

Approche interactive d'extraction de variables interprétables et explicatives pour la gestion des contraintes du réseau électrique français

Laure Crochepierre^{*,**}, Lydia Boudjeloud-Assala^{*}, Vincent Barbesant^{**}

^{*}Université de Lorraine, CNRS, LORIA, F-57000 Metz, France
{laure.crochepierre, lydia.boudjeloud-assala}@univ-lorraine.fr

^{**}Réseau de Transport d'Electricité (Rte) R&D, Paris, France
{laure.crochepierre, vincent.barbesant}@rte-france.com

Résumé. Les réseaux électriques sont des systèmes hautement surveillés, où les opérateurs synthétisent des informations complexes pour comprendre l'état du réseau. Nos travaux visent à faciliter cette synthèse en créant automatiquement des variables à partir de mesures de capteurs. Nous proposons une approche d'extraction de variables par évolution guidée par la grammaire, qui produit des variables interprétables et physiquement cohérentes. Nous introduisons des restrictions d'opérations sur les grandeurs physiques du problème grâce à une grammaire non contextuelle, construite interactivement avec des experts, qui assure la cohérence avec les lois de la physique, la cohérence des unités, et introduit de l'expertise technique dans l'apprentissage. Nous comparons notre approche à des méthodes d'extraction de variables de l'état de l'art, sur des données réelles du réseau électrique français et proposons également une évaluation de l'interprétabilité de la méthode d'un point de vue humain et fonctionnel.

1 Introduction

Le réseau de transport d'électricité est un système complexe surveillé et exploité en temps réel, 24h/24, 7j/7, par des opérateurs hautement qualifiés. Ils doivent s'assurer en permanence que l'ensemble du système, essentiel dans nos sociétés modernes, reste dans un état sûr pour acheminer l'électricité des centrales de productions vers les consommateurs. En particulier, ils veillent à ce que le flux électrique de chaque ligne reste en dessous de sa limite d'exploitation au-dessus duquel un court-circuit pourrait se produire, et mettrait en danger la sécurité des biens et des personnes à proximité. Cette surveillance est assurée par un grand nombre de capteurs placés sur l'ensemble du réseau et fournit des mesures relayées dans des salles de contrôle. A partir de ces mesures, les opérateurs font une synthèse d'informations pour planifier des actions préventives afin de réorienter les flux avant qu'ils n'atteignent leur seuil critique.

Cependant, même si les opérateurs sont à même de faire fonctionner le système électrique, les gestionnaires de réseaux de transport (GRT) comme le GRT français "Réseau de Transport d'Electricité" (Rte) ont constaté une augmentation de la complexité des opérations en temps réel (Donnot et al., 2017). En conséquence, les lignes électriques sont exploitées plus près de

leur limite, et les opérateurs doivent prendre des décisions plus rapidement afin de garder du temps pour gérer des situations critiques. Aujourd'hui, la synthèse d'information est assistée par des simulations massives et un suivi d'indicateurs. Historiquement créés par les opérateurs grâce à leur expertise, ces indicateurs ne sont pas exhaustifs et ne peuvent pas certifier la sécurité de tous flux électriques. Par conséquent, les opérateurs continuent à effectuer eux-mêmes certaines parties de cette synthèse en utilisant leur connaissance et les simulations.

Dans ce contexte, nous proposons une méthode *interactive d'extraction de variables* pour l'explication de flux électriques permettant d'insérer des *contraintes expertes et physiques* (loi d'Ohm par exemple) afin d'obtenir des solutions dont les unités physiques sont cohérentes. L'approche repose sur la programmation génétique guidée par la grammaire (McKay et al., 2010), abrégée en PGGG, pour extraire des combinaisons interprétables. Notre contribution est double. Tout d'abord, nous proposons une méthode de création de variables qui intègre des connaissances via une grammaire non contextuelle construite interactivement. In fine, les variables créées sont analysées et commentées par un expert humain qui fournit des retours utilisables lors des prochains apprentissages. La deuxième contribution est l'*évaluation de l'interprétabilité*. Suivant la terminologie utilisée par Doshi-Velez et Kim dans (Doshi-Velez et Kim, 2017), nous évaluons "fonctionnellement" et "humainement" l'approche proposée en la comparant à d'autres méthodes interprétables et en faisant analyser les résultats par un expert.

Nous utilisons la corrélation linéaire comme fonction fitness. Dans cet article, nous évaluons la méthode sur un jeu de données constitué de mesures du réseau électrique français, au pas de 5 minutes de janvier 2014 à décembre 2018. L'article est organisé comme suit. La Section 2 résume l'état de l'art, la Section 3 décrit les données, la Section 4 détaille la méthode, les expériences et résultats sont présentés en Section 5, et enfin, la Section 6 conclut ce document et propose des perspectives. Cet article est un résumé en français d'un article présenté dans le cadre de la conférence internationale ECML/PKDD 2020 (Crochepierre et al., 2020).

2 Etat de l'art

Les données réelles sont généralement mesurées en grande dimension, redondantes et ont, par conséquent, une structure sous-jacente représentables en plus petite dimension. La famille de méthode utilisée pour appréhender ce type de problème est la réduction de dimension (RD). La RD se fait par sélection ou extraction de variables (Khalid et al., 2014; Van Der Maaten et al., 2009). Là où la sélection ne conserve que les variables les plus informatives, l'extraction *combine* (linéairement ou non) les variables initiales pour en produire de plus expressives. L'extraction linéaire est considérée comme plus interprétable, alors que celle non linéaire représente souvent mieux les données d'origine. La RD est le plus fréquemment réalisée dans un paradigme non-supervisé. C'est une limitation dans notre cas, car il n'est alors pas possible de prendre en compte la connaissance métier dans l'apprentissage.

Le manque d'interprétabilité pose problème en RD. Bien que les méthodes linéaires soient considérées comme interprétables (même en additionnant des grandeurs d'unités physiques différentes), ce n'est souvent pas le cas pour les méthodes non-linéaires, rendant difficile leur utilisation dans des processus industriels. L'interprétabilité améliore aussi la confiance dans la solution et l'acceptation du modèle par l'utilisateur. L'interprétabilité se fait soit par des modèles interprétables soit post-hoc. Les modèles interprétables, tels que les arbres de décision (Breiman et al., 1984), produisent des résultats interprétables mais souvent sous-optimaux.

L'interprétabilité post-hoc, au contraire s'utilise après l'apprentissage pour comprendre le comportement de modèles plus complexes et plus précis tels que les réseaux de neurones. Cependant, Laugel et al. (Laugel et al., 2019) ont identifié le risque d'avoir des explications (post-hoc) qui seraient des artefacts appris par le modèle plutôt que de la connaissance réelle sur les données. L'interprétabilité est plus que jamais nécessaire, que ce soit pour la sécurité, l'équité ou la compréhension scientifique. Cependant, il n'existe pas encore de consensus méthodologique pour la définir et l'évaluer. Doshi-Velez et Kim (Doshi-Velez et Kim, 2017) proposent néanmoins une première méthode d'évaluation à trois niveaux : le premier est une évaluation "appliquée" où l'humain évalue l'interprétabilité sur une tâche réelle ; le second niveau "humain" où l'humain évalue une tâche plus simple ; enfin, le niveau "fonctionnel" où une tâche simple est évaluée sans aucun humain. Ce niveau le plus simple évalue, par exemple, plusieurs algorithmes interprétables afin d'identifier le meilleur vis-à-vis d'une métrique. Nous utiliserons cette taxonomie dans la Section 5.

Parmi les méthodes d'extraction non-linéaires, la programmation génétique (PG) (Koza, 1990) permet d'obtenir des résultats interprétables. Elle fait évoluer une population de programmes représentés par des *arbres*. Chaque arbre est composé de variables initiales (feuilles) combinées au moyen d'opérations (noeuds) prises dans une liste de fonctions licites. Comme dans certains cas l'espace de recherche est grand et empêche la convergence, des variantes ont été proposées telle que la programmation génétique guidée par la grammaire (PGGG) (McKay et al., 2010). En PGGG, l'espace de recherche est limité par des *règles* grammaticales comme celles de la Figure 1. La PGGG permet inclure de la connaissance experte dans l'apprentissage via par exemple des contraintes sur les *unités* physiques des variables (Cherrier et al., 2019). C'est ce qui nous a conduit à envisager cette méthode pour trouver des variables explicatives.

3 Description des données

Le réseau électrique peut être représenté par un graphe $G = (N, L)$ à N noeuds et L lignes. Dans les noeuds sont physiquement connectées les lignes qui transportent l'électricité. On mesure dans les noeuds $n \in N$ les puissances active p_n et réactive q_n . L'information contenue dans les lignes l correspond aux liaisons électriques ($connected_l^{or}$, $connected_l^{ex}$ la connexion à l'origine et extrémité de la ligne, $neighbor_id_l$ la liste des lignes voisines). Avec ces mesures en entrées, des simulations estiment pour chaque pas de temps la tension v_n et l'angle θ_n dans les noeuds $n \in N$ puis le flux aux extrémités des lignes (d'origine *or* et d'extrémité *ex*), i_l^{or} et i_l^{ex} , $\forall l \in L$. Les flux $i_l^{or,ex}$ constituent la sortie de notre problème. Le système étudié forme alors un *système fermé* sans dépendance temporelle car les variables cibles $i_l^{or,ex}$ ne dépendent que des entrées et des hyperparamètres du simulateur. Nous déplaçons les variables des noeuds vers chaque origine/extrémité de ligne connectée à ce noeud, ce qui nous permet maintenant de raisonner uniquement en termes de lignes électriques. Nous nous référerons aux variables mesurées/simulées par $X = (X_l)_{l \in L}$ et aux variables cibles $y = (y_l)_{l \in L}$ où :

$$\forall l \in L, \quad X_l = ((p, q, v, \theta)_{n_{or}(l), n_{ex}(l)}, connected_l^{or,ex}, neighbor_id_l) \\ y_l = (i_l^{or}, i_l^{ex})$$

Le réseau électrique français est composé de 6 500 noeuds et 12 000 lignes et a de nombreuses interactions avec les réseaux voisins et en son sein. Pour contrôler ces interactions dans l'analyse du réseau, on le divise généralement en sous-zones peu dépendantes. Nous nous

Approche interactive d'extraction de variables interprétables et explicatives

```

# 1) Grandeurs physiques autorisées en sortie
<expr> ::= <p> | <s> | <i> | <p>/<v>

# 2) Operations légales sur chaque dimension
<p> ::= <p>-<p> | <pop>(<p>, <p>) | <sop>(<p>) | <p_var>
<q> ::= <q>-<q> | <pop>(<q>, <q>) | <sop>(<q>) | <q_var>
<v> ::= <v>-<v> | <pop>(<v>, <v>) | <sop>(<v>) | <v_var>
<p2> ::= <p>*<p> | square(<p>)
<q2> ::= <q>*<q> | square(<q>)
<s> ::= sqrt(<p2> + <q2>) | <v> * <i>
<i> ::= <s>/<v> | <pop>(<i>, <i>) | <sop>(<i>) | <i_frontier_var>

# 3) Operations retournant la même dimension que les entrées
<pop> ::= sum | minimum | maximum # Fonctions à deux arguments
<sop> ::= abs | neg | pos # Fonctions à un seul argument

```

FIG. 1 – Exemple de grammaire simplifiée. “|” sépare chaque règle pouvant remplacer le symbole placé avant “ ::= ”. Les entrées sont $\langle p_var \rangle$, $\langle q_var \rangle$, $\langle v_var \rangle$ et $\langle i_frontiere_var \rangle$.

concentrons ainsi sur une vallée montagneuse de 69 noeuds et 92 lignes où l’escarpement limite intrinsèquement les interactions avec le reste du réseau. 9 lignes sont reliées à l’extérieur. En collectant des mesures de janvier 2014 à décembre 2018, nous obtenons 365 165 observations.

4 Méthode proposée

Etant donné les variables $X \in \mathbb{R}^D$, nous souhaitons extraire par ligne $l \in L$ des combinaisons $X'_l \in \mathbb{R}^d$ où $d \ll D$ et tel que X'_l soit pertinent pour expliquer y_l . Nous utilisons la PGGG avec une grammaire gérant des données électriques et une métrique de corrélation.

4.1 Description de la grammaire

La construction de la grammaire est cruciale pour une recherche efficace dans l’espace : une grammaire trop lâche chercherait dans un espace trop grand, tandis qu’une grammaire trop contraignante limiterait la recherche à des zones peu pertinentes. Pour des raisons de compréhension, nous fournissons une grammaire simplifiée dans la Figure 1. La grammaire complète utilise toutes les variables décrites dans la section 3. La grammaire est construite de manière itérative avec des experts qui retirent ou ajoutent des règles, variables ou opérations.

Nous définissons d’abord les grandeurs physiques du problème avec leurs unités : la puissance active p (unité : watt W), la puissance réactive q (volt-ampère réactif VAR), la puissance apparente s (volt-ampère VA), la tension v (tension V) et l’intensité i (ampère A) puis le carré de ces grandeurs (ex. p^2 le carré de p d’unité W^2). La grammaire définit en premier la structure de sortie $\langle expr \rangle$ pour la contraindre à des grandeurs physiques utiles pour l’expert. $\langle expr \rangle$ peut être une des grandeurs suivantes : q , p^2 , q^2 ou v^2 . Puis, elle décrit les opérations réalisables sur chaque grandeur et entre deux grandeurs différentes. Pour passer d’une unité à l’autre, des lois de la physique telles que le triangle de puissance $s = \sqrt{p^2 + q^2}$ ou une variante de la loi d’Ohm $i = \frac{s}{v}$ sont traduites en grammaire. Les variables d’entrée sont $\langle p_var \rangle$, $\langle q_var \rangle$,

Algorithme 1 : Algorithme de recherche évolutionnaire interactive

```

input : observations  $X$ , target  $y$ 
grammaire  $\leftarrow$  initialisation_de_la_grammaire()
while operateur_non_satisfait do
  population  $\leftarrow$  creation_de_la_population(grammaire)
  condition_d'arret_non_satisfait  $\leftarrow$  True
  while condition_d'arret_non_satisfait do
    parents, meilleur_individu  $\leftarrow$  selection_des_parents(population,  $X$ ,  $y$ )
    descendance  $\leftarrow$  croisement(parents)
    descendance  $\leftarrow$  mutation(descendance)
    population  $\leftarrow$  remplacement(population, descendance)
    condition_d'arret_non_satisfait  $\leftarrow$  test_des_conditions(meilleur_individu)
  grammaire  $\leftarrow$  mise_à_jour_de_la_grammaire(grammaire)
return grammaire, meilleur_individu

```

$\langle v_var \rangle$, et $\langle i_frontiere_var \rangle$ ($i_frontiere$ est le flux des lignes transfrontalières modélisant les interactions avec l'extérieur de la zone) de grandeur physique respective p , q , v , i . Les combinaisons de ces variables peuvent produire une nouvelle variable d'unité identique ($\langle p \rangle - \langle p \rangle$ d'unité p), ou différente ($\langle p \rangle / \langle p \rangle$ est sans unité alors que $\text{square}(\langle p \rangle)$ a pour unité p^2). Enfin, nous donnons les types d'opérations licites à un ou plusieurs arguments.

4.2 Description de l'apprentissage

Notre méthode utilise la PGGG avec une grammaire construite par des experts. Nous parcourons l'espace de recherche en itérant des opérations (sélection, croisement, mutation) sur une population d'individus, comme dans l'algorithme 1. La particularité de la PGGG est d'assurer que les individus suivent toujours les règles grammaticales à chaque opération. Afin de sélectionner les parents de la génération suivante, les individus de la génération courante sont évalués par un objectif d'optimisation appelé fonction fitness. L'algorithme s'arrête lorsque la condition d'arrêt pré-définie est rencontrée. A la fin de chaque exécution, nous demandons aux opérateurs d'examiner les variables créées, de les commenter pour proposer de nouvelles règles grammaticales ou les améliorer. Ces itérations ont permis de créer la grammaire en Figure 1.

La fonction fitness comporte deux termes : $\text{fitness} = r - 10^{-8} \times \text{depth}$. Le premier, la valeur absolue de la corrélation linéaire de Pearson r , décrit l'adéquation entre un individu et la variable cible y . Il vaut entre 0 et 1, où une valeur de 1 correspond à des prédictions \hat{y} ayant un comportement identique à la cible y . Il mesure la force de la relation linéaire entre deux variables et présente l'avantage de pouvoir comparer le comportement de variables d'ordre de grandeurs différents. Ceci est adapté à notre problème car nous ne nous intéressons pas à la prédiction de la valeur exacte du flux mais plutôt à la compréhension de la *relation* entre les variables d'entrée X et le flux y . De plus, pour l'opérateur, il est aussi intéressant d'examiner une variable F que $10 \times F$. Le deuxième terme régularise la profondeur de l'arbre représentant l'individu (i.e. le nombre maximum de noeuds de la racine aux feuilles) pour empêcher l'explosion de sa taille (Purohit et al., 2011). C'est une contrainte de parcimonie permettant de conserver des expressions lisibles par l'humain. Elle est faiblement pondérée pour ne supprimer que les noeuds redondants sans trop contraindre l'espace de recherche.

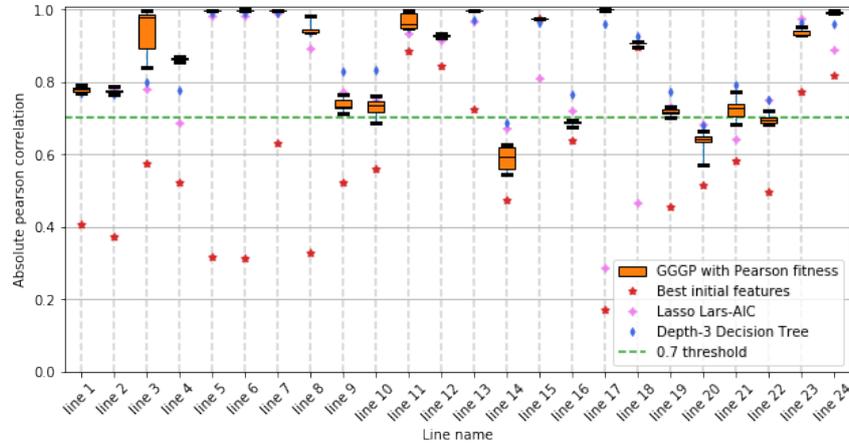


FIG. 2 – Comparaison entre la PGGG (boîte à moustaches orange) et l'état de l'art : Lasso Lars-AIC (rectangle rose) et arbre de décision (losange bleu). Pour chaque ligne étudiée, le nom correspondant est sur l'axe des abscisses et les résultats sont placés le long d'une ligne verticale en pointillés. La variable initiale la plus corrélée (étoile rouge) sert de référence.

5 Expérimentations et résultats

5.1 Protocole expérimental

Afin de ne pas examiner d'effets négligeables sur la variable cible, nous sélectionnons un sous-ensemble de lignes sur lesquelles réaliser l'extraction de variables. Pour cela, nous prenons les lignes les plus fréquemment chargées, c'est-à-dire se trouvant au-dessus d'un pourcentage $percent_i_threshold$ (100, 90, 80, et 70%) de la limite de la ligne $i_threshold$ pendant un pourcentage du temps total $percent_time$ (0,05 ou 0,1%). Par union de combinaison des hyperparamètres ($percent_time$, $percent_i_threshold$), nous identifions 24 lignes.

Nous souhaitons représenter le flux y des lignes présélectionnées par une nouvelle variable. Nous évoluons sur 200 générations une grande population (2 000 individus) en raison du nombre élevé de règles grammaticales. Le jeu de données initial est divisé en 80/20 entre entraînement et test. Pour chaque ligne, la recherche est lancée 30 fois avec une initialisation aléatoire, et seuls les individus les mieux notés sont conservés pour une inspection manuelle. Tous les hyperparamètres ont été sélectionnés en amont par validation croisée.

Nous utilisons l'implémentation parallèle open-source de PG en Python PonyGE2 (Fenton et al., 2017). Nous y insérons une métrique d'erreur basée sur la corrélation, de nouveaux traitement de données, d'évolution et la grammaire complète adaptée à notre problème.

5.2 Résultats

5.2.1 Comparaison avec d'autres méthodes

Cette expérience compare les individus évolués par corrélation avec des algorithmes d'interprétabilité similaires (Bayes Information criterion (Lasso Lars-BIC) (Zou et al., 2007) et un

arbre de décision de profondeur 3 (Breiman et al., 1984)) pour évaluer "fonctionnellement" la méthode (Doshi-Velez et Kim, 2017). Nous prenons comme référence la variable la plus corrélée des données initiales. Dans la Figure 2, les résultats associés à chaque ligne sont affichés le long d'une ligne verticale en pointillés, avec le nom de la ligne sur l'axe horizontal. Ainsi, pour la ligne 24 à droite, sont représentés de bas en haut les corrélations avec la cible de : la variable la plus corrélée des données initiales (marquée par une étoile rouge); la sortie de Lasso Lars-AIC (rectangle rose); la sortie de l'arbre de décision (losange bleu) et la boîte à moustaches des sorties de la PGGG (boîte orange).

Dans la Figure 2, les sorties de la PGGG sont plus corrélées avec la cible que les variables initiales. Nous constatons aussi que seuls 3 des 24 scores les plus élevés obtenus par PGGG sont inférieures au seuil de 0,7, en dessous duquel la corrélation n'est généralement pas considérée comme suffisamment forte pour être significative. Enfin, nous comparons notre approche avec d'autres approches interprétables, telles que le Lasso Lars ou un arbre de décision de profondeur limitée. Nous constatons que les résultats de la PGGG ayant une corrélation significative sont au moins aussi et parfois plus corrélés avec la cible que les autres méthodes.

5.2.2 Analyse experte des résultats

Les meilleures variables obtenues par PGGG sont finalement évaluées par des experts en réseau électriques afin d'identifier leur pertinence technique et physique. Cette expérience vise à confirmer l'interprétabilité d'un point de vue "humain" (Doshi-Velez et Kim, 2017). Les premières conclusions indiquent que toutes les variables au-dessus d'une corrélation de 0,8 sont pertinentes, définissant par ailleurs un seuil d'acceptation ou de rejet par l'expert. De plus, les variables de score très élevé (supérieur à 0,9) sont elles utiles en l'état. Toutefois, pour les variables inférieures à 0,8, les experts auraient trouvé intéressant d'intervenir pendant le processus d'apprentissage en supprimant, remplaçant/ajoutant des noeuds ou des feuilles des arbres pour augmenter leur score.

6 Conclusions et perspectives

En combinant interactivité et PGGG, nous obtenons des résultats prometteurs pour *extraire des variables interprétables* à partir de mesures du réseau électrique. La construction et la validation d'une grammaire non contextuelle permet d'inclure interactivement la *connaissance experte dans l'apprentissage*. Nos expériences évaluent également l'interprétabilité de notre méthode du point de vue "humain" et "fonctionnel". Les experts ont jugé des variables construites suffisamment pertinentes pour être incluses dans des outils d'hyper-vision. Cependant, certaines flux restent difficiles à expliquer avec une seule dimension. En effet, pour ces lignes, un modèle 1D est certainement trop restrictif, et nous envisageons de construire un espace multidimensionnel pour augmenter la représentativité du modèle. Nous utiliserons aussi des grammaires probabilistes pour mieux guider l'exploration.

Références

Breiman, L., J. H. Friedman, R. A. Olshen, et al. (1984). *Classification and Regression Trees*.

- Cherrier, N., J. Poli, M. Defurne, et F. Sabatié (2019). Consistent feature construction with constrained genetic programming for experimental physics. In *IEEE Congress on Evolutionary Computation*, pp. 1650–1658.
- Crochepierre, L., L. Boudjeloud-Assala, et V. Barbesant (2020). Interpretable Dimensionally-Consistent Feature Extraction from Electrical Network Sensors. In *Proceedings of ECML/PKDD*.
- Donnot, B., I. Guyon, M. Schoenauer, P. Panciatici, et A. Marot (2017). Introducing machine learning for power system operation support. In *IREP Symposium*.
- Doshi-Velez, F. et B. Kim (2017). Towards a rigorous science of interpretable machine learning. *stat 1050*, 2.
- Fenton, M., J. McDermott, D. Fagan, et al. (2017). Ponyge2 : grammatical evolution in python. In *GECCO (Companion)*, pp. 1194–1201.
- Khalid, S., T. Khalil, et S. Nasreen (2014). A survey of feature selection and feature extraction techniques in machine learning. In *2014 Science and Information Conference*, pp. 372–378.
- Koza, J. R. (1990). Concept formation and decision tree induction using the genetic programming paradigm. In *Parallel Problem Solving from Nature, 1st Workshop, PPSNI, Dortmund, Germany, Proceedings*, Volume 496, pp. 124–128.
- Laugel, T., M. Lesot, C. Marsala, et al. (2019). The dangers of post-hoc interpretability : Unjustified counterfactual explanations. In *IJCAI*, pp. 2801–2807.
- McKay, R. I., N. X. Hoai, P. A. Whigham, Y. Shan, et M. O’Neill (2010). Grammar-based genetic programming : a survey. *Genetic Programming and Evolvable Machines 11(3-4)*, 365–396.
- Purohit, A., A. Bhardwaj, A. Tiwari, et N. S. Chaudhari (2011). Handling the problem of code bloating to enhance the performance of classifier designed using genetic programming. In *IICAI*, pp. 333–342.
- Van Der Maaten, L., E. Postma, et J. Van den Herik (2009). Dimensionality reduction : a comparative review. *J Mach Learn Res 10(66-71)*, 13.
- Zou, H., T. Hastie, R. Tibshirani, et al. (2007). On the “degrees of freedom” of the lasso. *The Annals of Statistics 35(5)*, 2173–2192.

Summary

Electrical power networks are heavily monitored systems, requiring operators to perform intricate information synthesis before understanding the underlying network state. Our study aims at helping this synthesis step by automatically creating features from sensor data. We propose a feature extraction approach using a grammar-guided evolution, which outputs interpretable and physically consistent features. Operations restrictions on physical units are introduced in the learning process through interactively-built context-free grammars. They ensure coherence with physical laws, dimensional-consistency and also introduce technical expertise in the created features. We compare our approach to other state-of-the-art feature extraction methods on a real dataset taken from the French electrical network sensors and evaluate interpretability on human and functional levels.