UNIVERSITY OF ICELAND

Design and Evaluation of Parallel and Scalable Machine Learning Research in Biomedical Modelling Applications

Chadi Barakat

2023

FACULTY OF INDUSTRIAL ENGINEERING, MECHANICAL ENGINEERING AND COMPUTER SCIENCE

Design and Evaluation of Parallel and Scalable Machine Learning Research in Biomedical Modelling Applications

Chadi Barakat

Dissertation submitted in partial fulfillment of a
Philosophiae Doctor degree in Bioengineering

Supervisor
Prof. Dr.-Ing. Morris Riedel

Doctoral Committee
Prof. Dr.-Ing. Morris Riedel
Prof. Sigurður Brynjólfsson
Dr. med. M.Sc. Sebastian Fritsch
Prof. Dr. rer. nat. Andreas Schuppert

Opponents
Dr. Róbert Lovas
Dr. med. M.Sc. Maximilian Franz Schulze-Hagen

Faculty of Industrial Engineering, Mechanical Engineering and
Computer Science
School of Engineering and Natural Sciences
University of Iceland
Reykjavík, June 2023

Design and Evaluation of Parallel and Scalable Machine Learning Research in Biomedical Modelling Applications
(Parallel Machine Learning for Biomedical Modelling)

Dissertation submitted in partial fulfillment of a *Philosophiae Doctor* degree in Bioengineering

Faculty of Industrial Engineering, Mechanical Engineering and Computer Science
School of Engineering and Natural Sciences
University of Iceland
Tæknigarður - Dunhagi 5
107 Reykjavík
Iceland

Telephone: 525-4700

Bibliographic information:
Chadi Barakat (2023) *Design and Evaluation of Parallel and Scalable Machine Learning Research in Biomedical Modelling Applications*, PhD dissertation, Faculty of Industrial Engineering, Mechanical Engineering and Computer Science, University of Iceland, 104 pp.

ISBN 978-9935-9697-9-8

Copyright © 2023 Chadi Barakat
All rights reserved

Printing: Háskólaprent ehf.
Reykjavík, Iceland, June 2023

Abstract

The use of Machine Learning (ML) techniques in the medical field is not a new occurrence and several papers describing research in that direction have been published. This research has helped in analysing medical images, creating responsive cardiovascular models, and predicting outcomes for medical conditions among many other applications. This Ph.D. aims to apply such ML techniques for the analysis of Acute Respiratory Distress Syndrome (ARDS) which is a severe condition that affects around 1 in 10.000 patients worldwide every year with life-threatening consequences. We employ previously developed mechanistic modelling approaches such as the "Nottingham Physiological Simulator," through which better understanding of ARDS progression can be gleaned, and take advantage of the growing volume of medical datasets available for research (i.e., "big data") and the advances in ML to develop, train, and optimise the modelling approaches. Additionally, the onset of the COVID-19 pandemic while this Ph.D. research was ongoing provided a similar application field to ARDS, and made further ML research in medical diagnosis applications possible. Finally, we leverage the available Modular Supercomputing Architecture (MSA) developed as part of the Dynamical Exascale Entry Platform - Extreme Scale Technologies (DEEP-EST) EU Project to scale up and speed up the modelling processes. This Ph.D. Project is one element of the Smart Medical Information Technology for Healthcare (SMITH) project wherein the thesis research can be validated by clinical and medical experts (e.g. Uniklinik RWTH Aachen).

Útdráttur

Notkun vélnámsaðferða (ML) í læknavísindum er ekki ný af nálinni og hafa nokkrar greinar verið birtar um rannsóknir á því sviði. Þessar rannsóknir hafa hjálpað til við að greina læknisfræðilegar myndir, búa til svörunarlíkön fyrir hjarta- og æðakerfi og spá fyrir um útkomu sjúkdóma meðal margra annarra notkunarmöguleika. Markmið þessarar doktorsrannsóknar er að beita slíkum ML aðferðum við greiningu á bráðu andnauðarheilkenni (ARDS), alvarlegan sjúkdóm sem hrjáir um 1 af hverjum 10.000 sjúklingum á heimsvísu á ári hverju með lífshættulegum afleiðingum. Til að framkvæma þessa greiningu notum við áður þróaðar aðferðir við líkanasmíði, s.s. „Nottingham Physiological Simulator", sem nota má til að auka skilning á framvindu ARDS-sjúkdómsins. Við nýtum okkur vaxandi umfang læknisfræðilegra gagnasafna sem eru aðgengileg til rannsókna (þ.e. „stórgögn"), framfarir í vélnámi til að þróa, þjálfa og besta líkanaaðferðirnar. Þar að auki hófst COVID-19 faraldurinn þegar doktorsrannsóknin var í vinnslu, sem setti svipað svið fram og ARDS og gerði frekari rannsóknir á ML í læknisfræði mögulegar. Einnig nýtum við tiltæka einingaskipta högun ofurtölva, „Modular Supercomputing Architecture" (MSA), sem er þróuð sem hluti af „Dynamical Exascale Entry Platform" - Extreme Scale Technologies (DEEP-EST) verkefnisáætlun ESB til að kvarða og hraða líkanasmíðinni. Þetta doktorsverkefni er einn þáttur í SMITH-verkefninu (e. Smart Medical Information Technology for Healthcare) þar sem sérfræðingar í klíník og læknisfræði geta staðfest rannsóknina (t.d. Uniklinik RWTH Aachen).

*To my family for their unwavering support,
to my friends for keeping me grounded,
to my past self for his stubbornness,
and to my future self as a reminder
that even the toughest challenges
can be overcome.*

Contents

List of Figures

List of Tables

List of Original Publications

Paper I: **C. Barakat**, S. Fritsch, M. Riedel, S. Brynjólfsson, *An HPC-Driven Data Science Platform to Speed-up Time Series Data Analysis of Patients with the Acute Respiratory Distress Syndrome*, `https://ieeexplore.ieee.org/document/9596840/`, 2021

Paper II: **C. Barakat**, M. Riedel, S. Brynjólfsson, G. Cavallaro, J. Busch, R. Sedona, *Design and Evaluation of an HPC-based Expert System to speed-up Retail Data Analysis using Residual Networks Combined with Parallel Association Rule Mining and Scalable Recommenders*, `https://ieeexplore.ieee.org/document/9596796/`, 2021

Paper III: **C. Barakat**, S. Fritsch, K. Sharafutdinov, G. Ingólfsson, A. Schuppert, S. Brynjólfsson, M. Riedel, *Lessons learned on using High-Performance Computing and Data Science Methods towards understanding the Acute Respiratory Distress Syndrome (ARDS)*, `https://ieeexplore.ieee.org/document/9803320/`, 2022

Paper IV: **C. Barakat**, M. Aach, A. Schuppert, S. Brynjólfsson, S. Fritsch, M. Riedel *Analysis of Chest X-Ray for COVID-19 Diagnosis as a Use Case for an HPC-enabled Data Analysis and Machine Learning Platform for Medical Diagnosis Support*, `https://www.mdpi.com/2085284`, 2023

Paper V: **C. Barakat**, K. Sharafutdinov, J. Busch, S. Saffaran, D.G. Bates, J.G. Hardman, A. Schuppert, S. Brynjólfsson, S. Fritsch, M. Riedel *Developing an Artificial Intelligence-Based Representation of a Virtual Patient Model for Real-Time Diagnosis of Acute Respiratory Distress Syndrome*, Pending publication, 2023

Other Publications

Paper I: M. Riedel, R. Sedona, **C. Barakat**, P. Einarsson, R. Hassanian, G. Cavallaro, M. Book, H. Neukirchen, A. Lintermann, *Practice and Experience in using Parallel and Scalable Machine Learning with Heterogenous Modular Supercomputing Architectures*, `https://ieeexplore.ieee.org/document/9460702/`, 2021

Paper II: G. Marx, J. Bickenbach, S. Fritsch, J. Kunze, O. Maassen, S. Deffge, J. Kistermann, S. Haferkamp, I. Lutz, N. Voellm, V. Lowitsch, R. Polzin, K. Sharafutdinov, H. Mayer, L. Kuepfer, R. Burghaus, W. Schmitt, J. Lippert, M. Riedel, **C. Barakat**, A. Stollenwerk, S. Fonck, C. Putensen, S. Zenker, F. Erdfelder, D. Grigutsch, R. Kram, S. Beyer, K. Kampe, J. Gewehr,F. Salman, P. Juers, S. Kluge, D. Tiller, E. Wisotzki, S. Gross, L. Homeister, F. Bloos, A. Scherag, D. Ammon, S. Mueller, J. Palm, P. Simon, N. Jahn, M. Loeffler, T. Wendt, T. Schuerholz, P. Groeber, A. Schuppert, *Algorithmic surveillance of I CU patients with acute respiratory distress syndrome (ASIC): protocol for a multicentre stepped-wedge cluster randomised quality improvement strategy*, `https://bmjopen.bmj.com/content/11/4/e045589`, 2021

Paper III: M. Riedel, **C. Barakat**, S. Fritsch, M. Aach, J. Busch, A. Lintermann, A. Schuppert, S. Brynjólfsson, H. Neukirchen, M. Book *Enabling Hyperparameter-Tuning of AI Models for Healthcare using the CoE RAISE Unique AI Framework for HPC*, 2023

Abbreviations

AI	artificial intelligence
ANN	artificial neural network
API	application programming interface
ARDS	acute respiratory distress syndrome
ASIC	algorithmic surveillance of intensive care unit patients
BMBF	Federal Ministry of Education and Research
CASP	critical assessment of structure prediction
CC	cloud computing
CNN	convolutional neural network
CRISP-DM	cross industry standard process model for data mining
CXR	chest x-ray
DEEP	dynamic exascale entry platform
DL	deep learning
ECMO	extracorporeal membrane oxygenation
EHL	e*HealthLine
EHR	electronic health record
FIFO	first-in, first-out
F_iO_2	fraction of inspired oxygen
FLOPS	floating point operations per second
HPC	high-performance computing

GPU	graphical processing unit
GRU	gated recurrent unit
HTC	high throughput computing
ICU	intensive care unit
JSC	Jülich Supercomputing Centre
JUWELS	Jülich wizard for European leadership science
MAE	mean absolute error
MIMD	multiple instruction - multiple data
MIMIC-III	medical information mart for intensive care - III
ML	machine learning
MSA	modular supercomputing architecture
MSE	mean squared error
NLP	natural language processing
NPS	Nottingham physiology simulator
$\mathbf{P_aO_2}$	partial pressure of arterial oxygen
$\mathbf{P_aCO_2}$	partial pressure of arterial carbon dioxide
PBT	population-based training
PEPX	projection-expansion-projection-extension
P/F ratio	ratio of partial pressure of arterial oxygen (P_aO_2) to fraction of inspired oxygen (F_iO_2)
RNN	Recurrent Neural Network
RT-PCR	reverse-transcription polymerase chain-reaction
$\mathbf{R_{comp}}$	compartment resistance to flow
RL	reinforcement learning
SARS-CoV-2	severe acute respiratory syndrome coronavirus 2
SIMD	single instruction - multiple data
SL	supervised learning
SMITH	Smart Medical Information Technology for Healthcare
SVM	support vector machine

S_vO_2	venous oxygen blood saturation
TO	thesis objective
TPU	tensor processing unit
UL	unsupervised learning
VR_{comp}	compartment vascular resistance

Acknowledgments

Prof. Morris Riedel has been a source of support and guidance for the last four years, both at the University of Iceland and at the Jülich Supercomputing Centre, without which I could not have arrived at this point, and for that I owe him a debt of gratitude. I am also equally grateful for the support offered by the members of my doctoral committee, Prof. Sigurður Brynjólfsson, Prof. Andreas Schuppert, and Dr. Sebastian Fritsch. I thank the University of Iceland in general, and the Faculty of Industrial Engineering, Mechanical Engineering and Computer Science in specific, for the opportunity and the experience of completing this Ph.D.

The last four years working at the Jülich Supercomputing Centre (JSC), Dr. Gabriele Cavallaro has been a constant calming presence, Dr. Rocco Sedona has been a source of excellent wine and equally excellent camaraderie, and Josefine Busch has been a wonderful colleague and an absolute joy to work with, and I cannot overstate how thankful I am to them especially, but also to all the members of the *High-Productivity Data Processing* team. Similarly, the team members at the *Joint Research Centre for Computational Biomedicine* (JRC-Combine) have been extremely supportive and even though the end of the SMITH project signals the end of our collaboration, I am looking forward to future work with all of them. I thank them for coffee breaks and international cakes, and also for science days, journal clubs, and Monday morning jour fixe. Thanks go especially to Konstantin Sharafutdinov and Richard Polzin for being my partners in ASIC and in bowling and in coffee.

To Anna, Clara, Filip, and Nikola I am grateful for being a source of chaos, fun, and cosy dinners. To Jorge and Nina I am grateful for boardgames, dinners, and cocktails. Thank you Lisa for giving me puzzles to solve but also more reasons to dislike MATLAB, and Ariel for introducing me to the concept of Café-Office. Thank you Maëllia for setting the bar very high on amazing house parties, Ralph for reminders to take a break, and Reine for giving support when I needed it and for receiving it when I could give it. To Sam and Joy and Kai, thank you for being a welcome distraction and a reminder that life exists outside of the Ph.D.

Finally, my family have been an endless source of comfort and support, keeping me sane and grounded through both the Masters and the Ph.D. Mom, Dad, Samer, Tarek, Kalina, Mary, and Lucy, I love you all; this would not have been possible without you.

In loving memory of Joseph and Pierre Abdelnour.

Chapter 1

Introduction

1.1 Motivation

The onset of respiratory disease can be caused either through direct injury to the lungs (i.e. trauma, infection, toxic inhalations, etc.) or indirectly through the failure of other organ systems [1]. Though the causes may differ, the outcome is one: a reduced ability of the body to perform the gas exchanges necessary to maintain homeostasis, and thus a major threat to the affected patient's life. It has been shown that early detection of the symptoms of respiratory disease leads, and accordingly early treatment, is generally associated with positive outcomes for patients, which is especially true for intensive care unit (ICU) patients who are often at greater risk of developing these conditions than the general population [2, 3]. One common condition that affects a large number of ICU patients is acute respiratory distress syndrome (ARDS) [4]. Much research has been done in order to better understand the mechanisms through which ARDS is manifested and the treatment methods that produce the best outcomes, with no clear consensus [5, 6]. The "Berlin Definition" set down a number of markers for ARDS diagnosis and severity stratification and has been widely adopted in ICUs, paving the way for improvements in treatment [7].

On the other hand, as the technology to collect, store, and manage medical information becomes more accessible, large repositories of medical data become more commonplace, setting up the groundwork for "Medical Big Data" enabling large-scale research but also sparking discussions of patient privacy [8, 9, 10]. The availability of data does however provide the opportunity to gain knowledge concerning medical conditions and share it between different institutions and populations, allowing medical personnel to develop novel treatment methods and to discover underlying patterns that would have otherwise been difficult or impossible to detect.

Moreover, as the COVID-19 pandemic threatened to overwhelm healthcare institutions [11], it became evident that now more than ever there is a need for algorithmic diagnosis support to help mitigate the effects of such large scale contagion events which, research shows, are likely to become more common [12]. This culminated in several groups developing classification models with varying complexities and technologies for automating the COVID-19 diagnosis process [13, 14, 15, 16, 17].

Thus, there is a push from within medical centres towards improving their storage and computing capacities where possible, and collaborating with research centres in order to share expertise, while from the other side, researchers seek to use the available medical data to develop algorithmic diagnosis support systems with the aim of improving outcomes for patients and reducing stress on hospital staff [13, 18, 19, 20].

In one such attempt to build a diagnostic tool and to analyse the mechanisms of pulmonary disease progression, researchers at the University of Nottingham, namely Hardman *et al.*, developed the Nottingham physiology simulator (NPS) [21]. This mechanistic model represented the lungs as a series of differential equations, and was later expanded through the work of Das *et al.* to include a mathematical representation of the cardiovascular system [22]. Further validation of its performance in various applications were also done, which will further be discussed in later sections of this manuscript [23, 24].

Furthermore, research centres have pre-established frameworks for big data analytics within different domains that take advantage of high-performance computing (HPC) and efficient communications and storage hardware [25], which sheds light on opportunities for research using the available medical big data especially given promising results such as those presented by Kesselheim *et al.* [26]. Through the use of these large-scale resources, medical data analysis as well as diagnostic model development, training, and validation can be accelerated and scaled-up to unprecedented levels as was shown in the work done by Baek *et al.* and Jumper *et al.* [27, 28].

With the launch of the Smart Medical Information Technology for Healthcare (SMITH) project, established in 2017 by the German Federal Ministry of Education and Research (BMBF) to set up the communication framework between medical institutions and to test out digital solutions for healthcare, and especially through the use case algorithmic surveillance of intensive care unit patients (ASIC), the NPS is made available to the participating project members and put to use in developing a more portable version [29, 30]. Additionally, the COVID-19 pandemic offered a similar application field to ARDS, and resulted in a wide range of artificial intelligence (AI) tools for its detection becoming freely available for research purposes. Thus, the Jülich Supercomputing Centre (JSC) makes its HPC resources available within both of these use cases with the aim of setting up a data analysis and machine learning (ML) platform specifically designed with medical diagnosis support in mind.

Given the above, it is evident that there is a need for platforms through which cooperation between the medical field and AI experts can be made possible. The proposed platform would provide the necessary tools through which:

- data can be easily stored, cleaned, manipulated, and analysed.
- prospective diagnostic models can be developed, tested, and improved.
- data privacy aspects of dealing with medical information are taken into consideration.

- portability and interoperability with existing cloud computing (CC) frameworks and data science libraries are maintained.

This thesis presents the collection of research and applications done towards setting up the aforementioned platform, with special attention given to the knowledge gained along the way. Work on the platform is validated through model reduction performed on the NPS and experiments where the COVID-Net deep learning (DL) model, developed by Wang *et al.* for COVID-19 diagnosis, is retrained on new data. Additionally, cross-domain research is also described in this thesis through which were developed the skills that made later work possible.

1.2 Thesis Objectives

The main aim of this thesis is to study the implementation of HPC resources and ML algorithms for use in biomedical application fields. However, in order to arrive at the point where these algorithms and resources are implemented, a significant amount of preparatory work was necessary, especially analysing existing work done in the field and uncovering gaps within the research that need to be filled, picking up the skills, techniques, and tools necessary for performing the tasks at hand, and setting up the environments within which the work will need to be done. The end goal of this research is to build and validate a data science platform for the analysis of medical information and the development of ML-based diagnostic models that takes advantage of HPC resources. In order to gauge the progress towards this goal, a set of thesis objectives (TOs) are defined. This section provides an overview of these TOs.

TO1 — Build a knowledge base concerning ARDS and machine learning in biomedical applications

TO2 — Explore potential applications of machine learning algorithms in ARDS understanding and diagnosis

TO3 — Explore model conversion techniques on the ARDS simulation using different network architectures

TO4 — Perform retraining on the COVID-19 detection model using new data

TO5 — Speed up and scale up the developed techniques using the available HPC resources

Figure 1.1 below illustrates the timeline of the TOs from the beginning of the Ph.D. research until its end. The diagram highlights the steps taken towards achieving each TO and includes the necessary techniques, software, and data. The relationship between the TOs is given through the red horizontal arrows which represent the knowledge gained in one step being applied directly into the next. Furthermore, parallel research that does not directly fit into the presented TOs is represented here as "Transferable Knowledge from Cross-Domain Research". This work was vital to the completion of the later objectives, which justifies its inclusion in the Thesis outline. Finally, the flow diagram also provides a graphical representation of the TOs covered by each of the publications listed in Chapter 4.

1.3 Outline

This thesis follows the cumulative format where it highlights the completion of the TOs through the publications listed in the "List of Original Publications". A detailed description of the structure of this manuscript is thus provided in Section 1.3.1, followed by a list of the main publications relating to the objectives set out within this Thesis in Section 1.3.2. A more in-depth look into these publications is provided in Chapter 4 while the other publications not directly relating to the work done by the Thesis author or where their participation was marginal are provided in the "Other Publications".

1.3.1 Thesis Structure

This manuscript is organised as follows:

- **Chapter 1** introduces the scope of this thesis, sets down the thesis objectives, and describes the relationship between these objectives and the published works mentioned in later sections of the thesis.
- **Chapter 2** provides relevant information that sets the groundwork over which the work described in this thesis is built. This information is also part of the overall knowledge collected during work on **TO1**.
- **Chapter 3** describes relevant research that highlights similar approaches to the ones described in this thesis, and defines the state-of-the-art of the technology and approaches within the field.
- **Chapter 4** presents in-depth descriptions of the conference and journal publications, including those that are currently pending submission.
- **Chapter 5** summarises the manuscript and provides a brief outlook of future work.
- Afterwards, The publications listed in Section 1.3.2 are presented in full.
- The thesis ends with the bibliography section listing all the relevant literature.

1.3.2 Publications

The publications listed in the "List of Publications" are presented in full towards the end of this Thesis. These publications are as follows:

- **Paper I**
  C. Barakat, S. Fritsch, M. Riedel, and S. Brynjolfsson, 'An HPC-Driven Data Science Platform to Speed-up Time Series Data Analysis of Patients with the Acute Respiratory Distress Syndrome', in *2021 44th International Convention on Information, Communication and Electronic Technology (MIPRO)*, Opatija, Croatia, Sep. 2021, pp. 311–316. DOI: 10.23919/MIPRO52101.2021.9596840.

- **Paper II**
  C. Barakat, M. Riedel, S. Brynjolfsson, G. Cavallaro, J. Busch, and R. Sedona, 'Design and Evaluation of an HPC-based Expert System to speed-up Retail Data Analysis using Residual Networks Combined with Parallel Association Rule Mining and Scalable Recommenders', in *2021 44th International Convention on Information, Communication and Electronic Technology (MIPRO)*, Opatija, Croatia, Sep. 2021, pp. 248–253. DOI: 10.23919/MIPRO52101.2021.9596796.

- **Paper III**
  C. Barakat, S. Fritsch, K. Sharafutdinov, G. Ingólfsson, A. Schuppert, S. Brynjólfsson, M. Riedel, 'Lessons learned on using High-Performance Computing and Data Science Methods towards understanding the Acute Respiratory Distress Syndrome (ARDS)', in *2022 45th Jubilee International Convention on Information, Communication and Electronic Technology (MIPRO)*, Opatija, Croatia, Jun. 2022, pp. 368–373. DOI: 10.23919/MIPRO55190.2022.9803320.

- **Paper IV**
  C. Barakat, M. Aach, A. Schuppert, S. Brynjólfsson, S. Fritsch, and M. Riedel, 'Analysis of Chest X-ray for COVID-19 Diagnosis as a Use Case for an HPC-Enabled Data Analysis and Machine Learning Platform for Medical Diagnosis Support', *Diagnostics*, vol. 13, no. 3, 2023, DOI: 10.3390/diagnostics13030391.

- **Paper V**
  C. Barakat, K. Sharafutdinov, J. Busch, S. Saffaran, D.G. Bates, J.G. Hardman, A. Schuppert, S. Brynjólfsson, S. Fritsch, and M. Riedel, 'Developing an Artificial Intelligence-Based Representation of a Virtual Patient Model for Real-Time Diagnosis of Acute Respiratory Distress Syndrome', *Diagnostics*, Pending publication, 2023.

1.4 Contributions

As mentioned in Section 1.2, each publication relates directly to one or more TOs. It is however necessary to note that these publications were only possible through completion of literature reviews and coursework which were the central aspects of **TO1** and **TO2**. Additionally, knowledge gained through cross-domain research that does not directly relate to the TOs, especially concerning applications of DL and HPC in object detection from image data, was integral to the completion of the Thesis and is therefore included in the body of work. The relation between the publications and the TOs is presented in Table 1.1. In the remainder of this section, an in-depth discussion of the main contributions of this Thesis is provided, with emphasis on how the TOs were achieved and how they relate to the published material. Additionally, the Thesis author's contributions within each publication is also highlighted.

In order to accomplish **TO1**, a survey of the existing literature, tools, and approaches was done. As part of this step, several potential applications for ML and HPC within the scope of the SMITH project were determined. As this step is only concerned with understanding the use case and gaining knowledge of the available techniques and their feasibility, **TO1** is considered completed at the initial application of pre-processing and ML techniques on the available data.

As part of **TO2**, a deeper understanding of the data was achieved through a more thorough application of ML techniques and visualisation. At this step, it became possible to determine which potential applications from **TO1** were feasible. It was decided at this point to perform a model conversion on the NPS using knowledge with DL and CC methods acquired through coursework. This step also coincided with the onset of the COVID-19 pandemic which provided an added opportunity to test data processing and ML approaches. This in turn became **TO4**. The work done within **TO2**, that is setting up the environments, tools, and techniques, became the central aspect of the first publication provided below in Section 4.1.

Understanding the NPS was an integral part of successfully performing model conversion on it and replacing it with a purely ML-based surrogate model. In parallel to the work on model conversion, a second publication (presented in Section 4.2) was completed that described the development of ML and DL methods for retail

Table 1.1. Relation of publications to the TOs.

	Paper I	**Paper II**	**Paper III**	**Paper IV**	**Paper V**
TO1	×		×	×	×
TO2	×		×	×	×
TO3			×		×
TO4			×	×	
TO5				×	×
Transferable Knowledge		×	×	×	×

applications. The knowledge gained during this work was an integral part of the successful completion of **TO3** and **TO4**, as it provided further use cases to test out the available ML techniques, to become more aware of resource-efficient data-processing algorithms, and to advance knowledge in setting up environments that take advantage of parallel processing and graphical processing unit (GPU)-based resources. Moreover, **TO4** concerned the re-training of the COVID-Net model developed by Wang *et al.* over new data as a second use case for the established ML and data science platform. The publication presented in Section 4.3 thus concludes **TO3** and **TO4**.

Finally, an analysis of the potential speed-up and scale-up to improve the work done towards **TO3** and **TO4** was performed. The completion of this work was designated as **TO5** and its completion is related to the two publications presented in Sections 4.4 and 4.5 which describe the application of large-scale parallel hyperparameter tuning towards improving the COVID-Net model and the surrogate simulator, respectively.

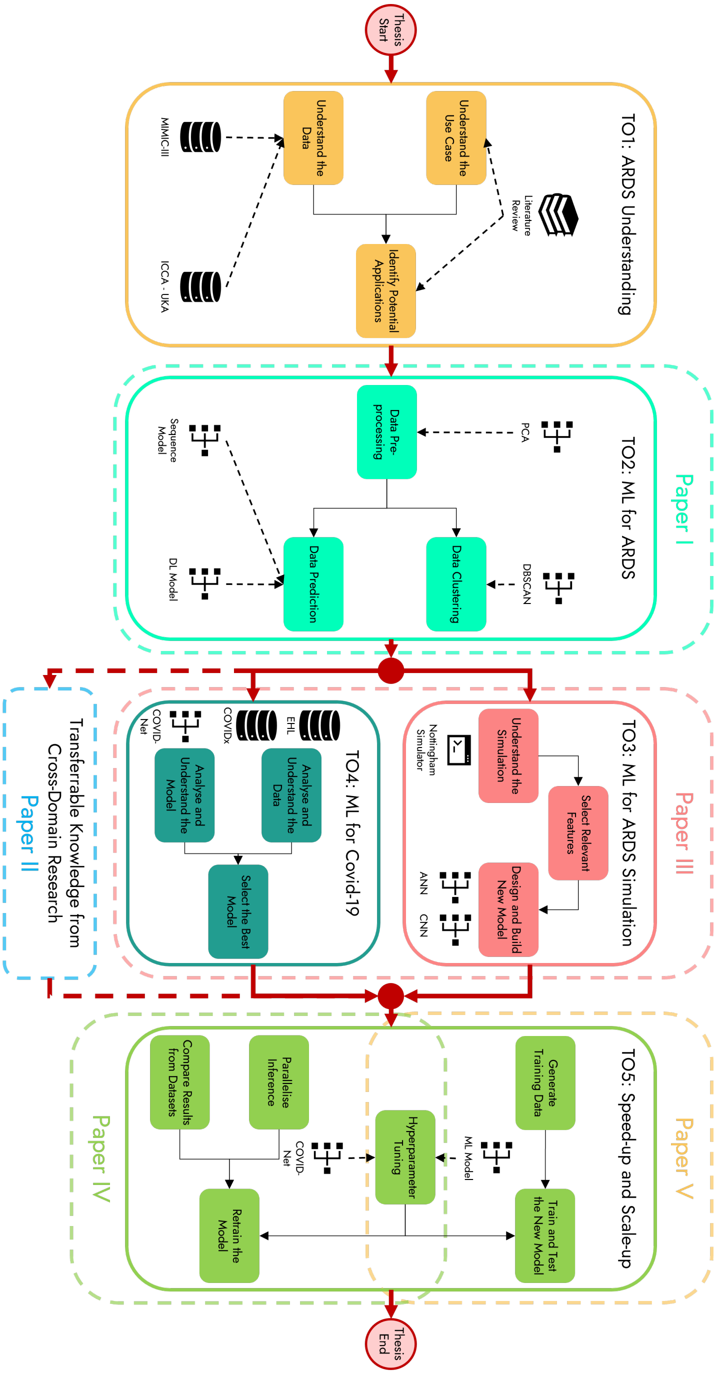


Figure 1.1. Flow diagram of the Thesis progression including the tasks performed and tools employed within each objective as well as the related publications.

Chapter 2

Background

2.1 Machine Learning: Theory and Application

By definition, computer programs are said to "learn" when they are able to independently improve their outputs with exposure to new inputs [31, 32]. Beyond this basic consensus, there is some debate concerning whether ML falls completely within the scope of AI or, as some argue, whether it has grown beyond these limits into a field of its own, building on a base of statistics and probability, and aiming to provide intelligent solutions for common problems rather than constructing intelligent machines with broad applications [33, 34]. Generally speaking, ML comprises many techniques through which it attempts to generalise information from a given input set [34]; these techniques can be divided into three main categories: unsupervised learning (UL), supervised learning (SL), and reinforcement learning (RL), which naturally also have some degree of overlap [35].

In UL, the algorithm explores the data and performs some calculations on it to extract useful information from it [36]. In this case, the "teacher" only participates in the learning process by defining the parameters of the algorithm and the organization of the input data. This approach to learning has seen extensive commercial applications for association rule mining tools and recommenders, among others [37]. On the other hand, in SL the "teacher" needs to provide input data and the expected outcomes as well as define a certain loss metric and optimization function. The combination of these parameters and curated data will allow the algorithm to slowly and iteratively adapt its internal structure until its predictions approach the expected outcomes [35]. artificial neural networks (ANNs) and support vector machines (SVMs) are commonly used types of SL techniques. Finally, RL algorithms perform their optimization task while trying to maximise a certain reward function. This reward increase as the algorithm moves closer to a desired target and decrease otherwise [38]. The key difference from SL is that RL does not require the inputs to be labeled, making it ideal for situations where the expected outcome is unknown and requires several steps to reach such as in robot motion.

The common aspect of all these approaches is the need for large amounts of input data which is one of the main reasons why ML is closely associated with the field of data mining [37]. In fact, since data mining is the process through which knowledge is extracted from databases, then it can be said that learning would be difficult or close to impossible without it [39]. In fact, it is only through the collection, curation, and analysis of large amounts of data that the major advancements in AI in the 21st Century have been possible, such as AlphaGo, AlphaFold, RoseTTAFold, and GPT-3, to name a few prime examples [40, 27, 28, 41].

2.2 High-Performance Computing

High-performance computing (HPC) is a term that describes the technological field where supercomputers - specially designed machines taking advantage of state of the art computation, storage, and communication hardware - are applied towards the solution of complex problems [42]. This is different from high throughput computing (HTC) where commonly available computing hardware is put to use performing basic jobs over long periods of time. Within the last 25 years, the HPC systems in operation have grown both in number and in power, leading to the establishment of the TOP500 list [1] where the current leader is the Frontier system at Oak Ridge National Laboratory. This system is the first to achieve 10^{18} floating point operations per second (FLOPS), and a clear marker of the direction in which the industry is heading. HPC has been applied towards solving many of the most computationally expensive tasks, including training AI models for biomedical applications such as DeepMind's AlphaFold [27], landcover classification [43], computational fluid dynamics simulations [44], and processing large scale data for particle physics experiments [45].

The basic principle of HPC builds on multi-core processors connected to a shared memory, with each processor receiving a portion of the data and either the totality of the program as is the case in single instruction - multiple data (SIMD), or only a portion of the program as in multiple instruction - multiple data (MIMD) [46]. This was adapted with the introduction of accelerators: hardware having a higher number of cores than basic processors, suitable for applications with many small computations in parallel. Eventually the modular supercomputing architecture (MSA) would be introduced where the amount of storage, processors, and accelerators recruited can be tuned depending on the task at hand [25]. In all the cases described above, computational power could be increased by increasing the number of nodes or the number of cores within each node, but this growth is limited by the communication overhead between the components and the minimum size that the cores themselves can physically be (i.e. towards the atomic scale). This is an active area of research with many proposed innovations aimed at overcoming the foreseen bottlenecks [47, 48, 49].

[1] https://www.top500.org/

2.3 Acute Respiratory Distress Syndrome

As mentioned in the introduction of this Thesis, early detection of respiratory disease onset is essential to improving outcomes for ICU patients [2, 3]. This is especially true for ARDS which is a common condition that has a high mortality rate, which is around 40% in the synopsis of the relevant studies [50]. This disease pattern was first described by Ashbaugh *et al.* in 1967 [4]. Its definition evolved over the time and is still under debate [51]. The current generally accepted and used definition is the "Berlin Definition", which was developed by the ARDS Definition Task Force in 2012 [7]. By these guidelines, patients with a ratio of partial pressure of arterial oxygen (P_aO_2) to fraction of inspired oxygen (F_iO_2) (P/F ratio) below 300 mmHg, bilateral opacities in chest x-ray (CXR) images, and an acute response following the initial onset of respiratory symptoms are diagnosed with ARDS. Other reason for hypoxemia like a cardiac failure or a hypervolemia have to be ruled out before. Additionally, the severity of the condition increases as the P/F ratio decreases.

Much research has been done towards fine-tuning diagnosis and treatment of ARDS with little consensus on the most effective method [5, 6, 52]. The most widespread treatment methods include maintained prone positioning and lung-protective ventilation, with the possibility of extracorporeal membrane oxygenation (ECMO) in extreme cases [53, 54, 55]. However, as digitisation of medical records grows, and as more data from ARDS cases becomes available for analysis, more information can be gleaned about this condition, its mechanisms, and the means of its prevention [6, 56].

2.4 COVID-19

At the time of writing this Thesis, almost all countries are still feeling the effect of, or still actively fighting, the COVID-19 pandemic [11]. The earliest evidence of the spread of this viral infection could be traced back to the wet markets of the city of Wuhan, in the Hubei Province in China [57]. The virus at the center of this global pandemic was the severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) [58]. With the increased pressure on the medical infrastructure, it became clear that quick diagnosis and triage of patients was necessary. These approaches were especially needed since the most widely accepted means of diagnosing the condition, reverse-transcription polymerase chain-reaction (RT-PCR), was time-consuming, required staff to be specially trained for sample collection, and was initially criticised for its heterogeneous performance [59, 60]. As COVID-19 predominantly manifested as pneumonia, it was expected that typical signs of the disease would be visible in CXR, which are routinely available diagnostic tools, thus leading to a breakthrough in diagnosis [61, 62].

Aside from CXRs, sonographic images were also used in COVID-19 diagnosis as described by Lugara *et al.* [17], while Elshennawy *et al.* proposed a method to predict COVID-19 patient mortality from routinely collected medical data [13]. At the same time many of these approaches built on or took advantage of ML techniques to some success [14, 15, 16], highlighting an application field for this technology that would simplify quick diagnosis and triaging of patients in the (likely) event of future pandemics [12].

Chapter 3

Related Work

3.1 HPC in Medicine

Given that electronic health records (EHRs) are a standard in the medical field, and finding research groups working on extracting information from physical records using natural language processing (NLP) [56], the amount of medical data that is available for analysis is growing. This growth is met with an increase in the use of HPC resources to process medical "Big Data". This is evident in the research done by Kesselheim *et al.* who set up and re-trained the ResNet152 DL network on the open-source COVIDx[2] dataset using the Jülich wizard for European leadership science (JUWELS)[3] supercomputing cluster and booster [26].

Similarly, the prediction of protein structure has been the focus of the biannual critical assessment of structure prediction (CASP) experiment. In the 2020 edition, Jumper *et al.* from DeepMind[4] implemented their two-stage DL network dubbed "AlphaFold2" which achieved the highest accuracy on all structure predictions [27]. In order to perform the training of AlphaFold2, the team took advantage of tensor processing units (TPUs), available through their parent company Google, to handle the large number of matrix operations. Baek *et al.* also developed an accurate structure prediction DL network dubbed "RoseTTAFold" with comparable performance to Alphafold2 in some tasks and improved results in protein-protein compounds [28]. The training of RoseTTAFold also took advantage of HPC resources at the University of Washington.

3.2 The Nottingham physiology simulator

The NPS was developed by Hardman *et al.* at the University of Nottingham in 1998 as a mathematical model of pulmonary function [21]. Later updates by Das *et al.*

[2] `https://www.kaggle.com/datasets/andyczhao/covidx-cxr2`

[3] `https://www.fz-juelich.de/en/ias/jsc/systems/supercomputers/juwels`

[4] `https://www.deepmind.com`

introduced aspects of the cardiovascular system and modelled the gas exchange in tissue, thus making the NPS a tool capable of simulating patients both under normal conditions and, more importantly, in disease states [22, 23].

The simulator is built within MATLAB[5] as a set of modular functions called by a main script. The user inputs the number of breathing cycles, number of alveolar compartments to model, some physiological initial values, and the duration of the simulation into the main script which in turn initiates a loop and performs iterative equilibration calculations for each compartment. The final output of the simulation is a matrix describing the patient state (including parameters of each alveolar compartment) at the end of every breathing cycle, allowing the use to visualise the patient's response. Additonally, the simulation also includes the option of simulating the effect of medication during the simulation.

The availability of a mechanistic model that could relatively accurately represent an ICU patient and highlight their response to different types of treatment is a boon for researchers aiming to better understand the mechanisms of ARDS and to experiment with novel treatment methods [24, 63].

3.3 ML and DL in Diagnosis Support

Lundervold and Lundervold conducted a survey of ML applications in the medical field with emphasis on medical imaging [18]. Their list is far from exhaustive, but the number of applications they highlight is indicative of the growth that this field has seen. This is partly due to the availability of data for analysis, especially with the advent of medical Big Data, and partly due to the ease with which models can be built and trained. That is not to say that the growth of EHRs does not have its drawbacks, especially in terms of threats to patient privacy [8, 10].

Shillan *et al.* provide a review of the current applications of machine learning in the analysis of ICU data, mainly to predict mortality or to aid in diagnosis, and highlight an increasing rate at which research in this field is being done [19]. One such research was conducted by Sun *et al.*, who developed a diagnostic model that can extract information from medical images and clinical transcripts, thus aiding in the analysis of medical Big Data and reducing workload on hospital staff [20].

This growth in medical ML applications has still not slowed down, especially with the onset of the COVID-19 pandemic, where quick diagnosis was necessary both to improve patient outcomes and to reduce the strain on the healthcare infrastructure. Many of the models built to assess risk or provide diagnostic assistance took advantage of ML and especially DL and trained on the large scale medical databases of medical [13, 14]. Of these models we specifically highlight COVID-Net, developed by Wang *et al.* in the next section.

[5] https://www.mathworks.com/products/matlab.html

3.4 COVID-Net

Wang *et al.* developed COVID-Net in 2020 at the height of the COVID-19 pandemic with the intention of providing a portable and accessible diagnosis support model for improved and accelerated patient screening [14]. Their approach builds on the residual architecture which was central to the success of ResNets at image recognition tasks [64], and leverages so-called projection-expansion-projection-extension (PEPX) design. The team also curated COVIDx[6]: a database of CXR images classified into healthy, COVID-19 Pneumonia, and non-COVID-19 Pneumonia patients.

COVID-Net is built using TensorFlow[7] version 1.13 and its build, training, and inference scripts are available online[8]. After pre-training the model on the ImageNet[9] dataset, the authors re-train on the COVIDx dataset and compare the performance to that of ResNet-50 and VGG-19. Their results show a performance improvement compared to the other networks in terms of accuracy, but also highlight the maintained portability of their model.

[6] https://www.kaggle.com/datasets/andyczhao/covidx-cxr2

[7] https://www.tensorflow.org/

[8] https://github.com/lindawangg/COVID-Net

[9] https://image-net.org

Chapter 4

Summary of Publications

In this section, a summary of the published papers relevant to the doctoral work is provided. Additionally, the TO(s) they relate to and their main contributions are described.

4.1 An HPC-Driven Data Science Platform to Speed-up Time Series Data Analysis of Patients with the Acute Respiratory Distress Syndrome

C. Barakat, S. Fritsch, M. Riedel, and S. Brynjolfsson, 'An HPC-Driven Data Science Platform to Speed-up Time Series Data Analysis of Patients with the Acute Respiratory Distress Syndrome', in *2021 44th International Convention on Information, Communication and Electronic Technology (MIPRO)*, Opatija, Croatia, Sep. 2021, pp. 311–316. DOI: 10.23919/MIPRO52101.2021.9596840.

*This publication fulfills the requirements of **TO2** concerning the analysis of potential applications of ML methods towards the understanding - and eventually diagnosis - of ARDS. Additionally, the knowledge gained during the first TOs was integral to the completion of the work done as part of this publication.*

The paper describes the initial development of the HPC-enabled Machine Learning and Data Science Platform within which much of the remainder of the doctoral work was done. This paper was presented at the 44th annual International Convention on Information, Communication and Electronic Technology (MIPRO) held remotely on September 27 to October 1, 2021.

Given the growth of EHR and the difficulty of analysing all the medical data that is being stored [9, 10], researchers are looking towards automation to be able to analyse and extract relevant information from these large databases in order to advance the healthcare sector and improve outcomes for patients. However, automation can not happen without the involvement of medical experts, who in turn can not be expected to become knowledgeable in data mining, ML, or HPC. For these reasons, this paper presents an HPC-enabled data science platform where data scientists can collaborate with medical personnel to store, manipulate, and visualise data, develop diagnostic models, and potentially uncover new treatments. The platform is validated through a comparative analysis of time-series prediction ML models with the intention to assist in the diagnosis of ARDS.

The data storage and analysis aspect of this research begins with making the medical information mart for intensive care - III (MIMIC-III) database available within the dynamic exascale entry platform (DEEP) cluster at JSC. Given the number of missing values in the data, it was necessary to apply some data manipulation steps including re-sampling and interpolation. Additionally, and in order to set up the ML part of the experiment, it was necessary to get acquainted with the proper modules to load within the cluster implementation of a Jupyter[10] Notebook. The data from selected parameters was then fed into different ML model designs to predict future values. The selected model designs were based on gated recurrent units (GRUs), 1-dimensional convolutional neural networks (CNNs), and a hybrid of both approaches. Figure 4.1 presents an outline of the different elements that make up the developed platform as well as the applications where some of these components were applied.

While setting up the platform, it was necessary to understand how to build and customize a Jupyter kernel with a specific goal in mind. This step would prove useful for later publications and their related TOs. Additionally, given the scale of the MIMIC-III database, storing and manipulating the data was greatly simplified through the use of a platform with access to HPC resources and online storage. The platform also simplified training the ML models due to the availability of GPUs within the compute nodes. The resulting models showed some predictive capabilities with the 1-Dimensions CNN being the simplest to build and fastest to train.

The results above provide a proof of concept for a platform for developing ML models for medical applications. Additionally, the use of GPU-enabled Jupyter Notebooks makes parallel computing more accessible to researchers interested in large scale medical data analysis without requiring them to gain knowledge in parallel programming concepts.

[10] `https://jupyter-jsc.fz-juelich.de`

4.2 Design and Evaluation of an HPC-based Expert System to speed-up Retail Data Analysis using Residual Networks Combined with Parallel Association Rule Mining and Scalable Recommenders

C. Barakat, M. Riedel, S. Brynjolfsson, G. Cavallaro, J. Busch, and R. Sedona, 'Design and Evaluation of an HPC-based Expert System to speed-up Retail Data Analysis using Residual Networks Combined with Parallel Association Rule Mining and Scalable Recommenders', in *2021 44th International Convention on Information, Communication and Electronic Technology (MIPRO)*, Opatija, Croatia, Sep. 2021, pp. 248–253. DOI: 10.23919/MIPRO52101.2021.9596796.

This publication does not directly fulfill specific TOs, but rather provided the practical experience necessary to perform efficient large scale data analysis, and to build, train, and test DL models and other ML techniques.

In this paper, a platform is built in the same way as the data science platform described in Section 4.1 above. The platform validation is done through analysis of data obtained from retail partners within the ON4OFF project, extracting relevant information to improve sales, and building models that would simplify the search function within the partners' websites. The results were presented at the 44_{th} International Convention on Information, Communication and Electronic Technology (MIPRO).

The COVID-19 pandemic forced brick-and-mortar stores to shut their doors to customers and bring their business onto the Internet or risk closing indefinitely. Thus the ON4OFF project was established to set up platforms where online services could be made available to previously offline stores include product recommendations and improved search.

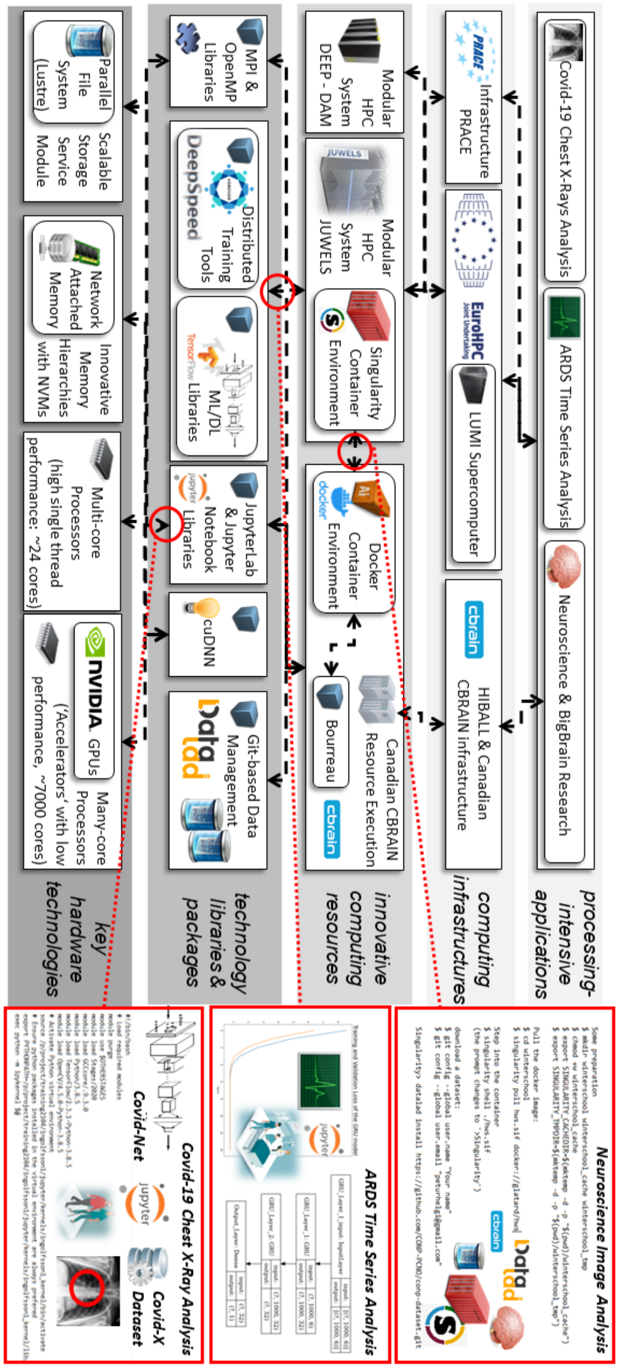
The data provided from retail partners included transaction data from their stores over a given period of time from which sales information would need to be extracted, and images of their products that would be used to improve their search function. The proposed solutions were to generate product recommendations from the transaction data using association rule mining algorithms while the image data would undergo two different analyses in order to extract metadata concerning shape and colour which would be used as tags for improved search functionality. In a similar approach to that discussed in Section 4.1, a platform is set up with the necessary hardware, storage space, and modules to perform the necessary tasks. The mlxtend[11] module is used to build frequent itemsets and perform the necessary calculations to extract association rules from the 8 million transactions available. Clustering is used to extract colour information from the product images while the ResNet50 DL model is fed these images to generate predictions concerning shape and recognisable features.

[11] `http://rasbt.github.io/mlxtend/`

Generating association rules from the available data served the purpose of highlighting efficient use of resources through the application of efficient data structures, namely sparse matrices. Given the task of finding association rules between thousands of products over millions of transactions was experimentally shown to be impossible without the use of the HPC infrastructure. It follows that the outputs from this task were packaged and made available to the project partners. On the other hand, applying clustering on the available image data was the best performing and most resource-friendly approach for color information extraction. Additionally, the pre-trained ResNet50 with a modified classification layer successfully generated predictions of object shapes. These two approaches were lightweight enough to be exported to the project partners and would not require further input from our side beyond the initial development and/or training.

Figure 4.2 represents the workflow of the retail project, with added information concerning the data received as input and how it was adapted at every step of the analysis.

Through the results discussed above, the HPC-enabled platform for retail applications was validated; it provided the necessary infrastructure to extract information and build models that would improve retail services. Additionally, this task was an exercise in resource-efficiency, organisation, collaboration, and proper programming technique, all of which were useful for the remainder of the doctoral work.


processing-intensive applications
Covid-19 Chest X-Rays Analysis
ARDS Time Series Analysis
Neuroscience & BigBrain Research
computing infrastructures
Infrastructure PRACE
PRACE
EuroHPC Joint Undertaking
LUMI Supercomputer
cbrain
HIBALL & Canadian CBRAIN infrastructure
innovative computing resources
Modular HPC System DEEP - DAM
JUWELS
Modular HPC System JUWELS
Singularity Container Environment
Docker Container Environment
docker
Canadian CBRAIN Resource Execution
Bourreau
cbrain
technology libraries & packages
MPI & OpenMP Libraries
DeepSpeed
Distributed Training Tools
TensorFlow
ML/DL Libraries
jupyter
JupyterLab & Jupyter Notebook Libraries
cuDNN
Data lad
Git-based Data Management
key hardware technologies
Parallel File System (Lustre)
Scalable Storage Service Module
Network Attached Memory
Innovative Memory Hierarchies with NVMs
Multi-core Processors (high single thread performance: ~24 cores)
NVIDIA. GPUs
Many-core Processors ('Accelerators' with low performance, ~7000 cores)
Neuroscience Image Analysis
cbrain
Data lad
ARDS Time Series Analysis
Training and Validation Loss of the GRU model
jupyter
GRU_Layer_1_input: InputLayer
input:
[(?, 1000, 6)]
output:
[(?, 1000, 6)]
GRU_Layer_1: GRU
input:
(?, 1000, 6)
output:
(?, 1000, 32)
GRU_Layer_2: GRU
input:
(?, 1000, 32)
output:
(?, 32)
Output_Layer: Dense
input:
(?, 32)
output:
(?, 1)
Covid-19 Chest X-Ray Analysis
Covid-Net
jupyter
Covid-X Dataset

```
Some preparation
$ mkdir winterschool winterschool_cache winterschool_tmp
$ chmod +w winterschool_cache
$ export SINGULARITY_CACHEDIR=$(mktemp -d -p "$(pwd)/winterschool_cache")
$ export SINGULARITY_TMPDIR=$(mktemp -d -p "$(pwd)/winterschool_tmp")

Pull the docker image:
$ cd winterschool
$ singularity pull hws.sif docker://glatard/hws

Step into the container
$ singularity shell ./hws.sif
(the prompt changes to `>Singularity`)

download a dataset:
$ git config --global user.name "Your name"
$ git config --global user.email "peturhelgi@gmail.com"

Singularity> datalad install https://github.com/CONP-PCNO/conp-dataset.git
```

```
#!/bin/bash
# Load required modules
module purge
module use $OTHERSTAGES
module load Stages/2020
module load GCCcore/.9.3.0
module load Python/3.8.5
module load TensorFlow/2.3.1-Python-3.8.5
module load OpenCV/4.5.0-Python-3.8.5
# Activate Python virtual environment
source /p/project/training2104/ingolfsson1/jupyter/kernels/ingolfsson1_kernel/bin/activate
# Ensure python packages installed in the virtual environment are always preferred
export PYTHONPATH=/p/project/training2104/ingolfsson1/jupyter/kernels/ingolfsson1_kernel/lib
exec python -m ipykernel $@
```

Figure 4.1. The structure of the developed machine learning and data science platform.

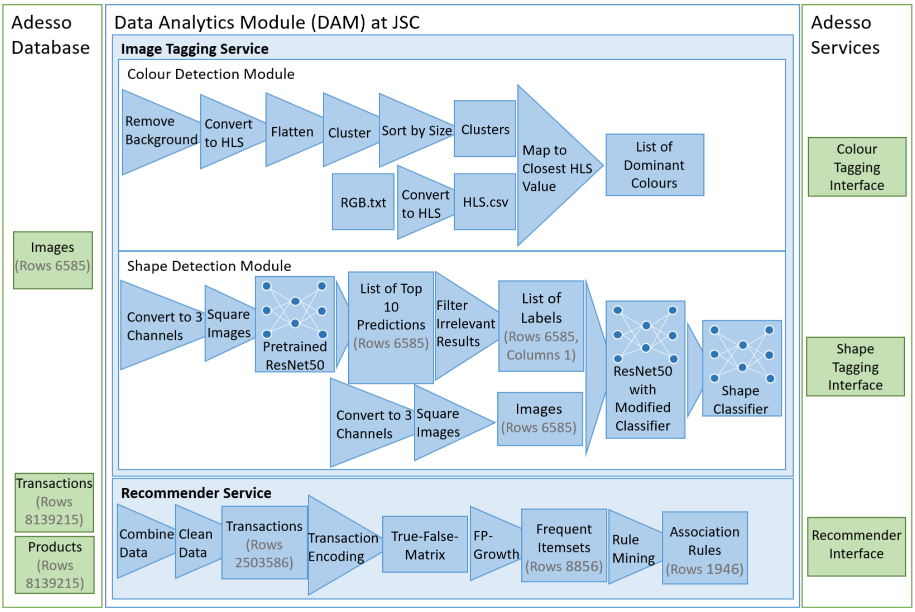


Figure 4.2. Flow diagram of the service modules within the developed platform for retail.

4.3 Lessons learned on using High-Performance Computing and Data Science Methods towards understanding the Acute Respiratory Distress Syndrome (ARDS)

C. Barakat, S. Fritsch, K. Sharafutdinov, G. Ingólfsson, A. Schuppert, S. Brynjólfsson, M. Riedel, 'Lessons learned on using High-Performance Computing and Data Science Methods towards understanding the Acute Respiratory Distress Syndrome (ARDS)', in *2022 45th Jubilee International Convention on Information, Communication and Electronic Technology (MIPRO)*, Opatija, Croatia, Jun. 2022, pp. 368–373. DOI: 10.23919/MIPRO55190.2022.9803320.

*The work described in this publication builds on the skills developed through completion of **TO1** and **TO2**, as well as the experimental approaches described in Section 4.2 above. Through this research, **TO3** and **TO4** would be achieved where an analysis of machine learning techniques for ARDS diagnosis is performed and where an initial re-training of the COVID-Net model on new data is done. Finally, the described experimental procedures set the scene for the work to be done towards completing **TO5**.*

Continuing the work described in 4.1, this paper discusses the work to be done in converting the NPS to a ML-based model, especially in terms of data preparation. In parallel, the results from an analysis of the COVID-Net model with newly obtained data from research partners e*HealthLine (EHL) are also presented. This publication was presented at the 45^{th} Jubilee International Convention on Information, Communication and Electronic Technology (MIPRO) which was held as a hybrid conference on 23 to 27 May, 2022.

Given a deeper understanding of how the NPS operates, especially in terms of the expected inputs and the generated outputs, as well as the knowledge gained from the literature reviews concerning the ARDS disease pattern, the path towards the diagnostic model became clear: the MATLAB-based simulation would be replaced with an equivalent, but more portable and open ML model. This conversion would not only require the model to be built, but also the data for its training to be generated since the MIMIC-III data is far from representative. On the other hand, and as part of supervising the Masters thesis work of G. Ingólfsson, the COVID-Net DL model performance on newly obtained data would be tested and then improved. This use case would highlight the developed data science and ML platform as an effective tool for developing diagnostic models in times of crisis.

The use of HPC in speeding up and scaling up the research was a central part of the paper. Thus, the MATLAB simulation would be converted to C, making it easily parallelisable on the supercomputing cluster, and setting it up as a tool to generate data for upcoming research. The process of generating synthetic inputs is also discussed, with special emphasis on the computational complexity of the task and the need for an efficient solution. In parallel, a separate environment is set up with

Table 4.1. Execution time of the simulation.

Platform	Execution Time
Original Simulation on Laptop	259.1 s
C in serial on DEEP with JupyterLab	108.8 s
C in parallel on DEEP on 48 CPUs	100.79 s

the necessary modules and storage for storing and running the COVID-Net model; several experiments are conducted following the cross industry standard process model for data mining (CRISP-DM) in order to select the best performing version of the model given the available data [65]. The chosen model is used to classify the newly obtained images, then retrained on a subset of this data to highlight how the performance is affected.

Converting the NPS cause an extreme reduction in the length of individual simulations as shown in Table 4.1. Furthermore, by running multiple simulations in parallel, the average duration of a simulation is further reduced, thus highlighting the speed-up that the platform can offer. Similarly, by analysing the available MIMIC-III data, a sampling method for generating synthetic data is proposed, although it poses the risk of overloading even the HPC system. This problem would require a solution that takes advantage of statistical methods and would be the subject of later research. Finally, the performance of the COVID-Net model before and after retraining is analysed and a clear difference is highlighted in Figure 4.3. This improvement is also shown to be extremely dependent on a number of parameters, which sets the path towards **TO5**.

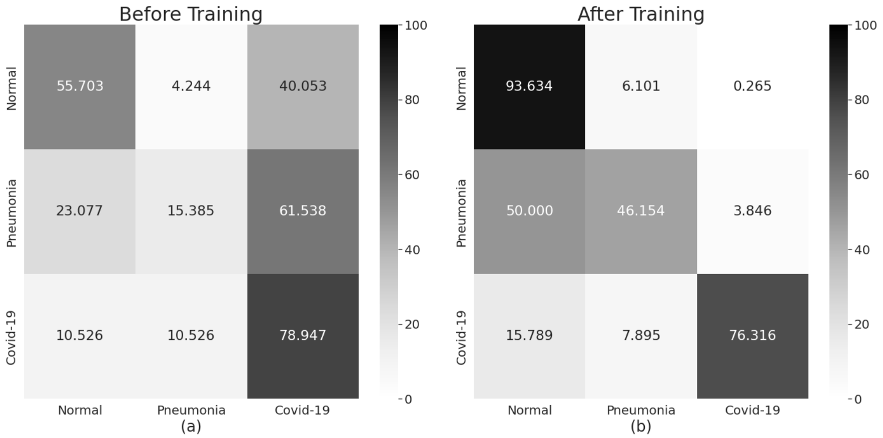


Figure 4.3. Prediction performance on a test set of the EHL dataset (a) before and (b) after training.

The paper described in this section provided the conclusive results that HPC is necessary for the research being done. The training of the COVID-Net model was accelerated using the available computing resource, while the parallelised NPS makes it possible to run multiple simulations simultaneously, thus generating more data for the proposed DL surrogate model.

4.4 Analysis of Chest X-Ray for COVID-19 Diagnosis as a Use Case for an HPC-enabled Data Analysis and Machine Learning Platform for Medical Diagnosis Support

C. Barakat, M. Aach, A. Schuppert, S. Brynjólfsson, S. Fritsch, and M. Riedel, 'Analysis of Chest X-ray for COVID-19 Diagnosis as a Use Case for an HPC-Enabled Data Analysis and Machine Learning Platform for Medical Diagnosis Support', *Diagnostics*, vol. 13, no. 3, 2023, DOI: 10.3390/diagnostics13030391.

*This journal article partially fulfills **TO5** especially concerning speed-up and scale-up of the work done on re-training the COVID-Net model with newly obtained data from industry partners which was done to fulfill **TO4**. Additionally, this work is the culmination of the other TOs and the knowledge collected along the way.*

The results from the paper summarised in Section 4.3 highlighted the potential for improvement in the performance of the COVID-Net model, leading to an analysis of available methods for model tuning that take advantage of the available supercomputing resources. It follows that several researchers have discussed the implementation of hyperparameter tuning to improve ML models such as Zhang *et al.* with their work on Alzheimer's Disease data [66], and Farag *et al.* with their applications on ResNets and Xception networks for COVID-19 diagnosis [67]. From there, it seems evident that further experimentation should be done towards scaling up the hyperparameter tuning process for the COVID-Net model. The goals of this analysis would be to highlight the speed-up of the tuning process, especially in terms of covering more ground in the parameter search process.

We consider the Ray[12] Tune library, and the KerasTuner[13] library for the hyperparameter optimization step due to prior successful experience within the team. Furthermore, given the extensive documentation available for Ray Tune and especially due to its compatibility with Tensorflow 1.13 (the version used to build COVID-Net), it is implemented in the remainder of this work. The parameters to tune are selected, namely the class weights, learning rate, and the COVID-19 percentage, and the tuning process is performed as a comparative analysis of 4 scheduling algorithms that are built into the Ray library. Aside from the default first-in, first-out (FIFO) scheduler used here as a control, each of the selected algorithms approaches the tuning process in a different manner: HyperBand and Asynchronous HyperBand rely on early stopping of underperforming trials to make the process more resource-efficient, with Asynchronous HyperBand also implementing a successive halving approach [68, 69], and population-based training (PBT) introduces perturbations at a set point during the process to expand the search space [70]. All of the tuning trials took advantage of parallel computing on the developed data science and machine learning platform discussed in Sections 4.1 and 4.3.

[12] `https://www.ray.io/`

[13] `https://keras.io/keras_tuner/`

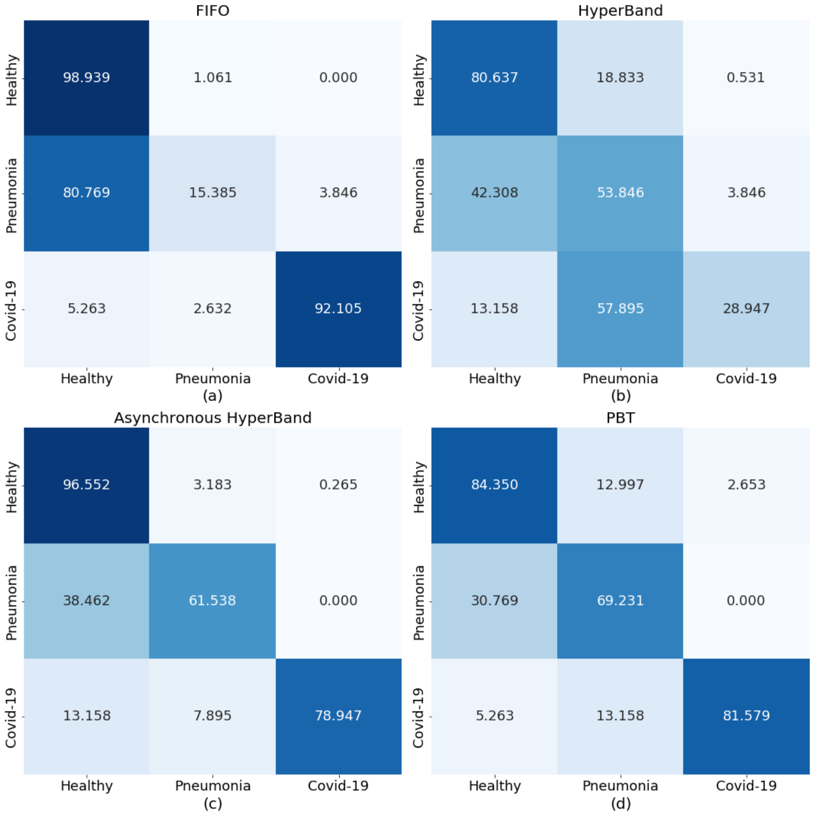


Figure 4.4. Prediction performance heatmaps for COVID-Net on the EHL dataset after re-training on the parameters chosen by (a) FIFO, (b) HyperBand, (c) Asynchronous HyperBand, and (d) PBT.

Figure 4.4 presents the prediction performance of the models trained using the best parameter combinations selected by each scheduler. The Asynchronous HyperBand and PBT schedulers produced models that improved the per-class accuracy of the COVID-Net model, especially in the case of pneumonia where the pre-tuning performance (Figure 4.3(b)) was noticeably improved. The remaining approaches under-performed, which highlighted the need to further analyse the chosen success metric over which the best trials are selected. A final experiment is performed to highlight the potential speed-up that can be achieved using the Horovod[14] distributed training framework. The training durations are presented in Figure 4.5 which highlights the fact that there is an upper limit at which the communication overhead between the recruited resources begins to affect performance.

The research presented in this paper was an initial foray into the use of hyperparameter optimization for improving the performance of ML models. Additionally, it sets up some of the groundwork for the final publication presented in Section 4.5 where a ML-based ARDS diagnosis model is developed, trained, and validated.

[14] `https://horovod.ai/`

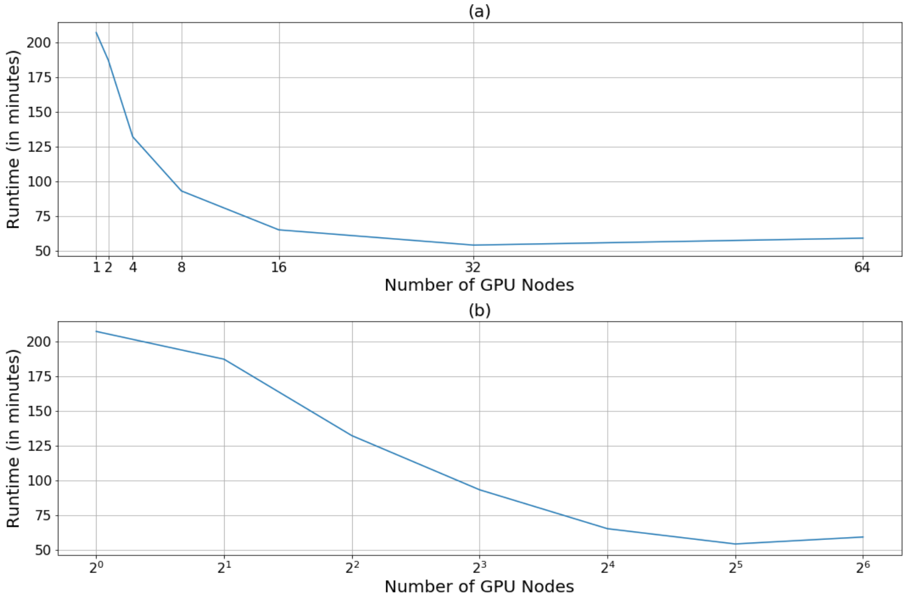


*Figure 4.5. Training duration (in minutes) as more GPU nodes are recruited, (**a**) on a linear scale and (**b**) on a logarithmic scale.*

4.5 Developing an Artificial Intelligence-Based Representation of a Virtual Patient Model for Real-Time Diagnosis of Acute Respiratory Distress Syndrome

C. Barakat, K. Sharafutdinov, J. Busch, S. Saffaran, D.G. Bates, J.G. Hardman, A. Schuppert, S. Brynjólfsson, S. Fritsch, and M. Riedel, 'Developing an Artificial Intelligence-Based Representation of a Virtual Patient Model for Real-Time Diagnosis of Acute Respiratory Distress Syndrome', *Diagnostics*, Pending publication, 2023.

This journal article fulfills the remaining requirements of ***TO5****, specifically in terms of large-scale parallel data generation for the ARDS model, the development and analysis of the surrogate ML model, and the analysis of potential improvements of the generated model through hyperparameter tuning methods. This publication is thus the culmination of the work done on the virtual patient model conversion and builds on all the knowledge gained through completing the previous TOs.*

Building on the results of the paper described in Section 4.3, and the technical knowledge in parallel hyperparameter tuning methods gained during work on the publication described in Section 4.4, the research presented in this publication revolves around the process by which different network architectures were tested in order to choose the best approach for the eventual ML-based surrogate virtual patient model. The article thus describes the following processes in detail:

- Generate data that mimics available real-world patient data.
- Run the C-based simulation using the generated data to produce the expected outputs.

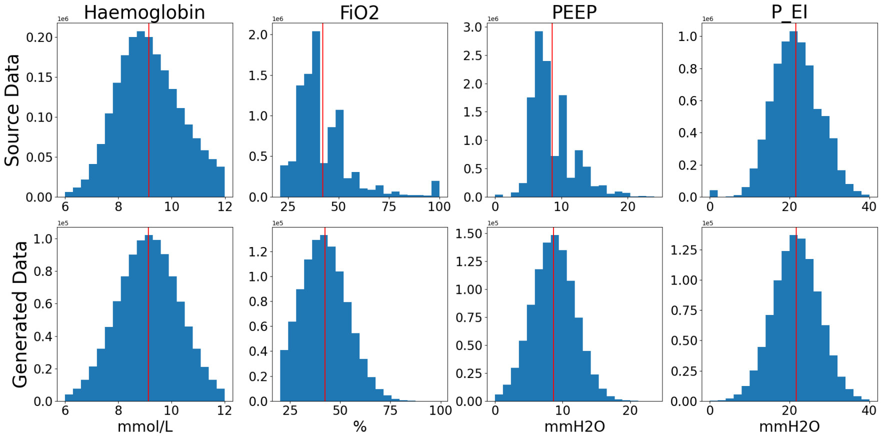


Figure 4.6. Histograms comparing the distribution of the generated input data with that of the original data. The red lines represent the means for each parameter.

- Perform preliminary testing with the generated inputs and expected outputs on different neural network architectures.
- Perform large-scale parallel hyperparameter optimization to uncover the best performing parameter combinations.
- Finally, train the best performing model and validate its performance.

It was necessary to generate data since the MIMIC-III patient data made available by Sharafutdinov *et al.* was not sufficient for adequate model training, or for covering a wide range of patient states [71]. Their analysis of the data does however provide an insight into the statistical distribution of physiological parameters which, combined with upper and lower bounds provided through the practical knowledge of medical personnel, could be used to generate biosimilar synthetic data. A comparison of the distributions of some parameters is provided in Figure 4.6.

Feeding the generated data as initial patient states into the C-based simulator produced output values of the parameters after a given time has passed. The outputs

Table 4.2. Input and output parameters of the C-ported virtual patient simulator.

	Parameter	**Description**
Input Parameters	v_sR, v_inR	used to calculate individual compartment resistance to flow (R_{comp}) values
	v_sVR, v_inVR	used to calculate individual compartment vascular resistance (VR_{comp}) values
	v_nc	number of closed compartments
	asht	anatomical shunt
	RQ	respiratory quotient
	VO_2	oxygen uptake
	VD_{phys}	volume of physiological deadspace
	CO	cardiac output
	I:E	inspiratory to expiratory ratio
	Hb	hæmoglobin
	F_iO_2	fraction of inspired oxygen
	PEEP	peak end-expiratory pressure
	P_{EI}	end-inspiratory pressure
	S_vO_2	venous oxygen blood saturation
	RR	respiratory rate
	V_t	tidal volume
	BE_a	arterial base excess
Output Parameters	P_aO_2	partial pressure of arterial oxygen
	P_aCO_2	partial pressure of arterial carbon dioxide
	HCO_3	bicarbonate concentration
	pH	blood acidity level

are indicative of pulmonary impairment and which can be used to determine ARDS onset, namely P_aO_2, partial pressure of arterial carbon dioxide (P_aCO_2), pH, and bicarbonate concentration. These inputs and expected outputs are combined into a training dataset that is next used to train several neural networks. Table 4.2 presents a description of the input and output parameters for the simulator.

Several neural network architectures were considered for building the surrogate model, although Recurrent Neural Networks (RNNs) were eventually excluded as the available data is not sequence data. Furthermore, given the experience applying ANNs and CNNs in the previously described research, they were chosen as the techniques for the task at hand. Additionally, the intention of this research is to develop the most portable, simplified, and lightweight approach to building a surrogate model, which further reinforces the choice of basic neural network architectures. It follows that the preliminary results showed that the CNN-based models often outperformed the ANN counterparts, reaching lower loss values in fewer training epochs with fewer instances of overfitting. It follows that a CNN model with hyperparameters chosen through trial and error was the most likely candidate for the next step of the experiment: large-scale parallel hyperparameter tuning. In this experiment, the parameters to be tuned dictated the network structure (presence or absence of an intermediate fully-connected layer at before the output layer), the rate of dropout within the network structure, training parameters such as the learning rate and the batch size, and finally the loss function (mean absolute error (MAE) or mean squared error (MSE)).

In similar fashion to the process described in the publication from Section 4.4, four different schedulers are selected to distribute the tuning trials between the available resources, and their performance is compared. The HyperBand-based schedulers were fastest at completing the tasks and, since they stop underperforming trials, they were deemed the most resource efficient. The PBT scheduler took the longest to finish due to it running the model training in several iterations with minor changes to selected parameters. Finally, the best performing parameter combination of each scheduler is used in training a model in full scale over the available data. Figure 4.7 showcases the performance of each model and presents the R^2 score for each of the output parameters as an accuracy metric.

The results show that even the best performing parameters chosen through hyperparameter tuning can lead to an model that overfits, which is the case for the parameter combinations produced by the FIFO and Asynchronous HyperBand schedulers. Additionally, the results achieved by these models do not greatly improve upon the results from the original trial and error approach in terms of loss reduction, although it is clear that an algorithmic search for best parameter combinations is substantially more resource-efficient than the trial and error approach, especially when early-stopping is implemented. Finally, all the trained models have a per-parameter R^2 score greater than 0.9 which, combined with their small footprint and ease of implementation, make them adequate diagnosis support and early warning systems for potential disease onset in ICU patients.

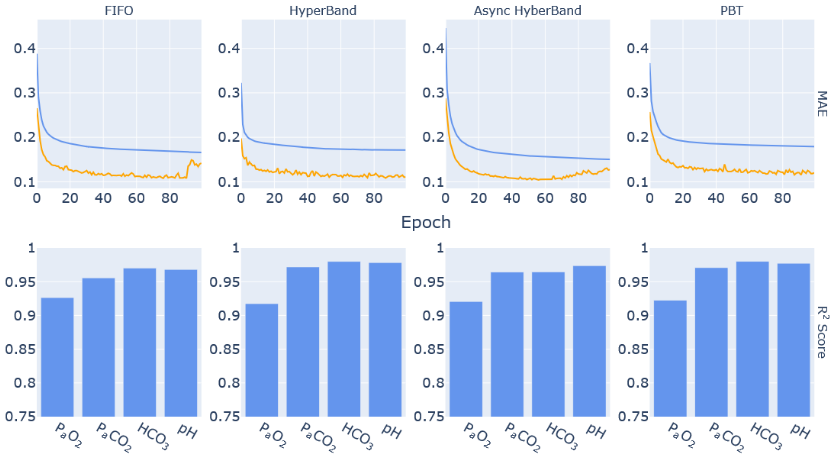


Figure 4.7. Learning curves of the training (blue) and validation (orange) MAE of the models using the best parameters as selected by each scheduling algorithm, accompanied by their respective per-parameter R^2 score bar graph.

To conclude the summary of this publication, it is necessary to discuss the added benefit of employing supercomputing resources in the model training process. The HPC-enabled ML and data science platform developed and discussed in previous publications provides an adequate and easily adaptable environment for data processing and model development, as it eliminates the resource limitations of personal computers and smaller scale architectures. This is also done without losing track of the portability and ease of implementation of the platform itself within different environments, including containers and cloud computing resources. Additionally, the specific application described herein serves as a definitive proof that the platform is applicable for the massively parallel development, training, and deployment of models; these in turn can be easily exported for offline use and implementation in clinical routine. This work thus paves the way for future research where larger and more representative models are developed and tested within the platform.

Chapter 5

Conclusions

Through the digitisation and collection of health records, the healthcare sector has set up the groundwork for Medical Big Data, a treasure trove for data scientists looking to extract new information and for ML researchers aiming to improve current technologies in the field and to develop the next generation of diagnostic support tools. This growth was made possible through the use of novel techniques and materials in electronic storage media and processing hardware which have also been the driving factors to the advancements in HPC. In fact, the race to exascale computing has already been won and the next target has already been selected. With the increase in computing power, more advanced ML models have been developed, many of which making their way into commonplace use such as in autonomous driving, improving search engines, and providing medical diagnosis assistance.

Additionally, with compute resources being made available to the general public through commercial CC services, and as more work is done towards the development of open-source software that take advantage of these resources, the research in this field is made more accessible to institutions that would otherwise not have access to these technologies. This Thesis compiles the research done within these fields towards the establishment of a modular, adaptable data science and ML platform within which novel diagnostic models and tools can be developed and improved, and which takes advantage of both HPC resources and open-source software to achieve the task. The validation of this platform was also presented herein as the central aspects of published material.

The research described in this manuscript is presented in the form of five TOs, the successive accomplishment of which designates progress through the doctoral research as well as the advancement of knowledge of the Thesis author in the field. These TOs are described below:

TO1 – Build a knowledge base concerning ARDS and machine learning in biomedical applications

TO2 – Explore potential applications of machine learning algorithms in ARDS understanding and diagnosis

TO3 – Explore model conversion techniques on the ARDS simulation using different network architectures

TO4 – Perform retraining on the COVID-19 detection model using new data

TO5 – Speed up and scale up the developed techniques using the available HPC resources

The main tasks of **TO1** were to collect information concerning the medical condition in question, namely ARDS, as well as the research done towards its diagnosis and treatment. Additionally, an analysis of the available approaches to data processing and clustering was done as well as a survey of available ML methods for information extraction, forecasting, and prediction. This work paved the way for application as part of **TO2** where the first steps towards setting up the aforementioned HPC-enabled ML platform were taken.

Completion of **TO2** was determined through the first publication, and provided a stepping stone for the experiments conducted towards **TO3** and **4** which concerned the implementation of the platform for model conversion and model improvement tasks, respectively. These two tasks were the core of a publication and would further be studied as part of the **final TO** revolving around the improvement of the models through the HPC resources within the established platform. **TO5** would yield two publications, one relating to the work done on improving the COVID-19 diagnosis model, and the other presenting the work done on converting the NPS to a ML simulation that would potentially aid in, and accelerate ARDS diagnosis.

Future Work

Given all of the above, it is self-evident that more work needs to be done to improve the platform for a wider spectrum of applications by finding new use cases and incorporating new tools and techniques. In particular, implementing the platform for the development or improvement of larger models is one of the main use cases expected within upcoming projects. That is not to say that work on improving ARDS diagnosis will stop; in fact, this work will be made available to other researchers within the scope of the SMITH project, thus making new clinical data available with which to test the model, and inviting scrutiny from experts in a wide variety of fields.

On the other hand, the majority of the ML models analysed or developed during this doctoral work were developed through Keras[15], an application programming interface (API) for TensorFlow. This approach may not be the fastest or most resource-efficient which makes the search for other ML frameworks the logical next step. During the doctoral work, PyTorch[16] was gaining in popularity as a ML framework, making it the most likely candidate.

Finally, although the platform discussed herein is designed to be portable, it would be an interesting exercise to implement it within different distributions or

[15] `https://keras.io/`

[16] `https://pytorch.org/`

compute resources in order to determine the elements that would need to be tuned or completely changed. This also highlights the need to compile some documentation for the platform in order to help in future implementations and troubleshooting.

Paper I

An HPC-Driven Data Science Platform to Speed-up Time Series Data Analysis of Patients with the Acute Respiratory Distress Syndrome

C. Barakat, S. Fritsch, M. Riedel, S. Brynjólfsson

https://ieeexplore.ieee.org/document/9596840/, 2021

This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/).

C. Barakat performed the data manipulation, platform preparation, and model development for the work described in this paper, as well as drafting the original manuscript.

An HPC-Driven Data Science Platform to Speed-up Time Series Data Analysis of Patients with the Acute Respiratory Distress Syndrome

C. Barakat*†, S. Fritsch†‡, M. Riedel*†, S. Brynjólfsson*
* School of Engineering and Natural Sciences, University of Iceland, Iceland
† Jülich Supercomputing Centre, Forschungszentrum Jülich, Germany
‡ Department of Intensive Care Medicine, University Hospital RWTH Aachen, Germany
c.barakat@fz-juelich.de, sfritsch@ukaachen.de, morris@hi.is, sb@hi.is

***Abstract*—An increasing number of data science approaches that take advantage of deep learning in computational medicine and biomedical engineering require parallel and scalable algorithms using High-Performance Computing systems. Especially computational methods for analysing clinical datasets that consist of multivariate time series data can benefit from High-Performance Computing when applying computing-intensive Recurrent Neural Networks. This paper proposes a dynamic data science platform consisting of modular High-Performance Computing systems using accelerators for innovative Deep Learning algorithms to speed-up medical applications that take advantage of large biomedical scientific databases. This platform's core idea is to train a set of Deep Learning models very fast to easily combine and compare the different Deep Learning models' forecast (out-of-sample) performance to their past (in-sample) performance. Considering that this enables a better understanding of what Deep Learning models can be useful to apply to specific medical datasets, our case study leverages the three data science methods Gated Recurrent Units, one-dimensional convolutional layers, and their combination. We validate our approach using the open MIMIC-III database in a case study that assists in understanding, diagnosing, and treating a specific condition that affects Intensive Care Unit patients, namely Acute Respiratory Distress Syndrome.**

Keywords*—*High-Performance Computing; MIMIC-III database; Acute Respiratory Distress Syndrome; modular supercomputing; data science platform

I. Introduction

The technology involved in collecting, storing, and processing information has advanced to such an extent that we have at our disposal data on almost every aspect of the world we can observe; this is true on a Universe[1,2], global[3], local[4], or personal level. This abundance of data means that Machine Learning (ML) experts can use new innovative tools to improve their sequence models, for example, improving Natural Language Processing (NLP) algorithms by processing open-access literature[5]. New ML methods are increasing image processing algorithms' accuracy with labelled open-source photographic data [1] and enhancing weather prediction protocols with long and detailed weather records that go back several decades[3]. Specifically, in the medical field, Electronic Health Records (EHRs) have made it easier to group data of many patients diagnosed with the same conditions from several hospitals, countries, and even time periods to highlight previously overlooked markers that could improve treatment or accelerate diagnosis [2]. Applying ML and Deep Learning (DL) techniques to this data has the potential of uncovering underlying correlations that would otherwise require several researchers several years to piece together [3]. All the above relevant methods and techniques for medical data sciences have in common that we observe a significant increase in the requirement of having larger computing capacity available (e.g., HPC for distributed training of deep learning networks).

This paper addresses the increased complexity that medical experts experience when interacting with High-Performance Computing (HPC) resources which are becoming more widely available in academic centers and accessible through public cloud resources as well. That also includes an increase in the power of HPC resources available through research institutions, clinics, and hospitals. Aside from their regular duties, medical experts have to learn to navigate these resources in order to perform their analyses as opposed to the traditional data analysis performed on personal computers. This paper thus describes one flexible platform approach wherein this problem is mitigated and there is no need for medical experts to pick up any specialised high-level programming skills. Furthermore, today, it is possible to scale medical applications of the above-mentioned DL and ML techniques in a way that fits the growing size of the data available through EHRs. But the quality of the data stored in EHRs represents another challenge for medical experts in the data analysis process. It varies between institutions due to different reporting standards or sensor configurations, while in parallel, several EHR standards are currently being used in hospitals, adding another layer of complexity to the equation and ultimately influencing the quality of any data analysis task.

This paper addresses these challenges by proposing an HPC-enabled platform that assists in data preparation and understanding to help medical professionals by taking advantage of algorithmic techniques and efficient computing and storage resources. We use one specific medical condition as a driving use case for the design and evaluation of this platform. Hence, our platform employs ML and DL models in the analysis of patient information to predict missing values in medical datasets while keeping the technical complexity to a

[1] https://exoplanetarchive.ipac.caltech.edu/index.html
[2] https://ai.googleblog.com/2018/03/open-sourcing-hunt-for-exoplanets.html
[3] https://www.ncdc.noaa.gov/cdo-web/
[4] https://www.europeandataportal.eu/data/datasets/10532954-7c62-44d4-826a-34642954e394?locale=en
[5] https://www.elsevier.com/connect/new-open-access-resource-will-support-text-mining-and-natural-language-processing

low degree. This model aims to assist in the understanding, diagnosis, and treatment of a specific condition that affects Intensive Care Unit (ICU) patients, namely Acute Respiratory Distress Syndrome (ARDS) while not losing sight that this platform can be used for other medical conditions.

The remainder of this paper is structured as follows. Related work is reviewed in Section II and Section III provides brief overviews on medical and technological methods required to understand the paper. Section IV describes the dynamic data science platform tailored to support clinical researchers in understanding ARDS. While Section V reveals our data analysis approaches, followed by our evaluations and findings. This paper ends with some concluding remarks.

II. Related Work

In this section we survey related works that are relevant in context (e.g., simulators of disease progression, machine and deep learning approaches, etc.). The research by Wang *et al.* showed the importance of using mathematical modeling in the treatment of chronic obstructive pulmonary disease patients (COPD). Their approach employed a physiological simulator of the cardiopulmonary system, tuned to replicate the responses of COPD patients, in order to test mechanical ventilation protocols *in silico*[4]. Their work builds on original work by Hardman *et al.* who initially developed a physiological simulator of the respiratory systems of a patient that was capable of accurately representing responses to changes in mechanical ventilation maneuvers[5]. Das *et al.* describe the development, testing, and validation of a virtual patient model that can accurately mimic the physiological state of ARDS patients[6]. Their work is a continuation of the work on the physiological simulator described in the work by Wang *et al.*

In terms of applications of machine and deep learning techniques in the context of ARDS analysis, Le *et al.* trained a gradient boosted tree model using the Medical Information Mart for Intensive Care - III (MIMIC-III) database that would provide an early prediction model for ARDS. Their model could accurately detect onset of ARDS, and had a relatively high predictivity of the condition up to 48 hours before onset[7]. Che *et al.* employed the MIMIC-III database, as well as synthetic data, in the development and testing of a novel Recurrent Neural Network (RNN)-based mortality prediction and classification model. Their GRU-D model is based on the Gated Recurrent Units (GRUs) discussed earlier in this paper, with an trainable decay mechanism and an application of "informed missingness" that take advantage of some of the inherent properties of medical timeseries data (i.e. homeostasis) in order to accommodate missing values[8].

Finally, Punn *et al.* fine-tuned and compared the performance of several current deep neural networks in diagnosing COVID-19 from chest X-ray images. The models were tested for binary classification in order to find out whether COVID-19 is detected or not, as well as for multi-class classification where the model would distinguish between healthy, COVID-19, and pneumonia patients, highlighting the NASNetLarge-based model as superior to the other proposed models [9].

III. Medical and Technological Methods

A. Acute Respiratory Distress Syndrome (ARDS)

ARDS is a medical condition that affects an average of 1-2% of mechanically-ventilated (MV) ICU patients and has a 40% mortality rate [10, 11]. At present, the leading protocol for diagnosing this medical condition is the Berlin definition that defines the onset of ARDS as a prolonged ratio of arterial oxygen potential to fraction of inspired oxygen (P/F ratio) of less than 300 mmHg, and the lower this value is determined to be, the more severe the diagnosis is [12]. Several papers have determined a correlation between early detection of the onset of ARDS and the patient's survival, highlighting the need for early detection and treatment of the condition before the onset of sepsis and multi-organ failure [11, 13, 14]. Also, several MV protocols stabilise and remedy the lung injury at the root of ARDS with the most promising being the 'low tidal volume' and 'high Peak End-Expiratory Pressure (PEEP)' ventilation [13, 15, 16]. However, these procedures depend significantly on the ICU personnel and are considerably subjective to each case. For this reason, an algorithmic approach seamlessly accessible through a platform that provides early warning and informs medical staff of mitigating procedures can be an extremely beneficial tool in the hands of ICU personnel.

B. Large Medical Databases for Scientific Research

The computationally powerful data platform's requirements are driven by our German Smart Medical Information Technology for Healthcare (SMITH)[6] project, with more than seven university hospitals and clinics taking part in it to deploy those solutions in daily medical care. The activities related to the realization of such a platform are part of the Algorithmic Surveillance of ICU patients (ASIC) use case in the SMITH project. The goal of ASIC relies in applying modern technologies to the healthcare system [17]. This use case's specific focus is to work with ARDS-related patient datasets, process them, analyse them, and understand the correlations between the features to predict outcomes from small changes in physiological parameters. The evaluation of the design of such an HPC-driven data science platform requires to access datasets that are very close to real datasets in those clinics. However, using the medical datasets directly from involved SMITH clinics is subject to many regulations (in terms of availability for research and anonymisation requirements), especially for publications. Instead, we take advantage of the freely available ICU patient data provided in the MIMIC-III database, compiled between 2001 and 2012 from patient admissions to the Beth Israel Deaconess Medical Center in Boston, MA [18]. Thus, the procedure is to build and test our platform using patient data from the MIMIC-III database, then verify our results using patient data collected from hospital participating in the SMITH project once available. After the platform and its models are assessed and found useful, the platform is rolled out with developed models for implementation in ICU for real-time usage subject to a more extensive medical certification foreseen in the SMITH project.

[6] https://www.smith.care/

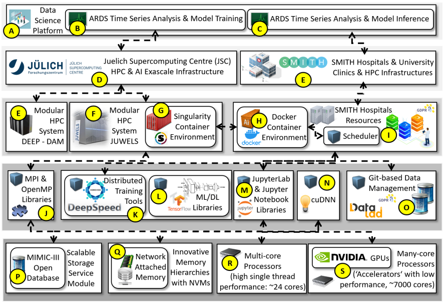

A
Data Science Platform
B
ARDS Time Series Analysis & Model Training
C
ARDS Time Series Analysis & Model Inference
JÜLICH Forschungszentrum
JÜLICH SUPERCOMPUTING CENTRE
Juelich Supercomputing Centre (JSC) HPC & AI Exascale Infrastructure
D
SMITH
SMITH Hospitals & University Clinics & HPC Infrastructures
E
E
Modular HPC System DEEP - DAM
F
JUWELS
Modular HPC System JUWELS
G
Singularity Container Environment
H
Docker Container Environment
docker
SMITH Hospitals Resources
GDPR
Scheduler
I
MPI & OpenMP Libraries
J
Distributed Training Tools
DeepSpeed
K
L
TensorFlow
ML/DL Libraries
JupyterLab & Jupyter Notebook Libraries
M
jupyter
N
cuDNN
Git-based Data Management
GDPR
DataLad
O
MIMIC-III Open Database
P
Scalable Storage Service Module
Q
Network Attached Memory
Innovative Memory Hierarchies with NVMs
Multi-core Processors
R
(high single thread performance: ~24 cores)
NVIDIA GPUs
S
Many-core Processors ('Accelerators' with low performance, ~7000 cores)

Figure 1. HPC-based data science platform design for medical applications for seamless access by non-technical medical experts.

C. Experimental HPC Setup for the Platform

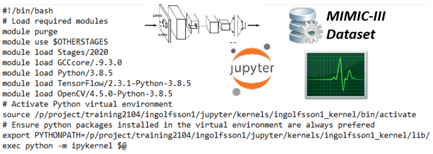
Given the use of sophisticated DL models, our HPC-driven data science platform's computational requirements are high for training models. Simultaneously, hospitals and clinics can run the platform locally with trained models to perform inference (i.e., much less computationally demanding) on real patients in the future. That avoids data transfers of critical patient datasets during the platform models' real usage and is a vital requirement. Our platform's HPC design elements take advantage of the Modular Supercomputing Architecture (MSA) [19] developed by the DEEP series of projects[7]. While the platform's experimental evaluation uses the Data Analytics Module (DAM) module (cf. Table I for selected technology specifications) of the MSA-based DEEP prototype[8], our platform can also leverage the MSA-based JUWELS[9] system to scale to larger models.

TABLE I. Specifications of the DEEP-EST DAM Prototype

CPU	16 nodes with 2x Intel Xeon Cascade Lake
Hardware Acceleration	16 NVIDIA V100 GPU 16 Intel STRATIX10 FPGA PCIe3
Memory	384 GB DDR4 CPU memory /node 32 GB DDR4 FPGA memory /node 32 GB HBM2 GPU memory /node
Storage	2x 1.5 TB NVMe SSD

IV. HPC-based Platform Design Elements

The HPC-based data science platform (see Fig. 1 A) can be seamlessly used by medical experts to perform essentially two different activities that are *'ARDS Time Series Analysis and Model Training'* (see Fig. 1 B) and *'ARDS Time Series Analysis and Model Inference'* (see Fig. 1 C). It is important to understand that the former performs model training on the Juelich Supercomputing Centre (JSC) HPC and AI Exascale infrastructure (see Fig. 1 D), while the latter is performing inference on patients on Hospitals moderate HPC infrastructure (see Fig. 1 E).



```
#!/bin/bash
# Load required modules
module purge
module use $OTHERSTAGES
module load Stages/2020
module load GCCcore/.9.3.0
module load Python/3.8.5
module load TensorFlow/2.3.1-Python-3.8.5
module load OpenCV/4.5.0-Python-3.8.5
# Activate Python virtual environment
source /p/project/training2104/ingolfsson1/jupyter/kernels/ingolfsson1_kernel/bin/activate
# Ensure python packages installed in the virtual environment are always prefered
export PYTHONPATH=/p/project/training2104/ingolfsson1/jupyter/kernels/ingolfsson1_kernel/lib/
exec python -m ipykernel $@
```

Figure 2. Covid-19 Chest X-Ray and ARDS Analysis Platform Environment.

A. Data Science Platform Design Blueprint

For model training our platform used the MSA-based DEEP DAM system (see Fig. 1 F) while we already started to use also the MSA-based JUWELS system (see Fig. 1 G). The deployment of trained models (see Fig. 1 top right) is foreseen to be done using the Singularity[10] container environment on JUWELS (see Fig. 1 H)[11] that is interoperable with Docker-based solutions. We expect to run Docker[12] container environments (see Fig. 1 I) at Hospital computational and storage resources (see Fig. 1 J). The platform uses a git-based data management system called DataLad[13] (see Fig. 1 P) to enable a transparent and trackable access to patient datasets on the premises at the hospital, while also the JSC infrastructure takes advantage of DataLad in context of MIMIC-III datasets stored in the Scalable Storage Service Module (SSSM) [19] of the MSA (see Fig. 1 Q). Hence, the model inference with real patient data will take place within the hospital moderate HPC environment that is not a problem because inference itself is not very much computational expensive (i.e., especially when only some patients of the ICU are daily analysed).

[7] https://www.deep-projects.eu/

[8] https://www.fz-juelich.de/ias/jsc/EN/Expertise/Supercomputers/DEEP-EST/_node.html

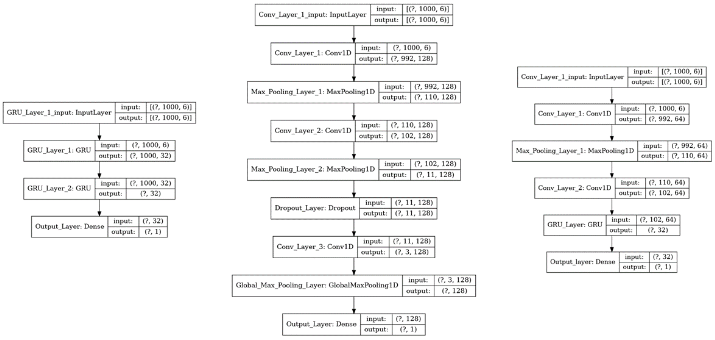
[9] https://www.fz-juelich.de/ias/jsc/EN/Expertise/Supercomputers/JUWELS/JUWELS_node.html

[10] https://singularity.lbl.gov/

[11] https://apps.fz-juelich.de/jsc/hps/juwels/container-runtime.html

[12] https://www.docker.com/

[13] https://www.datalad.org/



a) GRU Model Structure *b) 1D-Conv Model Structure* *c) Mixed Model Structure*

Figure 3. Comparison of the different model structures.

As it is difficult to obtain an *'informed consent'* as outlined in the General Data Protection Regulation (GDPR)[14] from many patients of the ICU our platform deployment approach convinces medical experts.

The platform supports the training of ARDS time series data (see Fig. 1 bottom right) via traditional machine learning models that are using MPI and OpenMP (see Fig. 1 K) via the Cluster modules [19] of the DEEP and JUWELS systems using high single-thread performance CPUs (see Fig. 1 S). Those models are used to exploit a new innovative platform approach of using Network Attached Memory (NAM) [19] for model sharing in teams (see Fig. 1 R) without the need to store data analysis results (e.g., hyper-parameter tuning or *'gridsearch'* results to disk). More notably, deep learning training is supported by offering cutting edge many-core processors and accelerators such as Nvidia GPUs (see Fig. 1 T) as part of the Booster modules [19] of the DEEP and JUWELS systems. The tensor cores of those systems are available by using libraries such as cuDNN (see Fig. 1 O) in conjunction with powerful deep learning libraries (see Fig. 1 M) such as pyTorch[15], TensorFlow[16], and Keras[17]. Our platform is even more powerful when considering that distributed deep learning training is possible via multiple GPUs using tools like Horovod[18] or DeepSpeed[19] which are available as modules within the HPC environment (see Fig. 1 L,N and Fig. 2).

B. Feature Selection with the Platform

Because our platform GUI is based on Jupyter notebook (see Fig. 1 M) it enables a seamless visual interface for medical experts to perform the necessary data preparation steps. Before performing *'feature selection'* on the available data, it is worth noting that of the ~44,000 patients in the original MIMIC-III database, we consider only the 24,947 patients that received mechanical ventilation during their ICU stays. Also, the patient data has many features with missing values and noise. Since we aim to predict missing values, we base our case study approach on the most represented features in our dataset. Hence, we first analyse the patient information (i.e., feature selection), drop the features that have missing values in all records, and try to determine which features have data in most patient records. In this way we also reduce the overall size of our data. Through this approach we find that six features are very well represented: Respiratory Rate (RR), Heart Rate (HR), Systolic Arterial Pressure (SAP), Diastolic Arterial Pressure (DAP), Mean Arterial Pressure (MAP), and Blood Oxygen Saturation (SpO_2). Knowing that (a) the Fraction of Inspired Oxygen (FiO_2) is a ventilator parameter that is set by ICU staff and is automatically recorded whenever it is adjusted and (b) that the Potential of Arterial Oxygen (PaO_2) is directly related to SpO_2, and keeping in mind that our final aim is to assist in the diagnosis of ARDS which is done by calculating the ratio of the two parameters mentioned above (P/F ratio), we centre our approach on predicting values of SpO_2 using our built DL models.

C. Medical Pre-processing Steps with the Platform

The platform key feature is to enable medical experts to interact with platform bringing in their medical expertise without being exposed to the underlying HPC technical difficulties. For example, the medical experts opted to disregard all patients

[14] https://www.eu-patient.eu/globalassets/policy/data-protection/data-protection-guide-for-patients-organisations.pdf

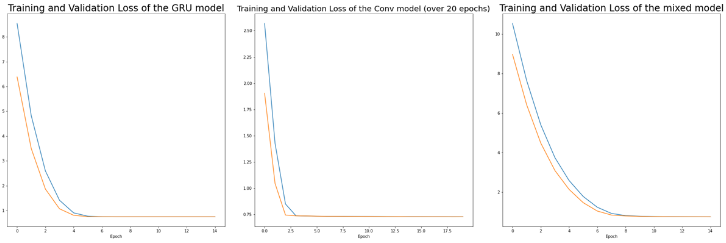
[15] https://pytorch.org/

[16] https://www.tensorflow.org/

[17] https://keras.io/

[18] https://horovod.ai/

[19] https://www.deepspeed.ai/



a) GRU Model Training

b) 1D-Conv Model Training

c) Mixed Model Training

Figure 4. Performance comparison of the different model structures.

having less than 70 recorded timesteps during their ICU stay. That reduced dataset at this point consists of 19,781 patients. Some patients had extremely long records (in the range of tens of thousands of timesteps), although this issue is resolved through down-sampling. The timestamps for the recordings are made consistent by up- or down-sampling each record. Resampling the data resulted in all patient files consisting of 1000 timesteps, although some patients still had columns of missing values which resulted in them being dropped from analysis. At the end of the resampling step the total number of patients available for analysis was 19,769. The resampled data is finally standardised and normalised. At the end of preprocessing all the features are within the same range, centred around 0.

V. Platform Evaluation Case Studies

This section describes the three learning and prediction approaches used to evaluate the usability and performance of the HPC-based platform described in Section IV. To enable a better comparison and understanding of the different structures, we summarize all models together in Fig. 3. Additionally, we present the training and testing performance of these models in Fig. 4.

A. Gated Recurrent Units (GRUs) Approach

The GRU model is built with two GRU layers with 32 units each, with dropout values of 0.2 and both kernel and recurrent regularization, followed by an output layer (Dense layer of size 1). 32 units were chosen for the layers after testing several sizes and tuning for the combination that produced lowest loss value. It is essential to mention here that this hyper-parameter tuning of the different layer structures requires HPC resources for our platform since layers' concrete structure is usually not known. The loss is calculated using the Mean Absolute Error (MAE) function and the optimisation is performed using the ADAM algorithm with a learning rate of 1e-4. Fig. 3 (a) shows the model structure and the shape of the tensors at each layer. The model had a total of 10,209 trainable parameters and was trained for 15 epochs, at which point the loss value stabilised at 0.7432. The training was completed in 405 seconds. This value, multiplied by the standard deviation of the feature in question, equates to an average difference of 1% difference from the expected value of SpO_2. The evolution of the training and validation losses is represented in Fig. 4 (a).

B. One-Dimensional Convolution Approach

Our One-Dimensional Convolution (1D-Conv) model is made up of three convolution layers with 128 filters each and a stride of 9, each followed by a 1D-maxpooling layer, except the last layer where we implement a GlobalMaxPooling1D layer to simulate "flattening" the data before the fully-connected output layer. Global maxpooling is used as it better takes into consideration the structure and sequence of the data than a normal Flatten() layer. To slowdown learning and try to avoid overfitting, we implement L1 and L2 kernel regularization at the input layer, and a 0.5 dropout layer before the final 1D-Conv layer. Also here the HPC-based platform features have been particularly effective in enabling multiple different quick runs to find the best hyper-parameter setups (e.g., dropout value). The structure of the 1D-Conv model is presented in Fig. 3 (b). The built model had a total of 302,337 trainable parameters. For this implementation, the learning rate of the ADAM optimiser was tuned to 5e-5 after several trials. At the end of training, which was completed in 40 seconds, the MAE plateaued at 0.725. The changes in training and validation losses during the 20 epochs are presented in Fig. 4 (b).

C. Mixed Approach

Finally, the hybrid model constitutes two 1D-Conv layers with 64 filters each, followed by a 32 unit GRU layer that leads into the fully-connected output layer. Optimisation during the training of this model was performed with a learning rate of 5e-5 which eventually produced the most promising results. The structure of the model is presented in Fig. 3 (c). This model had a total of 49,889 trainable parameters. This model brings together elements from both approaches described above and produce similar results. It performs much better than the GRU model in terms of speedup, completing training in 45 seconds, while its loss reduction is comparable. The loss results are presented in Fig. 4 (c).

D. Results Discussions

The results presented above highlight One-Dimensional Convolution networks as a promising approach to processing medical sequence data due to it's higher learning rate and better performance in terms of loss reduction. This is evident when we take into consideration that the two other models required more time to complete fewer training steps. That is especially beneficial in research if HPC time is limited. Similarly, these models are much easier to fine-tune and work with as medical experts, as they process the data in a similar fashion to 2D-convolution models. In other words, these initial phases of pre-processing and training are not difficult for the persons undertaking the task of building these models if they have only limited experience in using Convolutional Neural Network (CNN) as medical experts.

Using a pure GRU model for sequence data is a tried and true method, although our experience here made it clear that it is bulky, and quite sensitive to minute changes in parameters. It deserves more attention given that its loss reduction is still somewhat comparable to the convolution model, however its downfall is in the time required to train the model, which only increases as the network grows and becomes more complex.

Although the mixed model performs similarly to the 1D-Conv model in terms of speedup, its results are only as good as the basic GRU model. That is a positive aspect in that it offers the same loss reduction as a GRU model with reduced processing and training time, however it also suffers from the same sensitivity to minor changes in the parameters.

VI. Conclusion

In this paper we presented the design of a HPC-based platform for medical experts to perform analysis of different RNN approaches to analysing medical timeseries data, one purely based on GRUs, one using 1D-Conv, and a hybrid of both technologies. Medical experts have been able to seamlessly use the platform after having some short introduction and avoiding technical details of HPC and AI elements of our platform. The results of the platform case studies highlight that One-Dimensional Convolution as promising method of predicting missing values in time-series data. We can further conclude that for ARDS medical experts still some know-how is needed to understand some of the DL model elements despite the fact that the platform abstracts away all technical difficulties.

The next steps in our research will be to further understand the shortcomings of all three models and improve the data preparation procedures with more significant features with guidance from medical professionals using the platform. Additionally, more experimentation will be done on the available model in terms of increasing filters and the number of layers and observing how that affects the output predictions.

Acknowledgements

This work was performed in the SMITH Project receiving funding via the Medical Informatics Initiative from the German Federal Ministry of Educationand Research (BMBF), the Euro CC, and DEEP-EST projects receiving fund-ing from EU's Horizon 2020 Research and Innovation Framework Programme under grant agreement no. 951740 and no. 754304.

References

[1] J. Deng et al., "ImageNet: A large-scale hierarchical image database," in 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255.

[2] V. Huddar et al., "Predicting complications in critical care using heterogeneous clinical data," IEEE Access, vol. 4, pp. 1–1, January 2016.

[3] H. Sun, Z. Liu, G. Wang, W. Lian, and J. Ma, "Intelligent analysis of medical big data based on deep learning," IEEE Access, vol. 7, pp. 142022–142037, 2019.

[4] W. Wang et al., "Can computer simulators accurately represent the pathophysiology of individual COPD patients?," Intensive Care Medicine Experimental, vol. 2, no. 23, 2014.

[5] J. G. Hardman, N. Bedforth, A. Ahmed, P. Mahajan, and A. R. Aitkenhead, "A physiology simulator: validation of its respiratory components and its ability to predict the patient's response to changes in mechanical ventilation," British Journal of Anaesthesia, vol. 81, no. 3, pp. 327–332, 1998.

[6] A. Das et al., "Creating virtual ARDS patients," in 2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), August 2016, pp. 2729–2732.

[7] S. Le et al., "Supervised machine learning for the early prediction of acute respiratory distress syndrome (ARDS)," Journal of Critical Care, vol. 60, pp. 96–102, 2020.

[8] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, "Recurrent neural networks for multivariate time series with missing values," Scientific Reports, vol. 8, no. 6085, 2018.

[9] N. S. Punn and S. Agarwal, "Automated diagnosis of covid-19 with limited posteroanterior chest x-ray images using fine-tuned deep neural networks," Applied Intelligence, 2020.

[10] D. G. Ashbaugh, D. B. Bigelow, T. sL. Petty, and B. E. Levine, "Acute respiratory distress in adults," The Lancet, vol. 290, no. 7511, pp. 319 – 323, 1967, Originally published as Volume 2, Issue 7511.

[11] J. Villar et al., "The ALIEN study: incidence and outcome of acute respiratory distress syndrome in the era of lung protective ventilation," Intensive Care Medicine, vol. 37, no. 12, pp. 1932–1941, December 2011.

[12] The ARDS Definition Task Force, "Acute Respiratory Distress Syndrome: The Berlin Definition of ARDS," JAMA, vol. 307, no. 23, pp. 2526–2533, June 2012.

[13] A. Das, P. P. Menon, J. G. Hardman, and D. G. Bates, "Optimization of mechanical ventilator settings for pulmonary disease states," IEEE Transactions on Biomedical Engineering, vol. 60, no. 6, pp. 1599–1607, 2013.

[14] S. Kushimoto et al., "Relationship between extravascular lung water and severity categories of acute respiratory distress syndrome by the berlin definition," Critical Care, vol. 17, no. R132, 2013.

[15] A. Das et al., "Evaluation of lung recruitment maneuvers in acute respiratory distress syndrome using computer simulation," Critical Care, vol. 19, no. 8, 2015.

[16] J. G. Laffey et al., "Potentially modifiable factors contributing to outcome from acute respiratory distress syndrome: the LUNG SAFE study," Intensive Care Medicine, vol. 42, pp. 1865–1876, 2016.

[17] A. Winter et al., "Smart medical information technology for healthcare (SMITH)," Methods Inf Med, vol. 27, no. 1, 2018.

[18] A. E. Johnson et al., "MIMIC-III, a freely accessible critical care database," Scientific Data, vol. 3, no. 160035, May 2016.

[19] T. Lippert, N. Eicker, and E. Suarez, Modular Supercomputing Architecture: From Idea to Production, pp. 223–255, Imprint CRC Press, 1st ed. edition, 2019, Contemporary High Performance Computing.

Paper II

Design and Evaluation of an HPC-based Expert System to speed-up Retail Data Analysis using Residual Networks Combined with Parallel Association Rule Mining and Scalable Recommenders

C. Barakat, M. Riedel, S. Brynjólfsson, G. Cavallaro, J. Busch, R. Sedona

https://ieeexplore.ieee.org/document/9596796/, 2021

This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/).

C. Barakat performed the platform preparation, data cleaning, and model development for the retail work described in this paper in cooperation with J. Busch.

Design and Evaluation of an HPC-based Expert System to speed-up Retail Data Analysis using Residual Networks Combined with Parallel Association Rule Mining and Scalable Recommenders

C. Barakat*†, M. Riedel*†, S. Brynjólfsson*, G. Cavallaro†, J. Busch†, and R. Sedona*†
* School of Engineering and Natural Sciences, University of Iceland, Iceland
† Jülich Supercomputing Centre, Forschungszentrum Jülich, Germany
c.barakat@fz-juelich.de, morris@hi.is, sb@hi.is, g.cavallaro@fz-juelich.de,
j.busch@fz-juelich.de, r.sedona@fz-juelich.de

***Abstract*—Given the Covid-19 pandemic, the retail industry shifts many business models to enable more online purchases that produce large transaction data quantities (i.e., big data). Data science methods infer seasonal trends about products from this data and spikes in purchases, the effectiveness of advertising campaigns, or brand loyalty but require extensive processing power leveraging High-Performance Computing to deal with large transaction datasets. This paper proposes an High-Performance Computing-based expert system architectural design tailored for *'big data analysis'* in the retail industry, providing data science methods and tools to speed up the data analysis with conceptual interoperability to commercial cloud-based services. Our expert system leverages an innovative Modular Supercomputer Architecture to enable the fast analysis by using parallel and distributed algorithms such as association rule mining (i.e., FP-Growth) and recommender methods (i.e., collaborative filtering). It enables the seamless use of accelerators of supercomputers or cloud-based systems to perform automated product tagging (i.e., residual deep learning networks for product image analysis) to obtain colour, shapes automatically, and other product features. We validate our expert system and its enhanced knowledge representation with commercial datasets obtained from our ON4OFF research project in a retail case study in the beauty sector.**

Keywords*—High-Performance Computing; Expert Systems; Parallel and Distributed Algorithms; Retail data analysis; Accelerators; Deep Learning

I. Introduction

With the advent of online marketplaces, it has become easier to reach a broad customer base for many retailers and collect relevant information concerning shopping habits and seasonal trends. That became especially clear during the government-imposed lockdowns that came as a response to the Covid-19 pandemic. A wide variety of commercial sectors report a significant increase in online purchases since then. With this growth comes a benefit for data scientists as more information becomes available in a digital format, making it easier to extract, process, and analyse retail datasets. That is valid for both sales data from individual shopping transactions, and for image data of the products themselves to support a proper online presentation of products in online marketplaces.

On the other hand, the availability and abundance of collected datasets represent a significant challenge for store managers and data scientists being overwhelmed by the technical complexity of the *'big data analysis'*. That can be at least partly explained by the fact that the *'big data analysis'* requires scalable storage, memory, and computing at the intersection of traditional Data Mining (DM) and cutting-edge Machine Learning (ML) and Deep Learning (DL) algorithms. There is a need for an expert system that includes new forms of knowledge representations to seamlessly enable retail data analysis using cutting-edge technology.

This paper presents an architecture blueprint for such an expert system designed explicitly for retail applications (e.g., product image tagging, shop product placements, retail product recommendations). We address the challenging *'big data analysis'* requirements by exploiting cutting edge technologies such as modular High-Performance Computing (HPC) systems, containers, accelerators, Jupyter[1] notebooks, and open-source software stacks. Our lessons learned from the ON4OFF[2] retail project in using Europe No. 1 supercomputer Jülich Wizard for European Leadership Science (JUWELS)[3] and modular Dynamical Exascale Entry Platform (DEEP)[4] supercomputing prototypes complement this paper's general technical approach with practice and experience. By not losing sight of the interoperability with commercial cloud vendors, we outline conceptual pathways to encourage a broader uptake by retailers that require alternatives for academically-driven HPC infrastructures.

The remainder of this paper is structured as follows. Related work is reviewed in Section II followed by Section III where a summary of the DM and ML methods used to realise our expert system is presented. Section IV introduces the expert system architectural design, including its main features and necessary data preparation steps, including unique aspects of

This work was performed in the ON4OFF Project receiving funding from the EFRE.NRW programme and the Euro CC and DEEP-EST projects receiving funding from EU's Horizon 2020 Research and Innovation Framework Programme under grant agreement no. 951740 and no. 754304 respectively.

[1] https://jupyter.org/

[2] https://www.on-4-off.de/

[3] https://www.fz-juelich.de/ias/jsc/EN/Expertise/Supercomputers/JUWELS

[4] https://www.deep-projects.eu/

the HPC-based implementation. Finally, this paper ends with a brief summary and some concluding remarks.

II. Related Work

There is a lot of related research in association rule mining like by Sağın et al. in [1] to identify product groups sold together by DM transaction data of a hardware retailer. For this purpose, they grouped the products in categories and subcategories. Their approach was based on using DM software to extract association rules using both FP-Growth and Apriori. Kumar et al. [2] analyses the different state of the art parallel and distributed methods for DM. In addition to frequent itemset mining, they also investigate those methods regarding high utility itemset mining, uncertain itemset mining, and sequence pattern mining. They also analyse the difficulties in regards to big data and DM. Gassama et al. [3] developed a parallel approach to the FP-Growth algorithm, using the Apache Spark framework's in-memory computing capabilities. They compared their approach to the implementation of Apache Mahout[5] to evaluate the scalability of their algorithm. Min et al. [4] propose an FP-Growth algorithm, Grided FP-Growth, optimized for usage on a cluster. The Grided FP-Growth approach foregoes the construction of FP-trees and finds conditional pattern bases using the projection method, therefore preventing memory overflow problems. The processing of the conditional pattern bases is divided into subtasks and distributed to multiple nodes. They show that the Grided FP-Growth algorithm has better scalability and shorter computation time than FP-Growth. Venkatachari et al. [5] used the FP-Growth and Apriori algorithms to identify correlations between items in a grocery store. The goal was to use the gained knowledge to develop marketing strategies. They also compared the computational speed of generating frequent itemsets using the FP-Growth and Apriori algorithms. Khader et al. [6] analyses pharmacy data using association rule mining to help improve pharmacy management strategies. They compare a sequential and parallel approach to FP-Growth to find the system with the shortest execution time. They also analyse the difference between using data with only transactions that include more than one item and data that include single item purchase transactions.

Winlaw et al. [7] highlighted the effectiveness of using Nonlinear Conjugate Gradient (NCG) wrappers in Alternating Least Square (ALS)-based collaborative filtering algorithms in serial and parallel applications. They show a speedup and scalable performance that is also compatible with cloud-based environments such as Apache Spark. Jiang et al. [8] proposed a method to scale-up item-based collaborative filtering using Map-Reduce by distributing the compute-intensive components of the data analysis over parallel resources of an Apache Hadoop[6] cluster. They also implemented a partitioning method to reduce the communication cost as the dataset's size increases.

Loureiro et al. [9] used transaction data from a fashion retailer and a DL approach to predict sales numbers of clothing items. For the DL model, they used multiple physical characteristics of the items as well as opinions of domain experts. They also used traditional ML methods such as Support Vector Regression (SVR) and compared to the results of their DL approach. Advani et al. [10] developed an inference DL model for visual object recognition that is similar to our approach. In contrast, their DL model is used to identify relationships between objects in a given scene in retail environments and, therefore, add visual context. Fuchs et al. [11] analyses the potential of using Convolutional Neural Network (CNN) for object classification and multi-product object detection. They train the DL model with images of vending machines. They suggest running the image detection on mixed reality headsets which could provide customers with valuable information about products.

III. Data Analysis Methods

The *'association rule mining'* method is used to uncover underlying relationships between different retail products in transactions [12]. It identifies *'frequent itemsets'* that are items frequently occurring together in the transactions (i.e., products that customers often buy together). After fine-tuning parameters (i.e., support, confidence, and lift), the method result is typically a set of rules that can predict more products of interest for customers that already picked one or more certain products. As seen in Fig. 1, one example of applications that take advantage of these rules can be shop product placements or *'not personalized product recommendations'*. Two of the most commonly used algorithms are Apriori [13] and Frequent Pattern Growth (FP-Growth) [12] while the latter is more scalable since it uses a tree-based approach.

In contrast, the *'collaborative filtering'* [14] method identifies *'personalized product recommendations'* (cf. Fig. 1) out of a given transaction list using *'embeddings'*. The *'embedding space'* is an abstract representation common to both products and customers, in which similarity or relevance using a similarity metric is measured. Algorithms can learn *'embeddings'* automatically, which is the power of those models. Customers with similar preferences will be close together (i.e., a recommendation of a product to customer A based on the interests of a similar customer B). Used algorithms are *'matrix factorization (MF) models'* or *'singular value decomposition (SVD)'* while a comprehensive survey is given in [14].

Another processing-intensive application relevant for retailers is *'product image tagging'* (cf. Fig. 1) using innovative *'deep learning (DL)'* methods. *CNNs* are widely known for their effectiveness in analysing image data [15]. That is mostly due to 'CNN's' ability to uncover local patterns within a product image rather than trying to extract features from the whole as would be the case with traditional Artificial Neural Network (ANN)s [16]. But increasingly using deep layers is not limitless and gives rise to the *'vanishing gradient problem'* where the network will no longer train properly after reaching a certain complexity [16]. Another DL model

[5] https://mahout.apache.org/
[6] https://hadoop.apache.org/

to overcome this problem is a *'residual network'* [17] that introduces residual blocks within the network structure, which bypass convolutional layers during training and prevent the gradient from decreasing to zero.

We performed all data analysis with our HPC-enabled Jupyter-JSC[7] with JupyterLab (see Fig. 1 O), a Web-based interactive development environment for Jupyter notebooks.

IV. Expert System Architectural Design and Selected Implementation Details

Fig. 1 shows our architectural design of an expert system driven by three concrete retail applications that require different types of datasets and features in the data. The innovative HPC-based design implementation is realized by our modular supercomputing architecture (MSA) [18] that was developed in the last decade during the course of the DEEP series of projects. To use our HPC-based expert system for *'product image tagging'* (see Fig. 1 A), the retailer needs to access high-resolution images from products. But our experience working with many retailers in the German ON4OFF research project reveals that the availability of product image data is often limited. Reasons are copyright issues with the original brand owners or only too little number of pictures of products. Our practical experience also reveals that this limiting number of product images is challenging for offering sophisticated Graphical User Interfaces (GUI)s in our expert system in cases where product recommendations may not have a picture. As shown in Fig. 3, we used 6,585 product images of the beauty sector (e.g., perfume bottles, lipsticks, etc.) for our implementation evaluation example. Storing high-resolution image data raises the demand for a Scalable Storage Service Module (SSSM) [18] of our MSA architecture implemented in our Jülich Supercomputing Centre (JSC) infrastructure via the Lustre[8] parallel file system (see Fig. 1 B).

To use our expert system for *'shop product placements'* (see Fig. 1 C), the retailer needs *'simple transaction data'* without a vital link to customers (i.e., shopping baskets only). The input to association rule mining algorithms (see Fig. 1 D) require no customers' identity, and their rules are not personalized, thus making this method more suitable to optimize a store setup. Our research project example of this optimization is to position often brought together products at entirely different locations within the store to keep the customers as long as possible in a corresponding shop (i.e., expecting revenues by buying more products). As shown in Fig. 3, our expert system evaluation is using 8,139,215 transactions.

While association rule mining is also applicable in our example application of *'retail product recommendations'* (see Fig. 1 E) for customers, our experience reveals that personalized recommendation techniques like collaborative filtering (see Fig. 1 D) are more effective. However, these techniques require distinct customer IDs as part of the transactions. As such, the data needs to be often anonymized, making it difficult to include in the data analysis other relevant customer features (e.g., the street address of cities with high vs low-income regions). Our research reveals that stores often do not even have a system to link cashier transactions in stores to unique customer IDs. Only online shops and the use of (optional) customer loyalty cards overcomes this challenge, but also leads to a significant reduction of usable transactions.

A. Computational Infrastructure for Data Analysis

All DM and ML implementations of the HPC-based expert system are encapsulated as different services and hosted in a professional Service-Oriented Architecture (SOA) landscape (see Fig. 1 F) by our ON4OFF project partner Adesso[9]. While Adesso is a professional company able to host future commercial settings off the service landscape after the ON4OFF research project is over, one goal of the HPC-based expert system design is to enable conceptual interoperability with commercial cloud vendors (see Fig. 1 G). That enables decoupling the computing infrastructure from the services implemented in the project. Migration to Cloud services hosted by Amazon Web Services (AWS), MS Azure, or the Google Cloud is possible. Also, the SOA-based architecture design and this interoperability further enable retailers to complement the academically created service landscape with existing commercial services (e.g., AWS Sagemaker[10]).

Moderate size HPC systems are also available in Clouds today. But our proof-of-concept architecture implementation of the expert system uses academically-driven HPC systems to keep the research project's costs to a considerable level. It also enables us to fully exploit and conduct research with our unique MSA approach [18] since we use the DEEP modular supercomputer (see Fig. 1 H) in our studies and the JUWELS modular supercomputer (see Fig. 1 I). The DEEP Data Analytics Module (DAM)[11] comprises 16 nodes, each with 2 Intel Xeon Cascade Lake CPUs, 1 NVIDIA V100 Graphics Processing Unit (GPU), 1 Intel STRATIX10 Field-Programmable Gate Array (FPGA), and 446 GB of RAM, as well as a total of 2 TB of Non-Volatile Memory (NVM). Hence, with an aggregated 32 TB of NVM, this HPC module design is primarily driven to support big data analytics stacks like Apache Spark[12] (see Fig. 1 M) that require a high amount of memory to work fast. The module also has access to the SSSM module (see Fig. 1 B) of the cluster to support large-scale datasets and keep the local DAM storage available for memory-intensive applications. The JUWELS supercomputer, currently the fastest supercomputer in Europe and 7th fastest worldwide[13], consist of 2,583 and 940 nodes respectively, totalling 122,768 CPU cores and 224 GPUs in the cluster module, and 45,024 CPU cores and 3,744 GPUs in the booster module.

[7] https://jupyter-jsc.fz-juelich.de/hub

[8] https://www.lustre.org/

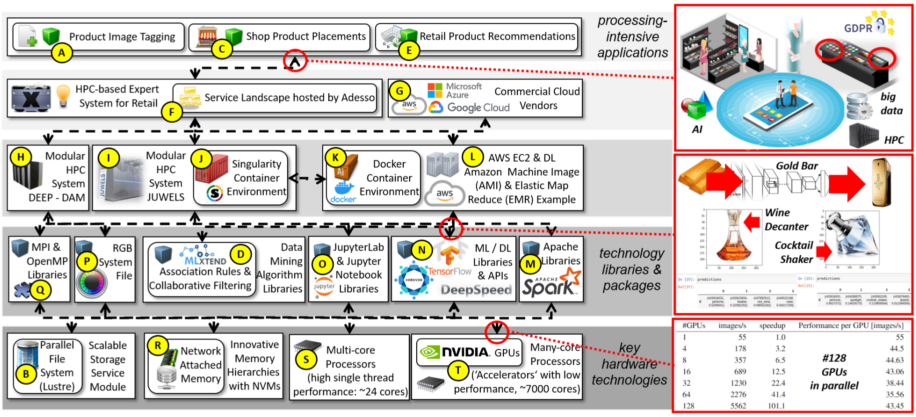
[9] https://www.adesso.de/en/index.jsp

[10] https://aws.amazon.com/sagemaker/

[11] https://www.fz-juelich.de/ias/jsc/EN/Expertise/Supercomputers/DEEP-EST

[12] https://spark.apache.org/

[13] https://www.top500.org/lists/top500/2020/11/


Product Image Tagging
A
Shop Product Placements
C
Retail Product Recommendations
E
processing-intensive applications
HPC-based Expert System for Retail
Service Landscape hosted by Adesso
F
G
aws
Microsoft Azure
Google Cloud
Commercial Cloud Vendors
H
Modular HPC System DEEP - DAM
I
JUWELS
Modular HPC System JUWELS
J
Singularity Container Environment
K
docker
Docker Container Environment
L
AWS EC2 & DL Amazon Machine Image (AMI) & Elastic Map Reduce (EMR) Example
MPI & OpenMP Libraries
Q
RGB System File
P
MLXTEND
D
Association Rules & Collaborative Filtering
Data Mining Algorithm Libraries
JupyterLab & Jupyter Notebook Libraries
O
jupyter
N
TensorFlow
HOROVOD
DeepSpeed
ML / DL Libraries & APIs
Apache Libraries
M
APACHE Spark
technology libraries & packages
Parallel File System (Lustre)
B
Scalable Storage Service Module
R
Network Attached Memory
Innovative Memory Hierarchies with NVMs
S
Multi-core Processors (high single thread performance: ~24 cores)
NVIDIA. GPUs
T
Many-core Processors ('Accelerators' with low performance, ~7000 cores)
key hardware technologies
GDPR
AI
big data
HPC
Gold Bar
Wine Decanter
Cocktail Shaker
In [37]: predictions
Out[37]:
(n03916031, perfume, 0.5255641)
(n02815834, beaker, 0.22582252)
(n07892512, red_wine, 0.08552182)
(n04522168, vase, 0.04217156)
In [15]: predictions
Out[15]:
(n03916031, perfume, 0.6627372)
(n04286575, spotlight, 0.14629175)
(n03062245, cocktail_shaker, 0.11080584)
(n03676483, lipstick, 0.012394556)
#128 GPUs in parallel

#GPUs	images/s	speedup	Performance per GPU [images/s]
1	55	1.0	55
4	178	3.2	44.5
8	357	6.5	44.63
16	689	12.5	43.06
32	1230	22.4	38.44
64	2276	41.4	35.56
128	5562	101.1	43.45

Figure 1. Architecture and design of a HPC-based expert system for retail being interoperable with commercial cloud vendors.

B. Product Image Tagging Application Evaluation

Customers of beauty stores participating in our ON4OFF project often ask salesclerks in the shops about specific shapes and colors of products (e.g., perfume bottle shapes and colors) without knowing the particular brands. Based on our research in the ON4OFF project, Fig. 2 shows selected examples of perfume bottles that are often asked for by customers given their unique characteristics. The basic knowledge representation of product data does not include those shapes and colors in the perfume producers' description. It focuses on fragrance strengths and types (e.g., fresh, floral, oriental, woody) that are sometimes even unknown by customers, especially when they search for new products never tried before. Our expert system GUI offers an advanced knowledge representation of the products available in a beauty store to salesclerks in the beauty shop to address the above-described problem.

As shown in Fig. 3, our expert system's *'Image Tagging Service'* (i.e., hosted in the Adesso service environment, see Fig. 1 F) consists of the colour detection module and shape detection module. It extracts information about the colour and shape of perfume products to produce previously unknown tags for these items (e.g., see Fig. 2 teddy bear) and store them together with other relevant product data, thus improving the search functionality in the expert system GUI. A salesclerk in the shop using our expert system with an electronic tablet facing a customer can now enter search strings that represent product shapes and colors. This approach works not only for perfume bottle examples, whereby perfume still represents the significant product portfolio of our use case in the beauty sector. Hence, it is essential to understand on the technical perspective that our implementation of the colour and shape detection modules are not customer nor salesclerk facing. Instead, our implementation using cutting-edge DL techniques enriches the product database to enable a better product knowledge representation, search, and management.

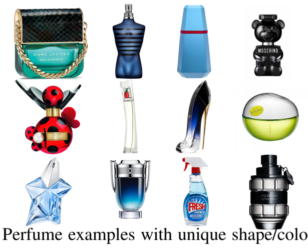


Figure 2. Perfume examples with unique shape/color features.

Our image tagging service shown in Fig. 3 consists of two separate modules that take the same input image data (i.e., 6,585 images of beauty products) and provide output to different service landscape interfaces. For the colour detection module to process the input, it applies a mask to the image and removes the background information, then converts it from the RGB (Red, Green, Blue) colour space to the HLS (Hue, Lightness, Saturation) colour space. Then the module flattens the data from a three-dimensional array to a single dimension. It then applies *'K-means clustering'* [12] to the resulting matrix with K=10, and the clusters are sorted by size, with the largest clusters representing the most frequently observed colours in the image. Knowing that most Linux distributions consist of an RGB colour reference system file (see Fig. 1 P), the obtained clusters' centroids are compared to an HLS conversion of that reference file. The nearest colours' names and HLS values to those centroids are returned to the service landscape via the Colour Tagging Interface of an Adesso landscape service updating the product database.

Our second *'shape detection module'* takes advantage of *'transfer learning'* [19], where a pre-trained network de-

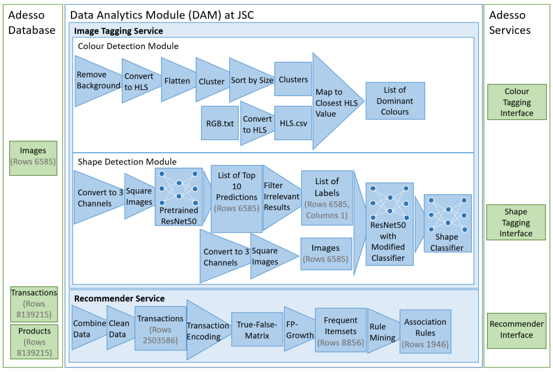


Figure 3. HPC-based expert system modules with implementation details about the different services and data analysis steps.

veloped to solve one problem is applied to solve a new, somewhat similar situation. Describing the shape of the products represented in the available product images is done by feeding the pre-processed images illustrated in Fig. 3 to a pre-trained ResNet-50 model [17] and collecting the top 10 decoded predictions for each image. These predictions are then post-processed to retain relevant shape descriptions (e.g., perfume, water bottle, website, lipstick... being subsequently dropped being too specific or too abstract). Further trimming is done on the list to select only one descriptive label for each image finally. We are using these labels and the original images to train a new ResNet-50 model with a reconstructed classification layer. Finally, the retrained model is stored on the service landscape as part of the Shape Tagging Interface (see Fig. 3).

The model ResNet-50 is available in DL packages available in our HPC module environment (i.e., Keras[14], TensorFlow[15], see Fig. 1 N) on JUWELS and DAM. Our experience reveals that python scripts from Keras and TensorFlow can be quickly migrated into clouds if needed by using the AWS EC2 combined with the Amazon Machine Images (AMI) that also offer images with the same set of DL packages (see Fig. 1 L). We also achieve interoperability by using container technologies such as Docker[16] (see Fig. 1 K) in Clouds and Singularity[17] on JUWELS (see Fig. 1 J) that can work with Docker files too[18]. Our approach is scalable to large quantities of product image data since we can use up to 128 GPUs on JUWELS (cf. Sedona et al. [20]) using Horovod[19] and DeepSpeed[20] (see Fig. 1 N). In terms of speedup during the re-training process of the ResNet-50-based image tag generator, we find that running the build, training, and validation scripts required a total of 603 seconds to complete using only the CPUs available in the HPC environment, while it completed within 70 seconds when using the available GPUs. This confirms the need to use the computational power of HPC resources in these applications, as well as the importance of applying the right tools for the right task at hand.

Finally, Horovod is using the Message Passing Interface (MPI) (see Fig. 1 Q) to communicate between GPUs. However, our experience reveals that scaling in commercial clouds is still challenging when using cutting-edge GPU types (see Fig. 1 T) required for DL because of high costs (e.g., AWS EC2 24 USD per hour rate for V100). For that reason we foresee that data manipulation and DL approaches can be performed using the modules on the HPC cluster, and migrating association rule and collaborative filtering scripts to the cloud.

C. Shop Product Placements Application Evaluation

Fig. 3 shows the implementation details and data analysis steps of our *'Recommender Service'* in our expert system. We use the FP-Growth MLxtend[21] library implementation (see Fig. 1 D) on our DAM HPC system with powerful CPUs (see Fig. 1 S) and take advantage of the large memory node setup, so that transaction data fits into memory. That can be combined by using new types of memory hierarchies that go beyond NVM using an innovative Network Attached Memory (NAM) [18] in DEEP (see Fig. 1 R). Larger transaction data can take advantage of Apache Spark (see Fig. 1 M) on the large-memory DAM nodes using the MLlib implementation of FP-Growth [22]. That also enables another conceptual interoperability with clouds since most offer Apache Spark with MLlib as part of their Hadoop ecosystem services (e.g., AWS Elastic Map Reduce service, see Fig. 1 L) too. We use the

[14] https://keras.io/
[15] https://www.tensorflow.org/
[16] https://www.docker.com/
[17] https://singularity.lbl.gov/
[18] https://apps.fz-juelich.de/jsc/hps/juwels/container-runtime.html
[19] https://horovod.ai/
[20] https://www.deepspeed.ai/
[21] http://rasbt.github.io/mlxtend/
[22] https://spark.apache.org/docs/latest/ml-frequent-pattern-mining.html

resulting rules in collaboration with store managers for product placements in beauty stores and the expert system GUI.

D. Product Recommendation Application Evaluation

Given the page restriction, we do not provide a detailed implementation for our *'Personalized Recommender Service'* in Fig. 3, because many steps overlap with those from our illustrated (unpersonalized) *'Recommender Service'*. In our approach we find that the data cleaning and association rule generation are only possible on the HPC cluster as the available memory was able to hold the generated one-hot encoded matrix of the 2.8 million transactions (size 289 GB). There is only a slight change in the number of transactions since we only use those with an associated CustomerID (i.e., beauty store loyalty cardholders). Instead of FP-Growth, we use the SVD algorithm mentioned above to perform personalized recommendations via collaborative filtering [14]. We use both Surprise[23] and the Apache Spark MLlib implementations of SVD [24] on our DAM nodes and in commercial clouds.

V. Conclusions

Based on the thorough survey of related work, we conclude that our HPC-based expert system is unique because it offers a comprehensive approach of seamlessly working with parallel and scalable methods on large quantities of data from the retail sector. Although there is a lot of research on parallel and scalable algorithms (e.g., FP-Growth, Apriori, DL), we observe that most retail-based solutions are instead purely algorithm-oriented or they do not offer an open-source solution like in our approach. We can further conclude that we do not validate our approach with only synthetic data as seen in many other algorithm-based retail systems, but on real retail data in the beauty sector through the ON4OFF project including real stores and store managers. While some related work addresses parallel and scalable methods, we are confident that our unique HPC-based expert system can scale to a very high amount of product image datasets or a very high number of transactions. Our approach also enables interoperability with Cloud-based systems and complements the existing services with other services relevant for retail and reusing this approach in other retail sectors such as bikes, wines, or book shops under the umbrella of the ON4OFF project. Another challenging future work is to enable reinforcement learning in our expert system that learns over time (i.e., through rewards and punishment) what recommender systems are performing good or bad in certain stores.

References

[1] A. N. Sağın et al., "Determination of association rules with market basket analysis: An application in the retail sector," Southeast Europe Journal of Soft Computing, vol. 7, no. 1, pp. 10–19, 2018.

[2] S. Kumar et al., "A review on big data based parallel and distributed approaches ofpattern mining," Journal of King Saud University – Computer and Information Sciences, 2019.

[3] A. D. dite Gassama et al., "S-FPG: A parallel version of FP-Growth algorithm under apache spark," in 2017 IEEE 2nd Int. Conference on Cloud Computing and Big Data Analysis (ICCCBDA), pp. 98–101.

[4] M. Chen et al., "An efficient parallel FP-Growth algorithm," in 2009 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, 2009, pp. 283–286.

[5] K. Venkatachari et al., "Market basket analysis using fp-growth and apriori algorithm: A case study of mumbai retail store," BVIMSR's Journal of Management Research, vol. 8, no. 1, pp. 56–63, 2016.

[6] N. Khader et al., "The performance of sequential and parallel implementations of fp-growth in mining a pharmacy database," in Proceedings of the 2015 Industrial and Systems Engineering Research Conference.

[7] M. Winlaw et al., "Algorithmic acceleration of parallel als for collaborative filtering: Speeding up distributed big data recommendation in spark," in 2015 IEEE 21st International Conference on Parallel and Distributed Systems (ICPADS), 2015, pp. 682–691.

[8] J. Jiang et al., "Scaling-up item-based collaborative filtering recommendation algorithm based on hadoop," in IEEE World Congress on Services, 2011, pp. 490–497.

[9] A. L. D. Loureiro et al., "Exploring the use of deep neural networks for sales forecasting in fashion retail," Decision Support Systems, vol. 114, pp. 81–93, 2018.

[10] S. Advani et al., "Visual co-occurrence network: Using context for large-scale object recognition in retail," in 2015 13th IEEE Symposium on Embedded Systems For Real-time Multimedia (ESTIMedia), pp. 1–10.

[11] K. Fuchs et al., "Towards identification of packaged products via computer vision: Convolutional neural networks for object detection and image classification in retail environments," in 9th International Conference on the Internet of Things (IoT 2019), 2019, pp. 1–8.

[12] P.-N. Tan et al., Introduction to Data Mining, Pearson, 2005.

[13] M. S. Mythili et al., "Performance evaluation of Apriori and FP-Growth algorithms," International Journal of Computer Application, vol. 79, no. 10, October 2013.

[14] D. Bokde et al., "Matrix factorization model in collaborative filtering algorithms: A survey," Procedia Computer Science, vol. 49, pp. 136–146, 2015, Proceedings of 4th International Conference on Advances in Computing, Communication and Control (ICAC3'15).

[15] J. Schmidhuber, "Deep learning in neural networks: An overview," Neural Networks, vol. 61, pp. 85–117, January 2015.

[16] F. Chollet, Deep Learning with Python, Manning, Shelter Island, NY, USA, 1st ed. edition, 2018.

[17] K. He et al., "Deep residual learning for image recognition," December 2015.

[18] T. Lippert et al., Modular Supercomputing Architecture: From Idea to Production, pp. 223–255, Imprint CRC Press, 1st ed. edition, 2019, Contemporary High Performance Computing.

[19] S. J. Pan and Q. Yang, "A survey on transfer learning," IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 10, pp. 1345–1359, October 2010.

[20] R. Sedona et al., "Scaling up a Multispectral RESNET-50 to 128 GPUs," in IEEE International Geoscience and Remote Sensing Symposium (IGARSS), 2020, to appear.

[23] http://surpriselib.com/

[24] https://spark.apache.org/docs/latest/mllib-dimensionality-reduction.html

Paper III

Lessons learned on using High-Performance Computing and Data Science Methods towards understanding the Acute Respiratory Distress Syndrome (ARDS)

C. Barakat, S. Fritsch, K. Sharafutdinov, G. Ingólfsson, A. Schuppert, S. Brynjólfsson, M. Riedel

https://ieeexplore.ieee.org/document/9803320/, 2022

This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/).

C. Barakat performed the platform preparation, data cleaning, and model development for the ARDS work described in this paper, provided guidance for the COVID-19 work done by G. Ingólfsson, and drafted the original manuscript.

MIPRO 2022, May 23 - 27, 2022, Opatija, Croatia

Lessons learned on using High-Performance Computing and Data Science Methods towards understanding the Acute Respiratory Distress Syndrome (ARDS)

C. Barakat[*][†], S. Fritsch[†][‡], K. Sharafutdinov[§], G. Ingólfsson[*],
A. Schuppert[§], S. Brynjólfsson[*], M. Riedel[*][†]
[*] School of Engineering and Natural Sciences, University of Iceland, Iceland
[†] Jülich Supercomputing Centre, Forschungszentrum Jülich, Germany
[‡] Department of Intensive Care Medicine, University Hospital RWTH Aachen, Germany
[§] Joint Research Centre for Computational Biomedicine, RWTH Aachen, Germany
c.barakat@fz-juelich.de, sfritsch@ukaachen.de, ksharafutdin@ukaachen.de, gii2@hi.is,
aschuppert@ukaachen.de, sb@hi.is, morris@hi.is

Abstract*—Acute Respiratory Distress Syndrome (ARDS), also known as noncardiogenic pulmonary edema, is a severe condition that affects around one in ten-thousand people every year with life-threatening consequences. Its pathophysiology is characterized by bronchoalveolar injury and alveolar collapse (i.e., atelectasis), whereby its patient diagnosis is based on the so-called 'Berlin Definition'. One common practice in Intensive Care Units (ICUs) is to use lung recruitment manoeuvres (RMs) in ARDS to open up unstable, collapsed alveoli using a temporary increase in transpulmonary pressure. Many RMs have been proposed, but there is also confusion regarding the optimal way to achieve and maintain alveolar recruitment in ARDS. Therefore, the best solution to prevent lung damages by ARDS is to identify the onset of ARDS which is still a matter of research. Determining ARDS disease onset, progression, diagnosis, and treatment required algorithmic support which in turn raises the demand for cutting-edge computing power. This paper thus describes several different data science approaches to better understand ARDS, such as using time series analysis and image recognition with deep learning methods and mechanistic modelling using a lung simulator. In addition, we outline how High-Performance Computing (HPC) helps in both cases. That also includes porting the mechanistic models from serial MatLab approaches and its modular supercomputer designs. Finally, without losing sight of discussing the datasets, their features, and their relevance, we also include broader selected lessons learned in the context of ARDS out of our Smart Medical Information Technology for Healthcare (SMITH) research project. The SMITH consortium brings together technologists and medical doctors of nine hospitals, whereby the ARDS research is performed by our Algorithmic Surveillance of ICU (ASIC) patients team. The paper thus also describes how it is essential that HPC experts team up with medical doctors that usually lack the technical and data science experience and contribute to the fact that a wealth of data exists, but ARDS analysis is still slowly progressing. We complement the ARDS findings with selected insights from our Covid-19 research under the umbrella of the European Open Science Cloud (EOSC) fast track grant, a very similar application field.

Keywords*—*High-Performance Computing; Acute Respiratory Distress Syndrome; modular supercomputing; data science platform; machine learning

I. Introduction

In their survey on the global impact of respiratory disease, the World Health Organization (WHO) highlighted the lungs' vulnerability to external disease vectors, and described the broad range of life-threatening conditions that can occur as a result of such exposures [1]. These conditions endanger the pathways through which the body collects oxygen and drains carbon dioxide, and would benefit greatly from early treatment, leading to more positive outcomes for patients. Generally speaking, diseases of the respiratory system can be either directly related to trauma or infection to the airways and lungs, or deferred through the failure of other organs (cardiovascular conditions, multi-organ failure). In the specific case of infections, part of the respiratory system can be affected and there is a generally observed distinction between upper respiratory tract infections (URTI) affecting the airways above the glottis and usually more benign, and lower respiratory tract infections (LRTI) where the condition can quickly become life-threatening [2]. One specific condition that affects a large fraction of mechanically-ventilated (MV) Intensive Care Unit (ICU) patients is Acute Respiratory Distress Syndrome (ARDS). It was first referred to in the literature by Ashbaugh *et al.* and has since been the subject of much research in order to determine means of diagnosis and treatment [3]. This condition is especially dangerous as it has a relatively high mortality rate, while early detection is generally associated with more positive outcomes for the patients [4, 5].

With the onset of the Covid-19 pandemic, it became clear that fast and accurate methods for diagnosis and prediction of disease progression are vital for hospitals as they strain under the large number of incoming patients. Seeing as infection with the Severe Acute Respiratory Syndrome coronavirus 2 (SARS-CoV-2) virus leads to a condition that is a similar application field to the work we are performing in ARDS prediction, we use the available resources and expertise to advance some work done in chest X-ray image analysis and attempt to expand it into new data provided under partnerships within the European Open Science Cloud.

The work described herein takes advantage of the information gained through work conducted on ARDS patient data as well as the collected knowledge of Covid-19 progression, the Modular Supercomputing Architecture (MSA) hardware resources available at the Jülich Supercomputing

Authorized licensed use limited to: NATIONAL UNIVERSITY OF ICELAND. Downloaded on June 28,2022 at 13:07:42 UTC from IEEE Xplore. Restrictions apply.

Centre (JSC), the previously developed High-Performance Computing (HPC)-enabled expert system [6], and the collaboration and collected expertise of Machine Learning (ML) specialists, medical doctors, data analysts, and statisticians to fulfill several goals as part of the overarching Smart Medical Information Technology for Healthcare (SMITH) project spanning several medical and research institutions in Germany, under the guidance of the Federal Ministry of Education and Research (BMBF) [7, 8]. The goals we set out to reach include (i) understanding ICU medical data made available through the collaboration between university clinics, (ii) using a patient simulator, made available by project partners, to generate output data that determine outcomes of patients based on selected inputs [9, 10], (iii) to leverage the available HPC and MSA resources at JSC to parallelise and optimise the process, (iv) to design, develop, train, and evaluate a ML-based model that can assist in ARDS diagnosis and is portable enough to be implemented in hospital ICUs, (v) to retrain a previously developed CNN-based approach to detect Covid-19 from patient chest X-rays, and subsequently, (vi) to validate the previously established HPC-enabled expert system in a clinical use case.

The remainder of this paper is structured as follows: related work is reviewed in Section II and Section III provides brief overviews on medical and technological methods required to understand the paper, Section IV presents the work done on the physiological model parallelisation and data preparation, while Section V describes the data preparation and model training for the predictive Covid-19 model. This paper ends with some concluding remarks.

II. Related Work

In this section we survey related works that are relevant in context (e.g., simulators of disease progression, machine and deep learning approaches, etc.).

Currently, the generally accepted method of diagnosing ARDS is the "Berlin Definition" which defines onset of the condition as a ratio of arterial oxygen to inspired oxygen of less than 300 mmHg, with increasing severity as the ratio decreases [11, 12]. The definition does not specify the duration of the reduced ratio, and diagnosis depends on the familiarity of the ICU staff and physicians with the condition. On the other hand, many treatment methods have been proposed to prevent or treat ARDS, although no concensus has been reached in the literature. These methods either revolve around lung protective ventilation in order to prevent ARDS, or lung recruitment through maintained inflation or high-PEEP/low tidal volume accompanied by treatment to reduce the associated infection [13, 14, 15]. In order to simplify the analysis of potential treatment methods, Hardman *et al.* and later on Das *et al.* worked on developing a mechanistic approach to simulate the pulmonary and cardiovascular system of a patient. Their model was built on available formulae that simulate air flow into the lungs, gas exchange through the alveoli, and hemodynamic equilibrium in the blood, and was shown to be accurate in its representation of patient trajectories based on selected input parameters [9, 16, 17, 10]. This model was also used to test the efficacy of several ventilation protocols to treat a simulated ARDS patient [14, 15, 18].

Work such as that done by Das *et al.* could only be possible as medical information becomes digitised, and patient data, after anonymisation, becomes more accessible and available for research [19, 20]. As Electronic Health Records (EHRs) become the standard for medical data storage while storage itself become more efficient, more medical information is available for analysis and research and we come into the age of medical "Big Data" [21]. This advancement mirrors the growth, increased efficiency, and expanding availability of computational resources and algorithms. Research institutions, universities, and medical centres are now more likely to have access to HPC resources on-site or through agreements with other institutions, cloud computing resources are available through private vendors (e.g. Amazon Web Services, Microsoft Azure), and finally, through worldwide collaboration, new and efficient open-source algorithms for ML and data processing that take advantage of the technological advancements are available online and are well-documented (e.g. Python[1], TensorFlow[2], PyTorch[3], etc.).

Covid-19 is the disease caused by infection with Severe Acute Respiratory Syndrome coronavirus 2 (SARS-CoV-2) and which has had a major effect on the international scale in terms of strain to medical infrastructures, as well as on an economic level[4] [22, 23]. Infected patients generally exhibit flu-like symptoms that in 5% of cases can lead to severe consequences such as shock, respiratory failure, and multi-organ dysfunction[5]. Currently, the standard and most effective diagnosis method is through Reverse Transcription-Polymerase Chain Reaction (RT-PCR) which is a time- and resource-consuming method[6]. The ability to quickly and accurately diagnose the condition at low cost and using standard equipment available at hospitals has been a goal for several researchers and Punn *et al.* present an analysis of developed Deep Learning (DL) methods to detect Covid-19 from chest X-rays [24]. Of these methods, COVID-Net developed by Wang *et al.* is considered in this paper, as an open-source network, trained on collected chest X-rays compiled within a open-source dataset (COVIDx[7]) [25]. This network leverages residual networks in a similar fashion to the ResNet50 developed by He *et al.* that outperformed its competitors in the ImageNet detection and localisation tasks in 2015 [26].

[1] https://www.python.org/

[2] https://www.tensorflow.org/

[3] https://www.pytorch.org/

[4] https://www.oecd.org/coronavirus/policy-responses/global-financial-markets-policy-responses-to-covid-19-2d98c7e0/

[5] https://www.cdc.gov/coronavirus/2019-ncov/hcp/clinical-guidance-management-patients.html

[6] https://ec.europa.eu/research-and-innovation/en/horizon-magazine/pcr-antigen-and-antibody-five-things-know-about-coronavirus-tests

[7] https://www.kaggle.com/andyczhao/covidx-cxr2

Authorized licensed use limited to: NATIONAL UNIVERSITY OF ICELAND. Downloaded on June 28,2022 at 13:07:42 UTC from IEEE Xplore. Restrictions apply.

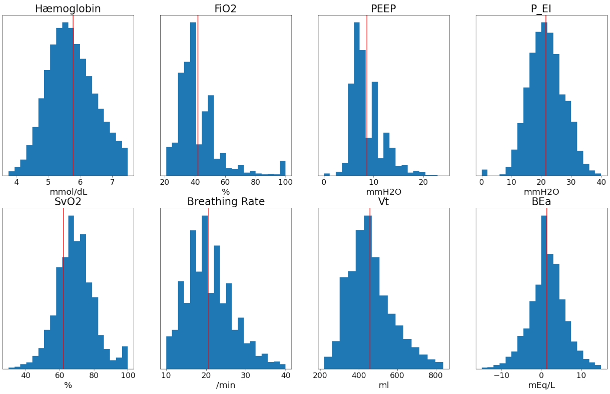


Figure 1. Parameter distribution over predetermined physiological ranges (mean in red). Parameters: **FiO2** - Fraction of inspired O_2, **PEEP** - Peak End-Expiratory Pressure, **Vt** - Tidal Volume, **P_EI** - End-Inspiratory Pressure, **SvO2** - Venous O_2 Saturation, **BEa** - Arterial Base Excess.

III. Medical and Technological Methods

A. *Machine Learning using State-of-the-Art Deep Learning*

Artificial Intelligence (AI) is a vast area of techniques and tools that enable computers to mimic human behaviour and thus also include an extensive range of approaches such as ML, DL, and robotics. ML is a specific subset of AI that is well understood through statistical learning theory [27] wherein valuable information can be extracted concerning model capacity, generalization, and the relevance of regularization and validation for model selection. More recently, DL emerged from ML as systems with the ability to learn underlying features in data using neural networks with specific dedicated types of layers tuned specifically for the tasks at hand such as image processing [28] or sequence data analysis [29]. DL is an active research topic with the number of publications grown exponentially [30]. The image recognition work described in this paper takes advantage of a specific type of DL network for image recognition tasks related to Covid-19 described in details in Sections II and V in more detail.

B. *Understanding the need for High-Performance Computing*

Using DL networks for image recognition tasks as required for Covid-19 prediction is very computational-intensive, requiring HPC or Cloud Computing (CC) resources. Parallelising DL algorithms on HPC resources happens at the level of numerical operations, at the level of the DL models themselves, and at the level of the training process. DL models transform n-dimensional tensors by applying element-wise operations (e.g. activation functions, convolution operations, or matrix multiplication) in fully-connected layers. Element-wise operations are easily parallelizable, but convolution operations and matrix multiplication require specialised parallelization strategies. Our work benefits from HPC systems using parallel matrix operations and convolutions using highly optimized libraries such as MKL[8], cuBLAS[9], and cuDNN[10]. More details on used HPC systems that are based on MSA[7] are described in Table I of Section IV.

C. *Selected Data Analysis Toolset*

A wide variety of toolsets enabled the work on both aspects of the project and simplified access to the HPC systems. The system JuDoor[11] enabled access to the HPC systems addressing issues such as resource access through the Secure Shell (SSH) protocol and account management, while the availability of Jupyter notebooks on HPC resources of the JSC[12] made it possible to test code and visualise results more efficiently. One particular challenge was switching between TensorFlow versions where using the Covid-19 prediction model required version 1.3 while the work on the virtual patient model is not version-restricted. Having access to both versions on the cluster greatly simplified the process. Additionally, the HPC systems we used provide an implementation of Horovod [31], a data-parallel framework for distributed training of DL networks with NCCL[13] as a communication framework. Finally, using COVID-Net required the use of the Open Source Computer Vision Library (OpenCV)[14] for image manipulation.

IV. Physiological Model - Results and Discussion

A. *Model Conversion and Parallelisation*

The physiological simulator is available to us as a Matlab[15] script. Given that (a) our intention is to parallelise the model,

[8] https://www.intel.com/content/www/us/en/developer/tools/oneapi/onemkl.html
[9] https://docs.nvidia.com/cuda/cublas/index.html
[10] https://developer.nvidia.com/cudnn
[11] https://judoor.fz-juelich.de
[12] https://jupyter-jsc.fz-juelich.de
[13] https://docs.nvidia.com/deeplearning/nccl/index.html
[14] https://opencv.org/
[15] https://www.mathworks.com/products/matlab.html

Authorized licensed use limited to: NATIONAL UNIVERSITY OF ICELAND. Downloaded on June 28,2022 at 13:07:42 UTC from IEEE Xplore. Restrictions apply.

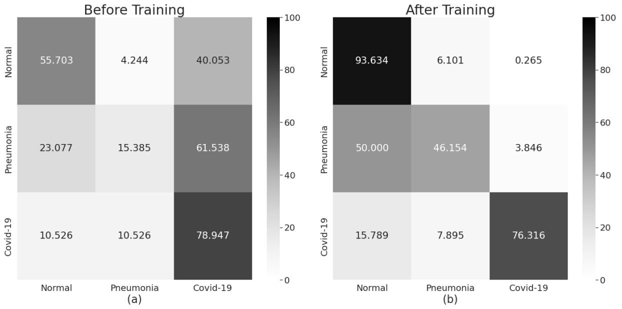


Figure 2. Prediction performance on a test set of the EHL dataset (a) before and (b) after training.

feed it automatically generated data, and produce from it outputs for selected parameters, and (b) the supercomputing clusters at JSC do not have an implementation of Matlab running in parallel, we opted to convert the model itself into a compilable and portable version in C. This was done using Matlab Coder[16] developed by Mathworks inc. On the cluster side, a python script was prepared that can read patient data, use it to populate a function call for the C-based simulator, compile it, and run it in order to generate outputs. The specific parameters to output after each simulation run will be selected at a later step as we progress further into the project.

Depending on which section of the supercomputing cluster we use, we are able to scale up the simulation both in terms of speed of execution of individual tasks and in terms of the number of tasks that can be executed concurrently. Table I shows the different configurations available. Accordingly, we tested the ported simulator using dummy data as input both within a serial JupyterLab implementation and in parallel using the Message Passing Interface (MPI) on the Dynamical Exascale Entry Platform (DEEP) cluster, and were able to achieve the speedup values presented in Table II.

Given these results, and after comparing the outputs with those from the original simulation, we show that the model can be scaled up proportionally to the number of processors recruited for the task at hand. Similarly, as running the simulations in parallel also reduces the run times, as shown in Table II, we can estimate the time it would take to run the large number of simulations possible using algorithmically generated inputs. The following section takes into consideration the methods through which the inputs for the simulator are generated.

B. *Defining Boundaries and Sampling*

For the approach described in this paper, the available patient data is used to validate the ranges within which our parameter generation methods will have to be bound. The boundaries themselves were selected based on the recommendations from ICU staff and medical practitioners participating in the work, and Figure 1 highlights the distribution of the data within these ranges. It is clear from the histograms that the data provided confirms the choice of upper and lower bounds for the parameters in question.

Aside from the parameters presented in Figure 1, the simulation also requires inputs related to the behaviour of the individual alveolar compartments within the respiratory model. These parameters will also be sampled within boundaries that were experimentally selected based on their distribution in the patient data provided by the clinics. Sampled values include the intra-compartmental airway resistance (R_{comp}), the physiological deadspace volume (V_{Dphys}), and the stickiness of the alveoli, among others.

We use the simulator in this manner to generate a large number of outputs that, along with their respective inputs, can be used to train a ML-based model. This model will be developed as an upcoming step within the scope of our work. We generate inputs by populating a range between the boundaries defined above, and randomly sampling over these values over several iterations. By limiting the range between boundaries to 10 values, we obtain 10^8 possible combinations of values for the patient parameters presented in Figure 1 and 10^{11} possible combinations for the variables that define compartment parameters. It follows that the number of combinations increases as the range between the boundaries increases, though it is worth noting that the list of generated parameter combinations grows in size to such an extent

[16] https://www.mathworks.com/products/matlab-coder.html

TABLE I. Partitions on the DEEP Prototype.

Partition	Nodes	CPUs	GPU
DP-DAM	16	96	✓
DP-ESB	75	16	✓
DP-CN	50	48	x

TABLE II. Execution time of the simulation.

Platform	Execution Time
Original Simulation on Laptop	259.1 s
C in serial on DEEP with JupyterLab	108.8 s
C in parallel on DEEP on 48 CPUs	100.79 s

Authorized licensed use limited to: NATIONAL UNIVERSITY OF ICELAND. Downloaded on June 28,2022 at 13:07:42 UTC from IEEE Xplore. Restrictions apply.

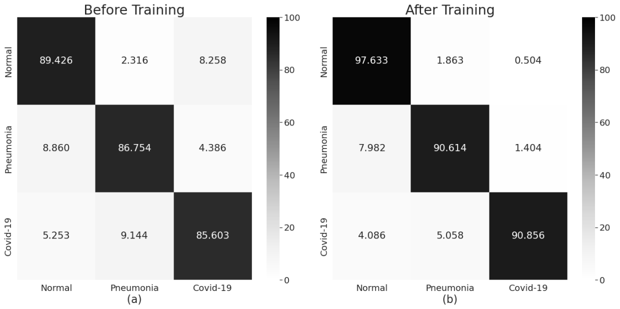


Figure 3. Prediction performance on a test set of the Fusion dataset (a) before and (b) after training.

that it would be difficult to keep in the available storage, even on the HPC cluster. Our approach to avoiding memory overload in this case is to apply the knowledge we have in using distributed memory and HPC to break down the task into smaller individual tasks. In doing these steps, we set the ground work both for building the ML model, and for generating the data required to train said model.

V. Covid-19 Model - Results and Discussion

A. Data Preparation and Distribution

Data from health partners in Europe is provided to assist in testing, training, and validating a model that builds on the work done by Wang *et al.* for COVID-Net [25]. Unlike the COVIDx dataset, in which the X-ray images are labelled across 3 categories (healthy, pneumonia, and Covid-19), the X-ray images provided by e-HealthLine (EHL) are further classified, as well as the above-mentioned three classes, into categories covering a wide range of conditions affecting the lungs (e.g. pulmonary edema, atelectasis, etc.).

For the time being, and as part of our attempt to validate the model and the platform with the available data, we restrict our approach to a 3 class prediction. Given that the EHL dataset is greatly reduced after removing images that do not fit the three categories mentioned above, we opt to create a Fusion dataset that merges the provided images with those available from COVIDx. Table III compares dataset sizes and constitutions.

B. Model Selection

Several iterations of COVID-Net exist in the original authors' public repository and they provide a comprehensive guide that sheds light on model accuracy after training. In our implementation of the model, we found that the latest greatest model (at the time of writing: COVID-Net-CXR4-A) performed badly on the data made available through EHL. This was due to the provided images being of a lower resolution than the COVIDx images. For that reason we opted to use an earlier version of the model (COVIDNet-CXR Large) which takes images of 224x224 pixel resolutions, and which has relatively high accuracy and Covid-19 sensitivity.

TABLE III. Dataset constitution

Dataset	Normal	Pneumonia	Covid-19
COVIDx	8,066	5,575	2,358
EHL	1,898	118	187
Fusion	9,964	5,693	2,542

The model inference performance was poor on both the EHL data and the Fusion dataset, with many images automatically being classified as having Covid-19. This highlights the difference between the available images and those that the model was trained on, as well as the presence of a built-in algorithmic weighting scheme that pushes the model towards detecting Covid-19 more often. Alternately, these results, presented in Figures 2(a) and 3(a), confirm the need to retrain the model altogether in order to increase prediction accuracy for the two other classes in our data.

C. Model Retraining

Training COVID-Net is done through a script provided by the original authors, though several parameters can be tuned. In our applications, we fine-tuned the class weights to leverage the Covid-19 and Pneumonia classes. This was done to make up for the class imbalance due to the healthy patient dataset being significantly larger than the other classes. We can see the improvement in the model's predictive capabilities in Figures 2(b) and 3(b). When training is performed on the EHL dataset alone, the model's ability to distinguish between Covid-19 and the other classes is much more pronounced. For pneumonia we see that the model is not able to accurately differentiate between it and healthy patients, but that might be due to the reduced number of images for this particular class. Alternatively, the model's performance improvement on the Fusion dataset is less pronounced but still clear as prediction accuracy increases for all three classes.

Seeing as the model was successfully trained on the data made available by EHL, which in turn was shown to be different than the COVIDx data the model was originally trained on, we can confirm that it is both robust and easy to train. This COVID-Net model that was retrained on the images from the EHL dataset is currently more attuned to the type of images that will be made available in the future

Authorized licensed use limited to: NATIONAL UNIVERSITY OF ICELAND. Downloaded on June 28,2022 at 13:07:42 UTC from IEEE Xplore. Restrictions apply.

by the participating hospitals, making it a good fit for their applications. On the other hand, and with the information gained through retraining the model, we can move on to the next step of the project where we to use the remainder of the labelled dataset and apply transfer learning on the model for prediction over further conditions.

VI. Conclusion

In this paper we described two methods where we leverage the HPC structure available through JSC to make possible or to accelerate work in medical data processing. On the one hand we facilitate the generation of data for simulating the pulmonary and cardiovascular system responses and pave the way for the development of a portable black-box model of human physiology. On the other hand we parallelise retraining of a DL classification model with new data to simplify Covid-19 diagnosis through chest X-rays, with the potential to expand into more conditions as data become more available and accessible. This work is presented both as stepping stones for future projects as well as validation of a pre-established HPC-enabled expert system for medical applications.

Acknowledgements

This work was performed in the SMITH Project receiving funding via the Medical Informatics Initiative from the German Federal Ministry of Education and Research (BMBF), and Icelandic HPC National Competence Center is funded by the EuroCC project that has received funding from the EU HPC Joint Undertaking (JU) under grant agreement No 951732, and the EOSC Covid-19 Fast Track. We also acknowledge the assistance provided by Dr. Hannah Mayer and Dr. Lars Küpfer for their assistance in obtaining the physiological simulator, and Dr. John G. Hardman *et al.* and Dr. Anup Das *et al.* for their work on the simulator.

References

[1] Forum of International Respiratory Societies and European Respiratory Society, The global impact of respiratory disease, WHO, 2 edition, 2017, OCLC: 999612837.

[2] P. V. Dasaraju and C. Liu, "Infections of the Respiratory System," in Medical Microbiology, S. Baron, Ed. University of Texas Medical Branch at Galveston, Galveston (TX), 4 edition, 1994.

[3] D. G. Ashbaugh, D. B. Bigelow, and B. E. Levine, "Acute Respiratory Distress in Adults," The Lancet, vol. 290, no. 7511, pp. 319–323, Aug. 1967.

[4] M. Confalonieri, F. Salton, and F. Fabiano, "Acute respiratory distress syndrome," European Respiratory Review, vol. 26, no. 144, 2017.

[5] S. Le, et al., "Supervised machine learning for the early prediction of acute respiratory distress syndrome (ards)," Journal of Critical Care, vol. 60, pp. 96–102, 2020.

[6] C. Barakat, S. Fritsch, M. Riedel, and S. Brynjolfsson, "An HPC-Driven Data Science Platform to Speed-up Time Series Data Analysis of Patients with the Acute Respiratory Distress Syndrome," in 2021 44th International Convention on Information, Communication and Electronic Technology (MIPRO), Opatija, Croatia, Sept. 2021, pp. 311–316, IEEE.

[7] E. Suarez, N. Eickert, and T. Lippert, "Modular Supercomputing architecture: from idea to production," in Contemporary High Performance Computing: From Petascale toward Exascale, J. Vetter, Ed., vol. 3, pp. 223–251. CRC Press, FL, USA, 1 edition, 2019.

[8] A. Winter, et al., "Smart Medical Information Technology for Healthcare (SMITH): Data Integration based on Interoperability Standards," Methods of Information in Medicine, vol. 57, no. S 01, pp. e92–e105, July 2018.

[9] J. G. Hardman, et al., "A physiology simulator: validation of its respiratory components and its ability to predict the patient's response to changes in mechanical ventilation," British Journal of Anaesthesia, vol. 81, no. 3, pp. 327–332, Sept. 1998.

[10] A. Das, et al., "Development of an integrated model of cardiovascular and pulmonary physiology for the evaluation of mechanical ventilation strategies," in 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Milan, Aug. 2015, pp. 5319–5322, IEEE.

[11] The ARDS Definition Task Force, "Acute Respiratory Distress Syndrome: The Berlin Definition," JAMA, vol. 307, no. 23, June 2012.

[12] L. Pisani, et al., "Risk stratification using SpO2/FiO2 and PEEP at initial ARDS diagnosis and after 24 h in patients with moderate or severe ARDS," Annals of Intensive Care, vol. 7, no. 1, pp. 108, Dec. 2017.

[13] J. Villar, et al., "The ALIEN study: incidence and outcome of acute respiratory distress syndrome in the era of lung protective ventilation," Intensive Care Medicine, vol. 37, no. 12, pp. 1932–1941, Dec. 2011.

[14] A. Das, et al., "Evaluation of lung recruitment maneuvers in acute respiratory distress syndrome using computer simulation," Critical Care, vol. 19, no. 1, pp. 8, Dec. 2015.

[15] A. Das, et al., "Hemodynamic effects of lung recruitment maneuvers in acute respiratory distress syndrome," BMC Pulmonary Medicine, vol. 17, no. 1, pp. 34, Dec. 2017.

[16] J. G. Hardman and A. R. Aitkenhead, "Estimation of Alveolar Deadspace Fraction Using Arterial and End-Tidal CO2 : A Factor Analysis Using a Physiological Simulation," Anaesthesia and Intensive Care, vol. 27, no. 5, pp. 452–458, Oct. 1999.

[17] A. Das, et al., "A systems engineering approach to validation of a pulmonary physiology simulator for clinical applications," Journal of The Royal Society Interface, vol. 8, no. 54, pp. 44–55, Jan. 2011.

[18] A. Das, L. Camporota, J. G. Hardman, and D. G. Bates, "What links ventilator driving pressure with survival in the acute respiratory distress syndrome? A computational study," Respiratory Research, vol. 20, no. 1, pp. 29, Dec. 2019.

[19] B. K. Beaulieu-Jones, P. Orzechowski, and J. H. Moore, "Mapping Patient Trajectories using Longitudinal Extraction and Deep Learning in the MIMIC-III Critical Care Database," preprint, Bioinformatics, Aug. 2017.

[20] K. M. Karunarathna, "Predicting ICU death with summarized patient data," in 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, Jan. 2018, pp. 238–247, IEEE.

[21] A. E. Johnson, et al., "MIMIC-III, a freely accessible critical care database," Scientific Data, vol. 3, no. 1, pp. 160035, Dec. 2016.

[22] T. Acter, et al., "Evolution of severe acute respiratory syndrome coronavirus 2 (sars-cov-2) as coronavirus disease 2019 (covid-19) pandemic: A global health emergency," Science of the Total Environment, p. 138996, 2020.

[23] G. French et al., "Impact of Hospital Strain on Excess Deaths During the COVID-19 Pandemic — United States, July 2020–July 2021," Morbidity and Mortality Weekly Report, vol. 70, no. 46, pp. 1613–1616, Nov. 2021.

[24] N. S. Punn and S. Agarwal, "Automated diagnosis of COVID-19 with limited posteroanterior chest X-ray images using fine-tuned deep neural networks," Applied Intelligence, vol. 51, no. 5, pp. 2689–2702, May 2021.

[25] L. Wang, Z. Q. Lin, and A. Wong, "Covid-net: a tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images," Scientific Reports, vol. 10, no. 1, pp. 19549, Nov 2020.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," arXiv:1512.03385 [cs], Dec. 2015, arXiv: 1512.03385.

[27] V. N. Vapnik, "An overview of statistical learning theory," IEEE transactions on neural networks, vol. 10, no. 5, pp. 988–999, 1999.

[28] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," Advances in neural information processing systems, vol. 25, pp. 1097–1105, 2012.

[29] B. Alipanahi, A. Delong, M. T. Weirauch, and B. J. Frey, "Predicting the sequence specificities of dna-and rna-binding proteins by deep learning," Nature biotechnology, vol. 33, no. 8, pp. 831–838, 2015.

[30] D. Zhang, et al., "The ai index 2021 annual report," arXiv preprint arXiv:2103.06312, 2021.

[31] A. Sergeev and M. Del Balso, "Horovod: fast and easy distributed deep learning in tensorflow," arXiv preprint arXiv:1802.05799, 2018.

Authorized licensed use limited to: NATIONAL UNIVERSITY OF ICELAND. Downloaded on June 28,2022 at 13:07:42 UTC from IEEE Xplore. Restrictions apply.

Paper IV

Analysis of Chest X-Ray for COVID-19 Diagnosis as a Use Case for an HPC-enabled Data Analysis and Machine Learning Platform for Medical Diagnosis Support

C. Barakat, M. Aach, A. Schuppert, S. Brynjólfsson, S. Fritsch, M. Riedel

https://www.mdpi.com/2085284, 2023

This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/).

C. Barakat performed the work described in the manuscript on the platform developed in previous work. He adapted the scripts with the assistance of M. Aach to implement hyperparameter optimization. Finally, he performed the result analysis, and final re-training of the COVID-Net model, based on which he drafted the original manuscript.

diagnostics

MDPI

Article

Analysis of Chest X-ray for COVID-19 Diagnosis as a Use Case for an HPC-Enabled Data Analysis and Machine Learning Platform for Medical Diagnosis Support

Chadi Barakat [1,2,3,*,†], Marcel Aach [1,2], Andreas Schuppert [3,4], Sigurður Brynjólfsson [1], Sebastian Fritsch [2,3,5,‡] and Morris Riedel [1,2,3,‡]

[1] School of Engineering and Natural Science, University of Iceland, 107 Reykjavik, Iceland
[2] Jülich Supercomputing Centre, Forschungszentrum Jülich, 52428 Jülich, Germany
[3] SMITH Consortium of the German Medical Informatics Initiative, 07747 Leipzig, Germany
[4] Joint Research Centre for Computational Biomedicine, University Hospital RWTH Aachen, 52074 Aachen, Germany
[5] Department of Intensive Care Medicine, University Hospital RWTH Aachen, 52074 Aachen, Germany
* Correspondence: c.barakat@fz-juelich.de
† Current address: Jülich Supercomputing Centre, Forschungszentrum Jülich, 52428 Jülich, Germany.
‡ These authors contributed equally to this work.

Abstract: The COVID-19 pandemic shed light on the need for quick diagnosis tools in healthcare, leading to the development of several algorithmic models for disease detection. Though these models are relatively easy to build, their training requires a lot of data, storage, and resources, which may not be available for use by medical institutions or could be beyond the skillset of the people who most need these tools. This paper describes a data analysis and machine learning platform that takes advantage of high-performance computing infrastructure for medical diagnosis support applications. This platform is validated by re-training a previously published deep learning model (COVID-Net) on new data, where it is shown that the performance of the model is improved through large-scale hyperparameter optimisation that uncovered optimal training parameter combinations. The per-class accuracy of the model, especially for COVID-19 and pneumonia, is higher when using the tuned hyperparameters (healthy: 96.5%; pneumonia: 61.5%; COVID-19: 78.9%) as opposed to parameters chosen through traditional methods (healthy: 93.6%; pneumonia: 46.1%; COVID-19: 76.3%). Furthermore, training speed-up analysis shows a major decrease in training time as resources increase, from 207 min using 1 node to 54 min when distributed over 32 nodes, but highlights the presence of a cut-off point where the communication overhead begins to affect performance. The developed platform is intended to provide the medical field with a technical environment for developing novel portable artificial-intelligence-based tools for diagnosis support.

Keywords: deep learning; COVID-19; high-performance computing; image-based diagnostics; medical diagnosis support

check for updates

Citation: Barakat, C.; Aach, M.; Schuppert, A.; Brynjólfsson, S.; Fritsch, S.; Riedel, M. Analysis of Chest X-ray for COVID-19 Diagnosis as a Use Case for an HPC-Enabled Data Analysis and Machine Learning Platform for Medical Diagnosis Support. *Diagnostics* **2023**, *13*, 391. https://doi.org/10.3390/diagnostics13030391

Academic Editors: Sivaramakrishnan Rajaraman, Zhiyun Xue and Sameer Antani

Received: 20 December 2022
Revised: 14 January 2023
Accepted: 18 January 2023
Published: 20 January 2023

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

As the COVID-19 pandemic threatened to break down medical infrastructure all over the world, it became evident that effective and efficient methods of diagnosis are necessary in order to improve outcomes and save the lives of hospital patients [1]. Especially during the early phase of the pandemic, when antigen-based rapid tests were not yet available, there was an urgent need for alternative diagnostic procedures. The standard approach using reverse-transcription polymerase chain-reaction (RT-PCR) required a lot of time, trained staff, and laboratory capacity and showed, especially at the beginning of the pandemic, very heterogeneous accuracy [2,3]. Since pulmonary involvement in particular posed a risk to patients with COVID-19, it was reasonable to examine conventional chest-X-ray

(CXR) images, which are a rapid and widely available diagnostic tool for COVID-19-specific changes [4]. Thus, early publications had already reported the presence of specific changes in thoracic imaging before a laboratory test yielded a positive result [5]. Focusing on readily available and inexpensive diagnostic procedures is especially meaningful as research predicts that such large-scale contagion events will happen at an increasing rate [6].

However, given the current advancements in high-performance computing (HPC) technology and the availability of commercial cloud computing (CC) resources to the general public, as well as large increases in online data storage and sharing capabilities, an increasing interest in machine learning (ML) and deep learning (DL) applications that put these resources to use in order to solve common problems can be observed [7–9]. Similarly, these techniques and resources are being employed towards extracting information from Big Data repositories that would otherwise require hundreds of researchers over several thousand hours [10,11]. More recently, the combination of HPC, Big Data, and ML have made headlines in the scientific community with the publication of two DL models, AlphaFold from DeepMind and RoseTTAFold from Baek et al., which match or even outperform existing methods for protein structure prediction [12,13].

It follows that several research groups have developed ML and DL methods for detecting COVID-19 from sonographic [14] and X-ray images of the thorax [15–17], or for predicting the mortality of COVID-19 patients from medical data [18], with all of the results highlighting how effective these models might be for quick triaging. In a similar application field, Rajaraman et al. merged several trained DL models to improve the diagnosis of pneumonia from CXR images with a higher success rate than conventional image recognition models [19]. Other researchers have made use of cutting-edge HPC resources, namely the Jülich Wizard for European Leadership Science (JUWELS) (https://www.fz-juelich.de/en/ias/jsc/systems/supercomputers/juwels (accessed on 19 December 2022)) cluster, one of Europe's fastest supercomputers to train advanced DL networks on Big Data from different fields, thus highlighting the need to make use of modular supercomputing architecture (MSA) to advance the field of artificial intelligence (AI) [20]. Furthermore, advanced automated hyperparameter tuning methods such as KerasTuner (https://keras.io/keras_tuner/ (accessed on 19 December 2022)) and Ray Tune (https://docs.ray.io/en/latest/tune/index.html (accessed on 19 December 2022)) have been developed, which simplify the parameter search process needed to fine-tune the training of neural networks, thus yielding the best performing model without major interventions from ML researchers [21].

Application of the available HPC resources in the medical field, thus contributing to the analysis of medical data and a timely and precise diagnosis, has the potential to reduce the amount of stress that medical personnel are exposed to during their work [22,23]. Similarly, the medical field presents a fertile ground for setting up frameworks that can be easily loaded, modified, and deployed where needed to help mitigate the effects of future epidemics and pandemics [24]. In the present paper, these approaches are thus validated in the application of the COVID-Net developed by Wang et al. on newly obtained CXR images that were provided by healthcare partner E*HealthLine (EHL) as part of the European Open Science Cloud (EOSC) Fast-Track grants for COVID-19 research.

The work presented in this article describes the culmination of work performed towards setting up a platform within which medical data can be stored, cleaned, and analysed, and easily used to train ML and DL models [25,26]. The platform makes use of highly specialised hardware and software available at the Jülich Supercomputing Centre (JSC) to develop and train these models in the most efficient manner. These include firstly the DEEP and JUWELS supercomputing clusters, and the storage made available through the related projects. Advanced hyperparameter tuning methods are also used to fine-tune the models to produce the best results.

The following sections go into the details of (a) training COVID-Net on newly acquired data, (b) performing large-scale hyperparameter tuning on the model in order to extract the parameter combinations that produce the best trained models, and (c) re-training the model to highlight the improvement achieved in per-class accuracy for each of these combinations.

Furthermore, resource scale-up is also performed in order to gauge the speed-up that can be achieved through the established platform.

Re-training the COVID-Net model in such a way serves as a preliminary proof-of-concept for the platform. Due to its easy adaptability to new use-cases and its portability on other academic or commercially available CC resources, this platform can support researchers in the medical field to create more complex models with better performance that would otherwise be impossible to develop due to a lack of computational resources and missing expertise in usage of HPC systems. Additionally, the models built and pre-trained within the platform rely on open-source data and software, making them easy to deploy on local machines in hospitals intensive care units (ICUs).

It is worth noting that several groups have applied hyperparameter optimisation to improve the results of DL-based COVID-19 diagnosis models [27–29]. However, comparison with these works cannot easily be undertaken, as the concept and specific innovation described in the present paper lies within scaling up the data storage, the model training, and the hyperparameter tuning processes through efficient use of HPC resources in order to cover more ground.

2. Materials and Methods

This section describes the hardware and software implemented within the developed data analysis and machine learning platform, as well as the methods and data through which the COVID-Net model, developed by Wang et al. [15], is re-trained on new data and its prediction performance is improved through large-scale hyperparameter tuning. Figure 1 presents a general overview of the re-training process and model improvement steps performed as part of the platform validation, and highlights how computationally expensive the hyperparameter tuning step is.

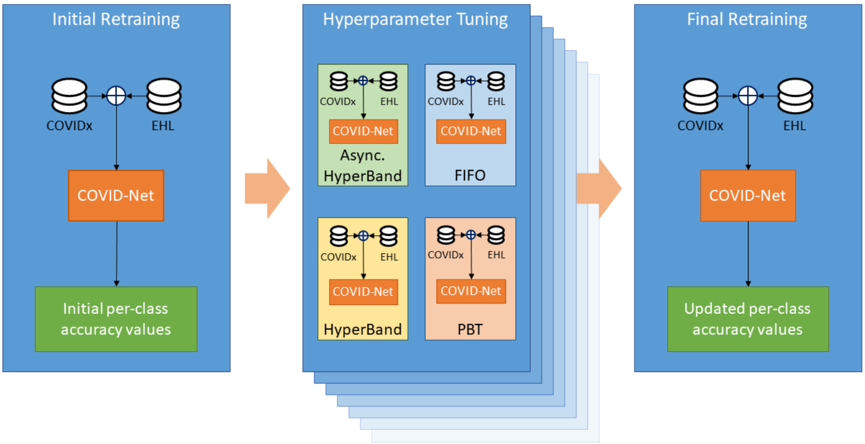


Figure 1. Block diagram representing the experimental process within the data analysis and machine learning platform. The different schedulers are represented as boxes within the hyperparameter tuning step. Due to the large amount of computations that it needs to perform, the hyperparameter tuning step requires significantly more resources than the remaining steps.

2.1. HPC Resources

In their presentation of a novel approach to build and organise HPC resources, Suarez et al. provide a thorough technical description of the hardware set up at JSC, with an emphasis on its modular aspects [30]. This is true in terms of the hardware dedicated to computation as well as that used for communication and for storage. In essence, the MSA allows for efficient scale-up as required by HPC researchers according to the tasks at hand.

The hardware is supported by the open-source scheduling software Simple Linux Utility for Resource Management (SLURM) (https://slurm.schedmd.com/(accessed on 19 December 2022)), which manages the workload over the available resources and leverages the scalability aspect of the modular system, but also reduces wasted computing time through intelligent prioritisation of tasks. Furthermore, aside from terminal access through SSH, users can directly access resources through an integrated Jupyter (https://jupyter-jsc.fz-juelich.de/ (accessed on 19 December 2022)) development environment, which can be adapted to the specific needs of the task at hand through pre-packaged data analytics and ML modules as well as personalised kernels and virtual environments.

2.1.1. DEEP

The DEEP series of projects has been setting up the path towards exascale computing since 2016, focusing on scaling available HPC resources through boosters [31]. These projects have received funding granted by the European Commission under the Horizon 2020 program and have so far had three iterations under the titles “DEEP”, “DEEP-Extended Reach” (DEEP-ER), and “DEEP-Extreme Scale Technologies” (DEEP-EST). A fourth iteration upcoming as “DEEP-Software for Exascale Architectures” (DEEP-SEA) was launched in 2021 with the aim of delivering a standardised programming environment for exascale computing for the European HPC systems.

At the hardware level, DEEP-EST introduced the concept of MSA, making the cluster-booster architecture more attuned for data analytics tasks [32]. Accordingly, the system itself is divided into several modules, each sporting the necessary hardware for specific tasks (i.e., numerical data processing, image processing, hyperspectral image processing). These modules are presented in Table 1.

Table 1. Partitions on the DEEP prototype.

Partition	Nodes	CPUs/Node	GPU
DEEP-Data Analytics Module	16	96	NVIDIA V100 + Intel Stratix10 FGPA
DEEP-Extreme Scale Booster	75	16	NVIDIA V100
DEEP-Cluster Module	50	48	n/a

2.1.2. JUWELS

The JUWELS supercomputer consists of two main parts: a cluster module and a booster module, commissioned in 2018 and 2020, respectively. The cluster module is a BullSequana X1000 system (https://atos.net/en/solutions/high-performance-computing-hpc/bullsequana-x-supercomputers/bullsequana-x1000 (accessed on 19 December 2022)) with 2583 nodes totalling 122,768 CPUs. Furthermore, several nodes are specialised for visualisation, large-memory, and accelerated computing tasks (https://apps.fz-juelich.de/jsc/hps/juwels/configuration.html (accessed on 19 December 2022)). The booster module, a Bullsequena XH2000 system (https://atos.net/wp-content/uploads/2020/07/BullSequana XH2000_Features_Atos_supercomputers.pdf (accessed on 19 December 2022)), expands on the available computing power by adding a total of 940 nodes totalling 3744 GPUs.

In essence, the cluster module is intended for general-purpose computation tasks while the booster module allows for scalable computing, making large-scale simulation and visualisation tasks more possible [20]. By making use of the available high-speed network connections and available storage, the booster module has reached a peak performance of 73 petaflop per second. Kesselheim et al. validated its performance for large-scale AI research on several DL network training tasks across different fields. Their results and the recorded peak performance earned the JUWELS booster the top position on the fastest supercomputers in Europe in 2021 as well as the 7th spot on the international TOP500 list and the 3rd spot on the Green500 list.

For the purposes described in this manuscript, the development phase is performed on the DEEP-EST cluster and the usage of the JUWELS cluster and booster is reserved for large-scale production applications of the developed models.

2.2. Datasets

To validate the established platform, two separate datasets were used in order to train a pre-built classification model. The first dataset is the open-source COVIDx dataset (https://github.com/lindawangg/COVID-Net/blob/master/docs/COVIDx.md (accessed on 19 December 2022)), which was compiled by Wang et al. from a collection of open repositories as listed in Table 2 [15]. At the time of preparing the data, the most current version was COVIDx V8A. This dataset is subdivided into 3 main classes: Healthy, Non-COVID-19 Pneumonia, and COVID-19.

Table 2. COVIDx V8A dataset sources.

Title	URL
Cohen	https://github.com/ieee8023/covid-chestxray-dataset
Figure 1	https://github.com/agchung/Figure1-COVID-chestxray-dataset
Actualmed	https://github.com/agchung/Actualmed-COVID-chestxray-dataset
Sirm	https://www.kaggle.com/tawsifurrahman/covid19-radiography-database/version/3
RSNA	https://www.kaggle.com/c/rsna-pneumonia-detection-challenge/data
RICORD	https://wiki.cancerimagingarchive.net/pages/viewpage.action?pageId=70230281

The second dataset was pre-compiled by industry partner EHL and made available through file transfer protocol (FTP). The dataset is subdivided into training and testing sets, each of which is further divided into different conditions including Healthy, Pneumonia, COVID-19, Atelectasis, and Cardiomegaly, among others. Further details about the dataset constitutions are presented in later sections of this manuscript, though it is worth mentioning that there was a considerable difference in the image resolutions between the two datasets as can be seen in Figure 2. Additionally, Table 3 describes the class distribution of images within each dataset.

Table 3. Number of images within each dataset.

Dataset	Healthy	Non-COVID-19 Pneumonia	COVID-19
COVIDx	8066	5575	2358
EHL	1898	118	187
Fusion	9964	5693	2542

Finally, in order to increase the robustness of the model to be re-trained, the two datasets were merged into a Fusion dataset, preserving the split structures shown in Tables 4 and 5. The Fusion dataset represents the relatively heterogeneous data usually received from different medical institutions in special circumstances [33]. The applicability of the platform and its intended use on heterogeneous data represents one of the most important advantages.

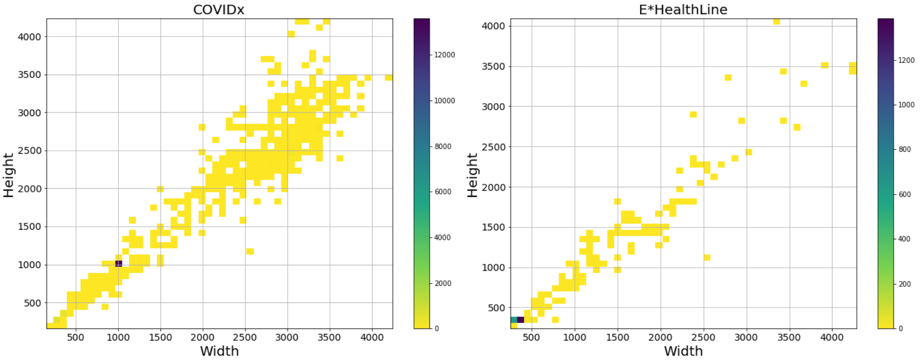


Figure 2. Range of image resolutions of the COVIDx (**left**) and EHL (**right**) datasets. A high concentration of images in the COVIDx dataset is centered around 1000 × 1000 pixels, but the majority of EHL images is below 480 × 480 pixels.

2.2.1. COVIDx Dataset

The process to obtain the COVIDx dataset is provided in detail as part of the COVID-Net Github (https://github.com/lindawangg/COVID-Net (accessed on 19 December 2022)) repository as it was compiled by Wang et al. [15]. The dataset was loaded into the online storage available at JSC and an analysis of the images was performed using the Open-Source Computer Vision Library (OpenCV) python package in order to verify that the dataset contains no duplicates or corruptions. The majority of the data provided in the COVIDx dataset are in the portable network graphics (PNG) image format. Table 4 presents the train-test split of the COVIDx dataset.

Table 4. COVIDx V8A dataset training and testing split.

Set	Healthy	Non-COVID-19 Pneumonia	COVID-19
Training	7966 (98.8%)	5475 (98.2%)	2158 (91.5%)
Testing	100 (1.2%)	100 (1.8%)	200 (8.5%)
Total	8066	5575	2358

2.2.2. EHL Dataset

The EHL dataset was made available through secure FTP and, similarly to the COVIDx dataset, loaded onto the online storage at JSC. The dataset is subdivided into several pulmonary and chest-related conditions, though for the purposes described in this manuscript solely the images within the Healthy, Non-COVID-19 Pneumonia, and COVID-19 directories were used. The remainder of the data will be used in a future transfer learning application of the available ML model.

After performing some verification steps on the data using OpenCV, it became evident that some images were duplicates of those available in the COVIDx dataset, which was traced back to the fact that one of the participating hospitals had made their data available as part of the Cohen dataset. These images were removed and the resulting distribution of data is presented in Table 5. The EHL dataset is made available as part of the European Open Science Cloud fast-track grant project and can be accessed online for research purposes (https://b2share.fz-juelich.de/records/aef5d3b8aa044485b9620b95b60c47a2 (accessed on 19 December 2022)). Evaluation of the trained models was performed using only the EHL dataset in order to verify these models' ability to predict over the new data.

Table 5. E*HealthLine dataset training and testing split.

Set	Healthy	Non-COVID-19 Pneumonia	COVID-19
Training	198 (10.4%)	21 (17.8%)	189 (65.4%)
Testing	1700 (89.6%)	97 (82.2%)	100 (34.6%)
Total	1898	118	289

2.3. COVID-Net Model

The COVID-Net deep learning model was developed and released by Wang et al. in May of 2020 in response to the COVID-19 pandemic to screen patients for COVID-19 using chest radiographs [15]. The model follows the current DL standard for image analysis of using convolutional neural networks (CNNs) with intermittently varying kernel sizes, but expands on it by employing the residual architecture that was introduced by He et al. in their pioneering work on residual networks for object detection in images [34]. COVID-Net was built using TensorFlow (https://www.tensorflow.org/ (accessed on 19 December 2022)) version 1.13.

The initial approach with COVID-Net within the scope of this project involved running inference using the pre-trained model on both available datasets in order to highlight their differences, before moving forward with the re-training attempts, which also served the purpose of highlighting the potential speed-up that can be achieved using the available MSA.

2.3.1. Model Selection

The Git repository for COVID-Net lists a number of models each with varying input image sizes and performance markers. At the time of performing this analysis, the best performing model was labelled “COVIDNet-CXR4-A”, which scales input images to a resolution of 480 × 480 pixels. Two other versions of the model exist that take inputs of lower resolution (224 × 224 pixels) with the best performing among them being “COVIDNet-CXR Large”. Both models are available for download from links in the repository.

Selecting the appropriate model for this application required an analysis of the resolutions of the available images, and since the majority of the images within the EHL dataset are below the threshold of 480 × 480 pixel resolution as can be seen in Figure 2, it became evident that the “COVIDNet-CXR Large” model would perform best. This decision is further supported by the initial inference results that will be presented below in Section 3, but follows the logic that down-sampling image data produces far less noise than up-sampling, which is more likely to generate artefacts by magnifying limited visual information.

2.3.2. Model Training

The repository for COVID-Net provides scripts and terminal commands for training the network. These scripts define the training parameters (learning rate, number of epochs, batch size, location of the pre-defined network weights) and the location of the datasets for training and testing. Accordingly, the parameters are adapted to the updated datasets being used in this application, and a range is defined over which the training will be parallelised.

Additionally, the training script is updated in order to introduce the possibility of many concurrent parallelised training runs, thus making use of the available HPC resources. The initial approach for parallelised training was through performing a grid-search of predefined parameters to tune and iteratively populating a job-script that would then be submitted to the HPC scheduler. Instead, hyperparameter tuning is implemented, as described in the next subsection, which can streamline the parameter search and potentially uncover hyperparameter combinations that would otherwise have been missed. Finally, a set of parameters is selected to train the model with an increasing number of nodes, using the Horovod (https://horovod.ai/ (accessed on 19 December 2022)) distributed DL framework, in order to determine the extent to which training can be accelerated as more resources are made available.

2.4. Hyperparameter Tuning

Hyperparameters are parameters which influence an algorithm's behaviour. These values are typically set by the user manually before the training of an algorithm. Choosing an optimal set of hyperparameters can significantly improve the performance of a model [35]. In order to easily find the best performing combination of parameters for training the COVID-Net model on the new and the combined datasets, the hyperparameter tuning library Tune, developed under the Ray framework, was employed [21,36]. This tuner takes a model and selected tunable parameters as input and performs an optimisation that highlights the combination of parameters that produces the best results according to a selected metric. Due to compatibility issues related to the earlier version of TensorFlow used in constructing COVID-Net, it was necessary to use version 0.6.2 of the Ray module.

The Ray framework employs schedulers that take advantage of parallel computing to scale up and speed up the task at hand; of these schedulers, population-based training (PBT), HyperBand, and Asynchronous HyperBand [37–39] are considered and compared to the default first-in, first-out (FIFO) scheduler. The comparison was performed by running the hyperparameter tuning process with each of the selected schedulers over the same parameter search space. The best-performing scheduler was selected based on runtime and the COVID-Net model's performance when re-trained using the optimal parameter combination that the tuning process output.

3. Results

3.1. Pre-Optimisation Analysis

Running inference with COVID-Net on the available images highlighted the differences between the two datasets. The network performance on COVIDx was in line with the results published by the original authors. However, the images from EHL were more likely to be misclassified. In fact, the results presented in Figure 3a highlight a bias towards predicting COVID-19.

After re-training the network on a combination of the newly acquired images and the original COVIDx dataset, the results achieved are presented in Figure 3b, where classification accuracy is improved. In order to achieve these results, several training runs were performed in parallel where the class weights (CWs) were adjusted, as well as the learning rate (LR), the batch size, the COVID-19 percentages (CPs), and the number of training epochs. Through these training runs the range of these parameters that are tuned on a larger scale in the next step was narrowed down.

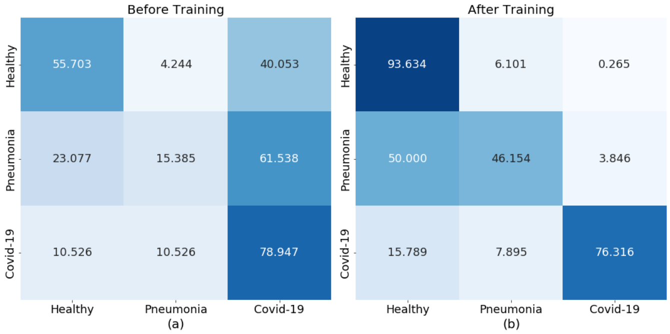


Figure 3. Prediction performance (in %) heatmaps for COVID-Net on the EHL dataset (**a**) before and (**b**) after initial re-training.

3.2. Hyperparameter Optimisation

The hyperparameter optimisation is performed on the DEEP-Extreme Scale Booster (ESB) partition, with 20 trials taking up 1 node each (see hardware configuration listed in Table 1). During these 20 trials the network is trained over 24 epochs, with each trial being assigned a different combination of the tunable parameters, in this case the COVID-19 percentage, the class weights, and the learning rate. The parameter values are chosen following a random uniform distribution in the case of the CWs and the CP, and a logarithmic uniform distribution for the LR. The selected schedulers distribute the tasks on the available nodes and in three of the four cases introduce further perturbations to the hyperparameters halfway through the training process. The specific experimental setup is further expanded in the below sections for each of the selected schedulers.

3.2.1. First-In First-Out

The default scheduling algorithm for the Ray library, first-in first-out (FIFO), performs the basic scheduling task of distributing the trials over the available nodes and does not update the tunable parameters during the training process. It is employed here as a benchmark to gauge the performance of the other schedulers.

Running all the trials in parallel took a total of 402 min to complete, after which the best performing combination of parameters was an LR of 0.00013, CWs of 1 for healthy, 1.38745 for pneumonia, and 6.1508 for COVID-19, and a CP value of 0.289. These parameters were used to re-train COVID-Net over 50 epochs and the prediction performance of the model re-trained using these parameters is highlighted in Figure 4a. The trained model in this case is very capable of detecting COVID-19 infections in CXRs, but pneumonia cases are almost always diagnosed as healthy.

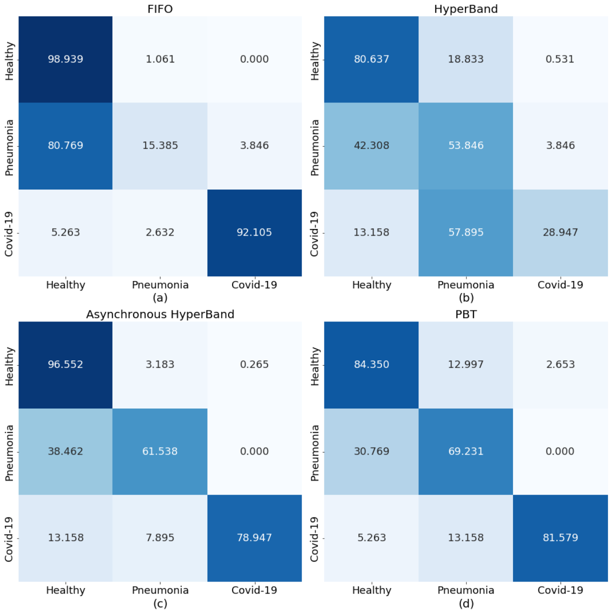


Figure 4. Prediction performance heatmaps for COVID-Net on the EHL dataset after re-training on the parameters chosen by (**a**) FIFO, (**b**) HyperBand, (**c**) Asynchronous HyperBand, and (**d**) PBT.

3.2.2. HyperBand

The HyperBand scheduler is activated in this case halfway through the training process, at which point it begins stopping tasks that underperform. The trials required a total of 421 min to complete, at which point stopped trials were discarded while the best performing trial was selected based on the overall accuracy, loss, and run time.

Interestingly, several of the trials that presented high accuracy at the end of tuning did not perform well when trained, showing a complete bias towards predicting one of the three conditions. The prediction performance of a model trained on the selected best parameters of LR = 0.0006, CW = [1, 5.0312, 3.4151], and CP = 0.081 is presented as a heatmap in Figure 4b. The trained model was unable to provide certain predictions when exposed to the images from the test set even after training for 50 epochs. The highest overall prediction accuracy is for healthy patients, but that is still at 80%.

3.2.3. Asynchronous HyperBand

Similarly to HyperBand, the Asynchronous HyperBand scheduler also implements early stopping, but does so while taking advantage of the available parallel processing power to distribute the tasks more efficiently.

Running the trials required a total of 422 min and the best performing model was chosen as having LR = 0.00012, CW = [1, 4.0981, 3.0387], and CP = 0.187. The outputs from the model trained on the best parameter combination from Asynchronous HyperBand are presented in Figure 4c. In this case, the generated parameters resulted in a trained model with improved results on the original re-trained COVID-Net presented in Figure 3b.

3.2.4. Population-Based Training

The PBT scheduler introduces perturbations to selected parameters at a set time during the tuning process. This introduces an extra layer of randomness to the hyperparameter tuning and potentially uncovers new combinations from the different trials running in parallel. In this case PBT is tasked to begin perturbing the LR halfway through the total training time.

The trials ran for a total of 419 min and from the results LR = 0.00024, CW = [1, 9.9599, 9.4996], and CP = 0.346 were selected to be used for re-training COVID-Net, the predictive performance of which is presented in Figure 4d. Similarly to the results obtained in the Asynchronous HyperBand trial, this model also presented an improved performance in detecting pneumonia and COVID-19 cases although the "Healthy" prediction was reduced to 84%.

Figure 5 compares the prediction performance of the original re-trained COVID-Net model with that of models retrained using the best performing hyperparameters from the tuning process with Asynchronous HyperBand and PBT.

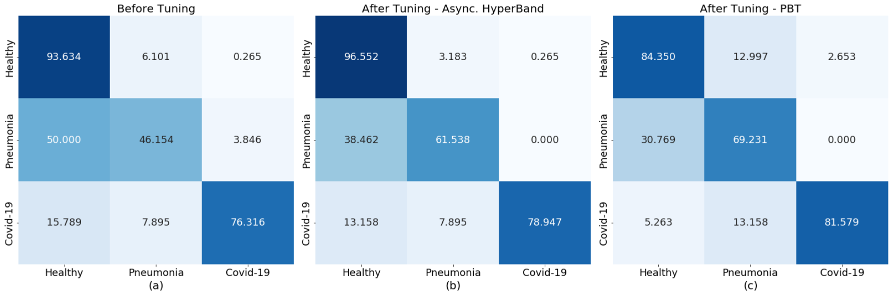


Figure 5. Comparison of trained COVID-Net prediction performance before (**a**) and after hyperparameter tuning with Asynchronous HyperBand (**b**) and PBT (**c**).

3.3. COVID-Net Re-Training

The Horovod framework was used to re-train the COVID-Net model based on parameters chosen from the previous results, while the resources available for training were iteratively increased. The graph presented in Figure 6a shows the change in training duration as more resources were made available.

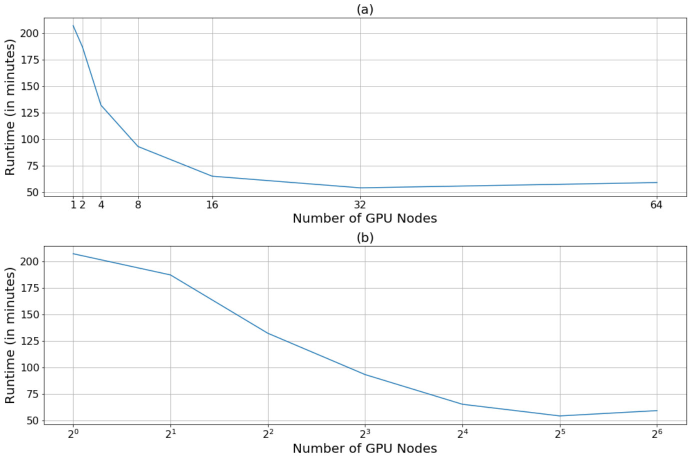


Figure 6. Training duration (in minutes) as more GPU nodes are recruited, (**a**) on a linear scale and (**b**) on a logarithmic scale.

The model trained significantly faster as the tasks were distributed among the increasing number of worker nodes. The time required to train over 25 epochs was reduced from 207 min on 1 node, to 54 min on 32 nodes. However, the rate of reduction decreased with resource increase as can be seen from the decreasing slope of Figure 6b. Ultimately, as the resources were increased to 64 nodes, the model training became slower and both curves switched to a positive slope, indicating that the cut-off point for speed-up had been reached.

4. Discussion

Through trial and error a set of parameters was selected to train the COVID-Net model on the Fusion dataset and the results obtained are shown in Figure 3b. In reality, several more parameters, including the batch size, the train-test split, the number of epochs, and freezing or unfreezing some layers from COVID-Net could have been tuned by hand in order to improve the results, but as the number of these parameters increases, so does the complexity of the optimisation problem. The results show that the model can be improved and highlight the fact that more effective tuning approaches are necessary.

Through four straightforward applications of a hyperparameter optimisation framework, it was possible to improve the predictive performance of COVID-Net on new data. The schedulers used for the optimisation took advantage of the available MSA and efficiently distributed the work over the available resources. In doing so, the framework was able to cover more ground and test more parameter combinations simultaneously in order to close in on the parameters with which the model would train more effectively. This process is not perfect, as can be seen from the results obtained from Hyperband, where the best-performing parameter combination yielded a model that underperformed, or through reducing the pneumonia class weights, the best performing parameters from the FIFO scheduler resulted in a model that was extremely good at finding COVID-19 patients, but completely incapable of predicting pneumonia. However, these results give insight into novel ways the parameters can be tuned and thus the model performance can be improved.

In the case of Asynchronous HyperBand and PBT, both resulting trained models performed more consistently than the original re-trained COVID-Net, with predictions trending towards true positives. The results also highlight the possibility of further improvement with longer training and further fine-tuning of the hyperparameters, both of which are made possible through the scale-up of the GPU resources on the compute clusters.

The reduction in training duration observed in Figure 6a is not infinite; in fact, as more nodes are recruited, the communication overhead between these nodes becomes more complex and more time-consuming, resulting in the flattening of the curve and ultimately the upward trend seen in Figure 6b. To counter this issue, it is important to understand the problem at hand and to recruit the appropriate hardware and software accordingly, while also performing many trials to pinpoint the cut-off at which training is the most efficient.

The work presented in this manuscript describes the large-scale re-training of COVID-Net as a use case to validate a modular medical diagnosis support platform built on an HPC infrastructure and taking advantage of novel and efficient ML algorithms. That is not to say that this work would not be possible without the specific HPC infrastructure used. In fact, the platform makes use of open-source software, making it easily portable onto commercially available cloud computing (CC) solutions. Similarly, the main aim is to develop the base infrastructure that takes advantage of the HPC resources to simplify the development of software that is lightweight enough to be easily deployed in most standard computers available in hospitals, making them a vital tool to support medical personnel.

Given that the medical field is regularly facing time-sensitive problems, this paper highlights the need for platforms that simplify access to cutting-edge resources for model training and development, and also for specially trained experts in the field of ML, data science, and HPC for medical applications, who would advise on applications, assist in setting up the problem solutions, and take part in the data analysis and the development of the diagnostic and treatment techniques of the future.

Finally, since the prototype platform described in this manuscript only used open-access data, there are no privacy risks and thus this issue was not addressed. As the platform moves towards production, and especially before dealing with restricted real-world data, its safety from outside threats will need to be assessed. Additionally, this process is still in its infancy and much work still needs to be done in order to test the robustness of this platform, and validate its performance in real-world use cases.

5. Conclusions

In the present manuscript, the re-training of a COVID-19 detection model was described as a use case through which an HPC-enabled data analysis and ML platform was validated. The MSA available at JSC, especially the scalable storage and computing resources, made it possible (1) to validate the performance of the COVID-Net model on the original COVIDx data as well as new data made available through research partners, (2) to perform large-scale hyperparameter tuning, through which the optimal training parameters for the model were uncovered, and (3) to re-train the model using the selected parameters and highlight the improvement that was achieved. Furthermore, the research also highlights the training speed-up that can be achieved using the platform.

The severity with which the COVID-19 pandemic struck worldwide, and research showing that such global phenomena may become more frequent, highlight the need for research platforms such as the one described in the present manuscript. These platforms would make use of highly efficient computing, communication, and storage technology, as well as open-source and interoperable software, and should be made available to assist the healthcare sector in order to simplify and accelerate the development of medical diagnosis support tools. This does not mean that medical institutions should be required to have access to HPC resources, which would put hospitals at a severe disadvantage, not only in developing countries. Rather, the models developed within these platforms ought to be more portable and easily implementable, while the communication channels between research institutions and medical centres ought to be strengthened, paving the way for effective medical and technological cooperation. Such platforms rely on the availability of data and the willingness of medical institutions to participate in the research, both of which are more likely to increase as the developed and validated models show beneficial effects in the field.

Author Contributions: Conceptualisation, C.B., M.R. and S.F.; methodology, C.B.; software, C.B. and M.A.; validation, C.B., M.A., M.R. and S.F.; formal analysis, C.B. and M.A.; investigation, C.B.; resources, M.R.; data curation, M.R.; writing—original draft preparation, C.B.; writing—review and editing, C.B., M.R. and S.F.; visualisation, M.A.; supervision, M.R., S.F., A.S. and S.B.; project administration, M.R. and S.F.; funding acquisition, M.R. All authors have read and agreed to the published version of the manuscript.

Funding: This work was performed in the Center of Excellence (CoE) Research on AI- and Simulation-Based Engineering at Exascale (RAISE), receiving funding from EU's Horizon 2020 Research and Innovation Framework Programme H2020-INFRAEDI-2019-1 under grant agreement no. 951733. The Icelandic HPC National Competence Center is funded by the EuroCC project that has received funding from the EU HPC Joint Undertaking (JU) under grant agreement no. 951732 and the EOSC COVID-19 Fast Track Grant under grant agreement no. 831644. This publication of the SMITH consortium was supported by the German Federal Ministry of Education and Research, grant no. 01ZZ1803M.

Institutional Review Board Statement: Not Applicable.

Informed Consent Statement: Not Applicable.

Data Availability Statement: The COVIDx dataset is available online at https://www.kaggle.com/datasets/andyczhao/covidx-cxr2 (accessed on 19 December 2022). The EHL dataset is available online at https://b2share.fz-juelich.de/records/aef5d3b8aa044485b9620b95b60c47a2 (accessed on 19 December 2022). COVID-Net is available at https://github.com/lindawangg/COVID-Net (accessed on 19 December 2022). The work described in this paper is available at https://github.com/c-barakat/covidnet_tune (accessed on 19 December 2022).

Acknowledgments: The authors acknowledge the support of E*HealthLine, who provided the data for re-training the COVID-Net model, as well as the Jülich Supercomputing Centre for providing access to the supercomputing resources including the DEEP-EST projects.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AI	artificial intelligence
CC	cloud computing
CNN	convolutional neural network
CP	COVID-19 percentage
CW	class weight
CXR	chest X-ray
DEEP	dynamic exascale entry platform
DL	deep learning
EHL	E*HealthLine
EOSC	European Open Science Cloud
ESB	extreme scale booster
FIFO	first-in, first-out
FTP	file transfer protocol
HPC	high-performance computing
ICU	intensive care unit
JSC	Jülich Supercomputing Centre
JUWELS	Jülich Wizard for European Leadership Science

LR	learning rate
ML	machine learning
MPI	message passing interface
MSA	modular supercomputing architecture
NumPy	Numerical Python
OpenCV	Open-Source Computer Vision Library
PBT	population-based training
PNG	portable network graphics
RT-PCR	reverse-transcription polymerase chain-reaction
SLURM	Simple Linux Utility for Resource Management
SSH	secure shell

References

1. French, G.; Hulse, M.; Nguyen, D.; Sobotka, K.; Webster, K.; Corman, J.; Aboagye-Nyame, B.; Dion, M.; Johnson, M.; Zalinger, B.; et al. Impact of Hospital Strain on Excess Deaths During the COVID-19 Pandemic—United States, July 2020–July 2021. *Morb. Mortal. Wkly. Rep.* **2021**, *70*, 1613–1616. [CrossRef] [PubMed]
2. Tahamtan, A.; Ardebili, A. Real-time RT-PCR in COVID-19 detection: Issues affecting the results. *Expert Rev. Mol. Diagn.* **2020**, *20*, 453–454. [CrossRef]
3. Teymouri, M.; Mollazadeh, S.; Mortazavi, H.; Naderi Ghale-noie, Z.; Keyvani, V.; Aghababaei, F.; Hamblin, M.R.; Abbaszadeh-Goudarzi, G.; Pourghadamyari, H.; Hashemian, S.M.R.; et al. Recent advances and challenges of RT-PCR tests for the diagnosis of COVID-19. *Pathol. Res. Pract.* **2021**, *221*, 153443. [CrossRef]
4. Roshkovan, L.; Chatterjee, N.; Galperin-Aizenberg, M.; Gupta, N.; Shah, R.; Barbosa Jr, E.M.; Simpson, S.; Cook, T.; Nachiappan, A.; Knollmann, F.; et al. The Role of Imaging in the Management of Suspected or Known COVID-19 Pneumonia. A Multidisciplinary Perspective. *Ann. Am. Thorac. Soc.* **2020**, *17*, 1358–1365. [CrossRef]
5. Ai, T.; Yang, Z.; Hou, H.; Zhan, C.; Chen, C.; Lv, W.; Tao, Q.; Sun, Z.; Xia, L. Correlation of chest CT and RT-PCR testing in coronavirus disease 2019 (COVID-19) in China: A report of 1014 cases. *Radiology* **2020**, 296, E32–E40. [CrossRef]
6. Marani, M.; Katul, G.G.; Pan, W.K.; Parolari, A.J. Intensity and frequency of extreme novel epidemics. *Proc. Natl. Acad. Sci. USA* **2021**, *118*, e2105482118. [CrossRef]
7. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255. [CrossRef]
8. Huddar, V.; Desiraju, B.K.; Rajan, V.; Bhattacharya, S.; Roy, S.; Reddy, C.K. Predicting Complications in Critical Care Using Heterogeneous Clinical Data. *IEEE Access* **2016**, *4*, 7988–8001. [CrossRef]
9. Erlingsson, E.; Cavallaro, G.; Galonska, A.; Riedel, M.; Neukirchen, H. Modular supercomputing design supporting machine learning applications. In Proceedings of the 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 21–25 May 2018; pp. 0159–0163. [CrossRef]
10. Sun, H.; Liu, Z.; Wang, G.; Lian, W.; Ma, J. Intelligent Analysis of Medical Big Data Based on Deep Learning. *IEEE Access* **2019**, *7*, 142022–142037. [CrossRef]
11. Sedona, R.; Cavallaro, G.; Jitsev, J.; Strube, A.; Riedel, M.; Benediktsson, J. Remote Sensing Big Data Classification with High Performance Distributed Deep Learning. *Remote. Sens.* **2019**, *11*, 3056. [CrossRef]
12. Jumper, J.; Evans, R.; Pritzel, A.; Green, T.; Figurnov, M.; Ronneberger, O.; Tunyasuvunakool, K.; Bates, R.; Žídek, A.; Potapenko, A.; et al. Highly accurate protein structure prediction with AlphaFold. *Nature* **2021**, *596*, 583–589. [CrossRef]
13. Baek, M.; DiMaio, F.; Anishchenko, I.; Dauparas, J.; Ovchinnikov, S.; Lee, G.R.; Wang, J.; Cong, Q.; Kinch, L.N.; Schaeffer, R.D.; et al. Accurate prediction of protein structures and interactions using a three-track neural network. *Science* **2021**, *373*, 871–876. [CrossRef] [PubMed]
14. Lugarà, M.; Tamburrini, S.; Coppola, M.G.; Oliva, G.; Fiorini, V.; Catalano, M.; Carbone, R.; Saturnino, P.P.; Rosano, N.; Pesce, A.; et al. The Role of Lung Ultrasound in SARS-CoV-19 Pneumonia Management. *Diagnostics* **2022**, *12*, 1856. [CrossRef] [PubMed]
15. Wang, L.; Lin, Z.Q.; Wong, A. COVID-Net: A tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images. *Sci. Rep.* **2020**, *10*, 19549. [CrossRef]
16. Lee, C.P.; Lim, K.M. COVID-19 Diagnosis on Chest Radiographs with Enhanced Deep Neural Networks. *Diagnostics* **2022**, *12*, 1828. [CrossRef] [PubMed]
17. Song, Y.; Liu, J.; Liu, X.; Tang, J. COVID-19 Infection Segmentation and Severity Assessment Using a Self-Supervised Learning Approach. *Diagnostics* **2022**, *12*, 1805. [CrossRef] [PubMed]
18. Elshennawy, N.M.; Ibrahim, D.M.; Sarhan, A.M.; Arafa, M. Deep-Risk: Deep Learning-Based Mortality Risk Predictive Models for COVID-19. *Diagnostics* **2022**, *12*, 1847. [CrossRef] [PubMed]
19. Rajaraman, S.; Guo, P.; Xue, Z.; Antani, S.K. A Deep Modality-Specific Ensemble for Improving Pneumonia Detection in Chest X-rays. *Diagnostics* **2022**, *12*, 1442. [CrossRef]

20. Kesselheim, S.; Herten, A.; Krajsek, K.; Ebert, J.; Jitsev, J.; Cherti, M.; Langguth, M.; Gong, B.; Stadtler, S.; Mozaffari, A.; et al. JUWELS Booster—A Supercomputer for Large-Scale AI Research. In *Proceedings of the High Performance Computing*; Jagode, H., Anzt, H., Ltaief, H., Luszczek, P., Eds.; Springer International Publishing: Cham, Switzerland, 2021; pp. 453–468.
21. Moritz, P.; Nishihara, R.; Wang, S.; Tumanov, A.; Liaw, R.; Liang, E.; Elibol, M.; Yang, Z.; Paul, W.; Jordan, M.I.; et al. Ray: A Distributed Framework for Emerging AI Applications. In Proceedings of the 13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18), Carlsbad, CA, USA, 8–10 October 2018; USENIX Association: Carlsbad, CA, USA, 2018; pp. 561–577.
22. Nijor, S.; Rallis, G.; Lad, N.; Gokcen, E. Patient safety issues from information overload in electronic medical records. *J. Patient Saf.* **2022**, *18*, e999–e1003. [CrossRef]
23. Manor-Shulman, O.; Beyene, J.; Frndova, H.; Parshuram, C.S. Quantifying the volume of documented clinical information in critical illness. *J. Crit. Care* **2008**, *23*, 245–250. [CrossRef]
24. Lundervold, A.S.; Lundervold, A. An overview of deep learning in medical imaging focusing on MRI. *Z. Med. Phys.* **2019**, *29*, 102–127. [CrossRef]
25. Barakat, C.; Fritsch, S.; Riedel, M.; Brynjólfsson, S. An HPC-Driven Data Science Platform to Speed-up Time Series Data Analysis of Patients with the Acute Respiratory Distress Syndrome. In Proceedings of the 2021 44th International Convention on Information, Communication and Electronic Technology (MIPRO), Opatija, Croatia, 24–28 May 2021; pp. 311–316. [CrossRef]
26. Barakat, C.; Fritsch, S.; Sharafutdinov, K.; Ingólfsson, G.; Schuppert, A.; Brynjólfsson, S.; Riedel, M. Lessons learned on using High-Performance Computing and Data Science Methods towards understanding the Acute Respiratory Distress Syndrome (ARDS). In Proceedings of the 2022 45th Jubilee International Convention on Information, Communication and Electronic Technology (MIPRO), Opatija, Croatia, 23–27 May 2022; pp. 368–373. [CrossRef]
27. Farag, H.H.; Said, L.A.A.; Rizk, M.R.M.; Ahmed, M.A.E. Hyperparameters optimization for ResNet and Xception in the purpose of diagnosing COVID-19. *J. Intell. Fuzzy Syst.* **2021**, *41*, 3555–3571. [CrossRef]
28. Adedigba, A.P.; Adeshina, S.A.; Aina, O.E.; Aibinu, A.M. Optimal hyperparameter selection of deep learning models for COVID-19 chest X-ray classification. *Intell.-Based Med.* **2021**, *5*, 100034. [CrossRef] [PubMed]
29. Arman, S.E.; Rahman, S.; Deowan, S.A. COVIDXception-Net: A Bayesian Optimization-Based Deep Learning Approach to Diagnose COVID-19 from X-Ray Images. *SN Comput. Sci.* **2021**, *3*, 115. [CrossRef] [PubMed]
30. Suarez, E.; Eickert, N.; Lippert, T. Modular Supercomputing architecture: From idea to production. In *Contemporary High Performance Computing: From Petascale toward Exascale*, 1st ed.; Vetter, J., Ed.; CRC Press: Boca Raton, FL, USA, 2019; Volume 3, pp. 223–251.
31. Eicker, N.; Lippert, T.; Moschny, T.; Suarez, E.; The DEEP Project. The DEEP Project An alternative approach to heterogeneous cluster-computing in the many-core era. *Concurr. Comput. Pract. Exp.* **2016**, *28*, 2394–2411. .: 10.1002/cpe.3562. [CrossRef]
32. Suarez, E.; Kreuzer, A.; Eicker, N.; Lippert, T. The DEEP-EST project. In *Porting Applications to a Modular Supercomputer-Experiences from the DEEP-EST Project*; Schriften des Forschungszentrums Jülich IAS Series; Forschungszentrum Jülich GmbH Zentralbibliothek, Verlag: Jülich, Germany, 2021; Volume 48, pp. 9–25.
33. Sharafutdinov, K.; Bhat, J.S.; Fritsch, S.J.; Nikulina, K.; Samadi, M.E.; Polzin, R.; Mayer, H.; Marx, G.; Bickenbach, J.; Schuppert, A. Application of convex hull analysis for the evaluation of data heterogeneity between patient populations of different origin and implications of hospital bias in downstream machine-learning-based data processing: A comparison of 4 critical-care patient datasets. *Front. Big Data* **2022**, *5*, 603429. [CrossRef]
34. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. *arXiv* **2015**, arXiv:1512.03385.
35. Luo, G. A review of automatic selection methods for machine learning algorithms and hyper-parameter values. *Netw. Model. Anal. Health Inform. Bioinform.* **2016**, *5*, 1–16. [CrossRef]
36. Liaw, R.; Liang, E.; Nishihara, R.; Moritz, P.; Gonzalez, J.E.; Stoica, I. Tune: A Research Platform for Distributed Model Selection and Training. *arXiv* **2018**, arXiv:1807.05118.
37. Jaderberg, M.; Dalibard, V.; Osindero, S.; Czarnecki, W.M.; Donahue, J.; Razavi, A.; Vinyals, O.; Green, T.; Dunning, I.; Simonyan, K.; et al. Population Based Training of Neural Networks. *arXiv* **2017**, arXiv:1711.09846. [CrossRef]
38. Li, L.; Jamieson, K.; DeSalvo, G.; Rostamizadeh, A.; Talwalkar, A. Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization. *J. Mach. Learn. Res.* **2018**, *18*, 1–52.
39. Li, L.; Jamieson, K.; Rostamizadeh, A.; Gonina, E.; Hardt, M.; Recht, B.; Talwalkar, A. A System for Massively Parallel Hyperparameter Tuning. *arXiv* **2018**, arXiv:1810.05934. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Paper V

Developing an Artificial Intelligence-Based Representation of a Virtual Patient Model for Real-Time Diagnosis of Acute Respiratory Distress Syndrome

C. Barakat, K. Sharafutdinov, J. Busch, S. Saffaran, D.G. Bates, J.G. Hardman, A. Schuppert, S. Brynjólfsson, S. Fritsch, M. Riedel

2023

This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/).

C. Barakat performed the work described in the manuscript on the platform developed in previous work. He adapted the scripts with the assistance of K. Sharafutdinov to adapt the MATLAB simulation into a C-based simulation and J. Busch to implement hyperparameter optimization on the ML-based model. Finally, he performed the final re-training of the model, based on which he drafted the original manuscript.

Article

Developing an Artificial Intelligence-Based Representation of a Virtual Patient Model for Real-Time Diagnosis of Acute Respiratory Distress Syndrome

Chadi Barakat[1,2,3,†,*], Konstantin Sharafutdinov[3,4], Josefine Busch[1], Sina Saffaran[5], Declan G. Bates[5], Jonathan G. Hardman[6], Andreas Schuppert[3,4], Sigurður Brynjólfsson[2], Sebastian Fritsch[1,3,7,‡], and Morris Riedel[1,2,3,‡]

[1] Jülich Supercomputing Centre, Forschungszentrum Jülich, 52428 Jülich, Germany
[2] School of Engineering and Natural Science, University of Iceland, 107 Reykjavik, Iceland
[3] SMITH Consortium of the German Medical Informatics Initiative, 07747 Leipzig, Germany
[4] Joint Research Centre for Computational Biomedicine, University Hospital RWTH Aachen, 52074 Aachen, Germany
[5] School of Engineering, University of Warwick, CV4 7AL Coventry, UK
[6] School of Medicine, University of Nottingham, NG7 2RD Nottingham, UK
[7] Department of Intensive Care Medicine, University Hospital RWTH Aachen, 52074 Aachen, Germany
* Correspondence: c.barakat@fz-juelich.de
† Current address: Jülich Supercomputing Centre, Forschungszentrum Jülich, 52428 Jülich, Germany.
‡ These authors contributed equally to this work.

Abstract: Acute Respiratory Distress Syndrome (ARDS) is a condition that endangers the lives of many Intensive Care Unit patients through gradual reduction of lung function. Due to its heterogeneity, this condition has been difficult to diagnose and treat, although it has been the subject of continuous research, leading to the development of several tools for modeling disease progression on the one hand, and guidelines for diagnosis on the other, mainly the "Berlin Definition". This paper describes the development of a deep learning-based surrogate model of one such tool for modeling ARDS onset in a virtual patient: the Nottingham Physiology Simulator. The model development process takes advantage of current machine learning and data analysis techniques, as well as efficient hyperparameter tuning methods, within a high-performance computing-enabled data science platform. The lightweight models developed through this process present comparable accuracy to the original simulator (per-parameter $R^2 > 0.90$). The experimental process described herein serves as a proof of concept for the rapid development and dissemination of specialised diagnosis support systems based on pre-existing generalised mechanistic models, making use of supercomputing infrastructure for the development and testing processes and supported by open-source software for streamlined implementation in clinical routine.

Keywords: High-Performance Computing; Machine Learning; ICU; ARDS; Surrogate Model; Virtual Patient

Citation: Barakat, C.; Sharafutdinov, K.; Busch, J.; Saffaran, S.; Bates, D.G.; Hardman, J.G.; Schuppert, A.; Brynjólfsson, S.; Fritsch, S.; and Riedel, M. Developing an Artificial Intelligence-Based Representation of a Virtual Patient Model for Real-Time Diagnosis of Acute Respiratory Distress Syndrome. *Diagnostics* **2022**, *1*, 0. https://doi.org/

Received:
Accepted:
Published:

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2023 by the authors. Submitted to *Diagnostics* for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Respiratory diseases endanger the ability of the respiratory system to supply the body with oxygen and to eliminate carbon dioxide sufficiently, potentially causing life-threatening consequences. These conditions are caused on one hand primarily by damaging the pulmonary tissue through, for instance, infection, toxic effects of inhaled gases or fluids or trauma. On the other hand, the lung can be affected indirectly as a side-effect of diseases of other organs [1]. Early diagnosis and treatment are essential to achieve positive outcomes for patients [2–5]. Critically ill patients who require treatment in an intensive care unit (ICU) are at high risk of developing respiratory disease, one of the most serious of which is Acute Respiratory Distress Syndrome (ARDS), a condition that was first described by Ashbaugh *et al.* [6]. ARDS is still the subject of intensive research due to its high incidence in ICU patients as reported by Confalonieri *et al* (10.4% of total ICU admissions), and high mortality rate as highlighted by Le *et al.* (30-55% of affected patients) [3,5].

ARDS is further characterised by its heterogeneity and the difficulty of its diagnosis, leading clinicians and researchers to establish of the "Berlin Definition" by which ARDS onset is defined as a ratio of Partial Pressure of Arterial Oxygen (P_aO_2) to Fraction of Inspired Oxygen (F_iO_2) (P/F ratio) of less than 300 mmHg in combination with bilateral opacities in pulmonary imaging and absence of hypervolemia and heart failure [7]. Furthermore, this definition classifies the severity of the condition to be inversely proportional to the value of the P/F ratio. Despite widespread research activities in this field, which were even intensified during the COVID-19 pandemic, effective treatment methods of ARDS are still lacking, resulting in a high mortality rate [3,5]. In fact, Bellani *et al.* highlight that ARDS diagnosis is still delayed or missed in two thirds of patients, leading to severe outcomes [8]. The management of ARDS patients, thus, usually remains supportive with lung-protective mechanical ventilation, prone positioning, and Extracorporeal membrane oxygenation (ECMO) treatment as *ultima ratio* [9–12].

In developing the Nottingham Physiology Simulator (NPS), Hardman *et al.* launched an in-silico tool for modelling pulmonary disease progression and determining the potential effectiveness of treatment methods [13]. This model was later improved upon by Das *et al.* and Saffaran *et al.* to include elements of the cardiovascular system and to improve its performance, which extended its usefulness even further [14,15]. The resulting virtual patient simulator was validated through generating outputs for initial conditions similar to real-world ARDS patients, and it was found that these model outputs were consistently comparable with the source clinical data [16]. With a tool such as the NPS, clinicians and biomedical engineers can consistently and accurately model individual patient states, predict the onset of disease, and formulate and validate potential treatment methods to guarantee the best outcomes for patients.

The development of models such as the NPS was simplified with the advent of Electronic Health Records (EHRs). Making large amounts of clinical data easily accessible has enabled a lot of research in healthcare, has helped highlight pathological patterns and uncover treatment methods, but has also sparked discussions about patient privacy and data security [17–19]. As these records grow into the realm of Medical Big Data, the need to develop more efficient storage for the data and more capable computing resources to process it grows at a similar rate [20–23]. Thus, it is essential to make High-Performance Computing (HPC) available for biomedical applications, and to develop the algorithms to take advantage of these resources in order to clean, process, analyse, and extract information from the available data.

It follows that several teams have already employed available HPC resources in the storage and analysis of medical big data or in training Machine Learning (ML) and Deep Learning (DL) models. Kesselheim *et al.* applied the Jülich Wizard for European Leadership Science (JUWELS) (https://www.fz-juelich.de/en/ias/jsc/systems/supercomputers/juwels (accessed on 03 February 2023)) supercomputing cluster and booster to perform pretraining of the ResNet-152 DL network. Their goal was to highlight the speedup achieved using the HPC resources and to eventually perform large-scale transfer learning using the publicly available COVIDx (https://www.kaggle.com/datasets/andyczhao/covidx-cxr2 (accessed on 03 February 2023)) dataset to develop a tool for rapid Covid-19 detection from Chest X-Rays (CXRs) [24]. The researchers also discussed using their supercomputing resources to improve the available ML methods for RNA structure prediction. In a similar vein, Baek *et al.* and Jumper *et al.* concurrently published their results for Artificial Intelligence (AI) models, RoseTTAFold and AlphaFold, that make use of the HPC clusters available at the University of Washington and at Google, respectively [25,26]. Both teams used an implementation of multi-track DL networks in an attempt to solve the protein folding problem, and in both cases the results were highly accurate. Finally, Zhang *et al.* made use of HPC to perform hyperparameter tuning on a ML model for Alzheimer's disease detection [27]. Their work highlights the speedup that can be achieved by making use of HPC, especially in situations where many trials need to be performed with minute changes in order to find the optimal parameter combination that produces the best results.

This paper describes the process by which a ML and data science platform, that takes advantage of Modular Supercomputing Architecture (MSA) available from the Jülich Supercomputing Centre (JSC), is used to build a surrogate model of the NPS with the intention of implementing it for streamlined ARDS diagnosis support [28–30]. In order to achieve this primary goal, several steps need to be completed as follows:

- Medical data collection, cleaning, analysis and visualisation.
- Data augmentation through statistical analysis of the available clinical data.
- Parallel simulation of patient states using a ported NPS.
- Parallel hyperparameter optimisation of the developed DL model using Ray Tune [31].
- Final training of the DL-based surrogate model and validation of the results with the original simulation.

As Gherman *et al.* highlighted, several researchers have already developed ML surrogate models from complex mechanistic models [32]. These surrogates benefit greatly from the high accuracy of the mechanistic models they emulate, while avoiding the computation overhead associated with equilibrating multiple complex differential equations. This aspect, coupled with the use of a pre-established HPC-enabled data science and ML platform that was validated in previously published work, represent the core innovations of the research described in this manuscript [28,29]. In this way, the HPC resources are instrumental to the accelerated development and testing of the surrogate.

This work is done as part of the use case Algorithmic Surveillance of Intensive Care Unit patients with ARDS (ASIC) which is part of the Smart Medical Information Technology for Healthcare (SMITH) project under the guidance of the German Federal Ministry of Education and Research (BMBF) [33,34]. Furthermore, the work described here paves the way for the future development of surrogate models from pre-established mechanistic disease representations, thus providing valuable tools to accelerate diagnosis in critical situations.

2. Materials and Methods

The experimental process leading towards completion of the research objective described in the Introduction is represented in Figure 1. The subsections below go further into the details of each step of the experimental process as well as the hardware and software implemented within them.

2.1. HPC Resources

The Dynamic Exascale Entry Platform (DEEP) series of projects (https://www.deep-projects.eu/ (accessed on 03 February 2023)) was set up to highlight the benefits of using heterogeneous architectures in HPC to pave the way towards exascale computing by introducing boosters alongside traditional supercomputing clusters [35,36]. The boosters, which run independently of the cluster nodes used for traditional supercomputing tasks, offer the option of expanding storage and compute power for specific tasks, including large-memory nodes for image processing tasks and multi-GPU nodes for accelerated DL tasks. Thus, the DEEP projects introduced the Modular Supercomputing Architecture (MSA) concept that would later be used in development systems such as the JUWELS cluster and booster, unveiled in 2018 and 2020, respectively [37]. The specific configuration

Table 1. Partitions on the DEEP Prototype.

Partition	Nodes	CPUs/Node	GPU
DEEP-Data Analytics Module	16	96	NVIDIA V100 + Intel Stratix10 FGPA
DEEP-Extreme Scale Booster	75	16	NVIDIA V100
DEEP-Cluster Module	50	48	n/a

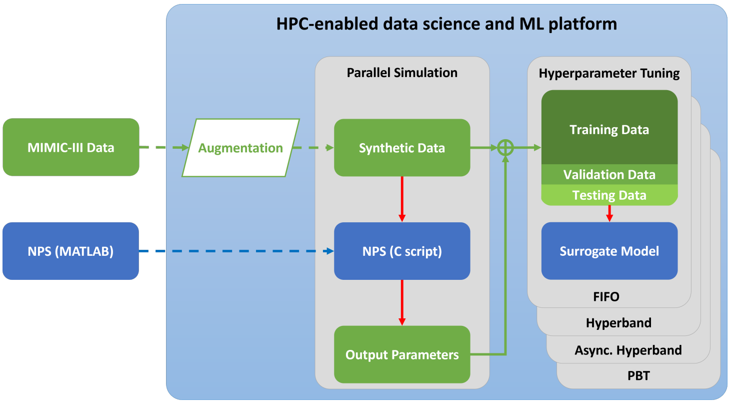


Figure 1. Flow diagram describing the data augmentation and surrogate model development steps within the data analysis and ML platform. The flow of data is represented in green, while the models are represented in blue.

of the cluster-booster prototype set up in the DEEP project and its subsequent projects, DEEP-Extended Reach (DEEP-ER) and DEEP-Extreme Scale Technologies (DEEP-EST), is presented in Table 1.

2.2. Software and Libraries

2.2.1. Nottingham Physiology Simulator

The NPS is made available as part of the SMITH project as a central MATLAB (https://www.mathworks.com/products/matlab.html (accessed on 03 February 2023)) script accompanied by peripheral functions written either in MATLAB, or in C-script and converted at initial startup into the MATLAB executable (.mex) format. Version 1.4 of the simulator was made available for this research as part of the SMITH project. Further updates to the NPS have already been implemented which improve its performance [15], however all of the experiments described in this manuscript concern the version mentioned above. The simulator loads patient data from prepared input files, then runs a preset number of cycles during which it solves a series of differential equations that model the gas exchange occurring during a breathing cycle.

Disease states can be modelled in the simulator through adjusting the input parameters, such as reducing oxygenation, reducing lung compliance, or changing the acid-base balance of the blood [16,38], which are typical pathophysiological alterations in ARDS patients [39]. Previous research has validated the performance of the NPS compared to the responses of real patients in the ICU [13,15,40].

Given all of the above, the NPS is certainly a valuable tool in the hands of clinicians aiming to understand medical conditions such as ARDS and to analyse potential treatment methods. It does however have specific shortcomings:

- the time required to run individual simulations makes it unfeasible to use the NPS in diagnosis support especially for more time-critical clinical situations.
- the outputs are broad and extremely detailed, requiring users to filter through them in order to extract the information useful for their specific task.
- it uses proprietary and license-based software which is a limiting factor for applications on a large scale, especially in remote clinics that would not have proper funding for it.

These shortcomings highlight the need to convert the NPS and to develop the surrogate model as described in the remainder of this manuscript.

2.2.2. Software Used in Model Conversion

As mentioned above, the NPS is built in MATLAB and thus is implemented on a local machine running MATLAB version R2019a within Windows 10 version 22H2. Additionally, the MATLAB Coder (https://www.mathworks.com/products/matlab-coder.html (accessed on 03 February 2023)) software plugin is used in order to export the simulation as a C-script and package it for implementation on the HPC cluster.

The remainder of the programming done for this project uses the Python (https://www.python.org/ (accessed on 03 February 2023)) programming language with additional packages installed through the built-in pip function or loaded from the list of pre-installed modules available on the HPC cluster. The packages include Numerical Python (NumPy) (https://numpy.org/ (accessed on 03 February 2023)) and Pandas (https://pandas.pydata.org/ (accessed on 03 February 2023)) for data structure manipulation, MatPlotLib (https://matplotlib.org/ (accessed on 03 February 2023)) for data visualisation, Keras (https://keras.io/ (accessed on 03 February 2023)) (running from within TensorFlow (https://www.tensorflow.org/ (accessed on 03 February 2023))) and Scikit-Learn (https://scikit-learn.org/ (accessed on 03 February 2023)) for performing the ML tasks, and mpi4py to bind to the Message Passing Interface (MPI) and handle the parallelisation aspect of some of the data manipulation tasks [41]. Hyperparameter tuning is done using Ray Tune, which in turn employs different scheduling algorithms in order to simplify the task of finding the optimal parameters for training the final model [31]. Finally, the HPC cluster employs the Simple Linux Utility for Resource Management (SLURM) scheduler (https://slurm.schedmd.com/ (accessed on 03 February 2023)) in order to distribute the submitted training and tuning jobs onto the available computing resources. The submission of jobs is done using shell scripts that define the environments to load and the resources to recruit for each specific job.

2.3. Model Preparation

In order to build the surrogate model, it is necessary to convert the NPS to a format that can more easily be run in parallel, which would then be used to generate data to train the DL model with. Exporting the model in C-script would be a simple task given its similarity to the MATLAB programming language, as well as the availability of the MATLAB Coder plugin. Accordingly, the various peripheral function files that make up the NPS are grouped into a single script as per the requirements of the MATLAB Coder, and the input parameters are defined according to the variables provided in the patient data.

Additionally, the original model outputs an array containing several parameters recorded over every time step of the simulation, which made exporting values difficult. Therefore, the output parameters are reduced to only include the final values of markers for a pulmonary impairment, which can be consistent with an ARDS onset (P_aO_2, Partial Pressure of Arterial Carbon Dioxide (P_aCO_2), pH, and Bicarbonate).

This converted model is tested locally on several patients and its outputs are compared to those from the original simulation in order to verify its integrity. The duration of each simulation is also recorded in order to evaluate the speed-up achieved through this conversion. Moreover, the same patient simulations are performed on the HPC cluster to both validate the outputs and to highlight the speed-up that can be achieved when running several instances concurrently.

2.4. Data

The data used in this research was collected from the open-source Medical Information Mart for Intensive Care - III (MIMIC-III) database as part of the research done by Sharafutdinov *et al.* also within the scope of the SMITH project [42–44]. Due to the limited number of patients and the inconsistent representation of their parameters, it was decided to generate simulated data based the statistical distribution of the original data extracted from the MIMIC-III database. In order to perform this data augmentation, the statistical distribution

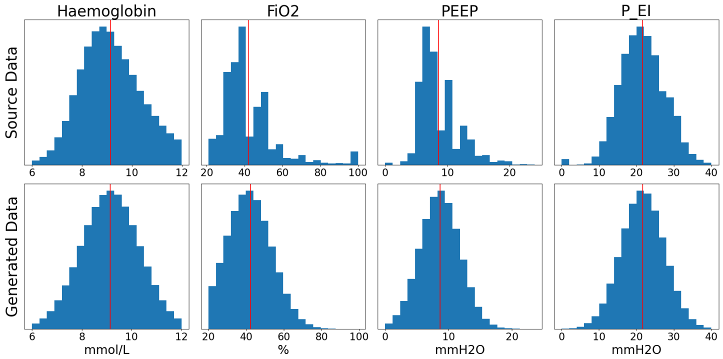


Figure 2. Histograms comparing the distribution of the generated input data with that of the original data. The red lines represent the means for each parameter.

of each parameter listed in Table 2 is analysed, and a generator is developed that outputs randomised snapshots of patient states emulating a wide range of real-world parameter combinations. The choice of these parameters was based on the input parameters required for proper functioning of the NPS. Matching the parameters from the simulation to their equivalent values in the MIMIC-III database was done by Sharafutdinov *et al.* in previous work [43,44]. Table 3 provides a statistical description of the data extracted from the source dataset while Figure 2 presents a comparison of the distributions of the source data and the generated data. In this case, the minimum and maximum cutoff values were chosen based on discussions with clinicians.

As this data is fed into the reduced simulation, the aforementioned markers of ARDS onset of these patients are generated. The end result of this data manipulation step is a collection of 1,000,000 initial states of patients made up of 19 input parameters and 4 associated expected outputs. The output parameters were chosen based on a sensitivity analysis done by Sharafutdinov *et al.* in previously published research [44], and are presented in Table 2. The generated patient states are further subdivided into 80% training / 10% validation / 10% testing datasets to be used to train the DL-based model described in the next section.

2.5. Model Design and Training

In order to select the model architecture that offers the greatest potential training performance, several different approaches are tested. However, the choice was limited by two major factors: first, the architecture does not need to be adapted for timeseries data since the inputs chosen are snapshots of patients' states, as described above, therefore Recurrent Neural Networks (RNNs) are excluded. Second, no advanced neural network architectures, such as residual layers or transformers, are to be used in order to maintain a reduced model complexity. Accordingly, the models tested out in this step were made up of stacked fully connected layers, convolutional layers, or a combination of both.

Several models of both architectures were tested, with varying depths and types of layers, including regularization, dropout, and normalization layers, and with different layer sizes, dropout rates, regularization factors, learning rates, batch sizes, and loss functions. This was done in order to uncover the hyperparameters that have a significant effect on the training process. Each of these architectures was trained for 50 epochs. After this initial testing phase, a provisional best performing model structure is decided based on a statistical comparison of the four output parameters listed in Table 2 (P_aO_2, P_aCO_2, pH, and

	Parameter	Description
Input Parameters	v_sR, v_inR	Used to calculate individual Compartment Resistance to Flow (R_{comp}) values
	v_sVR, v_inVR	Used to calculate individual Compartment Vascular Resistance (VR_{comp}) values
	v_nc	Number of Closed Compartments
	asht	Anatomical Shunt
	RQ	Respiratory Quotient
	VO_2	Oxygen Uptake
	VD_{phys}	Volume of Physiological Deadspace
	CO	Cardiac Output
	I:E	Inspiratory to Expiratory Ratio
	Hb	Hæmoglobin
	F_iO_2	Fraction of Inspired Oxygen
	PEEP	Peak End-Expiratory Pressure
	P_{EI}	End-Inspiratory Pressure
	S_vO_2	Venous Oxygen Blood Saturation
	RR	Respiratory Rate
	V_t	Tidal Volume
	BE_a	Arterial Base Excess
Output Parameters	P_aO_2	Partial Pressure of Arterial Oxygen
	P_aCO_2	Partial Pressure of Arterial Carbon Dioxide
	HCO_3	Bicarbonate Concentration
	pH	Blood Acidity Level

Table 2. Input and Output parameters of the C-ported virtual patient simulator.

HCO_3) with the outputs generated by the original simulation. Further improvements of this model are done through hyperparameter optimization as described in the next section.

2.6. Hyperparameter Tuning

Hyperparameters are the variables that affect the way in which a model is built or its training process, and can be altered either through a process of trial and error, or automatically using optimization algorithms [45,46]. In order the uncover potential hyperparameter combinations through which model training and performance can be improved, the Ray (https://www.ray.io/ (accessed on 03 February 2023)) framework is employed to perform hyperparameter tuning [31,47]. This framework can also take advantage of available HPC resources by distributing the tuning process over several nodes, thus reducing the time needed to run the trials and making the process more efficient.

The schedulers used by Ray Tune in the optimization process described in this manuscript are HyperBand, Asynchronous HyperBand, Population-Based Training (PBT), and the default First-In, First-Out (FIFO) [48–50]. These algorithms distribute the tuning task over the available resources and may interfere with the process by introducing perturbations as is the case for PBT, or by shutting down under-performing tasks as is the case for HyperBand and Asynchronous HyperBand. Aside from FIFO which was chosen to

	BE_a	Hb	V_t	PEEP	P_{EI}	F_iO_2	S_vO_2	RR
Unit	mmol/l	mmol/l	ml	cmH_2O	cmH_2O	%	%	
Min	-15	6	220	0	0	20	30	10
Max	15	12	840	24	40	100	100	40
Mean	1.37	9.15	463.80	8.64	21.68	41.08	68.81	20.83
SD	4.42	1.17	115.13	3.15	5.76	12.39	11.03	5.73

Table 3. Statistical description of the parameters extracted from the source dataset.

Table 4. Comparison of the average duration of the original MATLAB-based simulation with the ported C-code version.

	Short Simulation (run_time=60)	Long Simulation (run_time=120)
MATLAB Simulation	51 s	259.1 s
C-based Simulation on HPC	23.1 s	108.8 s

serve as a control in this experiment, the remaining schedulers were chosen based on their purported resource efficiency and accuracy. The comparison of the different algorithms is thus intended to highlight the most successful both in terms of resource use and accuracy of results for this specific application.

In this experiment, the tuned parameters are the learning rate, the batch size, the dropout rate, the loss function, and the presence of an intermediate fully-connected layer before the output layer in the network architecture. The choice of tuning these specific hyperparameters stemmed from the initial testing done in the model design and training phase described in Section 2.5 where changing these parameters had a significant effect on how the models performed. The tuning process is done to minimise the Validation error value, which serves to reduce the possibility of an overfitting model being selected as the best performing trial. After tuning, the best performing parameters for each scheduler are used to retrain the ML-based model and to highlight the improvement in its prediction performance. Best performance is thus based on the models with the most effective loss reduction, and where the output R^2 scores are closest to 1 for all output parameters. These scores quantify the deviation of the model results from the outputs generated by the original simulation.

3. Results

3.1. Performance of the C-based Model

The data generated as per Section 2.4 is used as input for the C-based simulation. To do that, it was necessary to hard code the information into an entry-point function for the simulation, which was done through Python. Additionally, and to take advantage of the available HPC resources, the process was automated through a jobscript that recruits the necessary resources and modules, then initialises the aforementioned python script that in turn scatters the data over the recruited CPUs using MPI. Each worker on the cluster generates its own copies of the entry-point function, compiles and then runs them, then collects the outputs and stores them. When all the tasks are completed successfully, the mother node gathers all the stored outputs, sorts them, and then appends them to the original inputs, before saving them as a Comma-separated values (CSV) file to be used for training the ML model.

Table 4 presents the average duration of a short (60 minute equivalent) and a long (120 minute equivalent) simulation in MATLAB and compares it to the average duration of those simulations using the C-based simulation on HPC, which highlights the speed-up that was achieved through this process.

3.2. Neural Network Architecture Choice

Different types of neural network architectures with varying depths were tested with the available data. The best performing were based on fully-connected layers and on 1D-convolutional layers. Further experiments with different depths of the two architectures were performed. From these experiments it was clear that models built with stacked fully-connected layers underperformed compared to the approaches using convolutional layers. Additionally, extending the training duration did not lead to improvement in the results, and in some cases led to overfitting.

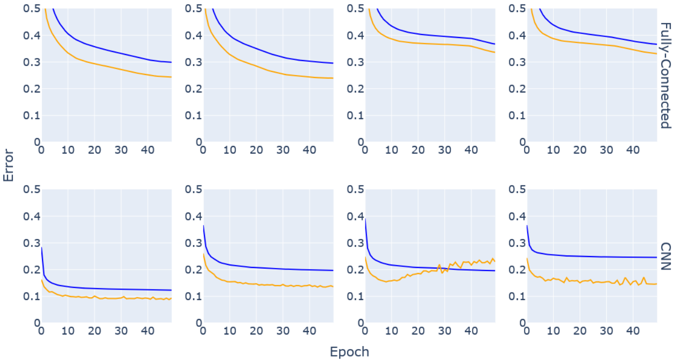


Figure 3. Training (blue) and Validation (orange) MAE for several neural networks built either using the fully-connected architecture or as CNNs.

Applying CNN-based models to the task at hand resulted in more consistent performance even with shallow architectures. Additionally, some of the Convolutional Neural Network (CNN) models did overfit, but in general the models using this architecture reached lower Mean Absolute Error (MAE) in fewer training steps than the fully-connected models. The evolution of the MAE during training and validation for several models of both network architectures are presented in Figure 3.

For both of these architectures, several iterations of testing were done during which the depths of the networks were varied, as well as the widths of their layers, and the addition of dropout and pooling layers into the network design. This process was done in combination with updating the learning rate, batch size, and regularization rates in order to find a rough estimate of the range of hyperparameters as well as the combination of layers that produced a promising model. Based on the results of these experiments, the parameters to be tuned during the hyperparameter tuning process were selected and the ranges over which the tuning would occur were estimated. Furthermore, the chosen network architecture would be based on CNNs with the possibility of adapting the architecture during the tuning process. Additionally, this network would have four 1-dimensional convolution layers, with kernel size of 64 for the first layer and 128 for the remaining layers. The output from the final convolution layer is flattened before being fed either to a intermediate fully-connected layer, or directly to the output layer.

3.3. Hyperparameter Tuning Results

Four different schedulers were used in the hyperparameter tuning step of this experiment in order to provide performance comparisons of the different applications. Figure 4 presents the training and validation MAE values of the different trials for each of the schedulers used.

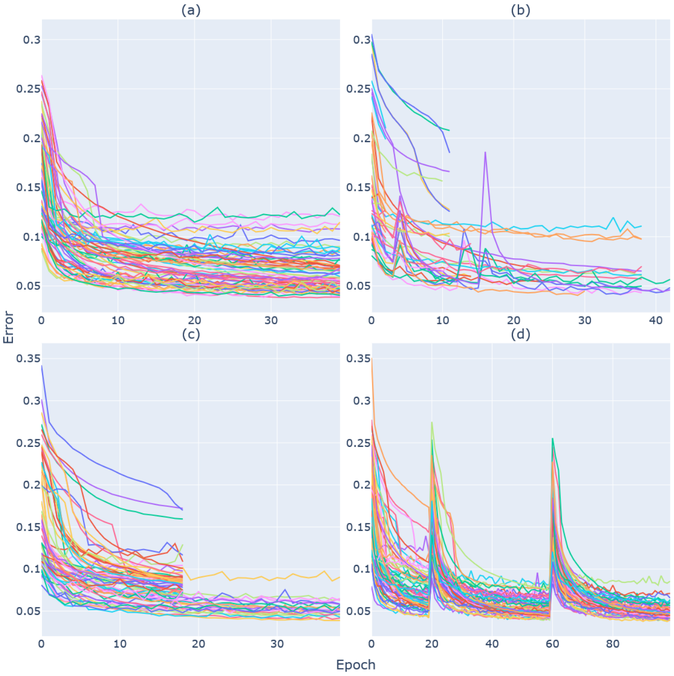


Figure 4. Curves showing the MAE for each of the 64 hyperparameter tuning trials. Each graph represents the trials for one scheduling algorithm: (a) FIFO, (b) HyperBand, (c) Asynchronous HyperBand, and (d) PBT.

Table 5. Parameters of the best performing trial from each Hyperparameter Tuning Scheduler.

Scheduler	Learning Rate	Loss Function	Dropout Rate	Batch Size	Additional Fully-Connected Layer
FIFO	4e-5	MSE	0.5	128	True
HyperBand	8e-5	MAE	0.52	64	True
Async. HyperBand	3e-5	MAE	0.5	128	True
PBT	5.8e-5	MSE	0.54	128	True

Running on 16 nodes of DEEP-ESB cluster, the FIFO scheduler provides a benchmark as it distributes the 64 available tuning jobs. In this approach, new trials cannot be started until the prescribed maximum number of training epochs of previously scheduled tasks are completed. Completing all the trials required a total of 78 minutes. The HyperBand scheduler performed early stopping on many trials that were underperforming, which allowed the tuning process to complete within a shorter duration (46 minutes). Asynchronous HyperBand performed in a similar manner, taking 64 minutes to complete all the trials. The early stopping is evident in the learning curves of these two scheduling algorithms (Figure 4(b) and (c)). Finally, PBT took the longest to complete due to the method with which it implements perturbations at specific times during the model training process. This

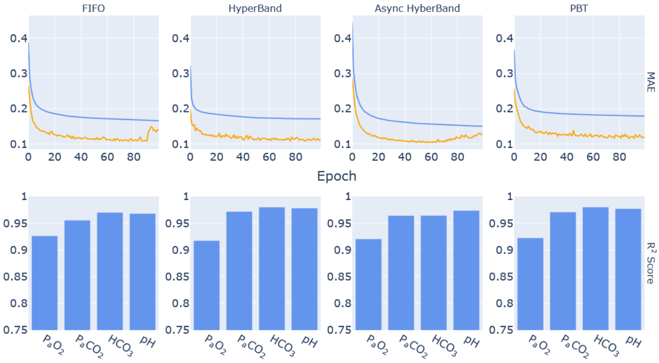


Figure 5. Learning Curves of the Training (blue) and Validation (orange) MAE of the models using the best parameters as selected by each scheduling algorithm, accompanied by their respective per-parameter R^2 score bar graph.

is visible in the spikes of the learning curves in Figure 4(d). At these points in the training process the scheduler reruns each trial with a slightly modified Learning Rate. This resulted in the hyperparameter tuning with the PBT scheduler taking 160 minutes to complete.

The best performing model parameters from each scheduler are presented in Table 5. A common aspect of the best performing models is the presence of the intermediate fully-connected layer before the output layer of the network. Similarly, the dropout rates and learning rates were all within close range for the four models.

3.4. Final Model Performance Analysis

The results from each parameter combination listed in Table 5 are presented in Figure 5. The learning curves for the networks trained on the parameters from the FIFO and the Asynchronous HyperBand trials both show some overfitting towards the second half of the training process. This is also reflected in the R^2 score graphs for the output parameters, where it can be seen that prediction performance for P_aCO_2, bicarbonate concentration, and pH is reduced compared to the networks from the HyperBand and PBT trials. Additionally, the R^2 score graphs show that P_aO_2 prediction accuracy is consistently lower than the remaining output parameters, although still above 0.90.

4. Discussion

Converting the NPS to C helped highlight the speed-up that can be achieved through the use of HPC resources. Running multiple simulations simultaneously as well as the increased efficiency and reduced overhead of C code reduced the code execution times and made it possible to generate more data with which to train the proposed ML models. In the end, the average duration of simulations was less than half the average duration of simulations in MATLAB. Additionally, processing and storing the output data was a compute- and communication-intensive process which was greatly simplified through the availability of online storage on the pre-established HPC-enabled platform for medical ML and data science [30].

The results of the DL model training step of this research highlighted the inherent differences between a fully-connected (i.e. traditional Artificial Neural Network (ANN)) architecture and a convolution-based approach. While ANNs are more likely to give value to every input parameter, CNNs are more adapted to uncover connections between the inputs and infer meaning from them, which might explain why these networks consistently performed better. The results in Figure 3 show that CNNs might overfit the data if the layers are not well tuned, but in most cases the performance surpassed that of ANNs and lower MAE values were reached in shorter training periods, which ultimately makes the convolutional approach more resource-efficient. Additionally, the curves consistently show the validation error being lower than the training error; this is due to the regularization and dropout layers introduced in the network architectures to reduce overfitting. These layers are active during the training process but inactive by design during validation and testing (https://keras.io/getting_started/faq/ (accessed 10 February, 2023)).

Similarly, when considering resource-efficiency, HyperBand and its successor Asynchronous HyperBand make the best use of the available resources to distribute the available tasks. Besides the reduction in compute time, these two approaches minimise stragglers, that is the number of allocated resources that are not effectively being used for computational tasks. Furthermore, the recommendations from the Ray framework highlight Asynchronous HyperBand as a more capable and efficient scheduler than the original HyperBand (https://docs.ray.io/en/latest/tune/api_docs/schedulers.html (accessed 10 February, 2023)). In the case of PBT, resource efficiency is secondary to uncovering more effective approaches through parameter perturbations. Although this approach could be beneficial for applications where minor changes of parameters might greatly alter the outcome of the experiment, the computational overhead necessary for PBT to complete the trials was judged too great for the research purposes described in this manuscript.

The performance of these models is comparable to the performance of the first CNN model in Figure 3, which shows that the best combination of parameters can be reached through a process of trial and error, although it required running several trials to find the best parameters was extremely time and resource consuming and the many combinations were difficult to keep track of. Making use of the hyperparameter tuning methods streamlined the process and had the added benefit of managing the compute resources and distributing the trials without much interference.

The models trained on the best parameters generated from the tuning process highlight the need to take advantage of early stopping during training. Such an approach might produce better predictive performance from the models trained on the parameters selected by FIFO and Asynchronous HyperBand where overfitting was a clear issue. The model trained on the parameters selected by HyperBand took the longest time to train due to the lower batch size but still had a performance similar to that of the PBT model in terms of R^2 scores for P_aCO_2, bicarbonate concentration, and pH. Moreover, it is clear from the results that all models have high prediction accuracy for the 4 output parameters (R^2>0.90), although the prediction of P_aO_2 was consistently lower. This could be due to possible physiological patterns that were not effectively represented within the data, although future tests with larger data sizes might shed more light on the issue.

These results highlight the fact that the surrogate model manages to accurately emulate the performance of the NPS within a statistically acceptable range. Although the performance of the models developed through this approach has not been compared with existing diagnostic support models, the surrogate model benefits greatly from the accuracy that is inherent to the original mechanistic simulation. On the other hand, in replacing the NPS with the DL-based surrogate model, the computational overhead due to nested calculation and equilibration loops is reduced. Additionally, following the experimental procedure described herein, further surrogate models can easily be developed from the NPS with the intent of diagnosing other conditions.

The results described in this research further showcase the benefits of building specialised surrogate models from existing complex medical mechanistic models, a process that is well established in many scientific fields as described by Gherman *et al.* [32]. Through this process, significantly representative, more easily applicable, and more lightweight models can be made available within hospital ICUs. This has the added benefit of not exposing ICUs to unnecessary external threats of data breaches, not requiring specialised and closed-source software, and at the same time not exposing the specific inner workings of the models themselves. Furthermore, this approach benefits from the portability of the developed models, as they can be trained within the platform and exported as offline regressors to be implemented within a container environment. These benefits come at the price of slightly reduced accuracy, although the resulting model predictions are still adequate for supporting clinicians in diagnosing potential disease onset and identifying the need for extra medical attention for a given patient. Another shortcoming of the research described herein is the fact that our surrogate is effectively a black box model. This goes against the current *modus operandi* of model development for clinical applications where explainable AI methods are recommended. It follows that developing explainable AI models for clinical diagnosis is one of the research focus points within the developed ML and data science platform described in this manuscript.

5. Conclusions

This article described the process by which a pre-established machine learning and data science platform was used to facilitate the conversion of a MATLAB-based virtual patient model. The process took advantage of available HPC infrastructure to parallelise the original model in order to generate synthetic data that was later used to train ML-based surrogate models. The performance of the models was improved through hyperparameter tuning which also took advantage of parallelisation. The resulting model performance closely mimics the performance of the original model, though with a massive improvement in the speed with which the results are generated. Additionally, the work shed light on the resource use as a means by which to improve efficiency; algorithmic finetuning of the models using parallel computing can efficiently uncover parameter combinations that would otherwise require a long process of trial and error. The work on model conversion is far from complete but offers a glimpse into the clinical applications of virtual patient simulators as real-time diagnostic support tools for clinicians and ICU personnel, especially in situations where early warning can greatly improve outcomes for patients.

Author Contributions: Conceptualization, C.B., S.F. and M.R.; methodology, C.B., S.F., M.R. and A.S.; software, C.B., J.B., K.S., S.S., D.G.B. and J.G.H,; validation, C.B., J.B., A.S., S.F. and M.R.; formal analysis, A.S., S.F. and M.R; investigation, C.B., J.B. and K.S.; resources, C.B., K.S., S.S., D.G.B., J.G.H, A.S., S.F. and M.R.; data curation, K.S., A.S., S.F. and M.R.; writing—original draft preparation, C.B.; writing—review and editing, S.F. and M.R.; visualization, J.B.; supervision, A.S., S.B., S.F. and M.R.; project administration, A.S. and M.R.; funding acquisition, A.S. and M.R. All authors have read and agreed to the published version of the manuscript.

Funding: This work was performed in the Center of Excellence (CoE) Research on AI- and Simulation-Based Engineering at Exascale (RAISE) receiving funding from EU's Horizon 2020 Research and Innovation Framework Programme H2020-INFRAEDI-2019-1 under grant agreement No. 951733. The Icelandic HPC National Competence Center is funded by the EuroCC project that has received funding from the EU HPC Joint Undertaking (JU) under grant agreement No. 951732. This publication of the SMITH consortium was supported by the German Federal Ministry of Education and Research, Grant No. 01ZZ1803M.

Institutional Review Board Statement: Not Applicable.

Informed Consent Statement: Not Applicable.

Data Availability Statement: The code for the deep learning models and visualising the outputs as well as the outputs from this experiment are available at https://github.com/c-barakat/ARDS_tune.git. The generated patient data is available under https://doi.org/10.23728/B2SHARE.B143C287BB69482A90ABABE7A5A8EB4A.

Acknowledgments: The authors acknowledge the support of the Jülich Supercomputing Centre for providing access to the supercomputing resources including the DEEP-EST projects, as well as Prof. Lars Küpfer and Dr. Hannah Mayer for making the Nottingham Physiology Simulator available as part of the SMITH project.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
ANN	Artificial Neural Network
ARDS	Acute Respiratory Distress Syndrome
ASIC	Algorithmic Surveillance of Intensive Care Unit patients with ARDS
BMBF	Federal Ministry of Education and Research
CC	Cloud Computing
CNN	Convolutional Neural Network
CP	Covid-19 Percentage
CW	Class Weight
CSV	Comma-separated values
CXR	Chest X-Ray
DEEP	Dynamic Exascale Entry Platform
DL	Deep Learning
ECMO	Extracorporeal membrane oxygenation
EHL	E*HealthLine
EHR	Electronic Health Record
EOSC	European Open Science Cloud
ESB	Extreme Scale Booster
FIFO	First-In, First-Out
F_iO_2	Fraction of Inspired Oxygen
FTP	File Transfer Protocol
HPC	High-Performance Computing
ICU	Intensive Care Unit
JSC	Jülich Supercomputing Centre
JUWELS	Jülich Wizard for European Leadership Science
MAE	Mean Absolute Error
MIMIC-III	Medical Information Mart for Intensive Care - III
ML	Machine Learning
MPI	Message Passing Interface
MSA	Modular Supercomputing Architecture
NPS	Nottingham Physiology Simulator
NumPy	Numerical Python
OpenCV	Open Source Computer Vision Library
PBT	Population-Based Training
PNG	Portable Network Graphics
P_aO_2	Partial Pressure of Arterial Oxygen
P_aCO_2	Partial Pressure of Arterial Carbon Dioxide
P/F ratio	Ratio of P_aO_2 to F_iO_2
R_{comp}	Compartment Resistance to Flow
RNN	Recurrent Neural Network
RT-PCR	Reverse transcription polymerase chain reaction
SLURM	Simple Linux Utility for Resource Management
SMITH	Smart Medical Information Technology for Healthcare
SSH	Secure Shell
S_vO_2	Venous Oxygen Blood Saturation
VR_{comp}	Compartment Vascular Resistance

References

1. Bellani, G.; Laffey, J.G.; Pham, T.; Fan, E.; Brochard, L.; Esteban, A.; Gattinoni, L.; Van Haren, F.; Larsson, A.; McAuley, D.F.; et al. Epidemiology, patterns of care, and mortality for patients with acute respiratory distress syndrome in intensive care units in 50 countries. *Jama* **2016**, *315*, 788–800.
2. Ramirez, P.; Lopez-Ferraz, C.; Gordon, M.; Gimeno, A.; Villarreal, E.; Ruiz, J.; Menendez, R.; Torres, A. From starting mechanical ventilation to ventilator-associated pneumonia, choosing the right moment to start antibiotic treatment. *Critical Care* **2016**, *20*, 1–7.

3. Confalonieri, M.; Salton, F.; Fabiano, F. Acute respiratory distress syndrome. *European Respiratory Review* **2017**, *26*, 160116. https://doi.org/10.1183/16000617.0116-2016.
4. Matthay, M.A.; Zemans, R.L.; Zimmerman, G.A.; Arabi, Y.M.; Beitler, J.R.; Mercat, A.; Herridge, M.; Randolph, A.G.; Calfee, C.S. Acute respiratory distress syndrome. *Nature Reviews Disease Primers* **2019**, *5*, 18. https://doi.org/10.1038/s41572-019-0069-0.
5. Le, S.; Pellegrini, E.; Green-Saxena, A.; Summers, C.; Hoffman, J.; Calvert, J.; Das, R. Supervised machine learning for the early prediction of acute respiratory distress syndrome (ARDS). *Journal of Critical Care* **2020**, *60*, 96–102. https://doi.org/10.1016/j.jcrc.2020.07.019.
6. Ashbaugh, D.G.; Bigelow, D.B.; Levine, B.E. Acute Respiratory Distress in Adults. *The Lancet* **1967**, *290*, 319–323. https://doi.org/10.1016/S0140-6736(67)90168-7.
7. The ARDS Definition Task Force. Acute Respiratory Distress Syndrome: The Berlin Definition. *JAMA* **2012**, *307*. https://doi.org/10.1001/jama.2012.5669.
8. Bellani, G.; Pham, T.; Laffey, J.G. Missed or delayed diagnosis of ARDS: a common and serious problem. *Intensive Care Medicine* **2020**, *46*, 1180–1183. https://doi.org/10.1007/s00134-020-06035-0.
9. Villar, J.; Blanco, J.; Añón, J.M.; Santos-Bouza, A.; Blanch, L.; Ambrós, A.; Gandía, F.; Carriedo, D.; Mosteiro, F.; Basaldúa, S.; et al. The ALIEN study: incidence and outcome of acute respiratory distress syndrome in the era of lung protective ventilation. *Intensive Care Medicine* **2011**, *37*, 1932–1941. https://doi.org/10.1007/s00134-011-2380-4.
10. Fan, E.; Brodie, D.; Slutsky, A.S. Acute respiratory distress syndrome: advances in diagnosis and treatment. *Jama* **2018**, *319*, 698–710.
11. Arrivé, F.; Coudroy, R.; Thille, A.W. Early Identification and Diagnostic Approach in Acute Respiratory Distress Syndrome (ARDS). *Diagnostics* **2021**, *11*. https://doi.org/10.3390/diagnostics11122307.
12. Ajibowo, A.O.; Kolawole, O.A.; Sadia, H.; Amedu, O.S.; Chaudhry, H.A.; Hussaini, H.; Hambolu, E.; Khan, T.; Kauser, H.; Khan, A. A Comprehensive Review of the Management of Acute Respiratory Distress Syndrome. *Cureus* **2022**, *14*.
13. Hardman, J.G.; Bedforth, N.M.; Ahmed, A.B.; Mahajan, R.P.; Aitkenhead, A.R. A physiology simulator: validation of its respiratory components and its ability to predict the patient's response to changes in mechanical ventilation. *British Journal of Anaesthesia* **1998**, *81*, 327–332. https://doi.org/10.1093/bja/81.3.327.
14. Das, A.; Gao, Z.; Menon, P.P.; Hardman, J.G.; Bates, D.G. A systems engineering approach to validation of a pulmonary physiology simulator for clinical applications. *Journal of The Royal Society Interface* **2011**, *8*, 44–55. https://doi.org/10.1098/rsif.2010.0224.
15. Saffaran, S.; Das, A.; Laffey, J.; Hardman, J.; Yehya, N.; Bates, D. Utility of Driving Pressure and Mechanical Power to Guide Protective Ventilator Settings in Two Cohorts of Adult and Pediatric Patients With Acute Respiratory Distress Syndrome: A Computational Investigation. *Critical Care Medicine* **2020**, *48*, 8. https://doi.org/10.1097/CCM.0000000000004372.
16. Laviola, M.; Das, A.; Chikhani, M.; Bates, D.; Hardman, J. Computer simulation clarifies mechanisms of carbon dioxide clearance during apnoea. *British Journal of Anaesthesia* **2019**, *122*, 395–401. https://doi.org/https://doi.org/10.1016/j.bja.2018.11.012.
17. Menachemi, N.; Collum, T.H. Benefits and drawbacks of electronic health record systems. *Risk Management and Healthcare Policy* **2011**, p. 47. https://doi.org/10.2147/RMHP.S12985.
18. Cowie, M.R.; Blomster, J.I.; Curtis, L.H.; Duclaux, S.; Ford, I.; Fritz, F.; Goldman, S.; Janmohamed, S.; Kreuzer, J.; Leenay, M.; et al. Electronic health records to facilitate clinical research. *Clinical Research in Cardiology* **2017**, *106*, 1–9. https://doi.org/10.1007/s00392-016-1025-6.
19. Hoerbst, A.; Ammenwerth, E. Electronic Health Records, A Systematic Review on Quality Requirements. *Methods of Information in Medicine* **2018**, *49*, 320–336. https://doi.org/10.3414/ME10-01-0038.
20. Lundervold, A.S.; Lundervold, A. An overview of deep learning in medical imaging focusing on MRI. *Zeitschrift für Medizinische Physik* **2019**, *29*, 102–127. https://doi.org/10.1016/j.zemedi.2018.11.002.
21. Shillan, D.; Sterne, J.A.C.; Champneys, A.; Gibbison, B. Use of machine learning to analyse routinely collected intensive care unit data: a systematic review. *Critical Care* **2019**, *23*, 284. https://doi.org/10.1186/s13054-019-2564-9.
22. Sun, H.; Liu, Z.; Wang, G.; Lian, W.; Ma, J. Intelligent Analysis of Medical Big Data Based on Deep Learning. *IEEE Access* **2019**, *7*, 142022–142037. https://doi.org/10.1109/ACCESS.2019.2942937.
23. Elshennawy, N.M.; Ibrahim, D.M.; Sarhan, A.M.; Arafa, M. Deep-Risk: Deep Learning-Based Mortality Risk Predictive Models for COVID-19. *Diagnostics* **2022**, *12*, 1847. https://doi.org/10.3390/diagnostics12081847.
24. Kesselheim, S.; Herten, A.; Krajsek, K.; Ebert, J.; Jitsev, J.; Cherti, M.; Langguth, M.; Gong, B.; Stadtler, S.; Mozaffari, A.; et al. JUWELS Booster – A Supercomputer for Large-Scale AI Research. In Proceedings of the High Performance Computing; Jagode, H.; Anzt, H.; Ltaief, H.; Luszczek, P., Eds.; Springer International Publishing: Cham, 2021; pp. 453–468.
25. Baek, M.; DiMaio, F.; Anishchenko, I.; Dauparas, J.; Ovchinnikov, S.; Lee, G.R.; Wang, J.; Cong, Q.; Kinch, L.N.; Schaeffer, R.D.; et al. Accurate prediction of protein structures and interactions using a three-track neural network. *Science* **2021**, *373*, 871—876. https://doi.org/10.1126/science.abj8754.
26. Jumper, J.; Evans, R.; Pritzel, A.; Green, T.; Figurnov, M.; Ronneberger, O.; Tunyasuvunakool, K.; Bates, R.; Žídek, A.; Potapenko, A.; et al. Highly accurate protein structure prediction with AlphaFold. *Nature* **2021**, *596*, 583—589. https://doi.org/10.1038/s41586-021-03819-2.
27. Zhang, F.; Petersen, M.; Johnson, L.; Hall, J.; O'Bryant, S.E. Hyperparameter Tuning with High Performance Computing Machine Learning for Imbalanced Alzheimer's Disease Data. *Applied Sciences* **2022**, *12*, 6670. https://doi.org/10.3390/app12136670.

28. Barakat, C.; Fritsch, S.; Riedel, M.; Brynjolfsson, S. An HPC-Driven Data Science Platform to Speed-up Time Series Data Analysis of Patients with the Acute Respiratory Distress Syndrome. In Proceedings of the 2021 44th International Convention on Information, Communication and Electronic Technology (MIPRO); IEEE: Opatija, Croatia, 2021; pp. 311–316. https://doi.org/10.23919/MIPRO52101.2021.9596840.
29. Barakat, C.; Fritsch, S.; Sharafutdinov, K.; Ingólfsson, G.; Schuppert, A.; Brynjólfsson, S.; Riedel, M. Lessons learned on using High-Performance Computing and Data Science Methods towards understanding the Acute Respiratory Distress Syndrome (ARDS). In Proceedings of the 2022 45th Jubilee International Convention on Information, Communication and Electronic Technology (MIPRO), 2022, pp. 368–373. https://doi.org/10.23919/MIPRO55190.2022.9803320.
30. Barakat, C.; Aach, M.; Schuppert, A.; Brynjólfsson, S.; Fritsch, S.; Riedel, M. Analysis of Chest X-ray for COVID-19 Diagnosis as a Use Case for an HPC-Enabled Data Analysis and Machine Learning Platform for Medical Diagnosis Support. *Diagnostics* **2023**, *13*. https://doi.org/10.3390/diagnostics13030391.
31. Moritz, P.; Nishihara, R.; Wang, S.; Tumanov, A.; Liaw, R.; Liang, E.; Elibol, M.; Yang, Z.; Paul, W.; Jordan, M.I.; et al. Ray: A Distributed Framework for Emerging AI Applications. In Proceedings of the 13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18); USENIX Association: Carlsbad, CA, 2018; pp. 561–577.
32. Gherman, I.M.; Abdallah, Z.S.; Pang, W.; Gorochowski, T.E.; Grierson, C.S.; Marucci, L. Bridging the gap between mechanistic biological models and machine learning surrogates. *PLOS Computational Biology* **2023**, *19*, 1–16. https://doi.org/10.1371/journal.pcbi.1010988.
33. Marx, G.; Bickenbach, J.; Fritsch, S.J.; Kunze, J.B.; Maassen, O.; Deffge, S.; Kistermann, J.; Haferkamp, S.; Lutz, I.; Voellm, N.K.; et al. Algorithmic surveillance of ICU patients with acute respiratory distress syndrome (ASIC): protocol for a multicentre stepped-wedge cluster randomised quality improvement strategy. *BMJ open* **2021**, *11*, e045589.
34. Winter, A.; Stäubert, S.; Ammon, D.; Aiche, S.; Beyan, O.; Bischoff, V.; Daumke, P.; Decker, S.; Funkat, G.; Gewehr, J.; et al. Smart Medical Information Technology for Healthcare (SMITH): Data Integration based on Interoperability Standards. *Methods of Information in Medicine* **2018**, *57*, e92–e105. https://doi.org/10.3414/ME18-02-0004.
35. Eicker, N.; Lippert, T.; Moschny, T.; Suarez, E.; project, f.t.D. The DEEP Project An alternative approach to heterogeneous cluster-computing in the many-core era. *Concurrency and Computation: Practice and Experience* **2016**, *28*, 2394–2411. https://doi.org/https://doi.org/10.1002/cpe.3562.
36. Suarez, E.; Kreuzer, A.; Eicker, N.; Lippert, T. The DEEP-EST project. In *Porting applications to a Modular Supercomputer - Experiences from the DEEP-EST project*; Forschungszentrum Jülich GmbH Zentralbibliothek, Verlag: Jülich, 2021; Vol. 48, *Schriften des Forschungszentrums Jülich IAS Series*, pp. 9–25.
37. Suarez, E.; Eickert, N.; Lippert, T. Modular Supercomputing architecture: from idea to production. In *Contemporary High Performance Computing: From Petascale toward Exascale*, 1 ed.; Vetter, J., Ed.; CRC Press: FL, USA, 2019; Vol. 3, pp. 223–251.
38. Das, A.; Camporota, L.; Hardman, J.G.; Bates, D.G. What links ventilator driving pressure with survival in the acute respiratory distress syndrome? A computational study. *Respiratory Research* **2019**, *20*, 29. https://doi.org/10.1186/s12931-019-0990-5.
39. Derwall, M.; Martin, L.; Rossaint, R. The acute respiratory distress syndrome: pathophysiology, current clinical practice, and emerging therapies. *Expert Review of Respiratory Medicine* **2018**, *12*, 1021–1029.
40. Wang, W.; Das, A.; Ali, T.; Cole, O.; Chikhani, M.; Haque, M.; Hardman, J.; Bates, D. Can computer simulators accurately represent the pathophysiology of individual COPD patients? *Intensive Care Medicine Experimental* **2014**, *2*. https://doi.org/10.1186/s40635-014-0023-0.
41. Dalcin, L.; Fang, Y.L.L. mpi4py: Status Update After 12 Years of Development. *Computing in Science & Engineering* **2021**, *23*, 47–54. https://doi.org/10.1109/MCSE.2021.3083216.
42. Johnson, A.E.; Pollard, T.J.; Shen, L.; Lehman, L.w.H.; Feng, M.; Ghassemi, M.; Moody, B.; Szolovits, P.; Anthony Celi, L.; Mark, R.G. MIMIC-III, a freely accessible critical care database. *Scientific Data* **2016**, *3*, 160035. https://doi.org/10.1038/sdata.2016.35.
43. Sharafutdinov, K.; Bhat, J.S.; Fritsch, S.J.; Nikulina, K.; E. Samadi, M.; Polzin, R.; Mayer, H.; Marx, G.; Bickenbach, J.; Schuppert, A. Application of convex hull analysis for the evaluation of data heterogeneity between patient populations of different origin and implications of hospital bias in downstream machine-learning-based data processing: A comparison of 4 critical-care patient datasets. *Frontiers in Big Data* **2022**, *5*. https://doi.org/10.3389/fdata.2022.603429.
44. Sharafutdinov, K.; Fritsch, S.J.; Iravani, M.; Ghalati, P.F.; Saffaran, S.; Bates, D.G.; Hardman, J.G.; Polzin, R.; Mayer, H.; Marx, G.; et al. Computational simulation of virtual patients reduces dataset bias and improves machine learning-based detection of ARDS from noisy heterogeneous ICU datasets. *IEEE Open Journal of Engineering in Medicine and Biology* **2023**, pp. 1–11. https://doi.org/10.1109/OJEMB.2023.3243190.
45. Diaz, G.I.; Fokoue-Nkoutche, A.; Nannicini, G.; Samulowitz, H. An effective algorithm for hyperparameter optimization of neural networks. *IBM Journal of Research and Development* **2017**, *61*, 9:1–9:11. https://doi.org/10.1147/JRD.2017.2709578.
46. Yang, L.; Shami, A. On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing* **2020**, *415*, 295–316. https://doi.org/https://doi.org/10.1016/j.neucom.2020.07.061.
47. Liaw, R.; Liang, E.; Nishihara, R.; Moritz, P.; Gonzalez, J.E.; Stoica, I. Tune: A Research Platform for Distributed Model Selection and Training. *arXiv:1807.05118 [cs, stat]* **2018**.
48. Li, L.; Jamieson, K.; DeSalvo, G.; Rostamizadeh, A.; Talwalkar, A. Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization. *Journal of Machine Learning Research* **2018**, *18*, 1–52.

49. Li, L.; Jamieson, K.; Rostamizadeh, A.; Gonina, E.; Hardt, M.; Recht, B.; Talwalkar, A. A System for Massively Parallel Hyperparameter Tuning. *arXiv:1810.05934* **2018**. https://doi.org/10.48550/ARXIV.1810.05934.
50. Jaderberg, M.; Dalibard, V.; Osindero, S.; Czarnecki, W.M.; Donahue, J.; Razavi, A.; Vinyals, O.; Green, T.; Dunning, I.; Simonyan, K.; et al. Population Based Training of Neural Networks, 2017. https://doi.org/10.48550/ARXIV.1711.09846.

References

[1] Giacomo Bellani, John G Laffey, Tài Pham, Eddy Fan, Laurent Brochard, et al. "Epidemiology, patterns of care, and mortality for patients with acute respiratory distress syndrome in intensive care units in 50 countries." In: *Jama* 315.8 (2016), pp. 788–800.

[2] Manuel Iregui, Suzanne Ward, Glenda Sherman, Victoria J Fraser, and Marin H Kollef. "Clinical importance of delays in the initiation of appropriate antibiotic treatment for ventilator-associated pneumonia." In: *Chest* 122.1 (2002), pp. 262–268.

[3] Paula Ramirez, Cristina Lopez-Ferraz, Monica Gordon, Alexandra Gimeno, Esther Villarreal, et al. "From starting mechanical ventilation to ventilator-associated pneumonia, choosing the right moment to start antibiotic treatment." In: *Critical Care* 20.1 (2016), pp. 1–7.

[4] David G Ashbaugh, D Boyd Bigelow, and Bernard E Levine. "Acute Respiratory Distress in Adults." en. In: *The Lancet* 290.7511 (Aug. 1967), pp. 319–323. DOI: 10.1016/S0140-6736(67)90168-7.

[5] Marco Confalonieri, Francesco Salton, and Francesco Fabiano. "Acute respiratory distress syndrome." In: *European Respiratory Review* 26.144 (June 2017), p. 160116. DOI: 10.1183/16000617.0116-2016.

[6] Sidney Le, Emily Pellegrini, Abigail Green-Saxena, Charlotte Summers, Jana Hoffman, et al. "Supervised machine learning for the early prediction of acute respiratory distress syndrome (ARDS)." In: *Journal of Critical Care* 60 (Dec. 2020), pp. 96–102. ISSN: 0883-9441. DOI: 10.1016/j.jcrc.2020.07.019.

[7] The ARDS Definition Task Force. "Acute Respiratory Distress Syndrome: The Berlin Definition." en. In: *JAMA* 307.23 (June 2012). ISSN: 0098-7484. DOI: 10.1001/jama.2012.5669.

[8] Nir Menachemi and Taleah H. Collum. "Benefits and drawbacks of electronic health record systems." en. In: *Risk Management and Healthcare Policy* (May 2011), p. 47. ISSN: 1179-1594. DOI: 10.2147/RMHP.S12985.

[9] Martin R. Cowie, Juuso I. Blomster, Lesley H. Curtis, Sylvie Duclaux, Ian Ford, et al. "Electronic health records to facilitate clinical research." In: *Clinical Research in Cardiology* 106.1 (Jan. 2017), pp. 1–9. ISSN: 1861-0692. DOI: 10.1007/s00392-016-1025-6.

[10] A. Hoerbst and E. Ammenwerth. "Electronic Health Records, A Systematic Review on Quality Requirements." EN. In: *Methods of Information in Medicine* 49.04 (Jan. 2018), pp. 320–336. ISSN: 0026-1270. DOI: 10.3414/ME10-01-0038.

[11] Geoffrey French, Mary Hulse, Debbie Nguyen, Katharine Sobotka, Kaitlyn Webster, et al. "Impact of Hospital Strain on Excess Deaths During the COVID-19 Pandemic — United States, July 2020–July 2021." en. In: *Morbidity and Mortality Weekly Report* 70.46 (Nov. 2021), pp. 1613–1616. DOI: 10.15585/mmwr.mm7046a5.

[12] Marco Marani, Gabriel G. Katul, William K. Pan, and Anthony J. Parolari. "Intensity and frequency of extreme novel epidemics." en. In: *Proceedings of the National Academy of Sciences* 118.35 (Aug. 2021), e2105482118. ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.2105482118.

[13] Nada M. Elshennawy, Dina M. Ibrahim, Amany M. Sarhan, and Mohamed Arafa. "Deep-Risk: Deep Learning-Based Mortality Risk Predictive Models for COVID-19." en. In: *Diagnostics* 12.8 (July 2022), p. 1847. ISSN: 2075-4418. DOI: 10.3390/diagnostics12081847.

[14] Linda Wang, Zhong Qiu Lin, and Alexander Wong. "COVID-Net: a tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images." en. In: *Scientific Reports* 10.1 (Dec. 2020), p. 19549. ISSN: 2045-2322. DOI: 10.1038/s41598-020-76550-z.

[15] Yao Song, Jun Liu, Xinghua Liu, and Jinshan Tang. "COVID-19 Infection Segmentation and Severity Assessment Using a Self-Supervised Learning Approach." en. In: *Diagnostics* 12.8 (July 2022), p. 1805. ISSN: 2075-4418. DOI: 10.3390/diagnostics12081805.

[16] Chin Poo Lee and Kian Ming Lim. "COVID-19 Diagnosis on Chest Radiographs with Enhanced Deep Neural Networks." en. In: *Diagnostics* 12.8 (July 2022), p. 1828. ISSN: 2075-4418. DOI: 10.3390/diagnostics12081828.

[17] Marina Lugarà, Stefania Tamburrini, Maria Gabriella Coppola, Gabriella Oliva, Valeria Fiorini, et al. "The Role of Lung Ultrasound in SARS-CoV-19 Pneumonia Management." en. In: *Diagnostics* 12.8 (July 2022), p. 1856. ISSN: 2075-4418. DOI: 10.3390/diagnostics12081856.

[18] Alexander Selvikvåg Lundervold and Arvid Lundervold. "An overview of deep learning in medical imaging focusing on MRI." en. In: *Zeitschrift für Medizinische Physik* 29.2 (May 2019), pp. 102–127. ISSN: 09393889. DOI: 10.1016/j.zemedi.2018.11.002.

[19] Duncan Shillan, Jonathan A. C. Sterne, Alan Champneys, and Ben Gibbison. "Use of machine learning to analyse routinely collected intensive care unit data: a systematic review." en. In: *Critical Care* 23.1 (Dec. 2019), p. 284. ISSN: 1364-8535. DOI: 10.1186/s13054-019-2564-9.

[20] Hanqing Sun, Zheng Liu, Guizhi Wang, Weimin Lian, and Jun Ma. "Intelligent Analysis of Medical Big Data Based on Deep Learning." en. In: *IEEE Access* 7 (2019), pp. 142022–142037. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2019.2942937.

[21] J G Hardman, N M Bedforth, A B Ahmed, R P Mahajan, and A R Aitkenhead. "A physiology simulator: validation of its respiratory components and its ability to predict the patient's response to changes in mechanical ventilation." en. In: *British Journal of Anaesthesia* 81.3 (Sept. 1998), pp. 327–332. ISSN: 00070912. DOI: 10.1093/bja/81.3.327.

[22] A. Das, Z. Gao, P. P. Menon, J. G. Hardman, and D. G. Bates. "A systems engineering approach to validation of a pulmonary physiology simulator for clinical applications." en. In: *Journal of The Royal Society Interface* 8.54 (Jan. 2011), pp. 44–55. ISSN: 1742-5689, 1742-5662. DOI: 10.1098/rsif.2010.0224.

[23] M. Laviola, A. Das, M. Chikhani, D.G. Bates, and J.G. Hardman. "Computer simulation clarifies mechanisms of carbon dioxide clearance during apnoea." In: *British Journal of Anaesthesia* 122.3 (2019), pp. 395–401. ISSN: 0007-0912. DOI: https://doi.org/10.1016/j.bja.2018.11.012. URL: https://www.sciencedirect.com/science/article/pii/S0007091218313011.

[24] Sina Saffaran, Anup Das, John Laffey, Jonathan Hardman, Nadir Yehya, and Declan Bates. "Utility of Driving Pressure and Mechanical Power to Guide Protective Ventilator Settings in Two Cohorts of Adult and Pediatric Patients With Acute Respiratory Distress Syndrome: A Computational Investigation." In: *Critical Care Medicine* 48 (Apr. 2020), p. 8. DOI: 10.1097/CCM.0000000000004372.

[25] Esteka Suarez, Norbert Eickert, and Thomas Lippert. "Modular Supercomputing architecture: from idea to production." In: *Contemporary High Performance Computing: From Petascale toward Exascale*. Ed. by Jeffrey Vetter. 1st ed. Vol. 3. FL, USA: CRC Press, 2019, pp. 223–251. ISBN: 978-1-351-03686-3.

[26] Stefan Kesselheim, Andreas Herten, Kai Krajsek, Jan Ebert, Jenia Jitsev, et al. "JUWELS Booster – A Supercomputer for Large-Scale AI Research." In: *High Performance Computing*. Ed. by Heike Jagode, Hartwig Anzt, Hatem Ltaief, and Piotr Luszczek. Cham: Springer International Publishing, 2021, pp. 453–468. ISBN: 978-3-030-90539-2.

[27] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, et al. "Highly accurate protein structure prediction with AlphaFold." en. In: *Nature* 596.7873 (Aug. 2021), pp. 583–589. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/s41586-021-03819-2.

[28] Minkyung Baek, Frank DiMaio, Ivan Anishchenko, Justas Dauparas, Sergey Ovchinnikov, et al. "Accurate prediction of protein structures and interactions using a three-track neural network." en. In: *Science* 373.6557 (Aug. 2021), pp. 871–876. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.abj8754.

[29] Alfred Winter, Sebastian Stäubert, Danny Ammon, Stephan Aiche, Oya Beyan, et al. "Smart Medical Information Technology for Healthcare (SMITH): Data Integration based on Interoperability Standards." en. In: *Methods of Information in Medicine* 57 (July 2018), e92–e105. ISSN: 0026-1270, 2511-705X. DOI: 10.3414/ME18-02-0004.

[30] Gernot Marx, Johannes Bickenbach, Sebastian Johannes Fritsch, Julian Benedict Kunze, Oliver Maassen, et al. "Algorithmic surveillance of ICU patients with acute respiratory distress syndrome (ASIC): protocol for a multicentre stepped-wedge cluster randomised quality improvement strategy." In: *BMJ Open* 11.4 (2021). ISSN: 2044-6055. DOI: 10.1136/bmjopen-2020-045589.

[31] Tom M. Mitchell. *Machine Learning*. McGraw-Hill Education, 1997. ISBN: 9780071154673.

[32] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. en. Cambridge: Cambridge University Press, 2014. ISBN: 978-1-107-29801-9. DOI: 10.1017/CBO9781107298019. URL: http://ebooks.cambridge.org/ref/id/CBO9781107298019.

[33] Pat Langley. "The changing science of machine learning." In: *Machine Learning* 82.3 (Mar. 2011), pp. 275–279. ISSN: 1573-0565. DOI: 10.1007/s10994-011-5242-y.

[34] Christopher M. Bishop. *Pattern recognition and machine learning*. eng. (Corrected at 8. printing). Information science and statistics. New York, NY: Springer, 2009. ISBN: 9780387310732.

[35] Stuart Russel and Peter Norvig. *Artificial Intelligence - A Modern Approach*. 2nd ed. Prentice Hall Series in Artificial Intelligence. New Jersey, USA: Pearson Education Inc., 2003. ISBN: 0-13-080302-2.

[36] G. Hinton and T.J. Sejnowski. *Unsupervised Learning: Foundations of Neural Computation*. Computational Neuroscience Series. MIT Press, 1999. ISBN: 9780262581684. URL: https://books.google.de/books?id=yj04Y0lje4cC.

[37] Ian Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. en. Elsevier, 2011. ISBN: 978-0-12-374856-0. DOI: 10.1016/C2009-0-19715-5. URL: https://linkinghub.elsevier.com/retrieve/pii/C20090197155.

[38] Martijn van Otterlo and Marco Wiering. "Reinforcement Learning and Markov Decision Processes." en. In: *Reinforcement Learning*. Ed. by Marco Wiering and Martijn van Otterlo. Vol. 12. Adaptation, Learning, and Optimization. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 3–42. ISBN: 978-3-642-27644-6. DOI: 10.1007/978-3-642-27645-3_1. URL: http://link.springer.com/10.1007/978-3-642-27645-3_1.

[39] Pang-Ning Tan, Michael Steinbach, and Vipin Kuman. *Introduction to Data Mining*. en. Pearson, 2006. ISBN: 9780321321367.

[40] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, et al. "Mastering the game of Go with deep neural networks and tree search." In: *Nature* 529.7587 (Jan. 2016), pp. 484–489. ISSN: 1476-4687. DOI: 10.1038/nature16961.

[41] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, et al. *Language Models are Few-Shot Learners*. 2020. DOI: 10.48550/ARXIV.2005.14165. URL: https://arxiv.org/abs/2005.14165.

[42] Georg Hager and Gerhard Wellein. *Introduction to High Performance Computing for Scientists and Engineers*. 1st. USA: CRC Press, Inc., 2010. ISBN: 143981192X. DOI: 10.1201/EBK1439811924.

[43] Rocco Sedona, Gabriele Cavallaro, Jenia Jitsev, Alexandre Strube, Morris Riedel, and Jón Benediktsson. "Remote Sensing Big Data Classification with High Performance Distributed Deep Learning." en. In: *Remote Sensing* 11.24 (Dec. 2019), p. 3056. ISSN: 2072-4292. DOI: 10.3390/rs11243056.

[44] *Exploiting GPU-Based HPC Architectures to Accelerate an Unsteady CFD Solver for Turbomachinery Applications*. Vol. Volume 10C: Turbomachinery — Design Methods and CFD Modeling for Turbomachinery; Ducts, Noise, and Component Interactions. Turbo Expo: Power for Land, Sea, and Air. June 2022. DOI: 10.1115/GT2022-82569.

[45] Ian Bird. "Computing for the Large Hadron Collider." In: *Annual Review of Nuclear and Particle Science* 61.1 (2011), pp. 99–118. DOI: 10.1146/annurev-nucl-102010-130059.

[46] Michael J. Flynn. "Some Computer Organizations and Their Effectiveness." In: *IEEE Transactions on Computers* C-21.9 (1972), pp. 948–960. DOI: 10.1109/TC.1972.5009071.

[47] M. Radosavljevic, G. Dewey, D. Basu, J. Boardman, B. Chu-Kung, et al. "Electrostatics improvement in 3-D tri-gate over ultra-thin body planar InGaAs quantum well field effect transistors with high-K gate dielectric and scaled gate-to-drain/gate-to-source separation." In: *2011 International Electron Devices Meeting*. 2011, pp. 33.1.1–33.1.4. DOI: 10.1109/IEDM.2011.6131661.

[48] Sasikanth Manipatruni, Dmitri E. Nikonov, and Ian A. Young. "Material Targets for Scaling All-Spin Logic." In: *Phys. Rev. Appl.* 5 (1 Jan. 2016), p. 014002. DOI: 10.1103/PhysRevApplied.5.014002.

[49] Frank Schwierz. "Graphene transistors — A new contender for future electronics." In: *2010 10th IEEE International Conference on Solid-State and Integrated Circuit Technology*. 2010, pp. 1202–1205. DOI: 10.1109/ICSICT.2010.5667602.

[50] Kathryn W. Hendrickson, Ithan D. Peltan, and Samuel M. Brown. "The Epidemiology of Acute Respiratory Distress Syndrome Before and After Coronavirus Disease 2019." In: *Critical Care Clinics* 37.4 (Oct. 2021), pp. 703–716. ISSN: 0749-0704. DOI: 10.1016/j.ccc.2021.05.001.

[51] Julian Kunze, Sebastian Fritsch, Arne Peine, Oliver Maaßen, Gernot Marx, and Johannes Bickenbach. "Management of ARDS: from ventilation strategies to intelligent technical support—connecting the dots." In: *Trends in Anaesthesia and Critical Care* 34 (2020), pp. 50–58.

[52] Anup Das, Oana Cole, Marc Chikhani, Wenfei Wang, Tayyba Ali, et al. "Evaluation of lung recruitment maneuvers in acute respiratory distress syndrome using computer simulation." en. In: *Critical Care* 19.1 (Dec. 2015), p. 8. ISSN: 1364-8535. DOI: 10.1186/s13054-014-0723-6.

[53] Eddy Fan, Daniel Brodie, and Arthur S Slutsky. "Acute respiratory distress syndrome: advances in diagnosis and treatment." In: *Jama* 319.7 (2018), pp. 698–710.

[54] Jesús Villar, Jesús Blanco, José Manuel Añón, Antonio Santos-Bouza, Lluís Blanch, et al. "The ALIEN study: incidence and outcome of acute respiratory distress syndrome in the era of lung protective ventilation." en. In: *Intensive Care Medicine* 37.12 (Dec. 2011), pp. 1932–1941. ISSN: 0342-4642, 1432-1238. DOI: 10.1007/s00134-011-2380-4.

[55] Abimbola O Ajibowo, Olasunkanmi A Kolawole, Haleema Sadia, Oyovwike S Amedu, Hassan A Chaudhry, et al. "A Comprehensive Review of the Management of Acute Respiratory Distress Syndrome." In: *Cureus* 14.10 (2022).

[56] Majid Afshar, Cara Joyce, Anthony Oakey, Perry Formanek, Philip Yang, et al. "A Computable Phenotype for Acute Respiratory Distress Syndrome Using Natural Language Processing and Machine Learning." en. In: *AMIA Annual Symposium Proceedings 2018* (Dec. 2018), pp. 157–165.

[57] Michael Worobey, Joshua I. Levy, Lorena Malpica Serrano, Alexander Crits-Christoph, Jonathan E. Pekar, et al. "The Huanan Seafood Wholesale Market in Wuhan was the early epicenter of the COVID-19 pandemic." In: *Science* 377.6609 (2022), pp. 951–959. DOI: 10.1126/science.abp8715.

[58] Alexander E. Gorbalenya, Susan C. Baker, Ralph S. Baric, Raoul J. de Groot, Christian Drosten, et al. "The species Severe acute respiratory syndrome-related coronavirus: classifying 2019-nCoV and naming it SARS-CoV-2." In: *Nature Microbiology* 5.4 (Apr. 2020), pp. 536–544. ISSN: 2058-5276. DOI: 10.1038/s41564-020-0695-z.

[59] Alireza Tahamtan and Abdollah Ardebili. "Real-time RT-PCR in COVID-19 detection: issues affecting the results." en. In: *Expert Review of Molecular Diagnostics* 20.5 (May 2020), pp. 453–454. ISSN: 1473-7159, 1744-8352. DOI: 10.1080/14737159.2020.1757437.

[60] Manoucher Teymouri, Samaneh Mollazadeh, Hamed Mortazavi, Zari Naderi Ghale-noie, Vahideh Keyvani, et al. "Recent advances and challenges of RT-PCR tests for the diagnosis of COVID-19." en. In: *Pathology - Research and Practice* 221 (May 2021), p. 153443. ISSN: 03440338. DOI: 10.1016/j.prp.2021.153443.

[61] Leonid Roshkovan, Neil Chatterjee, Maya Galperin-Aizenberg, Naraindar Gupta, Rosita Shah, et al. "The Role of Imaging in the Management of Suspected or Known COVID-19 Pneumonia. A Multidisciplinary Perspective." In: *Annals of the American Thoracic Society* 17.11 (2020), pp. 1358–1365.

[62] Tao Ai, Zhenlu Yang, Hongyan Hou, Chenao Zhan, Chong Chen, et al. "Correlation of chest CT and RT-PCR testing in coronavirus disease 2019 (COVID-19) in China: a report of 1014 cases." In: *Radiology* (2020).

[63] Wenfei Wang, Anup Das, Tayyba Ali, Oanna Cole, Marc Chikhani, et al. "Can computer simulators accurately represent the pathophysiology of individual COPD patients?" In: *Intensive Care Medicine Experimental* 2 (Sept. 2014). DOI: 10.1186/s40635-014-0023-0.

[64] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep Residual Learning for Image Recognition." en. In: *arXiv:1512.03385 [cs]* (Dec. 2015). arXiv: 1512.03385.

[65] Rüdiger Wirth and Jochen Hipp. "CRISP-DM: Towards a Standard Process Model for Data Mining." en. In: *Proceedings of the Fourth International Conference on the Practical Application of Knowledge Discovery and Data Mining* (Apr. 2000), pp. 29–39.

[66] Fan Zhang, Melissa Petersen, Leigh Johnson, James Hall, and Sid E. O'Bryant. "Hyperparameter Tuning with High Performance Computing Machine Learning for Imbalanced Alzheimer's Disease Data." en. In: *Applied Sciences* 12.13 (July 2022), p. 6670. ISSN: 2076-3417. DOI: 10.3390/app12136670.

[67] Hania H. Farag, Lamiaa A. A. Said, Mohamed R. M. Rizk, and Magdy Abd ElAzim Ahmed. "Hyperparameters optimization for ResNet and Xception in the purpose of diagnosing COVID-19." In: *Journal of Intelligent & Fuzzy Systems* 41.2 (2021), pp. 3555–3571. ISSN: 1875-8967. DOI: 10.3233/JIFS-210925.

[68] Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. "Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization." In: *Journal of Machine Learning Research* 18.185 (2018), pp. 1–52.

[69] Liam Li, Kevin Jamieson, Afshin Rostamizadeh, Ekaterina Gonina, Moritz Hardt, et al. "A System for Massively Parallel Hyperparameter Tuning." In: *arXiv:1810.05934* (2018). DOI: 10.48550/ARXIV.1810.05934.

[70] Max Jaderberg, Valentin Dalibard, Simon Osindero, Wojciech M. Czarnecki, Jeff Donahue, et al. *Population Based Training of Neural Networks*. 2017. DOI: 10.48550/ARXIV.1711.09846.

[71] Konstantin Sharafutdinov, Sebastian Johannes Fritsch, Mina Iravani, Pejman Farhadi Ghalati, Sina Saffaran, et al. "Computational simulation of virtual patients reduces dataset bias and improves machine learning-based detection of ARDS from noisy heterogeneous ICU datasets." In: *IEEE Open Journal of Engineering in Medicine and Biology* (2023), pp. 1–11. DOI: 10.1109/OJEMB.2023.3243190.