pour générer des explications contrefactuelles adaptées à la prévision de séries temporelles multivariées. La première, GENO-TOPSIS, combine un algorithme génétique et TOPSIS, tandis que la seconde, NSGA-II, est plus rapide avec des résultats comparables. Pour valider ces explications, nous introduisons l'Ontologie de Validation des Explications Contrefactuelles (CEVO), qui utilise des règles SWRL et des requêtes SPARQL pour assurer du respect des contraintes spécifiques au domaine. Nous illustrons l'efficacité de notre approche avec une étude de cas sur l'estimation de l'État de Charge (EDC) des cellules LFP, démontrant ainsi une meilleure explicabilité des modèles de séries temporelles.

1 Introduction

Ces dernières années, les modèles d'apprentissage profond ont été adoptés dans des secteurs critiques comme la finance, la santé et les véhicules autonomes, en raison de leur capacité à identifier des motifs complexes et à fournir des prévisions précises, y compris sur des séries temporelles Lim et Zohren (2021). Cependant, ces modèles sont souvent critiqués pour leur nature opaque, qualifiée de "boîte noire", rendant difficile la compréhension de leurs décisions. Ce manque d'explicabilité pose des défis importants dans les applications à enjeux élevés, où les résultats peuvent avoir des conséquences critiques. Pour y remédier, des méthodes post-hoc comme LIME (Local Interpretable Model-agnostic Explanations) Ribeiro et al. (2016) et SHAP (SHapley Additive exPlanations) Lundberg et Lee (2017) ont été développées pour expliquer les décisions des modèles sans nécessiter une connaissance détaillée de leur architecture. Néanmoins, dans les séries temporelles multivariées, il est crucial d'identifier les attributs influents et de comprendre quelles parties spécifiques affectent les prédictions et comment de légères modifications pourraient impacter les résultats. Comme le décrivent Wachter et al. (2017) les explications contrefactuelles (CFs) comblent cette lacune en identifiant des changements minimaux dans les données d'entrée qui pourraient produire des sorties différentes. Malgré leur potentiel, peu d'études ont appliqué ces explications à la prévision de

séries temporelles, et il reste difficile de générer des CFs valides respectant à la fois leurs propriétés clés et les contraintes spécifiques au domaine. Dans cet article, nous proposons deux méthodes pour générer des explications CFs dans une tâche de prévision de séries temporelles : un algorithme génétique combiné à TOPSIS (Technique de Préférence par Similarité à la Solution Idéale) Chakraborty (2022) et l'algorithme NSGA-II (Non-Dominated Sorting Genetic Algorithm) Deb et al. (2002). Ces méthodes visent à garantir que les CFs respectent trois propriétés clés des explications CFs. Pour valider leur réalisme, nous introduisons une ontologie en Web Ontology Language (OWL), appelée CEVO, utilisant des règles SWRL et des requêtes SPARQL pour garantir la conformité des CFs aux contraintes spécifiques au domaine. Nous démontrons l'efficacité de notre approche dans le domaine des batteries en utilisant l'ontologie BatINFO [1] et un modèle boîte noire pour estimer l'état de charge (EDC) des cellules Lithium FerroPhosphate (LFP). Nos principales contributions sont :

— la proposition de deux méthodes pour générer des explications CFs adaptées aux séries temporelles multivariées pour une tâche de prévision.
— la définition de l'ontologie CEVO pour valider les CFs en fonction des contraintes spécifiques au domaine.

2 Préliminaires

2.1 Explications contrefactuelles

Les explications CFs sont devenues un outil clé dans l'intelligence artificielle explicable (XAI), car elles montrent comment des changements mineurs dans certains attributs peuvent altérer la prédiction d'un modèle Wachter et al. (2017). Wachter et al. (2017) ont proposé une approche couramment adoptée pour générer des CFs. Ils formulent le problème comme une tâche d'optimisation intégrant des contraintes spécifiques basées sur des principes psychologiques, pour assurer l'interprétabilité et la réalisabilité. Verma et al. (2020) et Schleich et al. (2021) ont enrichi ces contraintes, en identifiant les propriétés essentielles des CFs suivantes :

— **Proximité** : Le CF doit être aussi proche que possible du scénario d'origine, avec des modifications minimales des valeurs des attributs.
— **Sparité** : Seuls quelques attributs clés doivent être modifiés facilitant ainsi la compréhension humaine.
— **Diversité** : Plusieurs CFs divers permettent d'explorer différentes alternatives pour influencer le résultat.
— **Validité** : Le CF doit modifier la prédiction du modèle.
— **Faisabilité** : Les changements proposés doivent être réalistes et réalisables dans la pratique, évitant des suggestions impossibles, comme l'altération de l'âge.

Peu d'études ont exploré les CFs dans le contexte des séries temporelles Karlsson et al. (2018); Ates et al. (2021); Wang et al. (2021); Delaney et al. (2021). Ces données introduisent une complexité supplémentaire en raison des dépendances temporelles. Modifier un point de données peut affecter les prédictions futures, rendant la génération de CFs valides plus difficile. Des travaux récents Wang et al. (2023) ont exploré la génération de CFs pour la prévision de séries temporelles en traitant la cohérence temporelle, souvent en utilisant des algorithmes génétiques

1. https://github.com/BIG-MAP/BattINFO

et des techniques d'optimisation pour respecter la structure séquentielle des données. Il est crucial que les CFs respectent les contraintes spécifiques au domaine. Si elles ne le font pas, elles risquent d'être impraticables ou non pertinentes. Bien qu'une validation manuelle par un expert soit possible, elle reste coûteuse. Une approche plus efficace consiste à utiliser une ontologie qui formalise les concepts et relations clés du domaine. En s'appuyant sur ces ontologies, les CFs peuvent être automatiquement vérifiés par rapport aux règles du domaine, garantissant leur validité et faisabilité. Cette approche réduit la dépendance à l'expertise humaine et accélère la vérification grâce à des contrôles de cohérence automatisés.

2.2 Ontologies

Les ontologies sont des cadres structurés qui représentent la connaissance sous la forme d'un ensemble de concepts au sein d'un domaine et des relations entre ces concepts. Les ontologies fournissent un cadre formel permettant aux machines d'interpréter et de raisonner sur les données avec un contexte enrichi. En définissant des entités, leurs propriétés et leurs relations, les ontologies facilitent la compréhension et l'organisation des informations complexes. Une norme largement utilisée pour représenter des ontologies est l'OWL, qui permet la création d'ontologies riches et descriptives, interprétables par les humains et les machines. Les ontologies représentées en OWL sont particulièrement utiles dans des domaines comme la santé, la biologie et l'ingénierie, où la cohérence et l'exactitude des données doivent être garanties. Les ontologies OWL, permettent aux systèmes de vérifier automatiquement si certaines conditions ou contraintes, telles que celles des explications CFs, sont respectés. Ceci contribue à garantir que les résultats générés sont valides dans le contexte spécifique du domaine Bellucci et al. (2024).

3 Méthodologie proposée

Dans cette section, nous décrivons notre approche proposée (Figure 1) pour générer des explications CFs pour l'estimation de l'EDC des batteries. Notre méthodologie repose sur l'utilisation de deux techniques basées sur l'optimisation, combinées à un processus de vérification basé sur une ontologie, afin de garantir que les CFs générés répondent aux propriétés requises et aux contraintes spécifiques au domaine.

3.1 Génération des contrefactuels

Étant donné que la génération des CFs peut être formulée comme un problème d'optimisation, nous employons deux méthodes différentes pour créer des candidats CFs :

3.1.1 Méthode Geno-TOPSIS

Cette approche combine la puissance d'un algorithme génétique (AG) avec la méthode TOPSIS pour générer et sélectionner des CFs Katoch et al. (2021). Plutôt que de se fier à l'approche traditionnelle de l'AG, qui se concentre uniquement sur l'optimisation d'une fonction de *fitness*, nous intégrons TOPSIS pour rendre le processus de sélection plus robuste et plus compréhensible. Le principe de la méthode TOPSIS est de choisir les meilleures alternatives

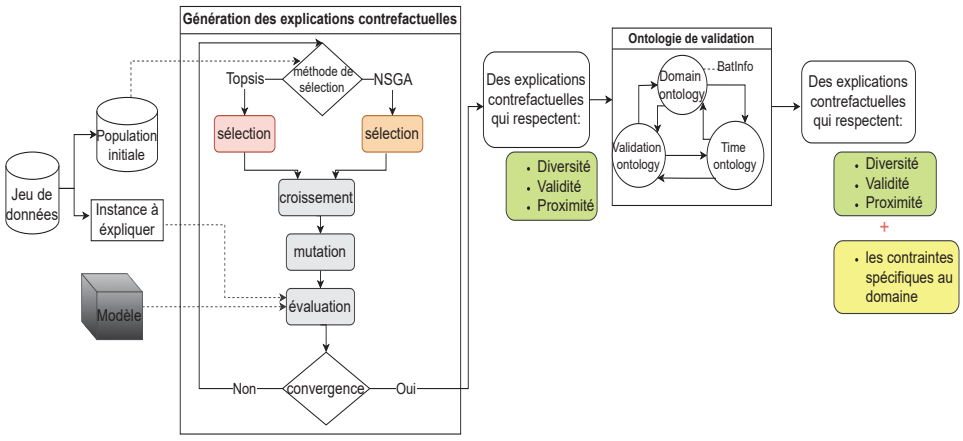

FIG. 1 – *Aperçu de l'approche proposée.*

en identifiant les solutions qui sont à la fois proches de la Solution Idéale Positive (PIS) et éloignées de la Solution Idéale Négative (NIS) Kim et al. (1997). Cela reflète le processus de prise de décision dans des scénarios réels, où les choix préférés sont souvent ceux qui se rapprochent le plus des résultats idéaux tout en évitant les options moins favorables. L'un des principaux avantages de cette approche est sa simplicité et sa nature intuitive, ce qui la rend facilement compréhensible par les décideurs Jun et al. (2017). Comme l'ont noté Behzadian et al. (2012), l'application de TOPSIS est devenue courante, en particulier dans le domaine de l'Aide à la Décision Multicritères (MCDA), en raison de son efficacité à équilibrer plusieurs critères conflictuels et à fournir des résultats exploitables. L'objectif de notre méthode est d'atteindre un équilibre entre trois propriétés clés des CFs : *diversité*, *validité* et *proximité*. En maximisant la diversité et la validité tout en minimisant la proximité, nous atteignons un compromis optimal, bien équilibré. La méthode TOPSIS garantit que les candidats CFs respectent ces critères. Avant de détailler comment TOPSIS permet d'atteindre ces objectifs, nous expliquons d'abord comment chaque propriété est calculée.

Préliminaires :

- x : le point d'intérêt original,
- x' : un point CF candidat,
- P : l'ensemble de la population contenant tous les candidats CFs,
- f : la fonction prédictive du modèle.

Objectif : Assurer les propriétés suivantes :

- **Diversité :** Les CFs doivent être significativement différents les uns des autres. Nous atteignons cela en maximisant la distance entre un CF x' et son plus proche voisin z dans la population P : $D(x') = \min_{\substack{z \in P \\ z \neq x'}} \|x' - z\|_2$, Nous avons opté pour la distance Euclidienne d'ordre 2, couramment utilisée pour comparer des séries temporelles Górecki et al. (2024).
- **Validité :** Un CF est considéré comme valide s'il modifie avec succès la prédiction du modèle. La métrique de validité mesure à quel point la prédiction pour x' est différente

de celle de x : $V(x') = \|f(x') - f(x)\|_2$. Une valeur plus élevée de $V(x')$ indique une plus grande probabilité que le CF ait modifié le résultat du modèle.

— **Proximité :** Pour garantir que le CF implique le moins de changements perturbateurs, nous visons à minimiser la distance entre x et x', rendant les modifications aussi petites que possible : $Pro(x') = \|x' - x\|_2$.

Cette méthode suit les étapes classiques d'un AG : *sélection*, *croisement*, *mutation* et *évaluation*, avec une probabilité de croisement fixée à 0.8 et de mutation à 0.2.

Sélection :

— **Étape 1 : Construire la matrice de décision**
Calculer les propriétés (*diversité*, *validité*, *proximité*) de chaque candidat $x'_i \in P$ et leur attribuer un identifiant, ID : $[ID, D(x'_i), V(x'_i), Pro(x'_i)]$.

— **Étape 2 : Normaliser la matrice de décision**
Normaliser chaque valeur de la matrice de décision pour garantir la comparabilité :

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{m} x_{ij}^2}}$$

— **Étape 3 : Déterminer la matrice de décision normalisée pondérée**
Appliquer des poids aux valeurs normalisées pour donner la priorité à certaines propriétés. Dans ce cas, la validité et la proximité sont données avec une priorité plus élevée par rapport à la diversité : $v_{ij} = w_j \cdot r_{ij} \quad \text{avec} \quad w = [0.1, 0.3, 0.6]$

— **Étape 4 : Identifier les solutions idéales et pires**
Identifier la solution idéale positive A^+ et la solution idéale négative A^- sur la base des propriétés :

$$A^+ = \begin{cases} \max(v_{ij}) & \text{si } j \text{ est un critère de bénéfice} \\ \min(v_{ij}) & \text{si } j \text{ est un critère de coût} \end{cases}$$

$$A^- = \begin{cases} \min(v_{ij}) & \text{si } j \text{ est un critère de bénéfice} \\ \max(v_{ij}) & \text{si } j \text{ est un critère de coût} \end{cases}$$

— **Étape 5 : Calculer la distance euclidienne aux solutions idéales et pires**

— **Étape 6 : Classer les alternatives**
classer les alternatives en fonction de leur proximité à la pire solution.

Croisement : Cette phase dépend du domaine, en particulier dans notre cas d'application aux batteries. Le modèle Hidouri et al. (2024) utilise trois caractéristiques en entrée : *le courant (I)*, la *tension (V)* et la *température (T)*, chacune d'une longueur 100, pour estimer 25 valeurs d'EDC. Ces caractéristiques sont corrélées, ce qui signifie que la modification d'une caractéristique implique des ajustements correspondants sur les autres pour garantir la cohérence. Dans des domaines avec des caractéristiques non corrélées, cette étape peut être modifiée pour permettre des changements sur une seule caractéristique à la fois. Soient x' et x'' deux points issus de P, avec trois caractéristiques, chacune de longueur 100, représentées comme suit : $x' = [x_1, x_2, x_3], \quad x'' = [x'_1, x'_2, x'_3]$. L'opération de croisement est effectuée de la manière suivante :

— Choisir un point de départ *starts* aléatoire dans l'intervalle $[0, 99]$.

— Choisir un point final *ends* dans l'intervalle $[\text{starts}, \min(\text{starts} + 20, 99)]$.
— Pour chaque caractéristique $i \in \{1, 2, 3\}$ et pour les indices j dans $[starts, ends]$, les segments correspondants de x' et x'' sont échangés. L'opération est définie comme suit : $x_i[j] \leftrightarrow x'_i[j] \quad$ pour $j \in [\text{starts}, \text{ends}] \quad$ et $\quad i \in \{1, 2, 3\}$

Mutation : Pendant la phase de mutation, des individus aléatoires de la population sont sélectionnés pour modification. La fonction choisit aléatoirement l'une des trois caractéristiques à muter. La valeur de chaque caractéristique est normalisée entre 0 et 1, grâce à la technique de normalisation *min-max* pour s'assurer que toutes les données se situent dans cette plage. Un nombre aléatoire détermine si la mutation va rapprocher la caractéristique de la limite inférieure $x_l = 0$ ou de la limite supérieure $x_u = 1$.

3.1.2 Méthode NSGA-II

Cette approche repose sur l'algorithme génétique NSGA-II Deb et al. (2002), qui est un algorithme d'optimisation multi-objectifs largement reconnu Ye et al. (2024). Contrairement à la méthode Geno-TOPSIS, NSGA-II génère des CFs par un processus d'optimisation dans un espace à plusieurs dimensions. Pour chaque population, NSGA-II divise les candidats en différents niveaux de domination, garantissant que la diversité est maintenue dans chaque génération. Par conséquent, cette méthode est moins affectée par des solutions convergentes. Le processus de sélection dans NSGA-II est guidé par le principe de dominance. Chaque solution dans la population est évaluée selon les objectifs correspondants aux critères de diversité, de validité et de proximité. Le choix des candidats est basé sur le principe suivant : une solution domine une autre si elle est meilleure dans au moins un aspect sans être pire dans les autres. Cela signifie que les candidats CFs générés par cette méthode ont une diversité inhérente et sont également plus susceptibles de respecter les propriétés souhaitées.

3.2 CEVO, ontologie de validation des explications contrefactuelles

L'ontologie proposée [2] met en évidence les concepts utilisés et aligne trois ontologies : BattINFO, Time [3] et la nouvelle Ontologie de Validation des Explications CFs (CEVO). La CEVO permet de valider des CFs générés pour un point d'intérêt basé sur des contraintes spécifiques au domaine. Elle prend en entrée un point d'intérêt, la prédiction du modèle pour ce point, et les CFs générés et leurs prédictions correspondantes. L'ontologie vérifie d'abord si ces CFs respectent les contraintes du domaine, les spécifications de la fiche technique d'une cellule LFP dans notre cas. Ces contraintes incluent :

1. *Tension nominale*, tension de coupure de charge et de décharge, garantissant l'utilisation du même type de batterie que celui utilisé dans le modèle de prédiction.
2. *Plage de température*, garantissant que la température se situe entre -30°C et 60°C.
3. *Plage de courant et de tension de charge*, garantissant que le courant se situe entre 1,5 A et 20 A, et que la tension est supérieure ou égale à 3,6 V.
4. *Plage de courant et de tension de décharge*, garantissant que le courant varie entre -30 A et 0 A, et que la tension est supérieure ou égale à 2 V.

2. https://github.com/arslimane/L-ontologie-de-validation-des-explications-\contrefactuelles.git
3. https://www.w3.org/TR/owl-time/

Voici quelques exemples de règles utilisées pour la validation de la :

Spécification de la tension nominale

```
BatteryCell(?bc) ^ hasNominalVoltage(?bc, 3.6) ^
    NominalVoltageSpecification(?nvs) ^
    followsBatteryCellSpecification(?bc, ?nvs)->
    NominalVoltageSpecification(?nvs) ^ isValid(?nvs, true)
BatteryCell(?bc) ^ hasNominalVoltage(?bc, ?value) ^ swrlb:notEqual(?
    value, 3.6) ^ NominalVoltageSpecification(?nvs) ^
    followsBatteryCellSpecification(?bc, ?nvs) ->
    NominalVoltageSpecification(?nvs) ^ isValid(?nvs, false)
```

Spécification de la plage de température

```
TimeSeriesData(?data) ^ followsDataSpecification(?data, ?trsp) ^
    TemperatureRangeSpecification(?trsp) ^ hasCollection(?data, ?
    collection) ^TimeSeriesCollection(?collection) ^ hasTemperature(?
    collection, ?temperature) ^ swrlb:lessThanOrEqual(?temperature,
    60) ^ swrlb:greaterThanOrEqual(?temperature,-30) ->
    TemperatureRangeSpecification(?trsp) ^ isValid(?trsp, true)
TimeSeriesData(?data) ^ followsDataSpecification(?data, ?trsp) ^
    TemperatureRangeSpecification(?trsp) ^ hasCollection(?data, ?
    collection) ^ TimeSeriesCollection(?collection) ^ hasTemperature
    (?collection, ?temperature) ^swrlb:lessThan(?temperature, -30) ->
     TemperatureRangeSpecification(?trsp) ^isValid(?trsp, false)
TimeSeriesData(?data) ^ followsDataSpecification(?data, ?trsp) ^
    TemperatureRangeSpecification(?trsp) ^ hasCollection(?data, ?
    collection)^TimeSeriesCollection(?collection) ^ hasTemperature(?
    collection, ?temperature) ^swrlb:greaterThan(?temperature, 60) ->
     TemperatureRangeSpecification(?trsp) ^isValid(?trsp, false)
```

Spécification de la plage de courant de charge

```
TimeSeriesData(?data) ^ followsDataSpecification(?data, ?trsp) ^
    TemperatureRangeSpecification(?trsp) ^ hasCollection(?data, ?
    collection) ^TimeSeriesCollection(?collection) ^ hasTemperature(?
    collection, ?temperature) ^ swrlb:lessThanOrEqual(?temperature,
    60) ^ swrlb:greaterThanOrEqual(?temperature,-30) ->
    TemperatureRangeSpecification(?trsp) ^ isValid(?trsp, true)
TimeSeriesData(?data) ^ followsDataSpecification(?data, ?trsp) ^
    TemperatureRangeSpecification(?trsp) ^ hasCollection(?data, ?
    collection) ^ TimeSeriesCollection(?collection) ^ hasTemperature
    (?collection, ?temperature) ^swrlb:lessThan(?temperature, -30) ->
     TemperatureRangeSpecification(?trsp) ^isValid(?trsp, false)
TimeSeriesData(?data) ^ followsDataSpecification(?data, ?trsp) ^
    TemperatureRangeSpecification(?trsp) ^ hasCollection(?data, ?
    collection) ^TimeSeriesCollection(?collection) ^ hasTemperature(?
    collection, ?temperature) ^ swrlb:greaterThan(?temperature, 60)
    -> TemperatureRangeSpecification(?trsp) ^isValid(?trsp, false)
```

Pour les propriétés CFs, nous proposons d'utiliser des requêtes SPARQL[4] pour vérifier les CFs générés par rapport aux contraintes prédéfinies. Ci-dessous, quelques exemples de requêtes qui démontrent la validation des propriétés de validité et proximité des CFs :

Validité des CFs

```
SELECT ?originalEDC ?counterfactualEDC ?originalTime
(IF(?counterfactualEDC != ?originalEDC, "Valid", "Invalid") AS ?
    differentValues)
WHERE {
?originalEDCData a cevo:OriginalEDCData ;
cevo:hasCollection ?originalCollection .
?originalCollection a cevo:TimeSeriesCollection ;
cevo:hasEDC ?originalEDC ;
time:hasTimeStamp ?originalTime .
?counterfactualEDCData a cevo:CounterfactualEDCData ;
cevo:hasCollection ?counterfactualCollection.
?counterfactualCollection a cevo:TimeSeriesCollection ;
cevo:hasEDC ?counterfactualEDC ;
time:hasTimeStamp ?originalTime .}
```

Proximité des CFs

```
SELECT (AVG(?difference) AS ?averageDifference)
WHERE {
?originalData a cevo:OriginalData .
?originalData cevo:hasCollection ?originalCollection .
?originalCollection a cevo:TimeSeriesCollection ;
cevo:hasTemperature ?originalTemperature ;
time:hasTimeStamp ?originalTime .
?counterfactualData a cevo:CounterfactualData .
?counterfactualData cevo:hasCollection ?counterfactualCollection.
?counterfactualCollection a cevo:TimeSeriesCollection ;
cevo:hasTemperature ?counterfactualTemperature ;
time:hasTimeStamp ?originalTime.
BIND(ABS(?counterfactualTemperature - ?originalTemperature) AS
?difference)}
```

4 Évaluation expérimentale

Pour évaluer les méthodes d'explication proposées, nous avons effectué des tests à cinq points clés durant le fonctionnement de la batterie : *début de la charge*, *à mi-charge*, *près de 100% d'EDC*, *et lors de la décharge à courant non constant*. Après avoir généré des CFs et les avoir validés à l'aide de l'ontologie CEVO, nous en avons sélectionné un pour la comparaison et l'analyse afin de comprendre les modifications nécessaires pour ajuster la prédiction du modèle. Le tableau 1 compare les méthodes GENO-TOPSIS et NSGA-II en fonction du

4. https://www.w3.org/TR/sparql11-query/

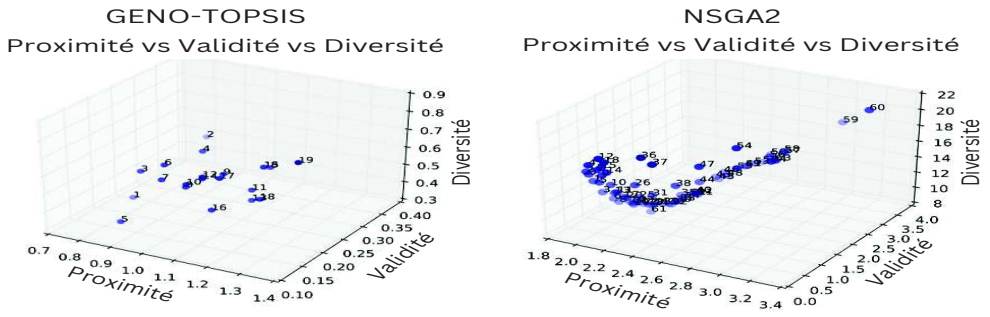

FIG. 2 – *La distribution des CFs à travers toutes les propriétés pour les deux méthodes.*

temps d'exécution et du nombre de CFs générés à partir d'une population initiale de 100 individus. La figure 2 montre la distribution des CFs générés pour le premier point d'intérêt, tandis que la figure 3 illustre les changements apportés par chaque méthode pour ajuster la prédiction. Les attributs modifiés et les moments précis de ces modifications sont observables. Après avoir comparé les résultats des deux méthodes, NSGA-II s'est révélé significativement plus rapide que GENO-TOPSIS, générant plus de CFs en moins de temps. Cependant, bien que les explications fournies par NSGA-II soient plus larges, elle sont moins raffinées que à celles de GENO-TOPSIS. Toutefois, elles capturent toujours des comportements essentiels et aboutissent souvent à des explications similaires.

Point d'intérêt	**Méthode**	**Temps (secondes)**	**CFs générés**
Début de la Charge	GENO-TOPSIS	5176	20
	NSGA-II	181	plus de 20
Milieu de la Charge	GENO-TOPSIS	1116	20
	NSGA-II	77	plus de 20
Près de 100% de EDC	GENO-TOPSIS	1085	20
	NSGA-II	179	plus de 20
Décharge (Courant Constant)	GENO-TOPSIS	1203	20
	NSGA-II	81	plus de 20
Décharge Instable	GENO-TOPSIS	3227	20
	NSGA-II	312	plus de 20

TAB. 1 – *Comparaison de la génération de CFs pour différents points d'intérêt en utilisant GENO-TOPSIS et NSGA-II.*

Observations clés :

1. **Absence d'impact du courant pendant les phases à courant constant :** Les méthodes n'ont pas ajusté les valeurs de courant, que ce soit en charge ou en décharge à courant constant, pour améliorer leurs prédictions. Ceci démontre que la valeur du courant n'est pas un facteur déterminant pour l'estimation de l'EDC durant ces phases.

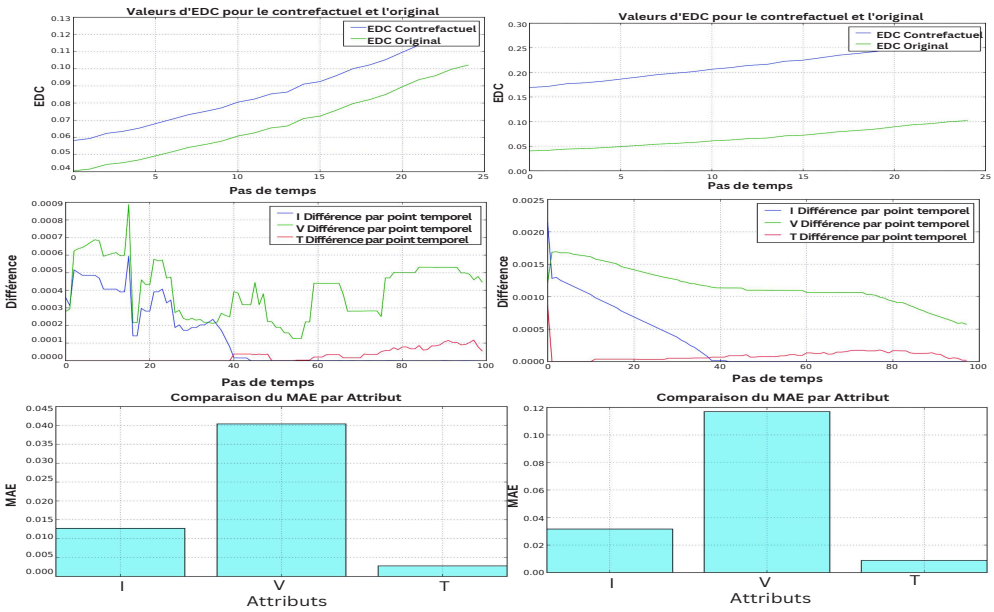

FIG. 3 – *Comparaison entre les changements effectués par la méthode GENO-TOPSIS et la méthode NSGA-II pour le premier point d'intérêt.*

2. **Importance de la température et de la tension :** Les méthodes ont donné la priorité à la *température* et à la *tension* pendant la charge et la décharge à courant constant, en se concentrant particulièrement sur la *température* lorsque l'EDC approchait 100%.
3. **Rôle du courant dans les phases à courant non constant :** Les méthodes ont accordé plus de poids au *courant* durant la décharge à courant non constant et dans les premières phases de la charge.

Ces observations sont conformes au comportement du système physique. Pendant la charge à courant constant, la *tension* et la *température* changent de manière significative, ce qui a été reconnu par les nôtres. La relation entre la tension à circuit ouvert (OCV) et l'EDC est évidente, car une augmentation de la tension est corrélée à une augmentation de l'EDC, comme le reflètent les explications CFs. Par exemple, une augmentation de la tension a entraîné des prévisions d'EDC plus élevées pendant la charge à courant constant.

5 Conclusion

Dans cet article, nous avons proposé deux méthodes pour générer des explications CFs adaptées aux séries temporelles. La première méthode, GENO-TOPSIS, combine un algorithme génétique avec la méthode TOPSIS, offrant un contrôle robuste sur la manière de sélectionner les candidats. La deuxième méthode, NSGA-II, fournit également des résultats comparables à ceux de GENO-TOPSIS, mais avec un temps d'exécution plus court. Les CFs générés ont été validés à l'aide de l'ontologie CEVO, garantissant leur conformité aux contraintes du

domaine et aux propriétés des CFs. Ce processus de validation confirme que les explications CFs générées sont à la fois réalistes et réalisables. Notre approche a été appliquée pour estimer l'EDC des cellules LFP, avec des résultats qui correspondent au comportement physique de la batterie. Les travaux futurs se concentreront sur l'amélioration de l'ontologie de notre approche et sur des tests dans divers scénarios d'application pour valider sa généralisabilité.

6 Remerciements

Cette recherche a bénéficié d'un financement partiel de l'Agence Nationale de la Recherche (ANR) française dans le cadre du projet "ANR-22-CE92-0007-02". De plus, un soutien a été apporté par l'Union Européenne à travers le programme Horizon Europe et le programme d'innovation sous la référence "GAP-101103667".

Références

Ates, E., B. Aksar, V. J. Leung, et A. K. Coskun (2021). Counterfactual explanations for multivariate time series. In *2021 International Conference on Applied Artificial Intelligence (ICAPAI)*, pp. 1–8.

Behzadian, M., S. Khanmohammadi Otaghsara, M. Yazdani, et J. Ignatius (2012). A state-of the-art survey of topsis applications. *Expert Systems with Applications 39*(17), 13051–13069.

Bellucci, M., N. Delestre, N. Malandain, et C. Zanni-Merk (2024). Towards counterfactual explanations for ontologies. *Semantic Web 15*(5), 1611 – 1636.

Chakraborty, S. (2022). Topsis and modified topsis : A comparative analysis. *Decision Analytics Journal 2*, 100021.

Deb, K., A. Pratap, S. Agarwal, et T. Meyarivan (2002). A fast and elitist multiobjective genetic algorithm : Nsga-ii. *IEEE Transactions on Evolutionary Computation 6*(2), 182–197.

Delaney, E., D. Greene, et M. T. Keane (2021). Instance-based counterfactual explanations for time series classification. In *International conference on case-based reasoning*, pp. 32–47.

Górecki, T., M. Łuczak, et P. Piasecki (2024). An exhaustive comparison of distance measures in the classification of time series with 1nn method. *Journal of Computational Science 76*, 102235.

Hidouri, A., S. Arbaoui, A. Samet, A. Ayadi, T. Mesbahi, R. Boné, et F. de Bertrand de Beuvron (2024). SOCXAI : leveraging CNN and SHAP analysis for battery SOC estimation and anomaly detection. In *Computational Science - ICCS 2024 - 24th International Conference, Malaga, Spain, July 2-4, 2024, Proceedings, Part VII*, pp. 177–191. Springer.

Jun, W., T. Lingyu, L. Yuyan, et G. Peng (2017). A weighted emd-based prediction model based on topsis and feed forward neural network for noised time series. *Knowledge-Based Systems 132*, 167–178.

Karlsson, I., J. Rebane, P. Papapetrou, et A. Gionis (2018). Explainable time series tweaking via irreversible and reversible temporal transformations. In *2018 IEEE International Conference on Data Mining (ICDM)*, pp. 207–216.

Katoch, S., S. S. Chauhan, et V. Kumar (2021). A review on genetic algorithm : past, present, and future. *Multimedia Tools and Application 80*, 8091–8126.

Kim, G., C. S. Park, et K. Yoon (1997). Identifying investment opportunities for advanced manufacturing systems with comparative-integrated performance measurement. *International Journal of Production Economics 50*(1), 23–33.

Lim, B. et S. Zohren (2021). Time-series forecasting with deep learning : a survey. *Philosophical Transactions of the Royal Society A : Mathematical, Physical and Engineering Sciences 379*, 2194.

Lundberg, S. et S.-I. Lee (2017). A unified approach to interpreting model predictions. *arXiv preprint arXiv :1705.07874*.

Ribeiro, M., S. Singh, et C. Guestrin (2016). "why should I trust you ?" : Explaining the predictions of any classifier. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics : Demonstrations*, pp. 97–101.

Schleich, M., Z. Geng, Y. Zhang, et D. Suciu (2021). Geco : Quality counterfactual explanations in real time. *arXiv preprint arXiv :2101.01292*.

Verma, S., V. Boonsanong, M. Hoang, K. Hines, J. Dickerson, et C. Shah (2020). Counterfactual explanations and algorithmic recourses for machine learning : A review. *ACM Computing Surveys*.

Wachter, S., B. Mittelstadt, et C. Russell (2017). Counterfactual explanations without opening the black box : Automated decisions and the gdpr. *Harvard Journal of Law & Technology 31*, 841.

Wang, Z., I. Miliou, I. Samsten, et P. Papapetrou (2023). Counterfactual explanations for time series forecasting. In *2023 IEEE International Conference on Data Mining (ICDM)*, pp. 1391–1396.

Wang, Z., I. Samsten, R. Mochaourab, et P. Papapetrou (2021). Learning time series counterfactuals via latent space representations. pp. 369–384.

Ye, L., C. Li, C. Wang, J. Zheng, K. Zhong, et T. Wu (2024). A multi-objective optimization approach for battery thermal management system based on the combination of bp neural network prediction and nsga-ii algorithm. *Journal of Energy Storage 99*, 113212.

Summary

Deep learning models are widely used in sectors like finance and healthcare for forecasting complex time series patterns. However, their "black box" nature raises explainability concerns. To address this, we propose two methods for generating counterfactual explanations: GENO-TOPSIS, combining a genetic algorithm with the TOPSIS method, and NSGA-II, offering faster execution with similar results. We validate these counterfactuals using the CEVO ontology, applying SWRL rules and SPARQL queries to meet domain-specific constraints. Our study focuses on estimating the State of Charge (SOC) for Lithium Iron Phosphate (LFP) cells, demonstrating the effectiveness of our approach in real-world applications.