ParCFD2022
33^{rd} International Conference on Parallel Computational Fluid Dynamics
May 25-27 2022, Alba Italy

LAGRANGIAN PARTICLE TRACKING DATA OF A STRAINING TURBULENT FLOW ASSESSED USING MACHINE LEARNING AND PARALLEL COMPUTING

R. HASSANIAN[*], M. RIEDEL[*,†], A. HELGADOTTIR[*], P. S. COSTA[*] AND L. BOUHLALI[‡]

[*]University of Iceland
The Faculty of Industrial Engineering, Mechanical Engineering and Computer Science
Saemundargata 2, 102 Reykjavik , Iceland
e-mail: seh@hi.is, web page: https://english.hi.is/staff/seh
e-mail: morris@hi.is, web page: https://english.hi.is/staff/morris
e-mail: asdishe@hi.is, web page: https://english.hi.is/staff/asdishe
e-mail: pcosta@hi.is, web page: https://english.hi.is/staff/pcosta

[†]Juelich Supercomputing Centre
52428 Jülich, Germany
e-mail: m.riedel@fz-juelich.de - Web page: http://www.morrisriedel.de

[‡]Reykjavik University
Department of Engineering, School of Technology
Menntavegi 1, 102 Reykjavik, Iceland
e-mail: lahcen09@ru.is

Key words: Lagrangian particle tracking, Parallel computing, Machine learning, Turbulent flow

Abstract. This study aimed to employ artificial intelligence capability and computing scalability to predict the velocity field of the straining turbulence flow. Rotating impellers in a box have generated the turbulence, subsequently subjected to an axisymmetric straining motion, with mean nominal strain rates of $4s^{-1}$. Tracer particles are seeded in the flow, and their dynamics are investigated using high-speed Lagrangian Particle Tracking at 10,000 frames per second. The particle displacement, time, and velocities can be extracted using this technique. Particle displacement and time are used as input observables, and the velocity is employed as a response output. The experiment extracted data have been divided into training and test data to validate the models. Support vector polynomial regression (SVR) and Linear regression were employed to see how extrapolation for the velocity field can be extracted. These models can be done with low computing time. On the other hand, to create a dynamic prediction, Gated Recurrent Unit (GRU) is applied with a high-performance computing application. The results show that GRU presents satisfactory forecasting for the turbulence velocity field and the computing scale performed on the JUWELS and DEEP-EST and reported. GPUs have a significant effect on computing time. This work presents the capability of the GRU model for time series data related to turbulence flow prediction.

1 INTRODUCTION

Turbulence flow is observed in many industrial and natural phenomena [1][2]. Smoke from the chimney, airflow around the car, airplane wing, wind speed, downstream flow in wind farms, and water flow in rivers and waterfalls are examples of the everyday turbulence around us [1][3][2]. The study of turbulence flow is essential because, firstly, to understand the phenomena in many applications. Secondly, it is necessary to know the turbulence flow in the next step and prediction in some cases. For example, the wind speed drives the wind turbine, and the output is electrical power. The generated force is proportional to wind speed, and the power grid needs to have short-term and long-term predictions [4][5][6][7] according to the power demand in the grid. This makes wind speed forecasting important, and, in this case, there is sometimes high-speed wind as an extreme event called gusts which could have an impact on the power grid and electricity production [8][9].

In fluid dynamics, many experiments have recorded the velocity fields. On the other hand, from the Computational fluid dynamics view, Direct Numerical Solution (DNS), Large-eddy Simulation (LES), and Reynolds-Averaged Navier Stokes (RANS) are capable methods to solve turbulence flow problems and extract the velocity field and other properties [10][11][12]. LES presents reliable result, and DNS dictate the exact solution, but these methods are costly, and it is necessary to run-on High-Performance Computing (HPC). The development in the HPC is rapid, but still, we are far from solving problems of enormous size [11][12]. This limitation tends us to use other methods to create interpolation functions from available data or predict the turbulent flow in the short or long term. Artificial Intelligence (AI) involves methods for interpolation or prediction [13]. The machine learning support vector machine (SVM) and linear regression can create a function and hypothesis with available data and build an interpolation function.

In some cases, we could use the created function for extrapolation. However, Deep learning (DL) networks can forecast time series data via a recurrent neural network (RNN). Long short-term memory (LSTM) is a successful approach for prediction, and many studies employed this method for time series, and there are combined hybrid methods with LSTM. LSTM is a comprehensive method but costly for large time series, and it consumes extensive computing time because of its architecture. Gated recurrent unit (GRU) is a next-generation LSTM that has a few distinctions from LSTM, and it is reported faster[14].

Turbulence flow is a high dimensional phenomenon, and all previous studies used DL ability via many layers to find the main hidden feature among the high dimensionality. Then the model has been created for the prediction via extracted main hidden features [11][12]. This study uses time-series data, including 2D velocity field and position corresponding to the specified time. This data was extracted from the experiment conducted at the laboratory. This work aims to present spatial and temporal features of the turbulent flow dataset to create a GRU model that gives a satisfactory prediction without encountering high dimensionality of the turbulence flow. The computing time for the GRU model is scaled, and the report showed GPUs could improve the computing time effectively for this model. The results from SVM and linear regression models are mentioned and compared.

2 DATA SET FROM EXPERIMENT

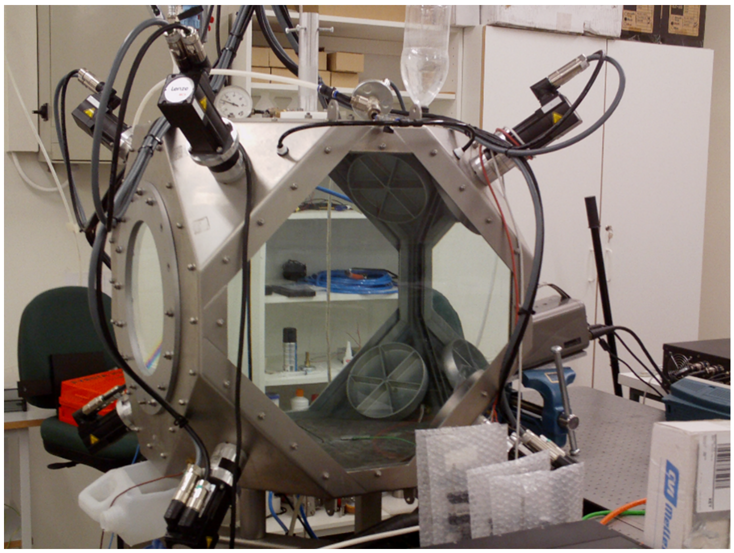


Figure 1: A photo of the turbulence box taken at the Laboratory for Fundamental Turbulence Research (LFTR) at Reykjavik University [15][16].

The turbulent flow is generated with the action of impeller rotors in the corners of our box turbulence facility (Figure 1). The turbulent flow is then strained in the vertical direction by the motion of flat circular plates. The fluid is seeded with buoyant (tracer) particles with median diameters 8-10 μm. The specific gravity for tracers was 1.1 gr/cm^3 (hollow glass). The flow field properties are obtained and studied through the Lagrangian Particle Tracking (LPT) method. Equation (1) describes the mean flow field in the facility:

$$\langle \mathbf{U} \rangle = (Sx, -2Sy, Sz), \tag{1}$$

Where $\langle \mathbf{U} \rangle$ is the mean velocity –2S is the primary strain rate in the Y-dir, and S is the strain rate for the other two directions. The straining flow cases were created in the experiment with mean strain rates, $2S = 4s^{-1}$. Equation (1) is based on the laminar flow; however, we know that velocity fluctuates in the turbulence flow[15]. The Lagrangian particle tracking (LPT) measurements were carried out in 2D, so the velocity components in x and y direction is obtained. The turbulence in this project depends on the size of the particles, on the rotation speed of the impeller in the tank and is applied at 1000 rpm. The mean strain rate $4s^{-1}$ was created using two circular plates - the plates driver motor moves according to an exponential velocity profile; the same as a particle would move by if it were moving along the y axis towards the center of the coordinate system. The detection system was set at 10 kHz (10000 fps) for well-resolved particle velocity and acceleration statistics. This very high temporal resolution (0.1-0.2 ms) is considerably smaller than the Kolmogorov time τ_η (35 - 99 ms) of the smallest eddies present in the flow; therefore, the properties of the dissipation range in the flow are solved. In general, there were 80000 frames for buoyant particles per experiment, and an in-house LPT code. The particle velocity and displacement data are extracted and are used to train the model. These data are divided between test data and training data.

3 MACHINE LEARNING MODELS

3.1 GATED RECURRENT UNIT (GRU)

The data series in this work involves velocity and position corresponding to specifies time. We aim to employ GRU to create a model with this time series to predict the next time step in a parallel computing setting. The empirical data is divided into 80% training and 20% test data. This modeling relies on the spatial and temporal features of the turbulent flow. We have not used a method to find specific features, and the practical elements are positioned during the time extracted via LPT. The advantage of this method is that we conduct the model for every velocity component separately and perform it for 2D and 3D velocity components depending on the data availability. Figure 2 represents the prediction result via GRU evaluated with the actual data.

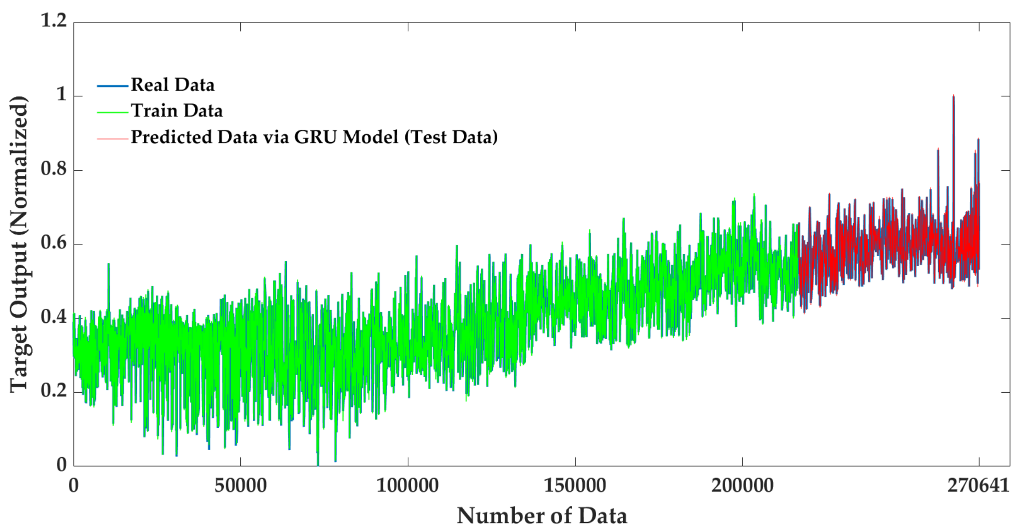
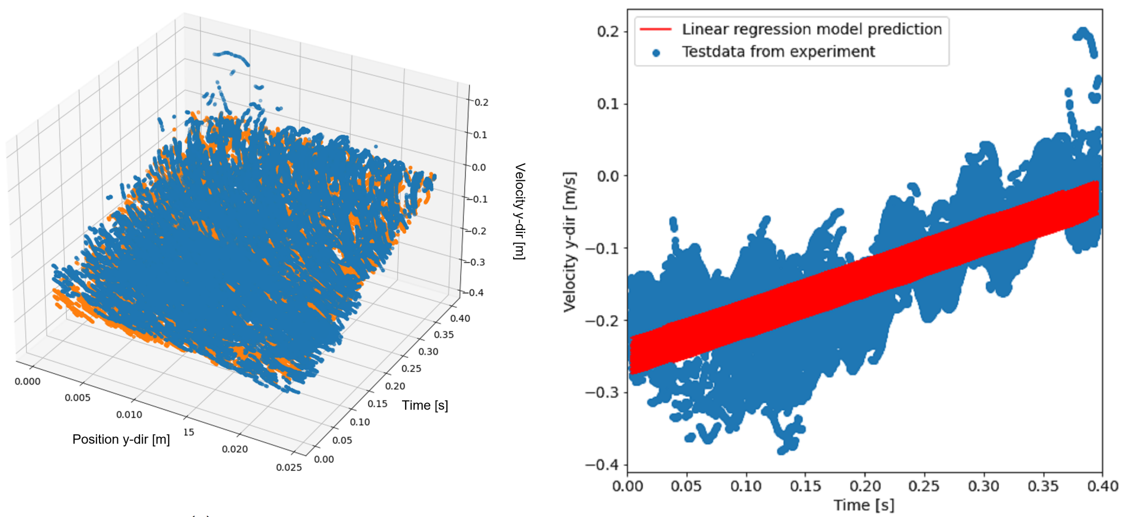


Figure 2: GRU model for turbulence flow velocity in y direction forecasts the next time step and assessed with actual data

3.2 REGRESSION AND SUPPORT VECTOR MACHINE (SVM)

Support vector regression polynomial (SVR) and linear regression were employed to create the model in a parallel computing setting. Figure 3 shows a prediction from linear regression for the velocity during the straining time, and the forecast from SVR is illustrated in Figure 4. It is well-known regression and SVR consuming low computing and very fast than deep learning methods such as LSTM and GRU. But the result dictates this method cannot create a prediction model for the turbulence flow velocity field with position and time. It may be necessary to use more features as input to provide a reliable model.



(a) (b)

Figure 3: A presentation is from the prediction model for particle velocity in y-dir were tracked during the straining flow time. (a) 3D view, (b) Y component velocity vs. time. Compared to test data from the experiment; The blue circle is experiment data. The red line has been predicted via the linear regression model.

4 PARALLEL COMPUTING

In this work to implement the GRU model training, two powerful parallel computing machines were employed to assess the scale computing. The first one is JUWELS Booster Module and the other one DEEP-EST, Data Analytics Module (DAM), both at the Jülich Supercomputing Centre (JSC) in Germany. JUWELS represents the fastest EU supercomputer with 122,768 CPU cores only in its cluster module. While JUWELS and multi-core processors offer tremendous performance, the particular challenge to exploiting this data analysis performance for ML is that those systems require specific parallel and scalable techniques. In other words, using JUWELS cluster module CPUs with Remote Sensing (RS) data effectively requires parallel algorithm implementations as opposed to using plain scikit-learn15, R16, or different serial algorithms [17][18]. The computing information in this study was carried out based on these parallel machines as below [18][19]:

• JUWELS Booster Module:

- 1 node involves 2× AMD EPYC Rome 7402 CPU, 2× 24 cores, 2.8 GHz
- 936 compute nodes
- 3,744 GPUs

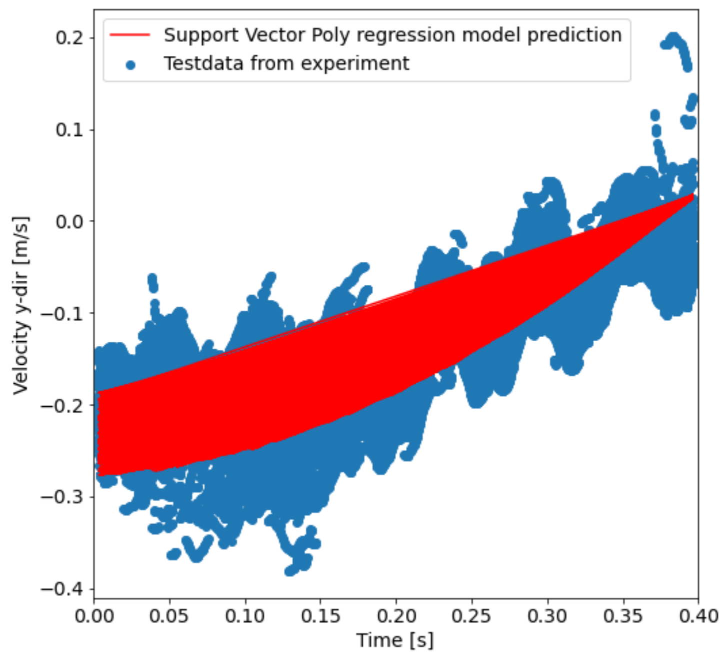


Figure 4: An illustration is from the prediction models for flow velocity in the y direction that were extracted from Lagrangian particle tracking (LPT) during the straining flow time. Compared to test data from the experiment; The blue circle is experiment data. The red-line has been predicted via the support vector polynomial regression (SVR) model.

• DEEP-EST, Data Analytics Module (DAM):
- 1 rack with 16 nodes, Nodes
- 2 x Intel (R) Xeon (R) CPU, 384 GB DDR4, 2 TB non-volatile DIMM (NVM), 1 Nvidia (R) V100 (R) 32GB HBM2 GPU 1 Intel (R) Stratix 10 FPGA 32GB DDR4
- Processors: 32 x Intel (R) Xeon (R) Platinum 8260M Scalable Processor @ 2.4GHz (768 cores total)

The scalability of these two parallel machines for training models is reported in the result section.

5 RESULTS

JUWELS booster module assessed with 1 to 4 GPUs with one node. The results in Figure 5 show GPUs made effective speed up for GRU model training in order to with 2 GPUs, 3 GPUs, and 4 GPUs the computing time reduces 50%, 100%, and 150% rather than one node with only 1 GPU.

Figure 6 illustrates the computing scale for described GRU model in this study on 1 node with 1 and 2 GPUs of the DEEP-EST DAM module machine. As it is expected the GPUs significantly increased the speed of computing by more than 50

Table 1 reports the computing time for the applied parallel computing machine. Also,

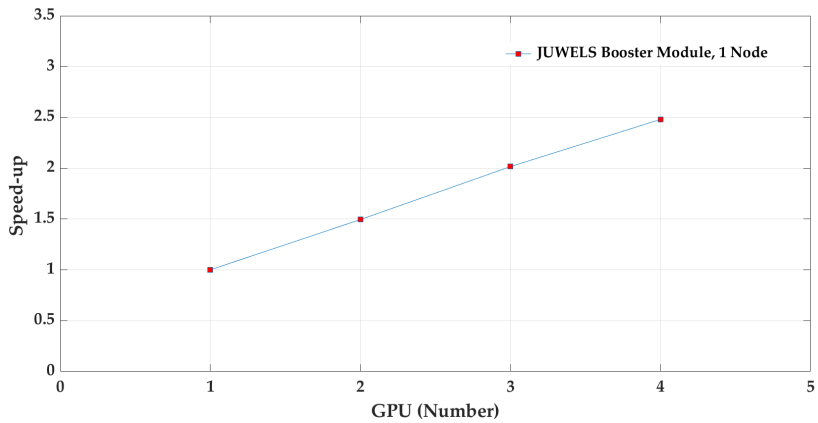


Figure 5: Computing scalability, JUWELS BOOSTER MODULE, 1 node assessed with 1 to 4 GPUs for GRU training model.

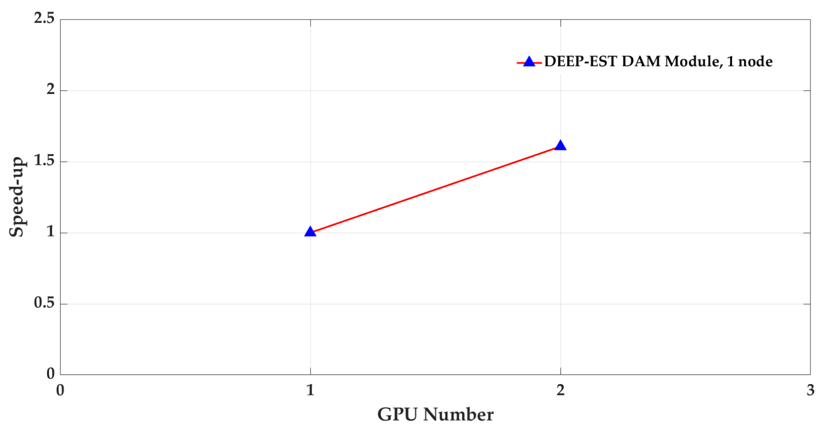


Figure 6: Computing scalability, DEEP-EST DAM MODULE, 1 node assessed with 1 to 2 GPUs for GRU training model.

Table 1: Parallel computing machine scalability to create the GRU model with GPUs

Parallel Machine/Module	Nodes	GPUs	Computing time [s]	Speed-up
JUWELS/ Booster Module	1	1	940.643	1.000
	1	2	629.094	1.495
	1	3	466.445	2.016
	1	4	379.135	2.481
DEEP-EST/ DAM Module	1	1	881.813	1.000
	1	2	548.754	1.606

table 2 shows the computing time for SVM and regression, which determines these two methods do not require parallel computing in contrast to the GRU model and can be

Table 2: Computing time to create the ML models; support vector polynomial regression and linear regression

Machine	Support vector poly. Regression model	Linear regression model
JUWELS/ Booster Module (1 node without GPUs) parallel computing [s]	17.165	0.016

implemented more straightforwardly and very fast. Still, they could not provide reliable and useful predictions with this data. GRU is significantly distinct and accurate.

6 CONCLUSIONS

This work used time-series data, including 2D velocity field, position, and time of a straining turbulence flow generated in the laboratory. Position coordinate during the time was concerned as the main feature to predict the velocity field. This statement determines the spatial and temporal properties of the fluid dynamics based on the Lagrangian framework. The results show the GRU model has a significant prediction for this data. For this approach, parallel computing scalability was investigated on two powerful machines. The GPUs can effectively speed up the GRU model, a subset of recurrent neural networks (RNN), with faster performance than LSTM and simpler architecture. The results mention this model could be suggested to employ recorded data for velocity fields such as wind speed, downstream wake profile for objects in the airflow, water flow in the pipe, and airflow in the channel and generate forecasting for the next step time.

Funding

This work was performed in the Center of Excellence (CoE) Research on AI and Simulation Based Engineering at Exascale (RAISE) and the EuroCC projects receiving funding from EU's Horizon 2020 Research and Innovation Framework Programme under the grant agreement no.951733 and no. 951740 respectively.

REFERENCES

[1] S. B. Pope, *Turbulent Flows.* Cambridge University Press, 2000.

[2] J. L. L. Henk Tennekes, *A First Course in Turbulence.* MIT Press, 1972.

[3] P. A. Davidson, *Turbulence: An Introduction for Scientists and Engineers.* Oxford University Press, 2004.

[4] D. Barbosa de Alencar, C. De Mattos Affonso, R. C. Limão de Oliveira, J. L. Moya Rodríguez, J. C. Leite, and J. C. Reston Filho, "Different models for forecasting wind power generation: Case study," *Energies*, vol. 10, no. 12, 2017.

[5] R. Jursa and K. Rohrig, "Short-term wind power forecasting using evolutionary algorithms for the automated specification of artificial intelligence models," *International Journal of Forecasting*, vol. 24, no. 4, pp. 694–709, 2008. Energy Forecasting.

[6] J. P. S. Catalao, H. M. I. Pousinho, and V. M. F. Mendes, "An artificial neural network approach for short-term wind power forecasting in portugal," in *2009 15th International Conference on Intelligent System Applications to Power Systems*, pp. 1–5, 2009.

[7] M. Carolin Mabel and E. Fernandez, "Analysis of wind power generation and prediction using ann: A case study," *Renewable Energy*, vol. 33, no. 5, pp. 986–992, 2008.

[8] A. Couto, P. Costa, and T. Simões, "Identification of extreme wind events using a weather type classification," *Energies*, vol. 14, no. 13, 2021.

[9] M. A. J. C.A.Garcia, P.G.Lind, "On the existence and characterization of extreme events in wind data," in *Proceedings of the 7. European Conf. on Renewable Energy Systems*, pp. 320–327, 2019.

[10] C.-M. Lee, Gylfason, P. Perlekar, and F. Toschi, "Inertial particle acceleration in strained turbulence," *Journal of Fluid Mechanics*, vol. 785, p. 31–53, 2015.

[11] P. A. Srinivasan, L. Guastoni, H. Azizpour, P. Schlatter, and R. Vinuesa, "Predictions of turbulent shear flows using deep neural networks," *Phys. Rev. Fluids*, vol. 4, p. 054603, May 2019.

[12] H. Eivazi, H. Veisi, M. H. Naderi, and V. Esfahanian, "Deep neural networks for nonlinear model order reduction of unsteady flows," *Physics of Fluids*, vol. 32, no. 10, p. 105104, 2020.

[13] S. L. Brunton, B. R. Noack, and P. Koumoutsakos, "Machine learning for fluid mechanics," *Annual Review of Fluid Mechanics*, vol. 52, no. 1, pp. 477–508, 2020.

[14] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, pp. 1735–1780, 11 1997.

[15] S. Hassanianmoaref, "An experimental study of inertial particles in deforming turbulence yedreza: in context to loitering of blades in wind turbines," m. eng. thesis, Reykjavik University, 2020.

[16] B. Lahcen, "On the effects of buoyancy on passive particle motions in the convective boundary layer from the lagrangian viewpoint," m. eng. thesis, Reykjavik University, 2012.

[17] M. Riedel, R. Sedona, C. Barakat, P. Einarsson, R. Hassanian, G. Cavallaro, M. Book, H. Neukirchen, and A. Lintermann, "Practice and experience in using parallel and scalable machine learning with heterogenous modular supercomputing architectures," in *2021 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pp. 76–85, 2021.

[18] J. S. Centre, "Juwels user documentation." https://apps.fz-juelich.de/jsc/hps/juwels/index.html, May 2022.

[19] J. S. Centre, "Deep-est - dynamical exascale entry platform - prototype system." https://www.fz-juelich.de/ias/jsc/EN/Expertise/Supercomputers/DEEP-EST/_node.html;jsessionid=1B6BD3B371D4C56131EB4BAF5C82FD96, May 2022.