

MODÈLE DE RÉSEAU DE NEURONES FIABLE POUR ACCÉLÉRER LES SIMULATIONS COUPLÉES DE THERMODIFFUSION

M.B Yahiaoui¹ & L. Giraldi¹ & G. Daniel² & C. Introïni¹ & J. Arbel³

¹ *CEA, DES, IRESNE, DEC, SESC, F-13108 Saint-Paul-lez-Durance, France
{mohamed-bahi.yahiaoui, loic.giraldi, clement.introïni}@cea.fr*

² *Université Paris-Saclay, CEA, SGLS, 91191, Gif-sur-Yvette, France
geoffrey.daniel@cea.fr*

³ *Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, 38000 Grenoble, France
julyan.arbel@inria.fr*

Résumé. Nous présentons une implémentation d’un modèle de substitution d’apprentissage profond pour prédire les contributions thermodynamiques (potentiels chimiques, mobilités) impliquées dans la résolution d’un problème d’inter-diffusion. Cette approche vise à accélérer les simulations, en tirant parti de la puissance des réseaux de neurones profonds. Le modèle de substitution est conçu pour reproduire le comportement du code de calcul d’équilibre thermodynamique, ici OpenCalphad, dans le cas simplifié d’un problème d’inter-diffusion pour un système ternaire. Les modèles d’apprentissage profond ont montré des performances remarquables sur notre cas d’étude sur nos données d’entraînement et de tests. Toutefois, il est nécessaire d’apporter des garanties sur l’erreur potentiellement commise par notre modèle. Nous abordons ainsi un aspect important dans l’utilisation des réseaux de neurones comme modèle de substitution et de leur fiabilité : les incertitudes. Nous présentons ainsi une méthode novatrice fondée sur l’étude du comportement des couches cachées des réseaux de neurones et visant à gérer efficacement l’incertitude dans l’apprentissage profond.

Mots-clés. Modèles de substitution, Apprentissage profond, Réseaux de neurones, Incertitudes, Surveillance en temps réel, Détection des données hors distribution, OpenCalphad, Thermodiffusion.

Abstract. We present an implementation of a deep learning surrogate model to predict the thermodynamic contributions (chemical potentials, mobilities) involved in solving an inter-diffusion problem. This approach aims to accelerate thermodiffusion simulations by leveraging the power of deep neural networks. The surrogate model is designed to replicate the behavior of the thermodynamic equilibrium calculation code, here OpenCalphad, in the simplified case of an inter-diffusion problem for a ternary system. Deep learning models have shown remarkable performance on our case study with our training and testing data. However, it is necessary to provide guarantees on the potential error made by our model. We thus address an important aspect in the use of neural networks as surrogate models and their reliability : uncertainties. We present an innovative method based on the study of the behavior of the hidden layers of neural networks and aiming to effectively manage uncertainty in deep learning.

Keywords. Surrogate models, Deep learning, Neural networks, Uncertainties, Runtime monitoring, Out-Of-Distribution detection, OpenCalphad, Thermodiffusion.

1 Introduction

Dans le contexte de la modélisation de systèmes complexes, la construction de modèles précis et efficaces est essentielle pour comprendre et prédire le comportement de ces systèmes. Cependant, la complexité de certains problèmes, par exemple en modélisation physique, rend souvent la simulation directe difficile. C’est ici que les modèles de substitution, parfois désignés sous le terme de métamodèles, prennent leur place.

Les modèles de substitution [1, 9] sont des approximations simplifiées du système d’origine pour remplacer efficacement des simulations coûteuses en temps et en ressources de calcul. Ces modèles sont généralement construits par des méthodes d’apprentissage via des données obtenues à partir de simulations du système d’origine ou d’observations expérimentales.

Dans ce travail, l’objectif est d’utiliser un modèle de substitution pour accélérer la simulation de l’évolution d’un système ternaire de diffusion d’éléments chimiques dans un solide sous l’effet d’un gradient de température (ici, un profil parabolique imposé). Dans le système d’équations aux dérivées partielles associé à ce problème, les flux de diffusion dépendent des mobilités et des potentiels chimiques des éléments. Ces contributions sont évaluées à chaque pas de temps, en chaque point du maillage, par un code de calcul d’équilibres thermodynamiques, OpenCalphad [8]. De manière générale, ce type de couplage entre un code de calcul thermodynamique et un modèle de transport diffusif peut s’avérer coûteux selon la complexité de la description thermodynamique (non décrite dans ce travail) et la raideur du problème numérique. Dans ces conditions, l’utilisation d’un modèle de substitution est pertinente pour réduire le surcoût des simulations provoqué par la détermination des potentiels chimiques et des mobilités par les calculs thermodynamiques locaux effectués par OpenCalphad. De plus, l’étude d’un système simplifié permet de mettre en place une démarche générale et reproductible pour des études plus complexes impliquant des systèmes multiphasiques multi-composants [4, 5].

Les réseaux de neurones sont des outils intéressants pour construire des modèles de substitution. L’un des principaux avantages de l’utilisation des réseaux de neurones dans ce cadre est leur capacité d’approximation universelle [6]. Par conséquent, ils peuvent gérer des relations complexes qui peuvent exister dans le système sous-jacent, sous réserve d’avoir un nombre de données suffisant. L’une des limites de l’utilisation des réseaux de neurones est leur incapacité à fournir nativement une incertitude liée à leur manque de connaissance ou d’information, couramment appelée incertitude épistémique. Le principal défi est la dépendance du modèle à la distribution des données d’entraînement. Lorsque le réseau de neurones est déployé dans un environnement de production, il est nécessaire de s’assurer que les nouvelles évaluations du métamodèle soient valides, surtout pour les entrées qui s’éloignent de la distribution d’entraînement. Par conséquent, il est nécessaire de disposer d’un domaine opérationnel bien défini qui puisse être utilisé pour identifier les potentielles erreurs de prédiction avant qu’elles ne se propagent à travers le système. Pour garantir sa fiabilité en déploiement, le modèle doit être capable de détecter des données pour lesquelles il fournit des prédictions potentiellement erronées, appelées données hors domaine du modèle, ou *Out-of-Model Scope (OMS) data* [2]. À cette fin, nous introduisons une nouvelle approche de détection des données hors domaine du modèle, nommée **BB-AS** (*Bounding Box with*

Anomaly Score). Elle consiste à comparer les états des couches intermédiaires avec un ensemble de boîtes englobantes [3].

Dans la section 2, nous décrivons brièvement le problème d’inter-diffusion pour un système ternaire et le développement d’un métamodèle pour le calcul des contributions thermodynamiques. Puis, dans la section 3, nous présentons la méthode BB-AS en décrivant les principes de son fonctionnement et son application au modèle de substitution d’OpenCalphad.

2 Métamodélisation du code de calcul OpenCalphad

2.1 Description du problème physique

Dans le cadre de ce travail, nous nous intéressons à la simulation de l’évolution d’un système ternaire U-Pu-O en phase solide. Plus spécifiquement, ces travaux visent à accélérer les simulations numériques de diffusion d’espèces, où les forces motrices sont exprimées en fonction des potentiels chimiques calculés par OpenCalphad.

La simulation nécessite d’abord la définition d’un état initial, comprenant les profils des quantités molaires initiales $N_e^0(\cdot)$ des éléments dans le système ainsi que le profil de température $T(\cdot)$ imposé. Les calculs d’équilibres thermodynamiques sont réalisés à une pression constante de 50 bars. Puis, l’algorithme de résolution est basé sur un schéma partitionné en deux étapes, illustré sur la Figure 1.

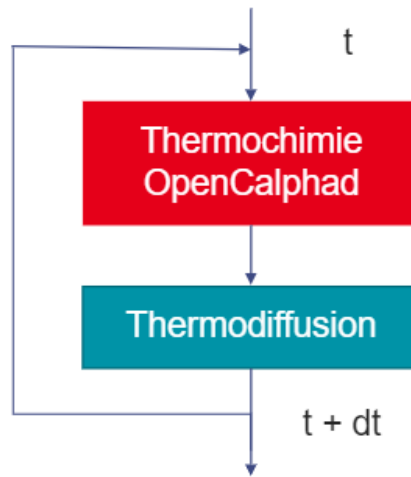


FIGURE 1 – Algorithme partitionné pour la résolution du problème d’inter-diffusion couplé au code de calcul thermodynamique OpenCalphad.

À chaque pas de temps et sur chaque nœud spatial du système, la première étape consiste à réaliser des calculs d’équilibres thermodynamiques OpenCalphad pour déterminer les potentiels chimiques μ_e et les mobilités M_e de chaque élément e (Figure 2). Ensuite, ces



FIGURE 2 – Entrées et sorties du composant OpenCalphad, que l’on cherche à métamodéliser

données thermodynamiques sont utilisées à la deuxième étape pour résoudre les équations d’interdiffusion suivantes (Éqs. 1) :

$$\frac{\partial x_O}{\partial t} = \nabla \cdot [M_{O,U} \nabla(\mu_O - \mu_U) + M_{O,Pu} \nabla(\mu_O - \mu_{Pu})], \quad (1a)$$

$$\frac{\partial x_U}{\partial t} = \nabla \cdot [M_{O,U} \nabla(\mu_U - \mu_O) + M_{U,Pu} \nabla(\mu_U - \mu_{Pu})], \quad (1b)$$

$$\frac{\partial x_{Pu}}{\partial t} = \nabla \cdot [M_{O,Pu} \nabla(\mu_{Pu} - \mu_O) + M_{U,Pu} \nabla(\mu_{Pu} - \mu_U)], \quad (1c)$$

où x_e est la fraction molaire de l’élément e , calculée à partir des quantités molaires des éléments, $M_{e,e'}$ le coefficient d’inter-diffusion entre les éléments e et e' . Le coefficient d’inter-diffusion est calculé à partir des mobilités M_e , $M_{e'}$ et des fractions molaires x_e , $x_{e'}$ de chaque élément :

$$M_{e,e'} = (M_e x_{e'} + M_{e'} x_e) x_e x_{e'}.$$

À l’issue de la deuxième étape, les quantités molaires de chaque élément sont mises à jour pour les calculs d’équilibres thermodynamique de la première étape du pas de temps suivant.

Les performances de ce couplage entre résolution d’un problème de diffusion et calculs d’équilibres thermodynamiques locaux en chaque nœud du maillage et à chaque instant sont pénalisés par l’appel direct au code de calcul thermodynamique. Par conséquent, pour réduire le surcoût engendré par les calculs thermodynamiques dans la simulation globale, nous développons un réseau de neurones pour se substituer au calcul des potentiels et des mobilités.

2.2 Développement du réseau de neurones

Pour générer les données d’entraînement, nous avons lancé 1000 simulations d’interdiffusion couplées au code de thermodynamique OpenCalphad. Pour toutes les simulations, nous avons imposé une symétrie coaxiale ainsi qu’un profil de température parabolique. Nous avons utilisé une grille de 40 points et avons exécuté la simulation pendant 1000 pas de temps, chacun représentant 0.5 s, pour un total de 500 s. Chaque simulation débute avec un mélange uniforme d’Uranium, de Plutonium et d’Oxygène à chaque point de la grille et la température minimale du système T_{\min} , que l’on doit spécifier.

Pour chaque simulation, nous avons donc tiré aléatoirement :

- La composition initiale :
 - $N_U^0 \sim U(0.6, 0.9)$
 - $N_O^0 \sim U(1.975, 1.999)$

- $N_{\text{Pu}}^0 = 1 - N_U^0$
- La température minimale du système : $T_{\text{min}} \sim U(800, 1100)$

Les données utilisées pour l'entraînement du réseau sont les paires entrée-sortie produites par OpenCalphad au cours de la simulation. 70 % des simulations sont utilisées pour l'entraînement, 15 % pour la validation et 15 % pour les tests.

Le réseau de neurones que nous avons développé est un réseau complètement connecté comprenant 16 couches, chacune de 512 neurones utilisant une connexion de saut (*skip connection*) à chaque couche et avec une fonction d'activation Tanh. Nous avons utilisé une fonction de perte \mathcal{L} incorporant la perte Mean Squared Error (MSE) classique, notée \mathcal{L}_α , et un terme supplémentaire \mathcal{L}_β qui prend en compte plus spécifiquement les différences de potentiels chimiques et termes d'inter-diffusion apparaissant dans le système Eq1.

$$\mathcal{L}(y, \hat{y}) = \mathcal{L}_\alpha(y, \hat{y}) + \mathcal{L}_\beta(y, \hat{y}),$$

Avec :

$$\mathcal{L}_\alpha(y, \hat{y}) = \sum_{e \in \{\text{O, Pu, U}\}} ((\mu_e - \hat{\mu}_e)^2 + (M_e - \hat{M}_e)^2),$$

et

$$\mathcal{L}_\beta(y, \hat{y}) = \frac{1}{2} \sum_{\substack{e, e' \in \{\text{O, Pu, U}\} \\ e \neq e'}} (((\mu_e - \mu_{e'}) - (\hat{\mu}_e - \hat{\mu}_{e'}))^2 + (M_{e, e'} - \hat{M}_{e, e'})^2).$$

Sur la base de test, nous avons obtenu des résultats prometteurs avec un coefficient de détermination R^2 supérieur à 99,99% sur toutes les quantités prédites. Nous montrons la pertinence de notre approche en déployant le modèle sur une simulation de 500 s, voir Figure 3.

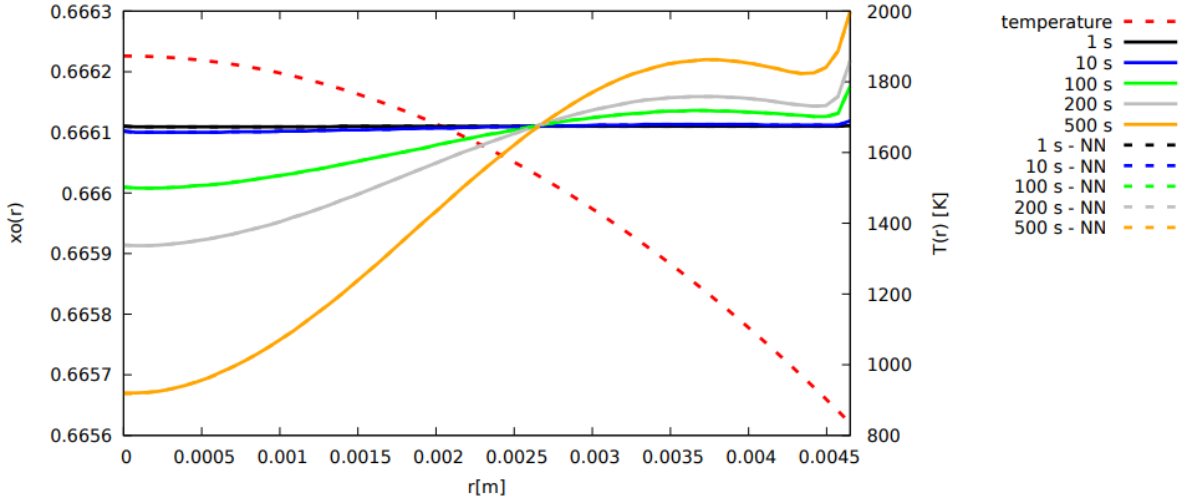


FIGURE 3 – Instantanés du profil de fraction molaire d'oxygène (x_{O}) par rapport au rayon du granule ($r[m]$). La ligne en pointillés rouges montre le profil de température fixe. Les lignes continues (resp. en pointillés) montrent l'évolution des fractions molaires dans la simulation couplée avec OpenCalphad (resp. le modèle de réseau de neurones). Les lignes continues et en pointillés sont superposées du fait que l'utilisation du réseau de neurones fournit des résultats très proches de ceux avec appel à OpenCalphad.

3 Détection des données hors du domaine du modèle

3.1 Méthodologie de détection

Malgré les performances de notre réseau de neurones obtenues sur les données de test, il est nécessaire de fournir des garanties sur sa fiabilité dans le cas d'un déploiement en production. La détection des données hors du domaine¹ du modèle [2] est une tâche importante pour répondre à cet objectif. Elle consiste à identifier les données d'entrée qui se situent en dehors du domaine où il est capable de fournir des prédictions correctes et fiables. Cela permet de rejeter le résultat du réseau de neurones si on considère que le niveau de confiance accordé à la prédiction n'est pas suffisamment élevé.

Notre objectif est de développer un système de surveillance en temps réel [2] pour la détection des entrées hors du domaine du modèle pour plusieurs raisons. Un système de surveillance en temps réel a pour but de détecter et de répondre rapidement aux déviations ou aux anomalies pendant l'exécution de la simulation, garantissant ainsi sa pertinence et son efficacité alors que le comportement du système physique évolue.

La méthode de surveillance que nous proposons est basée sur les boîtes englobantes (*Bounding boxes*), sur une approche similaire à celle proposée dans [3]. L'idée de base est de regrouper les données d'entraînement en clusters en fonction des valeurs des neurones avant activation des couches cachées du réseau. Puis, on établit le domaine d'entrée acceptable en définissant des bornes inférieures et supérieures sur les états des neurones pour chacun de ces clusters, construisant ainsi les boîtes englobantes. Le système de surveillance, ainsi construit dans [3], rejette une entrée si l'ensemble des états des neurones des couches cachés résultant de cette entrée ne se situent dans les bornes d'aucun cluster.

Cette méthode est très conservatrice, au sens où un seul dépassement des boîtes englobantes pour un seul neurone est réhibitoire. Nous proposons une extension de cette méthode, nommée **BB-AS** pour *Bounding box with Anomaly Score*, qui attribue un score d'anomalie consistant à calculer le nombre de dépassements des bornes définies sur les états des neurones sur chaque cluster et à conserver le nombre minimum de dépassement sur tous les clusters. Ainsi, le score d'anomalie d'une entrée varie entre 0 (lorsque l'entrée se situe dans les bornes d'un des clusters), et le nombre de neurones des couches cachées de réseau (lorsque l'entrée a dépassé tous les seuils de tous les clusters). La méthode BB-AS offre alors une souplesse qui permet de prendre en compte les cas où les données se situant hors de quelques bornes des boîtes englobantes peuvent malgré cela être pertinentes.

1. Cette tâche se distingue de l'approche plus classique qui consiste à identifier les données dites hors distribution. Celle-ci cherche en effet à détecter les données qui ne sont pas conformes à la distribution attendue ou normale de l'ensemble de données d'entraînement, mais cette formulation est ambiguë, car il est souvent difficile de définir avec précision ce qui constitue une donnée anormale.

3.2 Exemple d’application de la méthode BB-AS au métamodèle d’OpenCalphad

Afin de tester la méthode BB-AS sur le problème présenté dans la section 2, nous avons utilisé 1000 simulations d’inter-diffusion couplées au code de thermodynamique OpenCalphad. Pour ces simulations, nous nous sommes placés dans la même configuration que celle décrite en section 2. La principale différence est que nous avons élargi la distribution des paramètres initiaux de simulation afin d’inclure à la fois des données hors et dans le domaine du modèle. Le tirage aléatoire de ces paramètres est modifié comme suit :

- Composition initiale :
 - $N_{\text{U}}^0 \sim U(0.4, 0.95)$
 - $N_{\text{O}}^0 \sim U(1.95, 1.999)$
 - $N_{\text{Pu}}^0 = 1 - N_{\text{U}}^0$
- La température minimale du système : $T_{\text{min}} \sim U(750, 1100)$

Dans cette démarche, nous avons repris le même modèle préalablement entraîné.

Pour définir le domaine du modèle, nous avons utilisé un seuil supérieur sur la perte \mathcal{L} de 10^{-4} , calculée à partir des paires entrées-sorties produites par OpenCalphad et des prédictions du réseau. Nous considérons qu’une donnée dont la perte dépasse ce seuil est une donnée hors du domaine de modèle. La valeur de ce seuil a été définie empiriquement en remarquant que des simulations comportant des erreurs inférieures 10^{-4} ne divergent pas. Ainsi, pour construire nos boîtes englobantes, nous avons utilisé les données de l’ensemble d’entraînement en rejetant les instances où la perte dépasse ce seuil. Cette expérience n’a nécessité que l’utilisation d’un seul cluster pour regrouper les données, ce qui conduit finalement à une seule boîte englobante.

Nous avons comparé la méthode BB-AS à la méthode de référence *Deep ensemble* [7], en construisant un ensemble profond composé de 8 modèles de même architecture que notre modèle et entraîné sur les mêmes données. Nous avons utilisé le logarithme décimal² de la moyenne de la variance des sorties des huit réseaux comme critère de détection de données hors domaine du modèle. Nous avons également comparé la méthode BB-AS à l’approche originale [3], qui ne nécessite pas de calibration.

La comparaison des performances a été effectuée vis-à-vis de la capacité à accepter ou rejeter correctement des données hors du domaine du modèle, c’est-à-dire, menant à une perte \mathcal{L} supérieure à 10^{-4} . Nous calculons pour ce faire les métriques conventionnelles FPR@95 (False Positive Rate at 95% True Positive Rate), AUROC (Area Under the Receiver Operating Characteristic curve) et AUPR (Area Under the Precision-Recall curve).

La méthode BB-AS présente une amélioration significative par rapport à la méthode originale [3], qui rejette environ 42% des données situées dans le domaine du modèle dans notre cas de test. Les performances de la méthode BB-AS sont légèrement moins satisfaisantes que celles obtenues avec la méthode d’ensembles. Cependant, elle se distingue par le fait qu’elle ne nécessite que l’entraînement et l’inférence d’un seul modèle, le calcul du score d’anomalie ajoute une durée négligeable. La méthode d’ensemble nécessite l’entraînement puis l’inférence

2. Cette transformation était nécessaire car les valeurs de variance variaient sur des échelles différentes sur l’ensemble de test.

Méthode	FPR@95	AUROC	AUPR	Temps de calcul
Méthode de [3]	0.420	0.790	0.866	≈ 1
Méthode BB-AS	0.053	0.991	0.992	≈ 1
Méthode d'ensemble profond	0.025	0.996	0.996	8

TABLE 1 – Comparaison des performances entre les trois méthodes pour la détection de données hors domaine. Le temps de caclul est donné par référence à l'exécution d'un seul modèle neuronal.

de plusieurs modèles (huit dans notre cas). Cela ajoute à la fois une complexité supplémentaire au processus d'entraînement et une augmentation significative du temps de calcul à l'inférence, qui diminue les bénéfices apportés par l'utilisation d'un modèle de substitution à un code coûteux. Cet aspect est important à considérer pour notre cas d'utilisation.

3.3 Perspectives

Ce succès constitue un premier pas dans une approche à complexité croissante, et nous prévoyons d'étendre notre étude à des cas de dimensionnalité plus élevée à l'avenir. En continuant à explorer et à développer la méthode BB-AS, nous visons à adresser des problèmes plus complexes et à obtenir des résultats encore plus significatifs. Cette extension à des cas de plus grande dimensionnalité nous permettra d'évaluer la robustesse et la généralisation de notre approche dans des situations plus diversifiées et réalistes. En fin de compte, notre objectif est de proposer une solution innovante et efficace pour la détection des données hors domaine du modèle, tout en minimisant les exigences en termes de ressources et de temps de calcul.

Bibliographie

- [1] Alison Cozad, Nikolaos V. Sahinidis, and David C. Miller. Learning surrogate models for simulation-based optimization. *AICHE Journal*, 60(6) :2211–2227, 2014.
- [2] Joris Guerin, Kevin Delmas, Raul Ferreira, and Jérémie Guiochet. Out-of-distribution detection is not all you need. *AAAI Conference on Artificial Intelligence*, pages 14829–14837, 2023.
- [3] Thomas A. Henzinger, Anna Lukina, and Christian Schilling. Outside the box : Abstraction-based monitoring of neural networks. *CoRR*, abs/1911.09032, 2019.
- [4] C. Introïni, J. Sercombe, and Bo Sundman. Development of a robust, accurate and efficient coupling between pleiades/alcyone 2.1 fuel performance code and the openalphad thermo-chemical solver. *Nuclear Engineering and Design*, 369 :110818, 2020.
- [5] Clément Introïni, Jérôme Sercombe, Isabelle Ramière, and Romain Le Tellier. Phase-field modeling with the taf-id of incipient melting and oxygen transport in nuclear fuel during power transients. *Journal of Nuclear Materials*, 556 :153173, 2021.

- [6] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2) :251–257, 1991.
- [7] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles, 2017.
- [8] Bo Sundman, Ursula R Kattner, Mauro Palumbo, and Suzana G Fries. OpenCalphad-a free thermodynamic software. *Integrating Materials and Manufacturing Innovation*, 4(1) :1, 2015.
- [9] Laura von Rueden, Sebastian Mayer, Katharina Beckh, Bogdan Georgiev, Sven Giesselbach, Raoul Heese, Birgit Kirsch, Michal Walczak, Julius Pfrommer, Annika Pick, Rajkumar Ramamurthy, Jochen Garcke, Christian Bauckhage, and Jannis Schuecker. Informed machine learning - a taxonomy and survey of integrating prior knowledge into learning systems. *IEEE Transactions on Knowledge and Data Engineering*, page 1–1, 2021.