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

---

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.
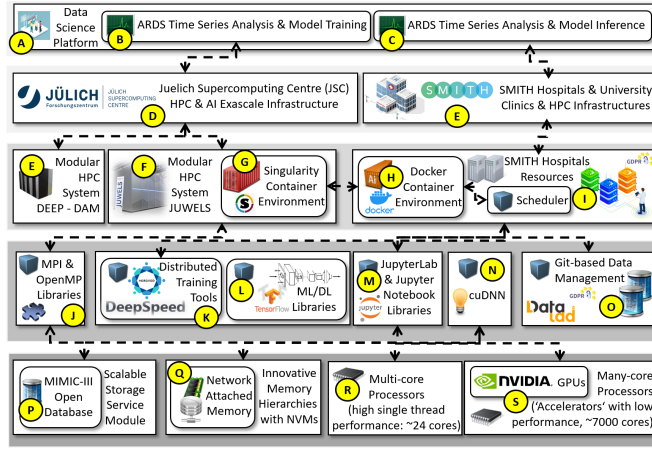
---

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

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
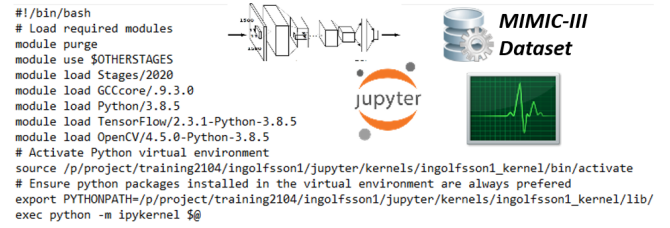

Figure 2. Covid-19 Chest X-Ray and ARDS Analysis Platform Environment.

infrastructure (see Fig. 1 D), while the latter is performing inference on patients on Hospitals moderate HPC infrastructure (see Fig. 1 E).

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/

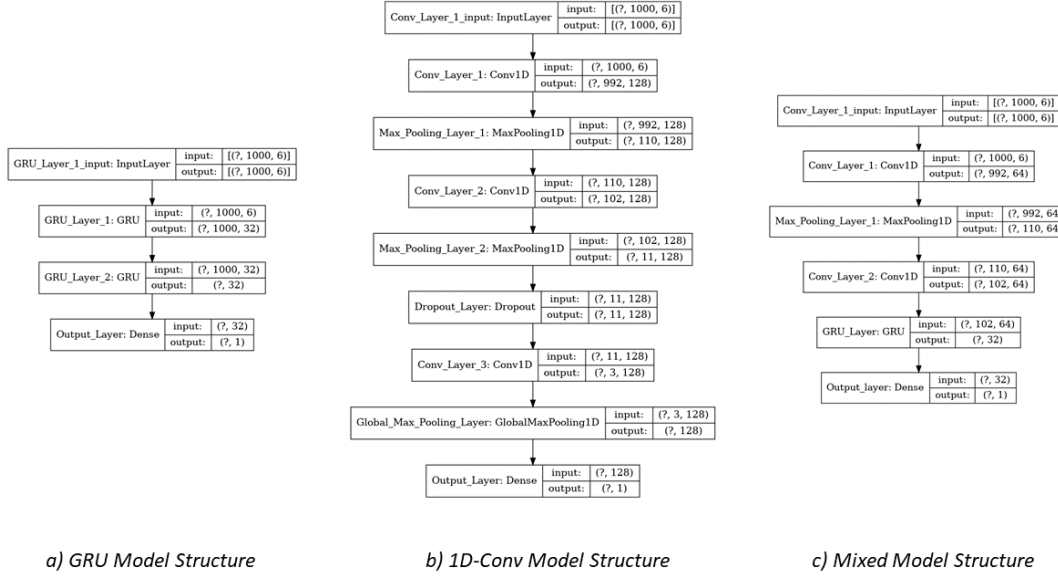a) GRU Model Structure b) 1D-Conv Model Structure c) Mixed Model Structure

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

a) GRU Model Training   b) 1D-Conv Model Training   c) Mixed Model Training
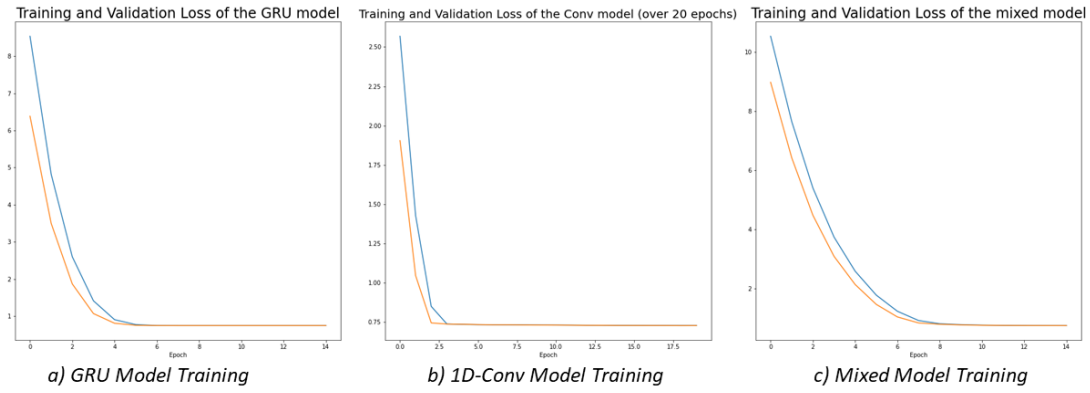
Figure 4. Performance comparison of the different model structures.

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 pre-processing 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

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.