Laboratoire d'Informatique de Paris 6

Faculté des Sciences et Ingénierie - Sorbonne université

Stage de recherche

XAI par approches logiques

Réalisé par

Rayane Nasri

Supervisé par

Marie-Jeanne Lesot

1 Introduction

Ces dernières années, il a été observé une montée exponentielle des modèles d'intelligence artificielle, notamment dans le domaine de l'apprentissage automatique. Actuellement, l'intelligence artificielle est omniprésente dans divers secteurs, parmi lesquels on peut citer la médecine (Gatza et al. 2010) (POLAT, GÜNEŞ, and ARSLAN 2008), la détection des spams et des virus (SONG, KOŁCZ, and GILES 2009), la reconnaissance optique de caractères (AYYAZ, JAVED, and MAHMOOD 2012) (CHACKO et al. 2012), l'identification biométrique (TAIGMAN et al. 2014), la cybersécurité (REGAINIA, SALVA, and ECUHCURS 2016), ainsi que d'autres applications variées. Les chercheurs ont été conduits à développer des modèles de complexité croissante, rendant la prise de décision opaque pour l'utilisateur. Cela a conduit à l'émergence de l'intelligence artificielle explicable (xAI), dont l'objectif est, comme son nom l'indique, de fournir des explications sur les décisions prises par un modèle. En effet, à la suite des récentes régulations du Règlement général sur la protection des données (RGPD) en 2018, qui exigent de fournir des explications sur les algorithmes de décision, ces explications sont devenues un droit pour l'utilisateur.

L'intelligence artificielle explicable (xAI) regroupe un ensemble d'algorithmes conçus pour permettre aux utilisateurs de comprendre et de valider les sorties générées par les modèles d'apprentissage automatique. Il existe deux approches: construire des modèles d'apprentissage automatique interprétables, tels que les arbres de décision (Hu, RUDIN, and SELTZER 2019), ou dériver des explications à partir de modèles d'apprentissage automatique complexes. Dans le cadre de ce stage, nous nous intéressons à la deuxième approche, qui se divise elle-même en deux catégories: Les méthodes ad-hoc tel que LIME (RIBEIRO, SINGH, and GUESTRIN 2016), SHAPE (MLUNDBERG and LEE 2017), et les méthodes formelles que nous allons détailler dans ce document.

Dans le cadre de ce stage, nous allons nous concentrer sur les méthodes formelles, plus précisément les méthodes logiques, notamment: la recherche des raisons suffisantes ayant conduit à une décision donnée (Darwiche and Hirth 2020) (Boumazouza 2024), les explications contrefactuelles (Boumazouza 2024), et les explications contrastives bi-factuelles (Miller 2020). Par ailleurs, dans le cadre de ce stage, nous n'aborderons pas en détail la transition du numérique au symbolique; nous supposerons que nous disposons déjà d'un classifieur représenté sous forme logique. En effet, l'autrice dans (Boumazouza 2024) a détaillé ce passage dans la section 3.2.

2 Introduction des notations

Les notations utilisés dans le cadre de ce stage sont exactement celles utilisés dans (Darwiche and Hirth 2020). Nous représentons un classifieur par une formule propositionnelle Δ dont les modèles (c'est-à-dire les affectations satisfaisantes) correspondent aux instances positives. La négation de la formule caractérise les instances négatives. Nous utilisons $\Delta(a)$ pour représenter la décision (0 ou 1) du

classifieur Δ sur l'instance a (c'est-à-dire $\Delta(a) = 1$ si et seulement si a satisfait Δ et $\Delta(a) = 0$ si et seulement si a satisfait $\neg\Delta$). Nous définissons également Δ_a comme étant égal à Δ si la décision est positive et à $\neg\Delta$ si la décision est négative.

Un **littéral** est une variable (littéral positif) ou sa négation (littéral négatif). Un **terme** est une conjonction satisfiable de littéraux. Le terme τ_j englobe le terme τ_i, écrit $\tau_j \models \tau_i$, si τ_j inclut les littéraux de τ_i. Par exemple, le terme $E \wedge \neg F \wedge G$ englobe le terme $E \wedge \neg F$. Nous traitons un terme comme l'ensemble de ses littéraux, donc nous pouvons écrire $\tau_i \subseteq \tau_j$ pour signifier que τ_j englobe τ_i. Parfois, nous appelons un littéral **une caractéristique** et un terme τ **une propriété** (d'une instance). Nous utilisons $\overline{\tau}$ pour désigner la propriété résultant de la négation de chaque caractéristique dans la propriété τ. Parfois, nous utilisons une virgule $(,)$ au lieu d'une conjonction $(\wedge)$ lors de la description des propriétés et des instances (par exemple, $E, \neg F$ au lieu de $E \wedge \neg F$). Un implicant τ de la formule propositionnelle Δ est un terme qui satisfait Δ, écrit $\tau \models \Delta$. Un implicant premier est un implicant qui n'englobe pas un autre implicant. Par exemple, $E \wedge \neg F \wedge G$ est un implicant de Δ mais n'est pas premier car il englobe un autre implicant $E \wedge \neg F$, qui lui est premier si l'on considère la formule: $\Delta = E \wedge (\neg F \vee G \vee W)$

3 Énumération des raisons suffisantes

Dans cette section, nous allons présenter les deux algorithmes utilisés lors du stage pour identifier les raisons suffisantes ayant declanché une décision.

Définition 1 (Raison suffisante). *Une raison suffisante pour une décision $\Delta(a)$ est un propriété de a qui est aussi un implicant premier de Δ_a.*

Nous avons mis en œuvre deux approches différentes pour énumérer les raisons suffisantes derrière une décision donnée. La première est celle présentée dans (Darwiche and Hirth 2020) et la deuxième est celle présentée dans (Boumazouza 2024).

3.1 Approche par la "Deterministic Decomposable Negation Normal From"

À partir d'un classifieur représenté par une formule propositionnelle, nous la transformons en forme normale conjonctive si ce n'était pas déjà le cas. Ensuite, la FNC est compilée en dDNNF à l'aide d'un compilateur tel que c2d[1]. Une fois que l'arbre représentant la dDNNF Δ est construit, nous construisons également $consensus(\Delta)$. Pour une instance a sur laquelle la décision a été prise $(\Delta(a) = 1)$, nous construisons le circuit $filter(consensus(\Delta), a)$, permettant ainsi de répondre à plusieurs requêtes en temps linéaire ou polynomial du nombre de nœuds de ce dernier.

[1] http://reasoning.cs.ucla.edu/c2d/

Définition 2 (Forme Normale de Négation). *Il s'agit d'un graphe acyclique dirigé où chaque nœud feuille est étiqueté par un littéral ou une constante vrai ou faux, et chaque nœud interne est étiqueté par une conjonction $\wedge$ ou une disjonction $\vee$.*

Une Forme Normale de Négation est dite vérifier la décomposabilité si et seulement si $vars(n_i) \cap vars(n_j) = \emptyset$ pour tous les enfants n_i et n_j d'un nœud "and_node". Elle est dite vérifier le déterminisme si et seulement si $\Delta(n_i) \wedge \Delta(n_j)$ est logiquement insatisfaisable pour tous les enfants n_i et n_j d'un nœud "or_node". Le compilateur c2d exprime chaque nœud "or_node" sous la forme $(X \wedge \alpha) \vee (\neg X \wedge \beta)$ où X est un symbol de variable et α β des Formes Normales de Négation. Ceci est crucial pour la construction du circuit $consensus(\Delta)$.

Définition 3 (consensus circuit). *On obtient le circuit consensus à partir d'un dDNNF en ajoutant un fils $\alpha \wedge \beta$ à chaque nœud "or_node" de la forme $(X \wedge \alpha) \vee (\neg X \wedge \beta)$.*

Définition 4 (circuit filtré). *Le filtrage d'un circuit consensus Γ par une instance a où $\Gamma(a) = 1$, est obtenu en remplaçant chaque littéral $l \notin a$ avec la constante faux.*

Théorème 1. *Considérons un circuit dDNNF Δ et une instance a telle que $\Delta(a) = 1$. Un terme τ est un implicant premier de Δ et $a \models \tau$ (ce qui signifie que τ est une raison suffisante) si et seulement si τ est un implicant premier de $filter(consensus(\Delta), a)$.*

Preuve. (Darwiche and Hirth 2020) page 7.

D'après le théorème, identifier les raisons suffisantes derrière une décision $\Delta(a) = 1$ revient à identifier les implicants premiers de $filter(consensus(\Delta), a)$. Notons que si $\Delta(a) = 0$, alors $\neg\Delta(a) = 1$, ce qui signifie que les implicants de $filter(consensus(\neg\Delta), a)$ nous permettront de lister les raisons suffisantes des instances dont la prédiction est négative. Nous allons maintenant présenter l'algorithme permettant de générer les raisons suffisantes derrière une décision. Avant cela, notons une légère différence entre l'algorithme présenté ici et celui décrit dans (Darwiche and Hirth 2020). Darwiche et Hirth, dans leur article, travaillent sur un 'decision-dDNNF', alors que nous utilisons un dDNNF. Bien que cette différence ne soit pas majeure, elle doit tout de même être prise en compte.

```
def Pi(decision: dDNNF, i= 0)-> list[set[int]]:
    """
    Generate the list of the decision's sufficient reasons.

    Parameters:
    ----------
    decision (dDNNF): The Filter(consensus(decision), a) circuit representing the decision.
```

```
    i (int): The index of the node we want to explain, default = 0.

    Returns:
    -------
    Explanation: A list of the decision's sufficient reasons.
    """

    r : list
    if (decision.label[i] == 'F'):
        r = []

    elif (decision.label[i] == 'T'):
        r = [set()]

    elif (decision.label[i] == 'A'):
        j = decision.children[i][0]
        r = Pi(decision, j)
        for j in range(1, len(decision.children[i])):
            r = prod_cart(r, Pi(decision, decision.children[i][j]))

    elif (decision.label[i][0] == 'O'):
        j = decision.children[i][0]
        r = Pi(decision, j)
        for j in range(1, len(decision.children[i])):
            r = r + Pi(decision, decision.children[i][j])

    else:
         r = [{int(decision.label[i])}]

    r = remove_subsumed(r)

    return r
```

3.2 Approche par le "Minimal unsatisfiable subset"

Considérons un classifieur représenté par une formule propositionnelle. Une instance qui ne satisfait pas cette formule. Nous construisons alors la formule $\Sigma = \Sigma_H \cup \Sigma_S$, où Σ_H représente les "Hards" clauses, celles qui doivent être absolument satisfaites, correspondant aux clauses du classifieur, tandis que Σ_S regroupe les "Softs" clauses, celles que l'on cherche à satisfaire autant que possible, représentant les clauses de l'instance. On calcule alors les "Minimal Unsatisfiable Subsets" de Σ, ce qui permet d'identifier les raisons suffisantes sous-tendant la décision.

Définition 5 (Minimal unsatisfiable subset MUS). *C'est un sous-ensemble*

minimal de clauses Γ *de la forme normale conjonctive* Σ *tel que* $\forall a \in \Gamma, \Gamma \setminus a$ *est satifiable.*

Définition 6 (Maximal satisfiable subset MSS). *C'est un sous-ensemble de clauses* $\phi \subseteq \Sigma$ *qui est satisfiable, tel que* $\forall a \in \Sigma \setminus \phi, \phi \cup a$ *est insatisfiable.*

Définition 7 (Minimal correction subset MCS). *C'est un ensemble de formules* $\psi \subseteq \Sigma$ *tel que son complément dans* $\Sigma, \Sigma \setminus \psi$ *est un "Maximal satisfiable subset MSS".*

Définition 8 (Partial Max-SAT). *Étant donnée une formule booléenne en forme normale conjonctive (CNF)* Σ *comprenant des "Hard" clauses et des "Soft" clauses, le problème de Partial Max-SAT consiste à trouver une affectation de vérité qui satisfait toutes les "Hard" clauses et le maximum "Soft" clauses.*

Théorème 2. *Considérons* f *un classifieur et* Σ_f *sa représentation sous forme normale conjonctive (CNF). Soit* x *une instance pour laquelle* f *prédit une sortie négative, c'est-à-dire* $f(x) = 0$*, et* $\Sigma_f \cup \Sigma_x$ *l'encodage correspondant en Partial Max-Sat. Désignons par* $SR(x, f)$ *l'ensemble des raisons suffisantes expliquant* $f(x)$*. Si l'on note* $MUS(\Sigma_f \cup \Sigma_x)$ *l'ensemble des "Minimal unsatisfiable subsets", alors* $SR(x, f) = MUS(\Sigma_f \cup \Sigma_x)$*.*

Preuve. (Boumazouza 2024) page 54.

Pour déterminer l'ensemble des "Minimal unsatisfiable subsets" $MUS(\Sigma_f \cup \Sigma_x)$, on commence par calculer l'ensemble des "Minimal correction subsets" $MCS(\Sigma_f \cup \Sigma_x)$. Puisque ces deux ensembles sont des "hitting set duals" l'un de l'autre, les $MUSes$ peuvent être obtenus en calculant les "minimal hitting sets" des $MCSs$. Pour déterminer l'ensemble des "Minimal correction subsets" on utilise l'outil **EnumELSRMRCache**[2] qui implémente l'algorithme "Boosting $MCSes$ enumeration" proposé dans (GRÉGOIRE, IZZA, and LAGNIEZ 2018).

Le code Python est présenté dans le notebook *xALogic.ipynb* (respectivement *xRLogic.ipynb*) pour l'approche utilisant la "Deterministic Decomposable Negation Normal Form" (respectivement "Minimal unsatisfiable subset") sur mon GitHub[3].

4 Énumération des explications contrefactuelles

Dans cette section nous allons présenté l'algorithme utilisé lors du stage pour identifier les explications contrefactuels pour une décision donnée.

Comme dans la section 3.2, nous construisons la formule $\Sigma = \Sigma_H \cup \Sigma_S$ mais cette fois nous calculons les "Minimal correction subsets" de Σ, ce qui permet d'identifier les explications contrefactuelles pour une décision.

Définition 9 (Explication contrefactuelle). *Une explication contrefactuelle* x *pour une décision* $\Delta(a)$ *est une propriété minimale de* a *telle que, si* b *est une*

[2] http://www.cril.univ-artois.fr/enumcs/
[3] https://github.com/RayaneNasri

instance qui diffère de a uniquement par x, alors $\Delta(b) = 1 - \Delta(a)$.

Théorème 3 *Considérons f un classifieur et Σ_f sa représentation sous forme normale conjonctive (CNF). Soit x une instance pour laquelle f prédit une sortie négative, c'est-à-dire $f(x) = 0$, et $\Sigma_f \cup \Sigma_x$ l'encodage correspondant en Partial Max-Sat. Désignons par $CF(x, f)$ l'ensemble des explications contrefactuelles expliquant $f(x)$. Si l'on note $MCS(\Sigma_f \cup \Sigma_x)$ l'ensemble des "Minimal correction subsets", alors $CF(x, f) = MCS(\Sigma_f \cup \Sigma_x)$.*

Preuve. (Boumazouza 2024) page 55.

Nous avons déjà expliqué comment identifier l'ensemble des "Minimal Correction Subsets" dans la section précédente.

5 Énumération des explications contrastives bi-factuelles

Dans les sections précédentes, nous avons présenté des méthodes d'explications non contrastives. Ces deux approches se limitent à expliquer un événement sans référence à d'autres alternatives. Les questions contrastives, en revanche, mettent en lumière ces alternatives. Dans notre étude, nous cherchons à identifier des explications contrastives bi-factuelles, visant à élucider pourquoi un événement s'est produit dans un contexte donné, tandis qu'un événement alternatif s'est produit dans un contexte similaire. Les travaux de Tim Miller, décrits dans (Miller 2020), ont permis d'établir une définition formelle des explications contrastives bi-factuelles. Toutefois, aucune proposition d'algorithme efficace pour les identifier n'avait été avancée à ce moment-là. Nous allons donc revisiter les définitions de Tim Miller en conservant nos notations, tout en introduisant notre propre algorithme.

Définition 10 (Cause réelle). *Soit a une instance. τ est une cause réelle si et seulement si,*

- $a \models \tau$ et $a \models \Delta_a$.
- $\exists w$ tel que $a = \tau \cup w$ et $\overline{\tau} \cup w \models \neg\Delta_a$.
- $\nexists x, x \subset \tau$ qui vérifie les deux premiers points.

Définition 11 (Cause partielle). *Une cause partielle est un sous-ensemble (inclus ou égal) d'une cause réelle.*

Dans ce qui suit, si c est un ensemble de symboles de variables, c_a désigne la conjonction des littéraux appartenant à a dont la variable est dans c.

Définition 12 (Cause contrastive bi-factuelle). *Soit a, b deux instances, et Δ un classifieur symbolique. c est une cause contrastive bi-factuelle si et seulement si,*

- c_a (respectivement c_b) est une cause partielle de Δ_a (respectivement Δ_b).
- $c_a \cap c_b = \emptyset$.
- $\nexists c'$, $c \subset c'$ et c' vérifie les deux premiers points.

Notre algorithme repose sur l'équivalence des définitions 9 et 10. En d'autres termes, la définition des explications contrefactuelles par Ryma Boumazouza dans (Boumazouza 2024) est équivalente à celle des "causes réelles" établie par Miller dans (Miller 2020). Cette équivalence est en effet valide et se démontre.

Proposition 1. Considérons un classifieur représenté sous forme normale conjonctive (CNF) Δ. Soit x une instance donnée. Soit $CF(\Delta, x)$ l'ensemble des explications contrefactuelles de $\Delta(x)$. Si $CS(\Delta, x)$ désigne l'ensemble des causes réelles de $\Delta(x)$, alors $CF(\Delta, x) = CS(\Delta, x)$.

Preuve. Nous démontrerons l'égalité par double inclusion. Soit $e \in CF(\Delta, x)$. Alors e est une propriété de x, donc $x \models e$. Par définition (Section 2), $x \models \Delta_x$ est vrai. Posons $w = x \setminus e$. Par la définition de l'explication contrefactuelle, nous avons $w \cup \overline{e} \models \neg\Delta_x$. Comme e est minimal, cela conclut que $e \in CS(\Delta, x)$. Pour l'autre sens, soit $e \in CS(\Delta, x)$. Posons $y = w \cup \overline{e}$, w est défini selon le point 2 de la définition 10. Ainsi $y \models \neg\Delta_x$ d'où $\Delta(y) = 1 - \Delta(x)$. La minimalité de e comme propriété de x est établie par les points 1 et 3 de la définition 10.

Proposition 2. Considérons un classifieur représenté sous forme normale conjonctive (CNF) Δ. Soient a et b deux instances. Désignons par $\chi(\Delta, a, b)$ l'ensemble des explications contrastives bi-factuelles. Notons $CF'(\Delta, a)$ (respectivement $CF'(\Delta, b)$) l'ensemble des explications contrefactuelles de Δ_a (respectivement de Δ_b) dont on a retiré tous les littéraux communs à a et b. Donc $\chi(\Delta, a, b) = \{c1 \cap c2 | c1_a \in CF'(\Delta, a), c2_b \in CF'(\Delta, b)\}*$

L'étoile dans l'énoncé de la proposition indique que les ensembles qui sont inclus dans d'autres ensembles ont été retirés.

Preuve. Nous démontrerons cette égalité par double inclusion. Soit $e \in \chi(\Delta, a, b)$, montrons que $e \in \{c1 \cap c2 \mid c1_a \in CF'(\Delta, a), c2_b \in CF'(\Delta, b)\}*$. e_a (respectivement e_b) est une cause partielle de Δ_a (respectivement Δ_b) et comme $e_a \cap e_b = \emptyset$, e_a et e_b ne sont pas affectés par le retrait des littéraux communs. Alors, il existe $c1$ et $c2$ tels que $c1_a \in CF'(\Delta, a)$ et $c2_b \in CF'(\Delta, b)$ où $e \subseteq c1 \cap c2$. Cependant, d'après le point 3 de la définition 12, e est maximal, donc $e = c1 \cap c2$ et par conséquent $e \in \{c1 \cap c2 \mid c1_a \in CF'(\Delta, a), c2_b \in CF'(\Delta, b)\}$. Par le même raisonnement, on arrive à $e \in \{c1 \cap c2 \mid c1_a \in CF'(\Delta, a), c2_b \in CF'(\Delta, b)\}*$. Pour l'inclusion dans l'autre sens, soit $e \in \{c1 \cap c2 \mid c1_a \in CF'(\Delta, a), c2_b \in CF'(\Delta, b)\}*$. e_a (respectivement e_b) est une cause partielle de Δ_a (respectivement Δ_b). $e_a \cap e_b = \emptyset$ puisque les littéraux communs ont été filtrés. De plus, puisque les ensembles inclus dans d'autres ensembles ont été retirés, le point 3 est évidemment vérifié. D'où $e \in \chi(\Delta, a, b)$.

Pour revenir à la définition de Tim Miller concernant les explications contrastives bi-factuelles, il suffit d'appliquer une fonction de mappage sur l'ensemble $\chi(\Delta, a, b)$, où chaque élément e est associé à $< e_a, e_b >$.

L'implémentation de cet algorithme se trouve dans le fichier causaLitycs.ipynb, dans le contexte spécifique de l'admission en Master.

6 Complexité théorique de l'algorithme

Il est manifeste que la complexité de cet algorithme est nettement meilleure que celle présentée par Faten Racha Said, Ahmed Abdelaziz Mokeddem et Yacine B.D. Chettab dans leur rapport de projet[4] qui est de nature factorielle. En notant n_1 (respectivement n_2) le nombre d'explications contrefactuelles pour la première (respectivement la deuxième) instance, v le nombre de variables et α la complexité de calcul des explications contrefactuelles, la complexité dans le pire des cas est $O(\alpha + n_1.n_2.v\check{s})$. Il est important de rester prudent et de ne pas se laisser emporter trop vite, car les valeurs de n_1, n_2 et α peuvent être exponentielles, même si cela reste préférable à une complexité factorielle. Cependant, dans le cas moyen, leurs valeurs sont généralement acceptables.

7 Conclusion

Pendant ce stage, nous avons réussi à implémenter des algorithmes existants pour expliquer les modèles d'apprentissage automatique (Sections 3 et 4). De plus, nous avons proposé un nouvel algorithme visant à identifier les explications contrastives bi-factuelles, répondant ainsi à la question : "Pourquoi le modèle a pris une décision Δ_a pour l'instance a et une décision Δ_b (probablement différente) pour l'instance b." Nous avons démontré formellement sa validité et l'avons également mis en œuvre. La prochaine étape consisterait à évaluer ce dernier en utilisant des volumes de données significatifs pour vérifier son efficacité expérimentale.

8 Remerciments

Tout d'abord, je souhaite exprimer ma gratitude envers madame Lesot Marie-Jeanne, ma tutrice de stage et professeure à Sorbonne Université, pour m'avoir offert l'opportunité de plonger dans le domaine de la recherche. En début de stage, j'étais totalement novice en matière de modèles de machine learning, et elle m'a guidé pas à pas jusqu'à ce que je sois capable de travailler de manière autonome. Évidemment, sans elle, rien de tout cela n'aurait été possible.

Je souhaite également exprimer mes sincères remerciements à toute l'équipe LFI avec laquelle j'ai collaboré durant ce stage. Sans oublier l'équipe administrative du LIP6 et de Sorbonne Université qui ont tout mis en œuvre pour rendre les démarches administratives aussi simples que possible.

Enfin, je tiens à remercier ma famille et mes amis pour leur soutien indéfectible tout au long de mon parcours académique et professionnel.

9 Bibliographie

AYYAZ, Muhammad Naeem, Imran JAVED, and Waqar MAHMOOD. 2012. "Handwritten Character Recognition Using Multiclass Svm Classification with

[4] https://github.com/chettabyacine/M1-S2-PLDAC/tree/main

Hybrid Feature Extraction." *Pakistan Journal of Engineering and Applied Sciences*.

Boumazouza, Ryma. 2024. "Predictive Models & Reasoning with Explanations."

CHACKO, Binu P, VR VIMAL KRISHNAN, G RAJU, and P BABU ANTO. 2012. "Handwritten Character Recognition Using Wavelet Energy and Extreme Learning Machine." *International Journal of Machine Learning and Cybernetics*.

Darwiche, Adnan, and Auguste Hirth. 2020. "On the Reasons Behind Decisions."

Gatza, Michael L., Joseph E. Lucas, William T. Barry, Jong Wook Kim, Quanli Wang, Matthew D. Crawford, Michael B. Datto, et al. 2010. "A Pathway-Based Classification of Human Breast Cancer." *Proceedings of the National Academy of Sciences*.

GRÉGOIRE, Éric, Yacine IZZA, and Jean-Marie LAGNIEZ. 2018. "Boosting Mcses Enumeration." 1309–15.

Hu, Xiyang, Cynthia RUDIN, and Margo SELTZER. 2019. "Optimal Sparse Decision Trees." *Advances in Neural Information Processing Systems*.

Miller, Tim. 2020. "Contrastive Explanation: A Structural-Model Approach." *The Knowledge Engineering Review*.

MLUNDBERG, Scott, and SuIn LEE. 2017. "Unified Approach to Interpreting Model Predictions," 30.

POLAT, Kemal, Salih GÜNEŞ, and Ahmet ARSLAN. 2008. "A Cascade Learning System for Classification of Diabetes Disease: Generalized Discriminant Analysis and Least Square Support Vector Machine." *Expert Systems with Applications*.

REGAINIA, Loukmen, Sébastien SALVA, and Cedric ECUHCURS. 2016. "A Classification Methodology for Security Patterns to Help Fix Software Weaknesses," 1–8.

RIBEIRO, Marco Tulio, Sameer SINGH, and Carlos GUESTRIN. 2016. ""Why Should I Trust You?" Explaining the Predictions of Any Classifier," 1135–44.

SONG, Yang, Aleksander KOŁCZ, and CLee GILES. 2009. "Better Naive Bayes Classification for High-Precision Spam Detection." *Software: Practice and Experience*.

TAIGMAN, Yaniv, Ming YANG, Marc'Aurelio RANZATO, and Lior WOLF. 2014. "Deepface: Closing the Gap to Human-Level Performance in Face Verification." *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1701–8.