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

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

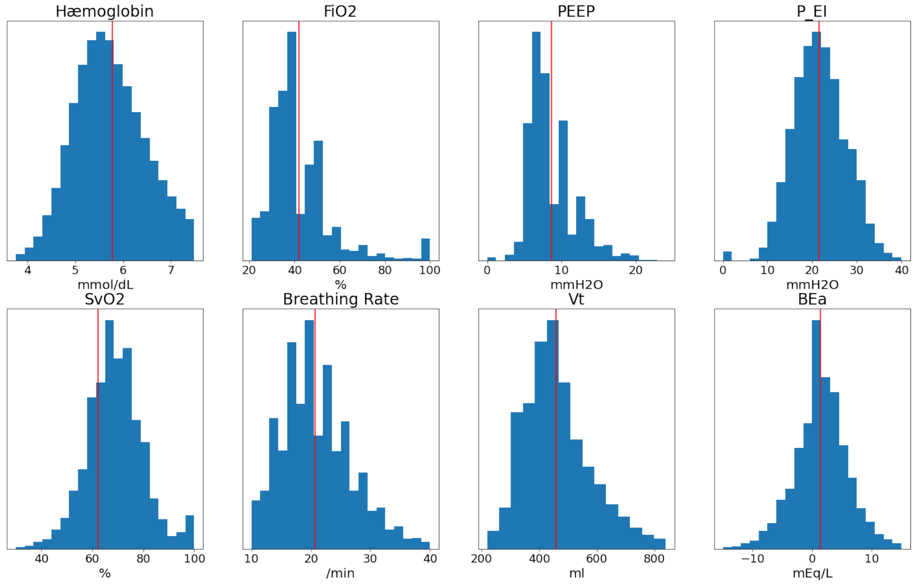

Figure 1. Parameter distribution over predetermined physiological ranges (mean in red). Parameters: **FiO2** - Fraction of inspired O_2, **PEEP** - Peak End-Expiratory Pressure, **Vt** - Tidal Volume, **P_EI** - End-Inspiratory Pressure, **SvO2** - Venous O_2 Saturation, **BEa** - Arterial Base Excess.

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

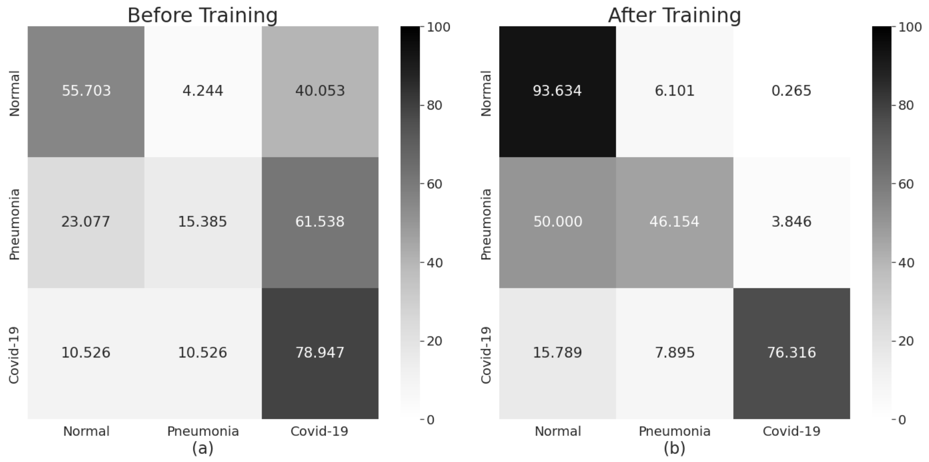

Figure 2. Prediction performance on a test set of the EHL dataset (a) before and (b) after training.

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

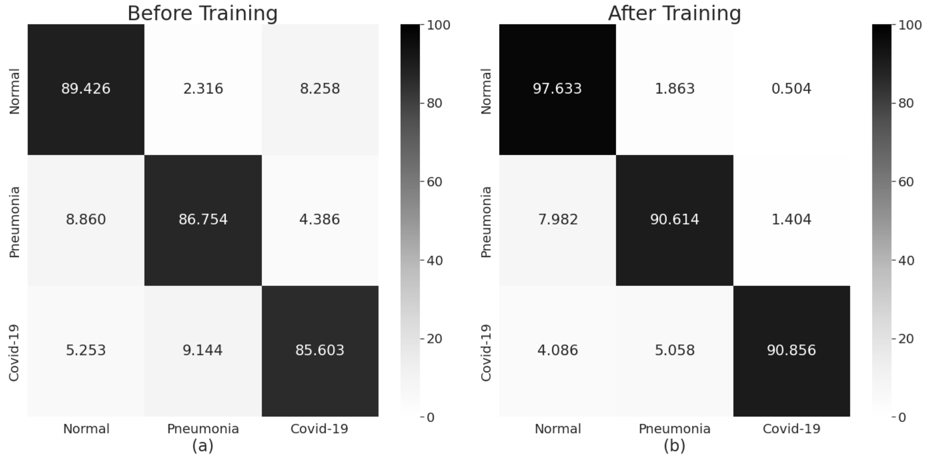

Figure 3. Prediction performance on a test set of the Fusion dataset (a) before and (b) after training.

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 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.