Accélération des Systèmes d'IA : Stratégies de Calcul Distribué et Parallélisation

Moez Krichen

Accélération des Systèmes d'IA : Stratégies de Calcul Distribué et Parallélisation

Moez Krichen

Laboratoire ReDCAD, Université de Sfax, Tunisie
moez.krichen@redcad.org

Résumé. L'évolution rapide des systèmes d'intelligence artificielle (IA) souligne la nécessité d'approches efficaces pour gérer la croissance exponentielle du volume de données et la complexité des modèles. Le calcul distribué et la parallélisation émergent comme des solutions clés, permettant une analyse et un traitement des données accélérés, ainsi qu'une amélioration de la vitesse d'entraînement et d'inférence des modèles d'IA. Ce document explore les principes, les architectures, et les avantages de ces approches, en mettant l'accent sur les techniques telles que le parallélisme des données et des modèles, les mécanismes de synchronisation, les systèmes de gestion de cluster, et l'exploitation des GPU et des accélérateurs. À travers cette analyse, nous démontrons comment le calcul distribué et la parallélisation peuvent surmonter les défis de scalabilité, favorisant ainsi l'avancement et l'efficacité des systèmes d'IA modernes.

1 Introduction

Le développement de l'apprentissage automatique (ML) et de l'intelligence artificielle (AI) a entraîné des changements significatifs dans plusieurs secteurs, menant à l'automatisation des emplois, à l'extraction de perspectives à partir de vastes ensembles de données et à la facilitation de processus de prise de décision avancés. L'utilisation des systèmes d'AI est de plus en plus répandue dans de nombreux domaines, englobant l'identification d'images, le traitement du langage naturel, les voitures autonomes et les recommandations personnalisées (60; 19). Néanmoins, la demande pour améliorer l'efficacité de ces systèmes d'AI a émergé comme un champ d'étude crucial, étant donné leur sophistication croissante et leurs besoins en ressources (41; 51).

Dans le domaine de l'intelligence artificielle, la capacité à traiter rapidement et efficacement de grandes quantités de données est un facteur clé qui détermine le succès et la faisabilité des modèles d'IA (28; 56; 4; 55; 13; 37; 53; 1; 25; 32; 5). Avec l'augmentation constante de la complexité des modèles et du volume des données, les chercheurs et les ingénieurs sont constamment à la recherche de méthodes pour accélérer le processus d'entraînement et d'inférence (47; 27; 38; 39; 22; 63; 6; 2; 12; 43; 18; 7; 36; 44). Le calcul distribué et la parallélisation se présentent comme des stratégies vitales pour relever ces défis, offrant une voie vers des systèmes d'IA plus rapides et plus scalables (48; 30; 31; 26; 24).

Le calcul distribué fait référence à l'utilisation de multiples ressources informatiques, souvent réparties géographiquement, pour travailler ensemble sur une tâche commune (50; 29; 40; 49; 3; 15; 14; 10; 20). Cette approche permet de diviser le travail en plusieurs parties plus petites, qui peuvent être exécutées simultanément sur différents nœuds, réduisant ainsi le temps total nécessaire pour atteindre le résultat. En parallèle, la parallélisation se concentre sur l'exploitation des architectures multicœurs et des accélérateurs matériels, tels que les GPU, pour exécuter simultanément de multiples opérations, accélérant encore davantage le traitement.

Ces méthodologies sont soutenues par des architectures systèmes sophistiquées et des mécanismes de synchronisation qui assurent une communication et une collaboration efficaces entre les nœuds de calcul. Le choix de l'architecture, qu'il s'agisse de modèles client-serveur, maître-esclave, pair-à-pair, ou de configurations hybrides, a un impact significatif sur la performance, la scalabilité et la tolérance aux pannes du système. De plus, le parallélisme des données et des modèles offre des moyens d'optimiser l'utilisation des ressources en distribuant les charges de travail de manière équilibrée et en réduisant les goulets d'étranglement.

La gestion efficace des clusters de calcul, grâce à des systèmes comme Apache Hadoop, Apache Spark, Kubernetes et Apache Mesos, joue un rôle crucial dans le déploiement à grande échelle des systèmes d'IA. Ces plateformes fournissent les outils nécessaires pour orchestrer les ressources de calcul, gérer les tâches, assurer la tolérance aux pannes et équilibrer les charges de travail, permettant ainsi aux applications d'IA de fonctionner de manière optimale dans des environnements distribués.

L'utilisation de GPU et d'autres accélérateurs, en parallèle avec des techniques de calcul de précision mixte, représente une autre dimension de la parallélisation, exploitant les capacités de calcul parallèle de ces dispositifs pour traiter efficacement les opérations d'IA. Cette approche est particulièrement pertinente pour les tâches intensives en calcul, comme l'entraînement de réseaux de neurones profonds, où elle peut réduire considérablement les temps d'entraînement.

La section suivante explore les principes, les méthodologies et les avantages du calcul distribué et de la parallélisation dans le cadre des systèmes d'IA (23). Avec la croissance continue du volume de données et l'augmentation de la complexité des modèles d'IA, l'utilisation du calcul distribué et de la parallélisation est apparue comme une approche viable pour relever les défis de scalabilité (57). Ces techniques permettent le traitement et l'analyse efficaces de grands ensembles de données et l'accélération des processus d'entraînement et d'inférence de l'IA. Grâce à l'utilisation de diverses ressources informatiques opérant en collaboration, ces méthodologies facilitent des calculs d'IA plus rapides et plus efficaces. Cette section se penche sur de multiples facettes du calcul distribué et de la parallélisation, incluant les architectures systèmes, le parallélisme des données, le parallélisme des modèles, les mécanismes de synchronisation, les systèmes de gestion de cluster, le parallélisme des GPU et des accélérateurs, et leurs applications dans l'entraînement et l'inférence distribués (61).

2 Architectures Systèmes

L'importance des conceptions de systèmes est cruciale pour permettre le calcul distribué et la parallélisation. Ces entités sont le cadre principal pour organiser et coordonner les ressources et les tâches distribuées. Diverses méthodologies architecturales, telles que client-serveur, maître-esclave, pair-à-pair et modèles hybrides, peuvent être utilisées dans la conceptualisation des systèmes distribués (59; 11; 8). Les cadres architecturaux sont responsables de l'établissement des protocoles et des méthodes qui fournissent la communication, la collaboration et l'échange de données et de tâches de calcul entre les nœuds informatiques. La détermination de l'architecture système repose sur plusieurs facteurs, y compris la scalabilité, la tolérance aux pannes, les schémas de communication et les besoins spéciaux inhérents à l'application d'IA.

3 Parallélisme des Données

Le parallélisme des données est une méthodologie computationnelle qui implique la réplication d'un modèle unique sur plusieurs nœuds informatiques (33; 54). Chaque nœud traite alors indépendamment un sous-ensemble unique des données disponibles en parallèle. Dans le calcul distribué, des nœuds distincts se voient attribuer des sous-ensembles spécialisés de données à des fins d'entraînement ou d'inférence. Ensuite, ces nœuds traitent individuellement les données qui leur sont attribuées, effectuant des calculs sur leurs sous-ensembles respectifs (9; 17). Après cela, les nœuds procèdent à l'échange de gradients ou de résultats pour permettre un traitement collaboratif. Le parallélisme des données est une stratégie hautement efficace, particulièrement lorsque l'ensemble de données peut être divisé en lots ou sous-ensembles plus petits pouvant être traités individuellement en parallèle sans interdépendances. Cette technologie permet un entraînement et une inférence distribués efficaces en partageant la charge de calcul parmi plusieurs nœuds, réduisant ainsi le temps global de traitement des données.

4 Parallélisme des Modèles

Le parallélisme des modèles est une technique qui implique la partition d'un modèle d'IA substantiel en composants plus petits et la distribution de ces composants sur plusieurs nœuds de traitement (42; 45). Chaque nœud individuel est chargé d'effectuer un certain sous-ensemble des tâches de calcul ou des couches dans le modèle. Le parallélisme des modèles est utilisé lorsque un nœud informatique unique manque de la mémoire ou des capacités de traitement nécessaires pour accueillir l'ensemble du modèle (62; 62). En partitionnant le modèle en composants plus petits et en les dispersant parmi des nœuds distincts, les demandes de calcul et de mémoire sont réparties, facilitant ainsi l'accommodation du modèle dans les ressources existantes. La mise en œuvre du parallélisme des modèles nécessite une coordination minutieuse et une communication efficace entre les nœuds pour garantir l'échange précis des résultats intermédiaires et la synchronisation du processus global.

5 Mécanismes de Synchronisation

Les méthodes de synchronisation sont cruciales pour coordonner le calcul et la communication parmi les nœuds dispersés dans les systèmes de calcul parallèle (58; 52). Ces procédures assurent la synchronisation appropriée des nœuds tout au long des processus d'entraînement et d'inférence, maintenant ainsi la cohérence et la précision. Des points de synchronisation sont mis en place pour faciliter l'alignement des calculs et permettre le flux d'informations entre les nœuds, incluant mais sans s'y limiter, les gradients, les paramètres du modèle et les résultats intermédiaires. Les méthodes couramment employées pour synchroniser les nœuds distribués et assurer des résultats cohérents comprennent la synchronisation par barrière, la moyenne des paramètres et l'agrégation des gradients. La mise en œuvre d'algorithmes de synchronisation efficaces est d'une importance capitale pour atteindre une performance optimale dans les systèmes de calcul parallèle tout en atténuant les risques associés aux situations de course et aux incohérences de données.

6 Systèmes de Gestion de Cluster

Les systèmes de gestion de cluster offrent une infrastructure complète et des outils qui simplifient le déploiement et la gestion des systèmes d'IA distribués (16; 46; 35). Ces systèmes gèrent l'allocation des ressources, planifient les travaux, assurent la tolérance aux pannes et équilibrent la charge de travail dans un environnement distribué. Les systèmes de gestion de cluster ont pour but d'abstraire l'infrastructure matérielle sous-jacente, offrant ainsi une interface consolidée pour gérer efficacement les ressources distribuées. Cette abstraction permet aux utilisateurs de se concentrer sur le développement et l'exécution des applications d'IA. Des exemples notables de systèmes de gestion de cluster incluent Apache Hadoop, Apache Spark, Kubernetes et Apache Mesos. Ces systèmes peuvent être mis à l'échelle, tolérer les pannes et montrer de la flexibilité, les rendant bien adaptés pour mener de vastes calculs d'IA distribués.

7 Parallélisme des GPU et des Accélérateurs

Les approches de parallélisation englobent plus que le calcul distribué ; elles comprennent également l'utilisation des GPU et d'autres accélérateurs pour exploiter les capacités de traitement (34). Les GPU et les accélérateurs matériels spécialisés, tels que les TPUs, possèdent d'importantes capacités de calcul parallèle qui ont le potentiel d'améliorer grandement la vitesse des opérations d'IA. En déléguant des opérations informatiquement exigeantes, telles que les multiplications matricielles ou les convolutions, aux GPU ou aux accélérateurs spécialisés, les modèles d'IA peuvent atteindre des améliorations significatives de performance. Des méthodes telles que le parallélisme des GPU et le calcul de précision mixte facilitent l'utilisation optimale des ressources matérielles, améliorant ainsi la vitesse des processus d'entraînement et d'inférence.

8 Entraînement et Inférence Distribués

Les techniques de calcul distribué et de parallélisation peuvent être efficacement employées dans les activités d'entraînement et d'inférence des modèles d'IA (21). L'entraînement distribué implique la répartition de la charge de calcul sur de nombreux nœuds, résultant en une convergence accélérée et une réduction de la durée d'entraînement. L'entraînement distribué s'avère avantageux dans les scénarios impliquant d'importantes ressources informatiques, telles que de grands ensembles de données ou des modèles complexes. Dans l'inférence distribuée, la répartition de la charge de travail parmi de nombreux nœuds permet un traitement accéléré et amélioré des demandes d'inférence. L'inférence distribuée joue un rôle crucial dans les applications d'IA nécessitant des capacités en temps réel ou à haut débit, privilégiant la faible latence et la scalabilité comme facteurs clés. L'utilisation du calcul distribué et de la parallélisation permet l'accélération des activités d'entraînement et d'inférence, facilitant ainsi la mise en œuvre des systèmes d'IA à grande échelle.

Le sujet englobe plusieurs domaines, incluant les Architectures Systèmes, le Parallélisme des Données, le Parallélisme des Modèles, les Mécanismes de Synchronisation, les Systèmes de Gestion de Cluster, le Parallélisme des GPU et des Accélérateurs, et l'Entraînement et l'Inférence Distribués. Ces techniques facilitent le traitement efficace, la scalabilité et l'accélération des systèmes d'IA en concevant des systèmes distribués, en répliquant des modèles, en partitionnant des tâches, en mettant en œuvre la synchronisation, en gérant des clusters, et en utilisant des GPU et des accélérateurs.

9 Conclusion

Le calcul distribué et la parallélisation sont des piliers essentiels pour surmonter les défis posés par la croissance explosive des volumes de données et la complexité des modèles dans le domaine de l'intelligence artificielle. En exploitant des architectures systèmes avancées, en optimisant le parallélisme des données et des modèles, et en utilisant des mécanismes de synchronisation efficaces, ces approches permettent une scalabilité, une performance et une efficacité accrues des systèmes d'IA. L'adoption de systèmes de gestion de cluster sophistiqués et l'utilisation stratégique des GPU et des accélérateurs matériel spécialisés ouvrent la voie à des avancées significatives dans l'entraînement et l'inférence des modèles d'IA, rendant les technologies d'IA plus accessibles et viables pour une gamme étendue d'applications. Alors que nous continuons à pousser les frontières de ce que l'IA peut accomplir, le calcul distribué et la parallélisation resteront au cœur des efforts pour rendre ces systèmes plus rapides, plus intelligents et plus efficaces.

Références

[1] Q Abu Al-Haija and M Krichen. A lightweight in-vehicle alcohol detection using smart sensing and supervised learning. computers 2022, 11, 121, 2022.

[2] Qasem Abu Al-Haija, Moez Krichen, and Wejdan Abu Elhaija. Machine-learning-based darknet traffic detection system for iot applications. *Electronics*, 11(4) :556, 2022.

[3] Qasem Abu Al-Haija and Moez Krichen. Analyzing malware from api call sequences using support vector machines. In *International Conference on Cybersecurity, Cybercrimes, and Smart Emerging Technologies*, pages 27–39. Springer International Publishing Cham, 2022.

[4] Omar Azib Alkhudaydi, Moez Krichen, and Ans D Alghamdi. A deep learning methodology for predicting cybersecurity attacks on the internet of things. *Information*, 14(10) :550, 2023.

[5] Hamoud Alshammari, Karim Gasmi, Ibtihel Ben Ltaifa, Moez Krichen, Lassaad Ben Ammar, and Mahmood A Mahmood. Olive disease classification based on vision transformer and cnn models. *Computational Intelligence and Neuroscience*, 2022, 2022.

[6] Hamoud Alshammari, Karim Gasmi, Moez Krichen, Lassaad Ben Ammar, Mohamed Osman Abdelhadi, Ammar Boukrara, and Mahmood A Mahmood. Optimal deep learning model for olive disease diagnosis based on an adaptive genetic algorithm. *Wireless Communications and Mobile Computing*, 2022 :1–13, 2022.

[7] Hashem Alyami, Wael Alosaimi, Moez Krichen, and Roobaea Alroobaea. Monitoring social distancing using artificial intelligence for fighting covid-19 virus spread. *International Journal of Open Source Software and Processes (IJOSSP)*, 12(3) :48–63, 2021.

[8] Sarina Aminizadeh, Arash Heidari, Shiva Toumaj, Mehdi Darbandi, Nima Jafari Navimipour, Mahsa Rezaei, Samira Talebi, Poupak Azad, and Mehmet Unal. The applications of machine learning techniques in medical data processing based on distributed computing and the internet of things. *Computer Methods and Programs in Biomedicine*, page 107745, 2023.

[9] Ignacio Arnaldo, Kalyan Veeramachaneni, Andrew Song, and Una-May O'Reilly. Bring your own learner : A cloud-based, data-parallel commons for machine learning. *IEEE Computational Intelligence Magazine*, 10(1) :20–32, 2015.

[10] Rubby Aworka, Lontsi Saadio Cedric, Wilfried Yves Hamilton Adoni, Jérémie Thouakesseh Zoueu, Franck Kalala Mutombo, Charles Lebon Mberi Kimpolo, Tarik Nahhal, and Moez Krichen. Agricultural decision system based on advanced machine learning models for yield prediction : Case of east african countries. *Smart Agricultural Technology*, 2 :100048, 2022.

[11] Ron Bekkerman, Mikhail Bilenko, and John Langford. *Scaling up machine learning : Parallel and distributed approaches*. Cambridge University Press, 2011.

[12] Wadii Boulila, Maha Driss, Eman Alshanqiti, Mohamed Al-Sarem, Faisal Saeed, and Moez Krichen. Weight initialization techniques for deep learning algorithms in remote sensing : Recent trends and future perspectives. *Advances on Smart and Soft Computing : Proceedings of ICACIn 2021*, pages 477–484, 2022.

[13] Zakaria Boulouard, Mariyam Ouaissa, Mariya Ouaissa, Farhan Siddiqui, Mutiq Almutiq, and Moez Krichen. An integrated artificial intelligence of things envi-

ronment for river flood prevention. *Sensors*, 22(23) :9485, 2022.

[14] Lontsi Saadio Cedric, Wilfried Yves Hamilton Adoni, Rubby Aworka, Jérémie Thouakesseh Zoueu, Franck Kalala Mutombo, Moez Krichen, and Charles Lebon Mberi Kimpolo. Crops yield prediction based on machine learning models : Case of west african countries. *Smart Agricultural Technology*, 2 :100049, 2022.

[15] Oumaima Chakir, Abdeslam Rehaimi, Yassine Sadqi, Moez Krichen, Gurjot Singh Gaba, Andrei Gurtov, et al. An empirical assessment of ensemble methods and traditional machine learning techniques for web-based attack detection in industry 5.0. *Journal of King Saud University-Computer and Information Sciences*, 35(3) :103–119, 2023.

[16] Christina Delimitrou and Christos Kozyrakis. Quasar : Resource-efficient and qos-aware cluster management. *ACM SIGPLAN Notices*, 49(4) :127–144, 2014.

[17] Nikoli Dryden, Tim Moon, Sam Ade Jacobs, and Brian Van Essen. Communication quantization for data-parallel training of deep neural networks. In *2016 2nd Workshop on Machine Learning in HPC Environments (MLHPC)*, pages 1–8. IEEE, 2016.

[18] Mourad Ellouze, Seifeddine Mechti, Moez Krichen, Vinayakumar Ravi, and Lamia Hadrich Belguith. A deep learning approach for detecting the behaviour of people having personality disorders towards covid-19 from twitter. *International Journal of Computational Science and Engineering*, 25(4) :353–366, 2022.

[19] Michael Haenlein and Andreas Kaplan. A brief history of artificial intelligence : On the past, present, and future of artificial intelligence. *California management review*, 61(4) :5–14, 2019.

[20] Olfa Hrizi, Karim Gasmi, Ibtihel Ben Ltaifa, Hamoud Alshammari, Hanen Karamti, Moez Krichen, Lassaad Ben Ammar, and Mahmood A Mahmood. Tuberculosis disease diagnosis based on an optimized machine learning model. *Journal of Healthcare Engineering*, 2022, 2022.

[21] Yanping Huang, Youlong Cheng, Ankur Bapna, Orhan Firat, Dehao Chen, Mia Chen, HyoukJoong Lee, Jiquan Ngiam, Quoc V Le, Yonghui Wu, et al. Gpipe : Efficient training of giant neural networks using pipeline parallelism. *Advances in neural information processing systems*, 32, 2019.

[22] Hajra Khan, Imran Fareed Nizami, Saeed Mian Qaisar, Asad Waqar, Moez Krichen, and Abdulaziz Turki Almaktoom. Analyzing optimal battery sizing in microgrids based on the feature selection and machine learning approaches. *Energies*, 15(21) :7865, 2022.

[23] Jin Kyu Kim, Qirong Ho, Seunghak Lee, Xun Zheng, Wei Dai, Garth A Gibson, and Eric P Xing. Strads : A distributed framework for scheduled model parallel machine learning. In *Proceedings of the Eleventh European Conference on Computer Systems*, pages 1–16, 2016.

[24] Moez Krichen. How artificial intelligence can revolutionize software testing techniques. In *International Conference on Innovations in Bio-Inspired Computing and Applications*, pages 189–198. Springer Nature Switzerland Cham, 2022.

[25] Moez Krichen. Les méthodes formelles sont-elles applicables à l'apprentissage automatique et à l'intelligence artificielle. 2022.

[26] Moez Krichen. Comment l'intelligence artificielle peut révolutionner les techniques de test de logiciels. 2023.

[27] Moez Krichen. Convolutional neural networks : A survey. *Computers*, 12(8) :151, 2023.

[28] Moez Krichen. Deep reinforcement learning. In *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pages 1–7. IEEE, 2023.

[29] Moez Krichen. Generative adversarial networks. In *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pages 1–7. IEEE, 2023.

[30] Moez Krichen. Renforcer la sécurité des contrats intelligents grâce à la puissance de l'intelligence artificielle. 2023.

[31] Moez Krichen. Strengthening the security of smart contracts through the power of artificial intelligence. *Computers*, 12(5) :107, 2023.

[32] Moez Krichen, Alaeddine Mihoub, Mohammed Y Alzahrani, Wilfried Yves Hamilton Adoni, and Tarik Nahhal. Are formal methods applicable to machine learning and artificial intelligence ? In *2022 2nd International Conference of Smart Systems and Emerging Technologies (SMARTTECH)*, pages 48–53. IEEE, 2022.

[33] Hao Li, Asim Kadav, Erik Kruus, and Cristian Ungureanu. Malt : distributed data-parallelism for existing ml applications. In *Proceedings of the tenth european conference on computer systems*, pages 1–16, 2015.

[34] Peilong Li, Yan Luo, Ning Zhang, and Yu Cao. Heterospark : A heterogeneous cpu/gpu spark platform for machine learning algorithms. In *2015 IEEE International Conference on Networking, Architecture and Storage (NAS)*, pages 347–348. IEEE, 2015.

[35] Darius Malysiak and Uwe Handmann. An efficient framework for distributed computing in heterogeneous beowulf clusters and cluster-management. In *2014 IEEE 15th International Symposium on Computational Intelligence and Informatics (CINTI)*, pages 169–178. IEEE, 2014.

[36] Seifeddine Mechti, Moez Krichen, Dhouha Ben Noureddine, and Lamia H Belguith. A decision system for computational authors profiling : From machine learning to deep learning. *Concurrency and Computation : Practice and Experience*, 34(7) :e5985, 2022.

[37] Saeed Mian Qaisar, Nehal Alyamani, Asad Waqar, and Moez Krichen. Machine learning with adaptive rate processing for power quality disturbances identification. *SN Computer Science*, 3 :1–6, 2022.

[38] Saeed Mian Qaisar, Dalila Say, Salah Zidi, and Krichen Moez. Automated categorization of multiclass welding defects using the x-ray image augmentation and convolutional neural network. 2023.

[39] Alaeddine Mihoub, Moez Krichen, Mohannad Alswailim, Sami Mahfoudhi, and

Riadh Bel Hadj Salah. Road scanner : A road state scanning approach based on machine learning techniques. *Applied Sciences*, 13(2) :683, 2023.

[40] Alaeddine Mihoub, Hosni Snoun, Moez Krichen, Riadh Bel Hadj Salah, and Montassar Kahia. Predicting covid-19 spread level using socio-economic indicators and machine learning techniques. In *2020 first international conference of smart systems and emerging technologies (SMARTTECH)*, pages 128–133. IEEE, 2020.

[41] Madison Milne-Ives, Caroline de Cock, Ernest Lim, Melissa Harper Shehadeh, Nick de Pennington, Guy Mole, Eduardo Normando, and Edward Meinert. The effectiveness of artificial intelligence conversational agents in health care : systematic review. *Journal of medical Internet research*, 22(10) :e20346, 2020.

[42] Sergio Moreno-Alvarez, Juan M Haut, Mercedes E Paoletti, and Juan A Rico-Gallego. Heterogeneous model parallelism for deep neural networks. *Neurocomputing*, 441 :1–12, 2021.

[43] Pierre Stanislas Birame Ndong, Wilfried Yves Hamilton Adoni, Tarik Nahhal, Charles Kimpolo, Moez Krichen, Abdeltif EL Byed, Ismail Assayad, and Franck Kalala Mutombo. A face-mask detection system based on deep learning convolutional neural networks. In *Advances on Smart and Soft Computing : Proceedings of ICACIn 2021*, pages 273–283. Springer Singapore Singapore, 2021.

[44] Dhouha Ben Noureddine, Moez Krichen, Seifeddine Mechti, Tarik Nahhal, and Wilfried Yves Hamilton Adoni. An agent-based architecture using deep reinforcement learning for the intelligent internet of things applications. In *Advances on Smart and Soft Computing : Proceedings of ICACIn 2020*, pages 273–283. Springer Singapore, 2021.

[45] Jay H Park, Gyeongchan Yun, M Yi Chang, Nguyen T Nguyen, Seungmin Lee, Jaesik Choi, Sam H Noh, and Young-ri Choi. {HetPipe} : Enabling large {DNN} training on (whimpy) heterogeneous {GPU} clusters through integration of pipelined model parallelism and data parallelism. In *2020 USENIX Annual Technical Conference (USENIX ATC 20)*, pages 307–321, 2020.

[46] Ju-Won Park and Jaegyoon Hahm. Container-based cluster management platform for distributed computing. In *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA)*, page 34. The Steering Committee of The World Congress in Computer Science, Computer ..., 2015.

[47] Saeed Mian Qaisar, Alaeddine Mihoub, Moez Krichen, and Humaira Nisar. Multirate processing with selective subbands and machine learning for efficient arrhythmia classification. *Sensors*, 21(4) :1511, 2021.

[48] Shalli Rani, Ali Kashif Bashir, Moez Krichen, Abdulaziz Alshammari, et al. A low-rank learning based multi-label security solution for industry 5.0 consumers using machine learning classifiers. *IEEE Transactions on Consumer Electronics*, 2023.

[49] Shashidhar Rudregowda, Sudarshan Patil Kulkarni, Gururaj HL, Vinayakumar Ravi, and Moez Krichen. Visual speech recognition for kannada language using vgg16 convolutional neural network. In *Acoustics*, volume 5, pages 343–353. MDPI,

2023.

[50] Dalila Say, Salah Zidi, Saeed Mian Qaisar, and Moez Krichen. Automated categorization of multiclass welding defects using the x-ray image augmentation and convolutional neural network. *Sensors*, 23(14) :6422, 2023.

[51] Terrence J Sejnowski. The unreasonable effectiveness of deep learning in artificial intelligence. *Proceedings of the National Academy of Sciences*, 117(48) :30033–30038, 2020.

[52] Amir Sepehr, Oriol Gomis-Bellmunt, and Edris Pouresmaeil. Employing machine learning for enhancing transient stability of power synchronization control during fault conditions in weak grids. *IEEE Transactions on Smart Grid*, 13(3) :2121–2131, 2022.

[53] Souhir Sghaier, Moez Krichen, Abir Othman Elfaki, and Qasem Abu Al-Haija. Efficient machine-learning based 3d face identification system under large pose variation. In *International Conference on Computational Collective Intelligence*, pages 273–285. Springer International Publishing Cham, 2022.

[54] Christopher J Shallue, Jaehoon Lee, Joseph Antognini, Jascha Sohl-Dickstein, Roy Frostig, and George E Dahl. Measuring the effects of data parallelism on neural network training. *arXiv preprint arXiv :1811.03600*, 2018.

[55] R Shashidhar, S Patilkulkarni, Vinayakumar Ravi, HL Gururaj, and Moez Krichen. Audiovisual speech recognition based on a deep convolutional neural network. *Data Science and Management*, 2023.

[56] Wiem Souai, Alaeddine Mihoub, Mounira Tarhouni, Salah Zidi, Moez Krichen, and Sami Mahfoudhi. Predicting at-risk students using the deep learning blstm approach. In *2022 2nd International Conference of Smart Systems and Emerging Technologies (SMARTTECH)*, pages 32–37. IEEE, 2022.

[57] Sujatha R Upadhyaya. Parallel approaches to machine learning—a comprehensive survey. *Journal of Parallel and Distributed Computing*, 73(3) :284–292, 2013.

[58] Tongfeng Weng, Xiaolu Chen, Zhuoming Ren, Huijie Yang, Jie Zhang, and Michael Small. Synchronization of machine learning oscillators in complex networks. *Information Sciences*, 630 :74–81, 2023.

[59] Hang Xu, Chen-Yu Ho, Ahmed M Abdelmoniem, Aritra Dutta, El Houcine Bergou, Konstantinos Karatsenidis, Marco Canini, and Panos Kalnis. Grace : A compressed communication framework for distributed machine learning. In *2021 IEEE 41st international conference on distributed computing systems (ICDCS)*, pages 561–572. IEEE, 2021.

[60] Caiming Zhang and Yang Lu. Study on artificial intelligence : The state of the art and future prospects. *Journal of Industrial Information Integration*, 23 :100224, 2021.

[61] Jilin Zhang, Hangdi Tu, Yongjian Ren, Jian Wan, Li Zhou, Mingwei Li, and Jue Wang. An adaptive synchronous parallel strategy for distributed machine learning. *IEEE Access*, 6 :19222–19230, 2018.

[62] Yonghao Zhuang, Lianmin Zheng, Zhuohan Li, Eric Xing, Qirong Ho, Joseph Gon-

zalez, Ion Stoica, Hao Zhang, and Hexu Zhao. On optimizing the communication of model parallelism. *Proceedings of Machine Learning and Systems*, 5, 2023.

[63] Salah Zidi, Alaeddine Mihoub, Saeed Mian Qaisar, Moez Krichen, and Qasem Abu Al-Haija. Theft detection dataset for benchmarking and machine learning based classification in a smart grid environment. *Journal of King Saud University-Computer and Information Sciences*, 35(1) :13–25, 2023.