UNIVERSITY OF ICELAND

Design and Evaluation of Parallel Machine Learning Approaches in Computational Fluid Dynamics Applications

Seyedreza Hassanianmoaref
(Reza Hassanian)

2024

FACULTY OF INDUSTRIAL ENGINEERING, MECHANICAL ENGINEERING AND COMPUTER SCIENCE

Design and Evaluation of Parallel Machine Learning Approaches in Computational Fluid Dynamics Applications

Seyedreza Hassanianmoaref
(Reza Hassanian)

Dissertation submitted in partial fulfillment of a
Philosophiae Doctor degree in Computational Engineering

Supervisor
Prof. Dr.-Ing. Morris Riedel

Doctoral Committee
Prof. Dr.-Ing. Morris Riedel
Associate Prof. Dr. Ásdís Helgadóttir
Assistant Prof. Dr. Pedro Simões Costa

Opponents
Prof. Dr.-Ing. Andrea Beck
Prof. Dr. Amir Hossein Shiravi

Faculty of Industrial Engineering, Mechanical Engineering and
Computer Science
School of Engineering and Natural Sciences
University of Iceland
Reykjavík, September 2024

Design and Evaluation of Parallel Machine Learning Approaches in Computational Fluid Dynamics Applications
(Parallel Machine Learning for Computational Fluid Dynamics)

Dissertation submitted in partial fulfillment of a *Philosophiae Doctor* degree in Computational Engineering

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

Telephone: 525-4000

Bibliographic information:
Seyedreza Hassanianmoaref
(Reza Hassanian) (2024) *Design and Evaluation of Parallel Machine Learning Approaches in Computational Fluid Dynamics Applications*, PhD dissertation, Faculty of Industrial Engineering, Mechanical Engineering and Computer Science, University of Iceland, 127 pp.

ISBN 978-9935-9769-2-5

Printing: Háskólaprent ehf.
Reykjavik, Iceland, September 2024

Abstract

The objective of this research is to investigate the potential of machine learning (ML) and deep learning (DL) within computational fluid dynamics (CFD) utilizing high-performance computing (HPC). Turbulent flow, a complex phenomenon prevalent in both natural sciences and industrial settings, poses a significant challenge in classical physics. Given its nonlinear and stochastic nature, DL emerges as a promising approach for comprehending turbulent flow dynamics.

This PhD Thesis introduces a novel data-driven methodology, leveraging experimental datasets to assess the efficacy and robustness of sequential DL models in turbulent flow analysis. The manuscript elaborates on the innovative aspects and advantages of employing DL within this context. Moreover, it outlines the necessity of HPC resources for executing cutting-edge DL models alongside the computationally demanding approaches for model tuning and hyperparameter optimization.

This study thoroughly followed a documented procedure to assess the efficacy of innovative DL modeling techniques. Initially, measured data obtained from experimental tests was analyzed in depth, establishing their relevance to DL models from a physical perspective. Subsequently, three DL model architectures— long short-term memory (LSTM), gated recurrent unit (GRU), and Transformer models—were systematically evaluated, and their outcomes were compared. Furthermore, a case study was undertaken in the wind energy sector within the realm of energy engineering, yielding promising results to validate the DL prediction model's applicability.

In addition, optimization techniques for hyperparameter tuning were explored utilizing new HPC methodologies (e.g., modular supercomputing architectures and cutting-edge graphical processing units (GPUs)) to enhance the performance of the DL models. This optimization process was rigorously executed and assessed within the setup of a data-driven model capable of training with statistics data of CFD or EFD approaches, resulting in improved and cutting-edge approaches within the CFD domains, augmenting findings of physics-driven models with DL models.

The findings of this research demonstrate a noteworthy breakthrough in predicting turbulent flow behavior, showcasing the effectiveness of the proposed DL models while not losing sight of leveraging cutting-edge HPC methodologies.

Útdráttur

Markmið rannsóknarinnar er að kanna mátt vélnáms (ML) og djúpnáms (DL) í tölulegri straumfræði (CFD) með aðstoð háafkasta tölva (HPC). Iðustreymi, margslungið fyrirbæri sem fyrirfinnst í bæði náttúrulegum aðstæðum og manngerðum iðnaði, er vel þekkt óleyst eðlisfræðilegt viðfangsefni. Þrátt fyrir að iðustreymi sé bæði ólínulegt og slembið fyrirbæri hefur djúpnám gefið góða raun um að vera fært um að segja til um hegðun iðustreymis.

Í þessari doktorsritgerð er kynnt ný gagnadrifin aðferðafræði, sem notar gögn frá tilraunum til að ná fram styrkleikum og áreiðanleika runubundins djúpnáms í iðustreymisgreiningu. Ritgerðin sýnir fram á helstu nýjungar þróaðar hér og kosti þess að nota djúpnám á þetta viðfangsefni. Einnig er farið í nauðsyn þess að nota háafkastatölvur til að beita djúpnámsaðferðum auk tölulega krefjandi aðferða til að stilla líkanið af og bestunar á yfirfæribreytu.

Í þessari rannsókn er lýst nákvæmlega stöðluðum aðferðum um hvernig má ná fram hámarksvirkni djúpnámsaðferða. Í fyrstu var farið vandlega yfir tilraunagögn og nýting þeirra til djúpnáms í raunverulegum aðstæðum skoðuð ítarlega. Næst eru þrjár mismunandi djúpnáms aðferðir – minnugt endurkvæmnis tauganet (LSTM), hlið endurtekin eining (GRU) og breytir (Transformer) – skoðaðar kerfisbundið í þremur fösum og niðurstöður þeirra bornar saman. Auk þess var gerð tilviksrannsókn úr vindorkugeiranum, þ.e. orkuverkfræði, sem sýndi niðurstöður sem lofa góðu til að sannreyna færni djúpnáms til að spá fyrir um framtíðarhegðun.

Í lokafasa rannsóknarinnar var bestunaraðferðum fyrir yfirfæribreytu beitt með því að nota skjákort (GPU) úr fremstu röð til að bæta virkni djúpnámslíkansins. Bestunin var framkvæmd af mikilli nákvæmni og metin út frá gagnadrifnum líkönum sem unnt var að þjálfa með tölfræðigögnum úr tölulegri straumfræði (CFD) og gögnum úr tilraunum. Það leiddi til bættra og nýrra framúrskarandi aðferða í tölulegri straumfræði þar sem niðurstöður módela byggðum á eðlifræði eru bætt með djúpnámi.

Niðurstöður rannsóknarverkefnisins eru bylting í að spá fyrir um hegðun iðustreymis og sýna klárlega virkni djúpnáms þegar fremstu háafkastatölvuaðferðum er beitt.

To my wife, Nashmin,
to my Parents and our children Mersana and Diana

Contents

List of Figures

List of Tables

List of Publications

Paper I: **R. Hassanian**, Á. Helgadóttir, L. Bouhlali, M. Riedel, *An Experiment Generates a Specified Mean Strained Rate Turbulent Flow: Dynamics of Particles*, American Institute of Physics (AIP) Physics of Fluids, vol. 35, no. 1, 2023 https://doi.org/10.1063/5.0134306, 2023

Paper II: **R. Hassanian**, M. Riedel, L. Bouhlali, *The capability of recurrent neural networks to predict turbulence flow via spatiotemporal features*, 2022 IEEE 10th Jubilee International Conference on Computational Cybernetics and Cyber-Medical Systems (ICCC), Reykjavik, Iceland, July, 2022, pp. 335–338 https://doi.org/10.1109/ICCC202255925.2022.9922754, 2022

Paper III: **R. Hassanian**, Á. Helgadóttir, M. Riedel, *Deep Learning Forecasts a Strained Turbulent Flow Velocity Field in Temporal Lagrangian Framework: Comparison of LSTM and GRU*, MDPI Fluids, 2022, 7(11), 344 https://doi.org/10.3390/fluids7110344, 2022

Paper IV: **R. Hassanian**, H. Myneni, Á. Helgadóttir, M. Riedel, *Deciphering the dynamics of distorted turbulent flows: Lagrangian particle tracking and chaos prediction through transformer-based deep learning models*, AIP Physics of Fluids 1 July 2023; 35 (7): 075118, https://doi.org/10.1063/5.0157897, 2023

Paper V: **R. Hassanian**, A. Shahinfar, Á. Helgadóttir, M. Riedel, *Optimizing Wind Energy Production: Leveraging Deep Learning Models Informed with On-Site Data and Assessing Scalability through HPC*, IEEE Hungary Section-Obuda Acta Polytechnica Hungarica Journal, no. 21: 9, 2024, https://doi.org/10.12700/APH.21.9.2024.9.4, 2024

Paper VI: **R. Hassanian**, M. Aach, A. Lintermann, Á. Helgadóttir, M. Riedel, *Turbulent Flow Prediction-Simulation: Strained flow with Initial Isotropic Condition Using a GRU Model Trained by an Experimental Lagrangian Framework, with Emphasis on Hyperparameter Optimization*, MDPI Fluids 9, no. 4: 84, 2024, https://doi.org/10.3390/fluids9040084, 2024

Other Publications

Paper A: **R. Hassanian**, Á. Helgadóttir, M. Riedel, *Iceland Wind Farm Assessment Case Study and Development: An Empirical Data from Wind and Wind Turbine*, https://doi.org/10.1016/j.cles.2023.100058, 2023

Paper B: **R. Hassanian**, M. Riedel, *Leading-Edge Erosion and Floating Particles: Stagnation Point Simulation in Particle-Laden Turbulent Flow via Lagrangian Particle Tracking*, https://doi.org/10.3390/machines11050566, 2023

Paper C: **R. Hassanian**, N. Yeganeh, M. Riedel, *Wind Velocity and Forced Heat Transfer Model for Photovoltaic Module*, https://doi.org/10.3390/fluids9010017, 2024

Paper D: **R. Hassanian**, Á. Helgadóttir, M. Riedel, *A wake loss model asymmetry induced by the circulation of a vertical axis wind turbine*, https://doi.org/10.1109/FES57669.2023.10182719, 2023

Paper E: **R. Hassanian**, Á. Helgadóttir, N. Yeganeh, R. Unnthorsson, M. Riedel, *Implicit Equation for Photovoltaic Module Temperature and Efficiency via Heat Transfer Computational Model*, https://doi.org/10.3390/thermo2010004, 2022

Paper F: **R. Hassanian**, H. Myneni, Á. Helgadóttir, M. Riedel, *Vertical Axis Wind Turbine Powers Telecom Towers: Green and Clean Configuration*, https://doi.org/10.1109/CEEGE58447.2023.10246601, 2023

Paper G: **R. Hassanian**, M. Riedel, *Buckling Assessment in the Dynamics Mechanisms, Stewart Platform Case Study: In the Context of Loads and Joints, Deflection Positions Gradient*, MDPI Computation 11, no. 11: 227, 2023 https://doi.org/10.3390/computation11110227, 2023

Paper H: **R. Hassanian**, N. Yeganeh, R. Unnthorsson, Á. Helgadóttir, M. Riedel, *A Practical Approach for Estimating the Optimum Tilt Angle of a Photovoltaic Panel for a Long Period—Experimental Recorded Data*, MDPI Solar 2021; 1(1):41-51, https://doi.org/10.3390/solar1010005, 2021

Abbreviations

AI	Artificial Intelligence
CFD	Computational Fluid Dynamics
CPU	Central Processing Unit
CNN	Convolutional Neural Network
DL	Deep Learning
DMD	Dynamical Mode Decomposition
DNS	Direct Numerical Simulation
EFD	Experimental Fluid Dynamics
GWEC	Global Wind Energy Council
GMM	Gaussian Mixture Model
GRU	Gated Recurrent Unit
GPU	Graphics Processing Unit
HAM	Hybrid Analysis and Modeling
HPC	High-Performance Computing
HPO	Hyperparameter Optimization
IEA	International Energy Agency
IMO	Icelandic Meteorological Office
JSC	Juelich Supercomputing Centre
JUWELS	Jülich Wizard for European Leadership Science
LES	Large Eddy Simulation
LSTM	Long Short-Term Memory
LPT	Lagrangian Particle Tracking
LUBE	Lower-Upper-Bound-Estimation

MAE	Mean Absolute Error
MSA	Modular Supercomputing Architecture
ML	Machine Learning
MLP	Multilayer Perceptron
MPI	Message Passing Interface
MTF	Multiphase Turbulence Flow
PBM	Physics-Based Model
PINNS	Physics-Informed Neural Networks
PIV	Particle Image Velocimetry
PLT	Particle-laden turbulence
POD	Proper Orthogonal Decomposition
RANS	Reynolds-Averaged Navier-Stokes
Re	Reynolds Number
RNN	Recurrent Neural Network
ROM	Reduced-Order Model
SGS	Subgrid-Scale
St	Stokes Number
TO	Thesis Objective

Acknowledgments

I extend my heartfelt appreciation to my esteemed supervisor, Prof. Dr. Morris Riedel, whose unwavering support has been instrumental throughout my doctoral journey. Without his guidance, this dissertation would not have been easy to complete. Prof. Dr. Morris Riedel generously afforded me the opportunity to teach, an experience that enriched my academic growth. Fond memories of shared moments with friends and colleagues, hosted by Prof. Dr. Morris Riedel, remain cherished.

My gratitude also extends to Dr. Ásdís Helgadóttir for her collaboration, research support, and invaluable guidance. Additionally, I am thankful to Dr. Ásdís Helgadóttir and Prof. Halldór Pálsson for entrusting me with teaching responsibilities in the fluid mechanics laboratory course.

Special thanks to Dr. Pedro Costa for his support in our collaborative works and mentorship. Professors Matthias Book and Helmut Wolfram Neukirchen from the computer science department deserve recognition for fostering an exceptional working environment and providing continuous support.

I appreciate the collaboration of Dr. Gabriele Cavallaro, Dr. Hemanadhan Myneni, Dr. Chadi Barakat, Dr. Rocco Sedona, and Marcel Aach in our research endeavors.

Gratitude is also owed to Dr. Clara Velte from DTU, Dr. Andreas Lintermann from Forschungszentrum Jülich, and Prof. Guillaume Houzeaux from the Barcelona Supercomputing Center for their guidance and support in publishing articles.

I am indebted to the friends within Prof. Morris Riedel's research group for their assistance and camaraderie during my research.

I am grateful for the support provided by the European High-Performance Computing Joint Undertaking, which funded my doctoral studies throughout the EuroCC 1 and 2 projects.

To my wife Nashmin, whose unwavering support and patience sustained me throughout my doctoral journey, I express profound gratitude. Without her, this dissertation would have been a far more daunting endeavor.

Last but not least, I am grateful to my parents for their unwavering support and sacrifices, which enabled me to reach this significant milestone in my life.

1 Introduction

1.1 Motivation

In fluid dynamics, alongside restricted analytical techniques, two approaches, Computational Fluid Dynamics (CFD) and Experimental Fluid Dynamics (EFD) methods, are employed to address scientific and engineering problems [1]. EFD is constrained by both the scale of the problem and its associated costs (e.g., experimental setup, wind tunnels, measurement devices, materials, security concerns, etc.), limiting its applicability across a wide range of scenarios [2]. Conversely, CFD's effectiveness enables understanding when increasing the problem scale and necessitates substantial computing resources [3]. As problem size increases, CFD encounters computational challenges. Additionally, validating CFD simulations requires comparison with experimental data, presenting another hurdle. Furthermore, proficiency in theoretical aspects of a problem is essential for utilizing CFD effectively.

Turbulent flow, a complex phenomenon in fluid dynamics, remains a challenge in classical physics [4]. Besides its randomness and nonlinear behavior, turbulent flow is prevalent in various natural sciences and industrial settings, indicating its widespread occurrence [1]. While EFD can address turbulent flow within a limited range, CFD offers three main categories for tackling turbulent flow problems: Reynolds-Averaged Navier-Stokes (RANS) [1], Large Eddy Simulation (LES) [5], and DNS [6]. RANS provides an average solution that is most applicable in the industry because it is the least computationally expensive but also the least accurate. LES creates a solution with better accuracy than RANS and its computing cost is higher than RANS and lower than DNS. DNS is the most expensive computing solution and provides the most exact solution. The continuous improvement and availability of High-Performance Computing (HPC) systems enhance the feasibility of more accurate simulations, yet many CFD problems, especially those involving intricate flow structures, remain computationally challenging [7, 8]. Additionally, empirical data validation is crucial in numerous CFD applications, posing another limitation [8].

These limitations raise the necessity for reliable tools to address such challenges and enable the analysis of turbulent flows across various scales. In recent years, Machine Learning (ML) and DL have demonstrated significant potential in studying nonlinear phenomena such as observed in fluid flow.

The subject of this study is therefore evaluating hybrid data-driven models of DL with the application of HPC for turbulent flow analysis, focusing on the simulation and prediction of future periods of turbulent flow. The primary concept of turbulent flow is

that it is composed of different sizes of eddies, a bunch of coherent molecules, which Richardson summarized in 1922 [9]. The eddies have various sizes, and this scale distinction has complicated the understanding of turbulence. G.I. Taylor, in 1935, established a fundamental statistical theory of turbulence [10]. A. N. Kolmogorov in 1941 proposed his hypotheses (K41) regarding the structure of turbulent flow [11]. Kolmogorov's hypotheses only determine the universal equilibrium range, composed of the dissipation and inertial ranges [1]. This thesis is, therefore, motivated by the fact that there is no comprehensive analytical understanding of turbulent flow, particularly on a large scale. Thus, predicting turbulent flow in the upcoming period hinges on understanding its characteristics, which is a complex task. Evaluating DL models within hybrid data-driven models could facilitate the analysis of turbulent flow and potentially serve as helpful tools for CFD applications. Two key lessons have been learned regarding large-scale structures: firstly, turbulence can manifest a broader spectrum of behaviors beyond turbulence typically accounted for by turbulence viscosity; secondly, turbulence operates non-locally in space and time [1]. Hence, turbulence possesses a prolonged memory, with its behavior at a given point being significantly influenced by flow conditions far removed from that point [1].

Similarly, there is no established theoretical framework for large-scale turbulent flow, and recent experiments challenge traditional assumptions regarding velocity difference statistics within the inertial range [12]. This study proposes an approach based on Lagrangian Particle Tracking (LPT) [13], dependent on the particle Stokes number(Stokes Number (St)), which is defined in the paper I and subsection 2.3, to investigate turbulent flow properties. The research also highlights the potential development of this approach in Multiphase Turbulence Flow (MTF) [1], incorporating particles, droplets, and bubbles [14]. Due to the lack of understanding of large eddies in turbulent flow, knowledge gaps persist regarding MTF [15, 16]. Additionally, the impact of gravity in MTF has not been thoroughly explored in CFD, with recent data suggesting caution when generalizing findings from experiments conducted in zero gravity environments [17]. Moreover, it is noted that the dynamics of finite-size particles in MTF cannot be solely determined by their response time [17].

Various methods have been developed to address the above-mentioned limitations to capture the key characteristics of turbulent flow through Reduced-Order Model (ROM). Prominent techniques such as Proper Orthogonal Decomposition (POD), Dynamical Mode Decomposition (DMD), and Koopman analyses are employed to construct ROMs [18, 19]. Additionally, within CFD, essential tasks include dimensionality reduction, feature extraction, super-resolution, application of ROMs, turbulence closure, shape optimization, and flow control [20]. However, these techniques tend to be intricate when dealing with real-scale and extensive fluid dynamics problems.

The thesis is further motivated by the fact that DL has demonstrated impressive proficiency in forecasting nonlinear and stochastic phenomena. Among the various DL architectures, the sequential model stands out (i.e., for investigating time series) for its potential in turbulent studies and is primarily researched in this thesis. As previously mentioned, turbulent flow operates on a large scale, affecting points both

[1] Particle-laden turbulence (PLT)

spatially and temporally, with each point influenced by flow conditions distant from it. Additionally, the Lagrangian framework [3, 2] establishes a link between time and local properties in fluid dynamics. Developing a DL model that incorporates both the Lagrangian framework and memory of turbulent flow could provide a foundation for predicting emerging turbulent flow features.

In the domain of fluid dynamics, the Lagrangian framework delineates spatial and temporal functionalities. One can employ a DL model trained solely on temporal, spatial features, or both within this framework. This approach has been adopted to construct a model capable of predicting turbulent flow without reliance on flow characteristic attributes like Re, St, strain mean rate, and gravity.

Employing a DL model trained on a turbulent flow dataset necessitates HPC resources. Moreover, HPO [21] was utilized to refine the DL models. Within the EuroCC 1 and 2 projects [2] and the National Competence Centre (NCC) for HPC and Artificial Intelligence (AI) of Iceland [3], the present research gained access to machines at the Juelich Supercomputing Centre (JSC) [4]. Therefore, the thesis findings could use cutting-edge HPC research resources such as DEEP [22] and JUWELS [23]. Those HPC systems are based on the innovative Modular Supercomputing Architecture (MSA) approach [24].

Utilizing a data-driven DL model necessitates a suitable dataset, particularly in fluid dynamics, to ensure the reliability and applicability of the proposed model in similar conditions and future studies. This thesis used a dataset from strained turbulence flow derived from a laboratory experiment, employing the LPT technique for flow feature extraction. This approach ensures that the proposed DL data-driven methods are founded on dependable experimental data, rendering them valid. Strained turbulence flow finds relevance across various applications. The distribution of pollutants in the atmosphere, the formation of rain in the cloud [25], and the spread of sediments in the ocean and rivers are a few instances [26]. Many industrial applications are related, including internal engine combustion, particle interaction in a mixing chamber [27], and leading-edge [28] erosion in compressors and turbines [29, 30]. The external flow over an airfoil [31, 32] and internal flow in a changeable cross-section pipe [33, 34] are instances of straining turbulent flow [35, 36].

Given the above, it is evident that there is a need for a data-driven DL approach through which integrating physical properties of turbulent flow and AI (i.e., DL and ML methods) capability can be made possible.

The proposed approach would provide the necessary flexibility through which:

- Dataset can be easily stored, cleaned, manipulated, and analyzed.
- Prospective prediction models can be developed, tested, and improved.
- Data must be valid in fluid dynamics.
- The DL model is flexible for different datasets with different inherent features and characteristics.

[2] EuroCC Access Webpage: https://www.eurocc-access.eu/
[3] NCC Webpage: https://www.ihpc.is/
[4] JSC Webpage: https://www.fz-juelich.de/en

- We foresee that a wide variety of other applications can use the findings of this thesis.
- DL and HPC are both essential components for implementing the current approach and could significantly benefit CFD applications.

1.2 Thesis Objectives

The overall objective of this thesis is to investigate the integration of cutting-edge HPC resources and innovative DL algorithms for CFD. However, achieving this integration requires extensive preparatory work, including dataset analysis for a data-driven approach, acquiring the necessary skills, techniques, and tools, and establishing suitable working environments. The ultimate aim of this research is to develop and validate a data-driven approach for analyzing and predicting turbulent flow in fluid dynamics, along with constructing DL-based models optimized for HPC resources utilization. To assess progress towards this overall thesis objective, a set of Thesis Objective (TO)s are outlined below.

TO1 – Constructing a dataset sourced from a credible experiment involving turbulent flow and describing the linkage between the Lagrangian framework and the architecture of sequential DL models.

TO2 – To develop an innovative data-driven approach using fluid dynamics data, enabling individual prediction of flow components (x, y, and z) as independent entities.

TO3 – Evaluating diverse data-driven DL models and analyzing their strengths and weaknesses within the proposed methodology, particularly emphasize on utilizing HPC resources.

TO4 – Employing the current state-of-the-art Transformer model within the context of turbulent flow and assessing its performance alongside pioneering sequential DL models such as LSTM variants.

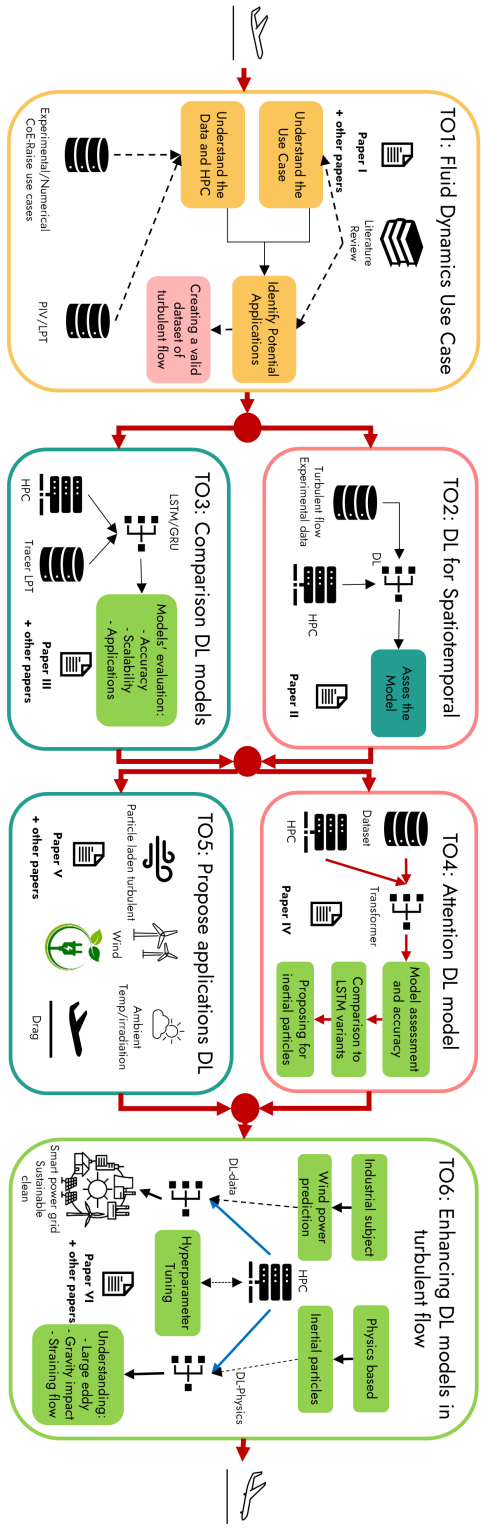
TO5 – Demonstrating the practical engineering application of integrating fundamental fluid dynamics studies with DL models in sustainable energy engineering, focussing on forecasting wind power production.

TO6 – Enhancing the data-driven model by increasing the volume of input data to assess the performance of DL models, coupled with HPO techniques in the context of HPC.

The diagram presented in Figure 1.1 delineates the chronological progression of the TOs from the commencement of the Ph.D. research to its conclusion. It showcases the sequential milestones undertaken to accomplish each TO, encompassing essential physics of the fluids, techniques, HPC methods, and data employed throughout the process.

The interconnection among the TOs is depicted by the red arrows, symbolizing the direct transfer of knowledge acquired in one phase to the subsequent step.

Finally, the flow diagram in Figure 1.1 offers a visual depiction of the TOs addressed in each of the publications outlined in Chapter 4.



6 *Figure 1.1. Flow diagram of the thesis outline and the achieved publications.*

1.3 Outline

This thesis adopts a cumulative format, emphasizing the achievement of the TOs via the publications listed in the "List of Publications". A comprehensive overview of the organization of this manuscript is available in Section 1.3.1, Subsequently, Section 1.3.2 presents a compilation of primary publications associated with the objectives outlined in this Thesis. Chapter 4 offers a detailed summary of these publications. Additional publications, not directly necessary but still somewhat relevant to the work undertaken by the thesis author or where their involvement was marginal, are listed in the "Other Publications".

1.3.1 Thesis Structure

This thesis is organised as follows:

- **Chapter 1** The introduction outlines the scope of the thesis, establishes the Thesis Objectives, and elucidates the connection between these objectives and the published works referenced in subsequent sections of the thesis.
- **Chapter 2** Offers insights into the foundational topics upon which the work detailed in this thesis is constructed. This information also contributes to the broader knowledge accumulated during the course of **TO1** and **TO6** endeavors.
- **Chapter 3** Reviews related work approaches and research showcasing analogous approaches to those expounded upon in this thesis and delineates the current state-of-the-art technology and methodologies within the field.
- **Chapter 4** presents comprehensive explanations of the conference and journal publications.
- **Chapter 5** provides a summary of the thesis and offers a concise overview of potential future endeavors.
- Subsequently, the publications enumerated in Section 1.3.2 are presented in their entirety.
- The thesis concludes with the bibliography Section, which catalogs all referenced literature.

1.3.2 Publications

- **Paper I**
  R. Hassanian, Á. Helgadóttir, L. Bouhlali, M. Riedel, 'An Experiment Generates a Specified Mean Strained Rate Turbulent Flow: Dynamics of Particles', in AIP *Physics of Fluids*, vol. 35, no. 1, 2023 https://doi.org/10.1063/5.0134306.

- **Paper II**
  R. Hassanian, M. Riedel, L. Bouhlali, 'The capability of recurrent neural networks to predict turbulence flow via spatiotemporal features', in *IEEE 10th Jubilee International Conference on Computational Cybernetics and Cyber-Medical Systems (ICCC)*, Reykjavik, Iceland, July, 2022, pp. 335–338 https://doi.org/10.1109/ICCC202255925.2022.9922754.

- **Paper III**
  R. Hassanian, Á. Helgadóttir, M. Riedel, 'Deep Learning Forecasts a Strained Turbulent Flow Velocity Field in Temporal Lagrangian Framework: Comparison of LSTM and GRU', in MDPI *Fluids*, vol. 7, no. 11, 2022, https://doi.org/10.3390/fluids7110344.

- **Paper IV**
  R. Hassanian, H. Myneni, Á. Helgadóttir, M. Riedel, 'Deciphering the dynamics of distorted turbulent flows: Lagrangian particle tracking and chaos prediction through transformer-based deep learning models', in AIP *Physics of Fluids*, vol. 35, no. 7, 2023, https://doi.org/10.1063/5.0157897.

- **Paper V**
  R. Hassanian, A. Shahinfar, Á. Helgadóttir, M. Riedel, 'Optimizing Wind Energy Production: Leveraging Deep Learning Models Informed with On-Site Data and Assessing Scalability through HPC', in *IEEE Hungary Section-Obuda Acta Polytechnica Hungarica Journal*, vol. 21, no. 9, 2024, https://doi.org/10.12700/APH.21.9.2024.9.4.

- **Paper VI**
  R. Hassanian, M. Aach, A. Lintermann, Á. Helgadóttir, M. Riedel, 'Turbulent Flow Prediction-Simulation: Strained flow with Initial Isotropic Condition Using a GRU Model Trained by an Experimental Lagrangian Framework, with Emphasis on Hyperparameter Optimization', in MDPI *Fluids*, vol. 9, no. 4: 84, 2024, https://doi.org/10.3390/fluids9040084.

1.4 Contributions

As mentioned in Section 1.2, each publication relates directly to one or more TOs as shown in Table 1.1. The central aspect of **TO1** which is reported in **Paper I** is to generate a reliable dataset in the fluid dynamics community to be used in a DL model with utilizing HPC. Therefore, **TO1** focuses on preprocessing data from an experiment and extracting the required information that could be used. **TO2** is a response to questions on how to find a connection between a data-driven DL architecture and the physical properties of turbulent flow. The core idea is established based on the Lagrangian framework, which connects the spatio-temporal feature, and it is published in **Paper II**. To assess the optimum model among DL sequential prediction models, in the context of performance and accuracy, the **TO3** enhances the strained turbulent flow prediction model with two most applicable DL sequential networks and compares their approaches and results in **Paper III**. To investigate the other DL models that could be employed in fluid dynamics and prove their capability, in **TO4**, an innovative attention mechanism integrated in cutting-edge DL models (i.e., transformer) are examined and introduced in **Paper IV**. **TO5** proposes and displays a use-case as an engineering application of the DL models that are successful in the fluid dynamics assessment in previous TOs and illustrates in **Paper V**. The study takes another step to enhance the DL models in a data-driven approach to predict the turbulent flow features combined with spatial and temporal features in **TO6**, which had more extensive data and uses hyperparameter tuning to optimize the model performance as reported in **Paper VI**. 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 to each publication are highlighted.

In order to enhance the DL and HPC applications in the fluid dynamics area, datasets are necessary based on the data-driven approach. Most of the flows are turbulent, and turbulent flow is still complex and has not been understood comprehensively. Therefore, guiding the application of DL and HPC in this way

Table 1.1. Relation of publications to the TOs.

	Paper I	Paper II	Paper III	Paper IV	Paper V	Paper VI
TO1	×					
TO2	×	×				
TO3	×		×			
TO4	×			×		
TO5					×	
TO6	×		×			×
Transferable Knowledge	×	×	×	×	×	×

to add an aid technique helping in turbulence study was chosen. Data from a laboratory experiment from the author's research is employed to follow up on that goal. The dataset was generated from strained turbulent flow with a particular range of Reynolds number (Re) [2] and specific mean strain rates. The experiment used a tracer particle to seed the flow and, with laser and high-speed camera, recorded the particle movement, which presented the flow streamline based on the Stokes number (St) [2] measurement. The experiment employed the Lagrangian particle tracking technique to extract the turbulent flow feature. The flow was three-dimensional, and the measurements were two-dimensional. The study focused on the velocity of the turbulent flow, which is fundamentally assumed to carry most of the flow features. **TO1** established the required data for next TOs. The **TO1** includes processing extensive measurement data to transfer and convert to applicable forms and files.

Utilizing a DL model for turbulent features needs to build a connection between the physics of the phenomenon and the DL architectures. In Fluid dynamics, the Lagrangian framework defines the flow properties as a function of time. Furthermore, the location is also connected to the time. This demonstration of the turbulent feature was an excellent design to apply sequential DL models to predict the flow features. Therefore, the recorded data from **TO1** has been used to train a sequential model as LSTM model and its variants and test its prediction. The first phase of this examination, which was performed in **TO2**, resulted in an impressive alignment between the predicted velocity of the flow and the measured velocity, exhibiting remarkable accuracy. Notably, the model's training relied solely on velocity time series data, devoid of any flow characterization information, enhancing its superiority within this domain. However, it was applied to a small dataset to assess this approach. The proposed approach is developed and leveraged in the next TOs.

Based on the literature, LSTM variants from Recurrent Neural Network (RNN) are excellent options for predicting sequential properties. In spite of the accuracy of the DL model prediction for turbulent flow in the present study, the HPC performance needs to be examined. Therefore, in **TO3** the data of the strained turbulent flow was used to train and predict the following period of the velocity with LSTM and GRU with different training and test ratios and the model performance was measured. The outcome of the **TO3** presents a perspective to select the proper DL model in fluid dynamics applications.

Among the DL models, the transformer incorporating from the attention mechanism is state of the art and, at the time of the current study, had not been examined in fluid dynamics, particularly in a turbulent regime in the data-driven approach. Therefore, **TO4** presents a novel application of the transformer model to predict the turbulent flow velocity with remarkable accuracy that was observed in the test. The outcome of **TO4** opens a window to develop and leverage transformer applications in the fluid dynamics area.

TO5 studies an engineering application of the achievement in **TO1** to **TO4**. In wind energy production, the prediction of power production, which is a function of the wind speed and technical parameters of the wind turbine and land surface type, is relevant to the wake loss effect [37, 38]. Thus, obtaining a prediction model of the wind speed with nonlinear and randomness behavior could lead to great development in wind power prediction. Smart power production will assist in

development of a smart power grid and cities. **TO5** applied GRU model trained with wind velocity from the Icelandic Meteorological Office (IMO) to predict the following period of the wind speed. The study proposed the idea to split the data seasonally and predict the wind speed in the seasonal category. The result was successful and had remarkable accuracy, and it can be developed for practical application in the industry. Also, the **TO5** proposes the local HPC cluster in wind farm supplied by the wind farm production, making the clean and green resources for HPC applications.

The employed dataset in the present work is obtained from strained turbulent flow seeded by tracer particles and LPT used to extract the flow features. In the examination among **TO1** to **TO5** the DL only trained with time series of the velocity and was not informed about the turbulence intensity, strain rate, and particle size–density. This shows that the model only relies on the time series of the velocity. In order to assess the model with larger data and more information, the location of the particles corresponding to the velocity is also trained to help the model predict the velocity. A GRU model is used and trained in the spatial and temporal approach. The results of this design are presented in **TO6**. Furthermore, HPO on cutting-edge HPC resources was used to improve the prediction model's performance. The author's contribution to the published papers in his PhD thesis involves writing the original draft, conceptualizing, developing methodology, developing software, validating, formal analysis, investigation, visualizing, and corresponding author.

2 Background

2.1 Deep Learning Models for Sequential Datasets

DL, a subset of both ML and more broadly, AI, has emerged as a cornerstone technology in today's Fourth Industrial Revolution (4IR or Industry 4.0) [39]. Leveraging its ability to extract insights from data, DL, rooted in artificial neural network (ANN) principles [39], has captured widespread attention across scientific and engineering domains and finds extensive utility in sectors such as healthcare, visual recognition, text analytics, cybersecurity, among others. Yet, crafting effective DL models presents a challenge, given the ever-evolving DL model architectural design complexities and capabilities and nuances of real-world problems and data. Furthermore, the opacity inherent in DL methodologies, lacking transparent comprehension, can impede progress and standardization efforts, rendering them akin to black-box systems. That is particularly challenging to use in physics applications that are often driven by causality and based on physical laws using often fully understandable numerical methods.

Still, DL methodologies have experienced a remarkable surge in performance across a broad spectrum of applications, also within physics and engineering, where they excel at unraveling intricate structures within high-dimensional datasets. Consequently, these techniques wield significant potential in constructing intelligent, data-centric systems tailored to contemporary requirements, owing to their adeptness at extracting insights from historical data (i.e., DL models can perform feature learning apposed to traditional ML models were inputs often had to be manually feature engineered). As a result, DL revolutionizes both the global landscape and individuals' daily routines through its capacity for learning. Given its intersection with AI, ML, and data science featuring advanced analytics with enormous needs of computing capacity and capability, DL technology holds profound relevance within the realm of computer science, particularly within the context of today's intelligent computing models [39].

An increasingly favored neural network model, the RNN, specializes in handling sequential or time-series data. It operates by utilizing the output from the preceding step (.e.g, time step) as input for the current phase [40, 41]. RNN derive their learning from training input, but what sets them apart is their "memory," enabling them to influence current input and output by leveraging information from past inputs. The output of an RNN depends on previous elements within the sequence. However, RNN faces the challenge of vanishing gradients [42], complicating the learning process for lengthy data sequences.

LSTM stands out as a favored RNN architecture employing specialized units to

address the vanishing gradient issue, as pioneered by Hochreiter et al. [42]. In an LSTM unit, a memory cell possesses the ability to retain data for extended durations, with the flow of information regulated by three gates (see Figure 2.1).

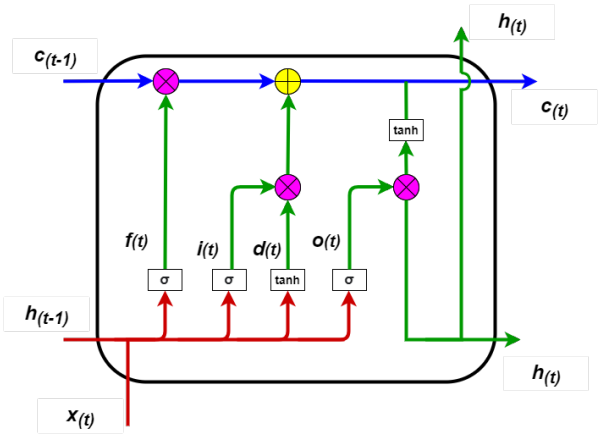


Figure 2.1. The LSTM–model architecture [43].

For example, the 'Forget Gate' decides which information from the prior state cell to retain and what to discard as obsolete. Meanwhile, the 'Input Gate' governs the entry of information into the cell state, and the 'Output Gate' dictates and manages the outputs. Addressing the challenges inherent in training recurrent networks, the LSTM network is widely regarded as one of the most effective RNN architectures. They are also known DL models that handle certain missing data in sequence data relatively well. Given the many gates and corresponding weight matrices to train, LSTM training is computationally expensive.

GRU emerges as another widely adopted variation of the RNN architecture, employing gating mechanisms to regulate the flow of information among cells within the neural network, as pioneered by Cho et al. [44]. The GRU bears a resemblance to an LSTM but boasts fewer parameters. It incorporates a reset gate and an update gate but lacks the output gate, distinguishing it from the LSTM structure.

Therefore, the primary contrast lies in the number of gates: a GRU features two (reset and update gates), while an LSTM incorporates three (input, output, and forget gates). The streamlined architecture of the GRU enables it to effectively capture dependencies within extensive data sequences while adaptively preserving information from earlier segments. Consequently, the GRU serves as a more efficient variant, often delivering comparable performance with significantly faster computational processing and learning [45]. The GRU architecture is illustrated in Figure 2.2 and is usually thus a bit less computationally expensive to train.

Even though GRUs have demonstrated improved performance on particular smaller and less frequent datasets [45, 46]. Both versions of RNN have demonstrated their effectiveness in generating results in a wide variety of scientific and engineering applications.

More recently, the transformer is a DL network structured with an encoder-

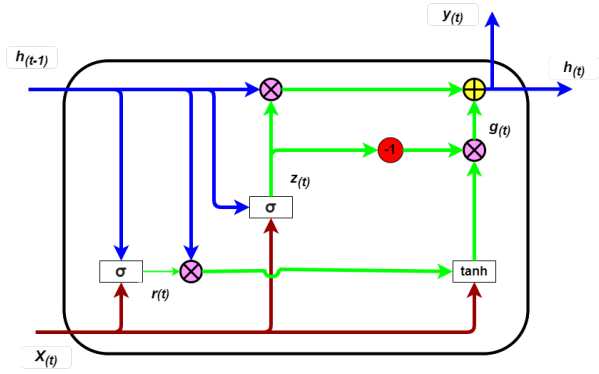


Figure 2.2. The GRU–model architecture [43].

decoder architecture [47]. Input data is processed through the encoder layers, and the decoder generates the resulting output [48]. This process involves several steps. The transformer architecture works effectively if the number of encoder layers precisely matches the number of decoder layers (see Figure 2.3). Positional embedding [49] is introduced into the input vectors to encode both the sequence and positional information within the input data. These positional embedded vectors serve as the input for the initial encoding layer, and the output from each encoder layer serves as the input for the subsequent layers.

As shown in Figure 2.3 each encoder layer is structured into two sublayers. Initially, input data traverse through a multi-head attention sublayer within the first encoder. This sublayer considers dependencies among all inputs to construct weight matrices. Subsequently, the outputs from the multi-head attention sublayer flow into the feed-forward sublayer as shown in Figure 2.3. An intermediate Add&Norm sublayer resides between these two, adding the inputs of the multi-head sublayer to its original input and normalizing the result.

Within the feed-forward sublayer, data is processed independently at each position, allowing for parallel and autonomous processing. After this, the outputs from the feed-forward sublayer undergo the Add&Norm intermediate sublayer in the same manner. Thus, data is progressively processed through each encoder layer before proceeding to the subsequent layers.

The quantity of encoder layers is not bound by fixed or magical numbers [49]; rather, it needs to be carefully selected and tailored for each specific application during the DL model architecture design phase. In the inception of the Transformer architecture [49], for instance, it was initially structured with merely six encoder-decoder layers, showcasing significant accomplishments. This aspect of the Transformer architecture highlights its impressive ability to address sequential data challenges efficiently with fewer layers. Decreasing the number of layers in DL models has the potential to reduce computational complexity in processing, but also enables a better parallel computing approach.

After traversing all encoder layers, the data proceeds to the decoder layers for output embedding. The decoder layer comprises two sublayers: multi-head and feed-

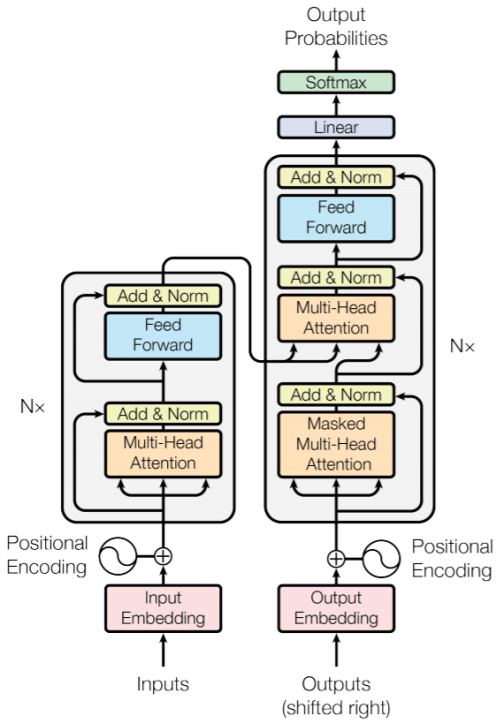


Figure 2.3. The Transformer–model architecture [49].

forward, akin to those in the encoder layer, positioned after a masked sublayer. The initial sublayer in the decoder, known as the masked multi-head sublayer, establishes a masking layer for the embedded outputs, ensuring dependency only on preceding data. This masking prevents the influence of subsequent data sequences.

Upon passing through the Add&Norm intermediate sublayer, the output from the masked multi-head sublayer flows through the multi-head sublayer. During training, three weight matrices—Query, Key, and Value [49]—are generated in the encoding layers. While the Key and Value matrices from the preceding encoder feed directly into the multi-head sublayers of decoder layers, the Query matrix for each decoder layer originates from the earlier masked sublayer.

Subsequent steps in the decoder layers mirror those in the encoder layers. Eventually, the output progresses through linear and softmax layers in the final decoder layer. The linear layer, a fully connected neural network, transforms the vector produced by the decoder stack into a larger vector known as a logit vector. Then, the softmax function converts scores from the linear vector into probabilities, ensuring all are positive and sum up to 1.0. The output for the current time step is selected based on the cell with the highest probability.

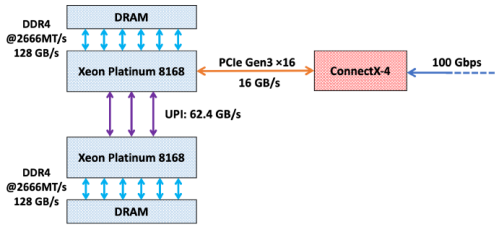
2.2 High-Performance Computing and Parallel Computing

Over the past few decades, computational technology has made significant advancement that favour the use of ML and DL models and enlarging their success in a broader set of applications., with even commonplace computers now capable of executing tens of billions of operations. Specialized machines designed for scientific endeavors further amplify this capability. The discipline that delves into the workings of such disruptive machines and their application to research and engineering problems is known as HPC or supercomputing. The distinction between HPC and regular computing relies on the fact that HPC is a research discipline on its own, always striving for new architectures and innovative methods incorporating groundbreaking new processing capabilities (e.g., accelerators) at an unprecedented scale. The augmented computational capability of supercomputers largely stems from conducting operations in exceedingly brief intervals and leveraging numerous computing units to perform operations simultaneously, known also as parallel processing or parallel computing. [50].

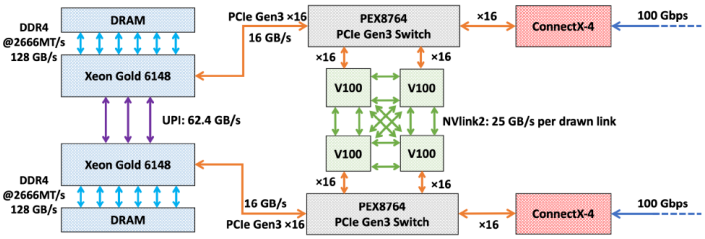
As a relatively new research area to engage in solutions facing substantial heterogeneity in user needs, the MSA integrates compute modules with varying hardware and performance characteristics into a unified heterogeneous system. Each module operates as a parallel, clustered system, potentially of considerable size, while each module focuses on specific needs (e.g., offering cutting-edge Central Processing Unit (CPU) with high single-thread performance or offering a high number of accelerators like Graphics Processing Unit (GPU)s or neuromorphic devices in the future). These module-specific interconnects are linked through a federated network. This configuration significantly benefits heterogeneous applications and workflows, as each component can run on the most suitable system, enhancing both time to solution and energy efficiency while offering still users a broad spectrum of computing solutions. Consequently, it is ideal for supercomputer centers managing mixed heterogeneous applications, offering higher throughput and better energy efficiency. Hence, the MSA provides system operators with valuable flexibility, enabling them to tailor the modules and their sizes to the center's specific portfolio and usage requirements.

To provide an example, the JUWELS system employs a MSA. Its JUWELS Cluster Module, a BullSequana X1000 supercomputer, follows a scalable hierarchical cell-based design. The cluster comprises ten Sequana X1000 cells: nine cells with 279 compute nodes each in the CPU-only partition and a tenth cell with 48 GPU-accelerated compute nodes. The CPU-only partition includes 2,511 compute nodes, each equipped with two Intel Xeon Skylake Platinum 8168 CPUs, offering 24 cores each and a base frequency of 2.7 GHz. The GPU partition features 56 compute nodes based on the BullSequana X1125 accelerator blade, with each node housing two Intel Xeon Gold 6148 processors (20 cores each) and 192 GB of main memory (cf. Figure 2.4) [23]. Hence, the Cluster Module offers cutting-edge CPUs for processing-intensive computing with high single-thread performance.

In contract, the JUWELS Booster is a BullSequana XH2000 supercomputer



(a) JUWELS Cluster compute node (CPU partition).



(b) JUWELS Cluster compute node (GPU partition).

Figure 2.4. Annotated block diagrams of JUWELS compute nodes [51].

composed of 936 X2415 compute blades. Each blade features a single node with 512 GB of memory and two AMD EPYC 7042 processors, each with a base frequency of 2.8 GHz and 24 cores, matching the core count of the Cluster partition for user convenience. Despite the robust capabilities of these processors, the standout feature of the JUWELS Booster nodes is their GPUs. Each compute node houses four NVIDIA Ampere A100 GPUs in the SXM4 form factor, each with 6912 CUDA cores and 40 GB of HBM2 memory. These GPUs communicate with each other bi-directionally at 200 GB/s via an NVLink3 bus. Another significant difference between the GPU nodes in the Cluster partition and those in the Booster is the interconnect. The complete topology of the nodes is illustrated in Figure 2.5 [23]. To sum up, the Booster Module is specifically designed for GPU-intensive workloads such as the training of ML and DL models relevant to this thesis.

A substantial element of research in HPC is parallelism, which poses challenges for both computer architecture hardware vendors, HPC systems designers, and software developers, as efficient data and message transfer between different parts of the computer is essential. Specific programming interfaces, such as Message Passing Interface (MPI) and OpenMP, facilitate this process [50]. It is important to note that running a parallelizable program across n_c computing units rarely results in an n_c times speedup compared to running it on just one unit. This limitation arises from two primary factors.

Firstly, most algorithms necessitate data and message among computing units, which can impede calculations as the devices responsible for this transfer operate at a slower pace than the computing units themselves, especially at a high rate of scalability of applications covering many different compute nodes and transfers

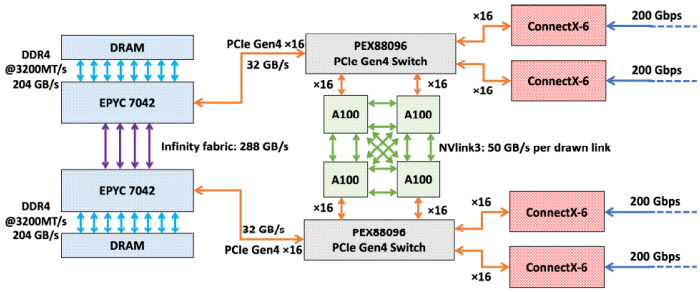


Figure 2.5. JUWELS Booster compute node [51].

involving thus thousands of processors. Secondly, algorithms often cannot be fully parallelized; only certain portions can be shared among multiple computing units, while the remainder remains inherently serial or must be executed with a core count lower than the available number. Despite these challenges, high-performance machines have become indispensable in contemporary scientific endeavors [50] and are able to overcome the limits (e.g., processing power, memory capacity, storage capability, etc.) in using just one workstation client or one desktop pc for solving scientific and engineering problems.

DL models are characterized by a multitude of parameters, and trained weights, which are refined during the training process. Additionally, each network incorporates additional hyperparameters (e.g., learning rate of optimization process, number of neurons in layer, etc.) that necessitate user configuration. For instance, adjusting the learning rate and batch size is essential to ensure the model converges to a local optimum of the training loss. However, simply minimizing the training loss to set these hyperparameters can lead to overfitting of the training data.

One DL model is defined by millions of parameters that need to be learned from large amounts of data. This process is computationally intensive and time-consuming. Often, the dataset is too large to be stored on a single machine, making it crucial to develop parallel and distributed algorithms to significantly reduce training times (e.g., distributed deep learning training [21]). When it is not feasible to store the entire dataset or model on a single machine, the data or model must be distributed across multiple machines [52]:

- Data Parallelism: The data is distributed across multiple machines. This approach is useful when the dataset is too large for a single machine or to achieve faster training by processing data in parallel.
- Model Parallelism: If the model is too large to fit into a single machine, it can be split across multiple machines. For instance, each layer of the model can be assigned to a different machine, and forward and backward propagation involve sequential communication of outputs between machines. Model parallelism is primarily used when the model size exceeds the capacity of a single machine, rather than to accelerate training.

Hyperparameters of DL models are often manually tuned through a trial-and-error approach, constituting a time-consuming and challenging aspect of ML workflows. Moreover, most hyperparameters have no straightforward rules of thumb, often requiring expert knowledge or a manually repeated random selection to determine sensible values.

HPO algorithms offer a systematic and automated solution to this DL modeling challenge [53], framing it as a global optimization problem. Typically, the default objective is to minimize error on a hold-out validation dataset, though alternative business metrics can also be considered. Furthermore, secondary objectives such as training time, inference time, or model complexity can be incorporated or constrained within the optimization process [54]. HPO thus enables an intelligent approach to reduce the search space, finding the optimal parameters for a specific DL modeling approach.

2.3 Turbulent flow in fluid dynamics

A Lagrangian framework is an exploration of fluid motion that keeps track of the velocity vector and displacement vector of each flow point, called a fluid particle [1, 3]. A fluid particle is a point that moves with the local fluid velocity, and, therefore, it identifies the position at the time t of a fluid particle [3]. The definition of a fluid particle arithmetically is [1]:

$$x_i = x_i(t, x_{i,0}), \tag{1}$$

$$U_i = U_i(t, x_1(t, x_{1,0}), x_2(t, x_{2,0}), x_3(t, x_{3,0})), \tag{2}$$

where the fluid particle position and velocity in 3D coordinates are determined by notations 1 and 2, respectively, x is the position, U is the velocity, t is the time, and i specifies the vector component.

To illustrate the Lagrangian framework conceptually, tracer particles are seeded into the fluid via experimental methods. These particles navigate through the fluid, influenced by its inherent characteristics such as turbulence intensity, deformation, and boundary conditions. By monitoring the trajectory of these particles within the fluid, it becomes possible to extract key properties of the fluid, including its instantaneous velocity, which is called Lagrangian tracking particle [3]. Thus, here the above-mentioned characterizations are described:

- The Re is the ratio of inertial forces to viscous forces. It serves as a dimensionless parameter for classifying fluid regimes, particularly where viscosity significantly influences velocity or flow patterns. In turbulent flow, characterized by high Re, inertial forces prevail [3]:

$$Re = \frac{\rho V L}{\mu}, \tag{3}$$

  where ρ is the fluid's density, V is the flow velocity, L is the characteristic length and μ is the fluid dynamic viscosity.

- The St is characterizing the behavior of particles suspended in a fluid flow. The St significantly greater than 1 ($St \gg 1$) describes particles that are unaffected by a fluid velocity change and continue their original trajectory; if ($St \ll 1$), the particle will follow the fluid's local velocity and it is defined [55]:

$$S_t = \tau_p / \tau_0, \tag{4}$$

  where τ_p is the Stokes' relaxation time and τ_0 is the characteristic time of the flow. Stokes' relaxation time τ_p is in turn calculated by equation 5 [55]:

$$\tau_p = \rho_p d_p^2 / 18 \mu_f, \tag{5}$$

  where ρ_p is the particle density, d_p is a spherical particle diameter, and μ_f is the dynamic fluid viscosity.

- In turbulent flow undergoing an axis-symmetric expansion (deformation) in the direction of Y, the mean flow field is described by [56]:

$$\langle \mathbf{U} \rangle = (Sx, -2Sy, Sz), \tag{6}$$

where x, y and z are the location and S is the mean strain rate $S = \frac{1}{\sqrt{6}}(\bar{S_{ij}}\bar{S_{ij}})^{\frac{1}{2}}$, $\bar{S_{ij}} = \frac{1}{2}(\frac{\partial U_i}{\partial xj} + \frac{\partial U_j}{\partial xi})$ is the mean rate of strain tensor with $i = 1,2,3$ and $j = 1,2,3$.

As previously mentioned, there remains a lack of experimentally validated hypotheses for anisotropic turbulent flow, particularly concerning large scales [57, 4]. Recent observations reveal discrepancies between existing theories and experimental findings [12]. The experiment finding shows that the measured second-order velocity difference statistics become independent of the Reynolds number, suggesting a universal behavior of decaying turbulence [12]. Utilizing data from experiments, excluding unproven theories, and training a DL model to forecast turbulent flow without informing it of turbulence characteristics—such as turbulence intensity, mean strain rate, and the as-of-yet-unknown effects of gravity [17]—represents a significant advancement in turbulent flow research. This approach holds broad applicability in both industrial and natural contexts. This thesis, therefore, is partly motivated by that fact, and thus, experiments are also incorporated into the thesis studies.

2.4 Data-Driven DL Model for Turbulent Flow

According to the Lagrangian perspective, time series data for fluid particles captures their position and velocity at specific points in time. Specifically, in turbulent flow—characterized by its lack of a known equation and often studied through statistical methods [4, 31]—a sequential dataset from the Lagrangian viewpoint could be utilized for forecasting models. Achieving accurate predictions for turbulent flow velocity without the need for preprocessing to extract hidden features poses a significant challenge.

Certainly, over the recent decades, big data has become a prevalent aspect of fluid mechanics research [58], owing to advancements in HPC architectures (i.e., storage and memory capacity and increasing processing power) and experimental measurement capabilities. Throughout the last 50 years, numerous techniques have emerged to manage such data, spanning from sophisticated algorithms for data processing and compression to fluid mechanics databases [59, 60]. Nonetheless, the examination of fluid mechanics data has predominantly leaned on domain expertise, statistical analysis, and heuristic algorithms [61].

Today, the proliferation of data spans across various scientific fields, ushering in a new era where deriving insights and actionable information from data is both a scientific pursuit and a commercial endeavor. Our era is marked by an unparalleled convergence of factors: (a) the exponential growth of data volume; (b) advancements in computational hardware, coupled with reduced costs for computation, data storage, and transfer; (c) the refinement of sophisticated algorithms; (d) the abundance of open-source software and benchmark problems; and (e) substantial and ongoing investments by industry in data-driven problem-solving. These developments have reignited interest and progress in DL as a means to extract knowledge from this deluge of data [17].

DL offers a flexible and adaptable modeling framework that can be customized to tackle various complexities in fluid mechanics, including reduced-order modeling, experimental data analysis, shape optimization, turbulence closure modeling, and control [62, 63]. As the focus of scientific investigation transitions from traditional first principles to data-driven methodologies, a comparison can be drawn with the evolution of numerical methods in the 1940s and 1950s for solving fluid dynamics equations [64]. The field of fluid mechanics stands poised to gain from the application of learning algorithms, while simultaneously presenting challenges that could propel the advancement of these algorithms, complementing human insight and engineering intuition [17].

Besides highlighting achievements, it is crucial to grasp the functioning of learning algorithms and discern when these methods thrive or falter. While it is essential to appreciate the potential of DL, it is equally vital to acknowledge that its integration into fluid mechanics remains an ongoing and demanding endeavor. Within this framework, it has been emphasized that the value of infusing domain expertise in fluid mechanics into learning algorithms. It is envisaged that the fluid mechanics community can play a pivotal role in driving advancements in DL, akin to the progress witnessed in numerical methods during the previous century [17, 65].

Wiewel et al. recently employed the LSTM model to predict the temporal changes

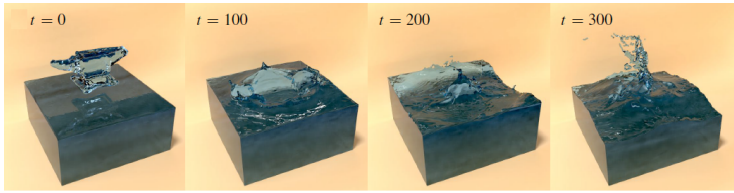


Figure 2.6. A test simulation with the DL model. The initial anvil shape was not part of the training data, but DL successfully generalizes to unseen shapes such as this one. [66].

in the pressure field of fluid flow. They showed that dense 3D+time functions of a physical system can be effectively predicted within the latent spaces of a DL model (cf. Figure 2.6). This breakthrough led to the creation of a DL-based simulation algorithm that provides substantial practical speed-ups. The results demonstrated the method's capabilities through a series of complex liquid simulations and single-phase buoyancy simulations. With trained models, this approach was more than two orders of magnitude faster than traditional pressure solvers [66]. Still, it has to be noted that DL models are still not fully interpretable even when generalizing well, and as such, they are not in competition with physical modeling, but rather a complementary approach leader to new insights. For example, a relatively new approach in this context if also the methodology of physics-informed DL model learning [67].

3 Related Work

3.1 HPC in Computational Fluid Dynamics (CFD)

CFD methods offer a convenient avenue for simulating turbulent flows, primarily through DNS and LES [1]. While LES is generally less accurate than DNS, both approaches require substantial computational resources [1] on HPC systems. RANS equations serve as a cost-effective method widely adopted in industry, albeit at the expense of accuracy compared to LES or DNS [68]. The continuous expansion of capabilities, availability, and scalability of HPC systems enable increasingly detailed simulations. Nevertheless, existing numerical techniques still fall short in tackling every CFD problem, particularly those featuring intricate and highly complex flow structures [7]. Moreover, many CFD applications necessitate validation of solutions against empirical data, posing another challenge [6, 5].

3.1.1 DNS and HPC

Using the DNS approach requires a large amount of the HPC resources and a proper meshing setup. For example, Hosseini et al. [69] applied DNS to study the flow around a wing section at a moderate Reynolds number Re; thus the mesh, which comprises around 3.2 billion grid points, was optimized to resolve all relevant scales in the flow properly. Figure 3.1 displays the resolved scales.

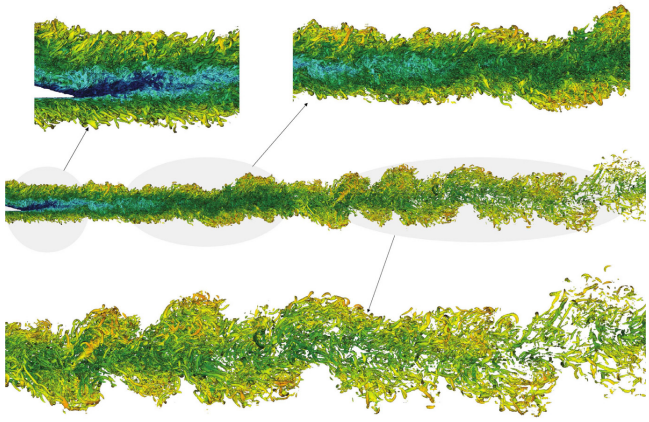


Figure 3.1. Presentation of the instantaneous vortical structures resolved by DNS for flow around a wing section at a moderate Reynolds numbe Rer [69].

The incompressible spectral-element Navier–Stokes solver Nek5000 [70] has been used to carry out the simulation shown in Figure 3.1. The work has been performed in parallel computations on 16,384 processors. Figure 3.2 illustrates a strong scaling for problem sizes. For visualization purposes, 35 million core hours have been spent collecting full turbulence statistics, time history data, and flow snapshots [69].

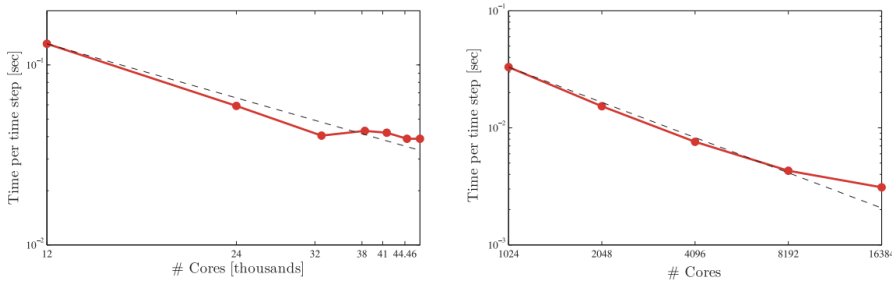


Figure 3.2. Strong scaling for problem sizes of (left) 3.2 billion and (right) 120 million grid points vs. the number of cores. The dashed line shows the linear scaling [69].

3.1.2 Fluid dynamics Use cases and HPC

In fluid dynamics, most use cases have various scenarios, which makes the studies challenging. For example, Lawson et al. study the flow around a helicopter landing on a ship [71]. The study of interactions between aircraft and ship wakes using detailed CFD methods is driven by the necessity to operate vertical take-off and landing aircraft from decks on both civil and military vessels. It is well-established that the ship's superstructure creates a highly unsteady wake [72], along with strong, organized vertical structures emanating from the ship's stern. The interaction of the ship's wake with the helicopter rotor depends on the ship and wind directions, as well as the specific features of the ship's superstructure. This interaction can negatively impact the helicopter's rotor loading and overall performance. Due to the challenges of accurately measuring flow details during sea trials and the difficulty of setting up experiments with representative Reynolds numbers Re and blade speeds in wind tunnels, numerical simulation using CFD is likely the most feasible method to study this interaction. Figure 3.3 displays the visualization of the CFD application for such use cases.

The CFD simulation has utilized HPC clusters: HECToR [5] [73] with 44,544 cores in the UK and JuRoPA [6] [74] with 17,664 cores at JSC in Germany. For this use case, the generated grid contains approximately 103 million cells. Due to the vastly different scales involved, the computation time has been extremely large. Lawson et al. [71] mentioned that the code used was highly portable and demonstrated that

[5] HECToR: High-End Computing Terascale Resource: UK National Supercomputing Service
[6] JuRoPA: Juelich Research on Petaflop Architectures, Forschungszentrum Jülich GmbH, Germany

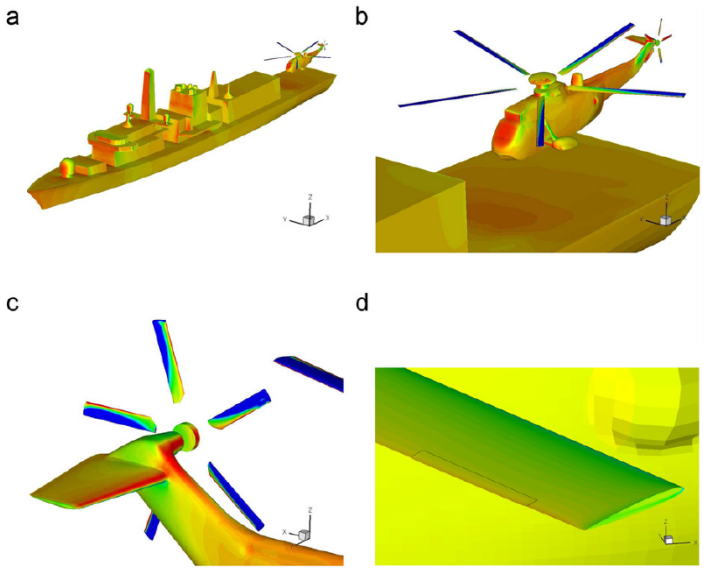


Figure 3.3. Ship and Helicopter Computation: (a) Ship and Helicopter, (b) Helicopter, (c) tail rotor, and (d) active flaps on the main rotor [71].

it could scale effectively up to 24,576 cores [71]. Given the extensive use cases in CFD, HPC is crucial, encompassing both numerical solution methods and DL applications. This thesis leverages HPC resources to train the proposed DL models.

3.2 DL in Fluid Dynamics

The research field of fluid dynamics was traditionally based on mechanical models or numerical models based on known physical laws. More Recently, DL models have shown significant capabilities in extracting latent features from nonlinear phenomena and generating predictions across various domains [18, 20]. Furthermore, researchers have explored the potential of DL models in fluid dynamics applications, such as Coletti et al. [17] and Eivazi et al. [18].

3.2.1 DL applications in aerodynamics

Duru et al. [75] leveraged DL to predict transonic flow patterns around airfoils. They developed a DL model to forecast pressure and Mach fields around airfoils across various angles of attack. The model's predictive capabilities were demonstrated through flow field contours and validated using several quantitative accuracy metrics. The performance was tested by examining the impacts of angle of attack and airfoil shape variations on the flow field. Comparisons with CFD simulations showed that the model predictions were highly accurate in capturing the effects of these variations. Figure 3.4 shows the DL model prediction of the flow on NACA0012. The proposed model significantly reduces the need for time-consuming CFD simulations, with only a slight loss in accuracy. After a 360-hour training process using two TESLA V100 GPUs, the model's prediction time for a single case is approximately 1 second, whereas the CFD solver takes about 85 minutes using 32 Xeon E5-2690 CPU cores. It demonstrates that GPU cores combined with DL models in HPC resources enables a significant progress in the field in aeronautics in particular and fluid flow in general.

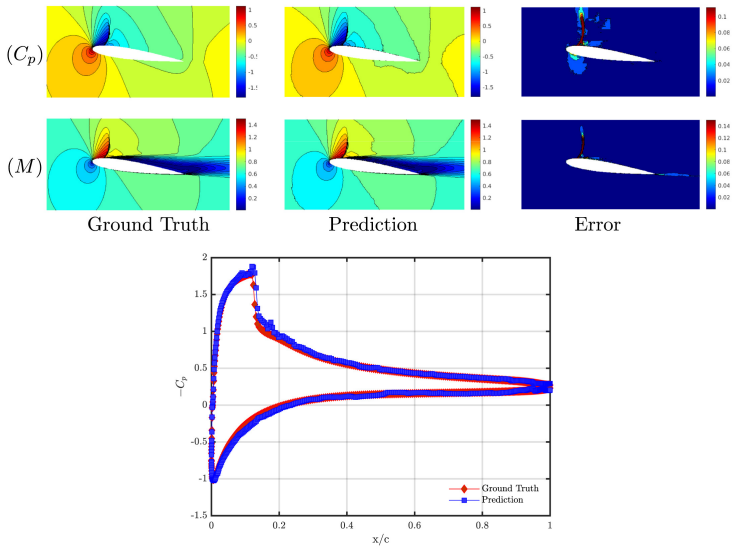


Figure 3.4. DL model predictions on NACA0012 with angle of attack 8^o [75].

3.2.2 DL model in shear flow

Srinivasan et al. [20] employ the Multilayer Perceptron (MLP) and DL model to forecast turbulent shear flow using equations derived from a Moehlis model [76]. They tested several neural network architectures by varying the number of layers, units per layer, input dimension, and weight initialization and activation functions to obtain the best configurations for flow prediction. Hence, they did not follow a systematic HPO process as performed in this thesis. Still, the LSTM led to excellent predictions of turbulence statistics and the system's dynamic behavior. Figure 3.5 represents the prediction comparison for turbulence statistics. Hence, this DL model in shear flow indicates that model tuning using hyperparameter optimization accelerated through HPC systems is necessary for good model performance.

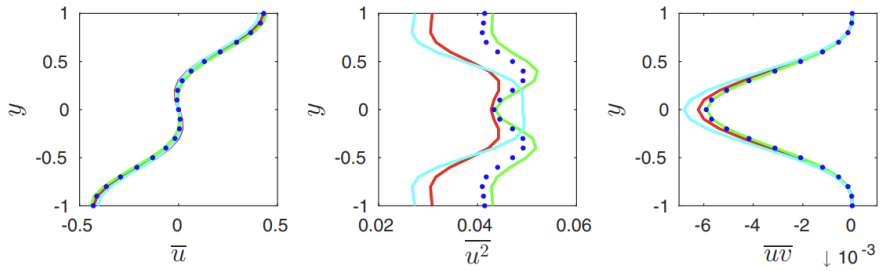


Figure 3.5. Turbulence statistics:(left) mean profile, (middle) velocity fluctuations, and (right) Reynolds shear stress. Blue dots are a nine-equation model, and prediction models: (red) LSTM1, (green) LSTM2, and (cyan) LSTM3 [20].

As shown in Figure 3.5, Srinivasan et al. examined three LSTM models with different setups which varying the number of layers, units per layer, input dimension, and weight initialization and activation functions: LSTM1, LSTM2, and LSTM3. They reported that using a standard workstation (Intel(R) Core(TM) i7-4930K CPU at 3.4 GHz), training the LSTM1 model with 10,000 time series took approximately 70 hours. After the training phase, generating 500 time series, which is the required amount for obtaining converged statistics, took around 12 minutes. In contrast, producing the same amount of data by integrating the nine-equation model by Moehlis et al. [76] took about 6 minutes on the same workstation. Therefore, after the initial training investment, the computational time needed for the DL model to predict the flow is about twice that of resolving a nine-equation model. Moreover, once trained, DL models can be summarized into smaller, more computationally efficient networks, significantly reducing prediction time complexity [77]. In CFD approaches for turbulent flow, LES offers an effective solution by directly resolving the large-scale motions and modeling the small-scale motions using the Subgrid-Scale (SGS) model [78]. Srinivasan et al. considered that the computational cost of evaluating the DL is sufficiently low to be an efficient alternative for predicting instantaneous variables, such as those in SGS models.

Due to the above mentioned drawbacks of LSTM models, hybrid models such as autoencoders- LSTM [79], LSTM/ RNN [79], and Convolutional Neural Network

(CNN)-LSTM [79] have been developed. Eivazi et al. [18] present a DL application for nonlinear model reduction in unsteady flows. Gu and Li [79] report on the use of an LSTM network for predicting turbulent wind speeds. Bukka et al. [7] employ a hybrid, deeply reduced model for predicting unsteady flows.

The majority of fluid flow investigations employing DL utilize data derived from CFD computations [20]. Given this data availability, DL models are often a bit easier to use than in use cases with real measurement datasets. Moreover, many studies incorporate pre-processing procedures aimed at identifying dominant features, such as POD or DMD [20, 18]. This thesis proposes a novel approach to predict unknown turbulent flow patterns using a DL model trained on raw measurement data, while the DL model is not informed of the turbulent flow characteristics.

3.3 Applications in Green Energy Engineering

The significant increase in energy consumption is primarily driven by emerging technological demands and evolving lifestyles, leading to the challenges of climate change as reported by International Energy Agency (IEA) [80]. Scientific reports consistently highlight the escalating temperature trends, particularly during the summer months [7] [81]. To combat rising temperatures, many countries rely on air conditioning systems in residential and commercial buildings, further straining electricity grids. Consequently, energy producers face the challenge of accurately predicting such demand spikes.

There is a global consensus that harnessing renewable energy offers the most promising solution to mitigate climate change. For instance, the proliferation of electric vehicles underscores the increasing attractiveness of utilizing green energy, such as that generated by wind turbines [80, 82]. The benefits of wind energy have been extensively researched, leading to the development of various forecasting methods—ranging from very short-term to long-term predictions—utilizing wind energy models or wind speed profiles (cf. Figure 3.6).

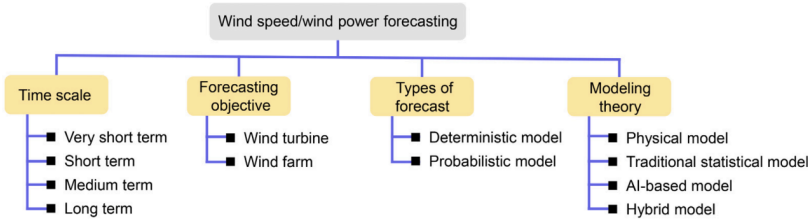


Figure 3.6. Classification of Wind Speed/Wind Power forecasting [83].

Table 3.1. Summary of DL models in wind power forecasting applications [79].

Applications	Time Step	Location	Model Used	Type
Wind speed	4 h	Onshore	1-D CNN	24 h, Real-time
Wind power	10/30/60/20-min	Onshore	Stacked ensemble	Short-term/very short-term
Wind power	24 h	-	LSTM - GMM [8]	Short-term
Wind power	-	Coast	LSTM + LUBE [9]	Short-term
Wind farm cluster power	-	Onshore	CNN + LSTM	
Surface wind	12 h	River	Real-time 4D assimilation	Short-term

[7] Global Wind Energy Council (GWEC)
[8] Gaussian Mixture Model (GMM)
[9] Lower-Upper-Bound-Estimation (LUBE)

One example of the impact of DL models using HPC in these green energy domains based on sequential frameworks was shown by wind energy prediction models [79]. The DL model, built upon sequential frameworks, demonstrated its adeptness in forecasting nonlinear phenomena [79]. To enhance the precision of the DL model in predicting wind velocity, it is crucial to select the appropriate time period and dataset size. Moreover, the wind speed exhibits distinct patterns across monthly, seasonal, bi-annual, and annual datasets, contingent upon the location of the wind park.

In order to provide an overview of related work, Table 3.1 presents a variety of DL models employed for wind power and wind speed forecasting. The table highlights the differences in time-step intervals among the models, a crucial parameter in wind speed prediction. Additionally, it demonstrates that most models are hybrid and designed for short-term forecasting. The wind speed or wind power dataset is essential for training these DL models. According to the literature, there is currently no universal model capable of predicting wind speed and wind power; the models function on a local basis [79].

Recently, it has come to light that the wake loss effect influences the weather conditions of the land downstream of offshore wind farms. DL models offer a potential avenue for exploring this phenomenon [84, 85]. This observation underscores the significance of weather forecasting [86]. DL models grounded in sequential concepts shown valuable progress in this regard. Furthermore, the potential of DL models could be harnessed in wind energy research to delve into the elusive wake loss phenomenon associated with vertical axis wind turbines [38]. This thesis examined measured wind speed data to evaluate wind power production potential by performing research studies on a suitable DL model architecture to predict wind speed, with a focus on seasonal datasets. Moreover, while many horizontal wind turbines have been studied, this thesis also researched vertical-axis wind turbines.

4 Summary of Publications

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

4.1 An Experiment Generates a Specified Mean Strained Rate Turbulent Flow: Dynamics of Particles

R. Hassanian, Á. Helgadóttir, L. Bouhlali, M. Riedel, 'An Experiment Generates a Specified Mean Strained Rate Turbulent Flow: Dynamics of Particles', in AIP *Physics of Fluids*, vol. 35, no. 1, 2023, DOI: https://doi.org/10.1063/5.0134306.

*This publication fulfills the requirements of **TO1** concerning the preparation of a reliable dataset from turbulent flow which was built experimentally. The dataset extract to be used in DL models in the data-driven approach.*

The paper describes properties of strained turbulent flow generated in a laboratory facility with a range of turbulence intensity identified by Taylor micro-scale Reynolds number Re_λ, mean strain rate of deformation, and tracer particles that were seeded the flow to display the flow characteristics. Furthermore, in separate experiments, the flow was seeded with inertial particles to study the particle movement. Table 1 in paper I displays the obtained flow parameters and the details of the measurements of the mean strain rate, the strain move, a dimensionless parameter for two different strain rates, and calculating Re_λ as well as other information are also documented in appendix A. The paper points to observations regarding the impacts of the turbulence intensity, strain rate, and gravity on the tracer and inertial particle distract their behavior. The St number for the particles was measured to illustrate the tracer particle presenting the turbulent flow streamlines. The LPT technique was employed to record the particles move, and their properties include velocity and location used in next TOs to utilize DL models in the fluid dynamics area. Figure 4.1 is the presentation of the tracer particle move in the generated turbulent flow with particle characterization. The measurement data supporting this paper's findings are available from the PhD student and his supervisor upon reasonable request.

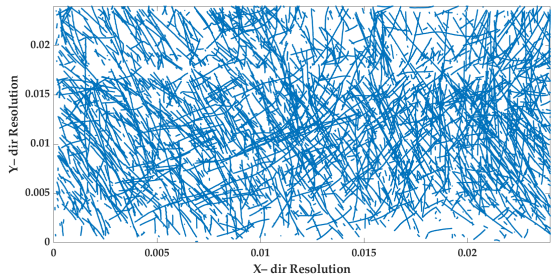


Figure 4.1. Real view of the path traveled by the particles, obtained from the video recordings for a data set that included 4000 images (resolution of 512 $\times$512 pixels), obtained from one of the 20 individual and independent videos observing the same experimental condition.

4.2 The capability of recurrent neural networks to predict turbulence flow via spatiotemporal features

R. Hassanian, M. Riedel, L. Bouhlali, 'The capability of recurrent neural networks to predict turbulence flow via spatiotemporal features', in *2022 IEEE 10th Jubilee International Conference on Computational Cybernetics and Cyber-Medical Systems (ICCC)*, Reykjavik, Iceland, July, 2022, pp. 335–338, DOI: https://doi.org/10.1109/ICCC202255925.2022.9922754

*This publication meets the criteria outlined in **TO2** by establishing a fresh connection between flow characteristics in the Lagrangian framework and the architecture of DL models.*

The paper introduces a new interpretation of how turbulent flow properties within the Lagrangian framework intersect with the GRU architecture. Furthermore, the model was designed to predict velocity for each component of turbulent flow individually, enabling its application to both 2D and 3D datasets.

In Figure 4.2, the model capability in the prediction is presented.

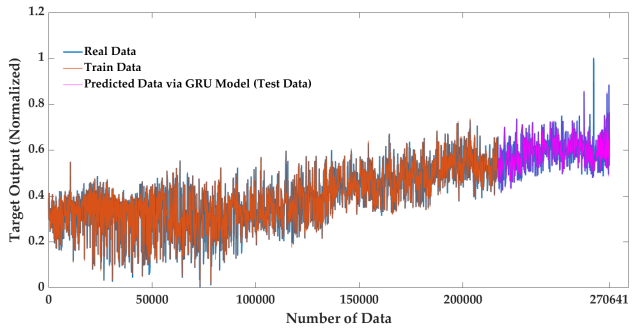


Figure 4.2. GRU model for turbulent flow velocity in y direction with spatial-temporal features.

4.3 Deep Learning Forecasts a Strained Turbulent Flow Velocity Field in Temporal Lagrangian Framework: Comparison of LSTM and GRU

R. Hassanian, Á. Helgadóttir, M. Riedel, 'Deep Learning Forecasts a Strained Turbulent Flow Velocity Field in Temporal Lagrangian Framework: Comparison of LSTM and GRU', in MDPI *Fluids*, vol. 7, no. 11, 2022, DOI: https://doi.org/10.3390/fluids7110344.

*This paper meets the criteria outlined in **TO3** by employing two DL models, namely LSTM and GRU, to analyze velocity prediction in turbulent flows. It systematically compares the performance of these models to fulfill and address **TO3***

In this study, a legitimate dataset was experimentally generated from **TO1** to investigate the predictive capabilities of LSTM and GRU models thus addressing **TO3** regarding the elusive patterns of turbulent flow velocity. Emphasizing the temporal dimension, the model aimed to establish and retain the sequential nature of the data within sequential DL frameworks.

Figure 4.3 and 4.4 depict the model's velocity prediction performance for two flow components using 60% and 80% of the training data, respectively. The MAE = 0.001-–0.002 and the R^2 score is in the range of 0.983–-0.987 for both models. The HPC speedup as a critical requirement to perform these models is also measured.

From a physical perspective, the notable advantage of the proposed models lies in their ability to make predictions without requiring information on turbulence intensity, mean flow rate, or the influence of gravity. This characteristic renders the model applicable across a wide spectrum of industrial and natural contexts, particularly where raw velocity and other flow properties data are typically available.

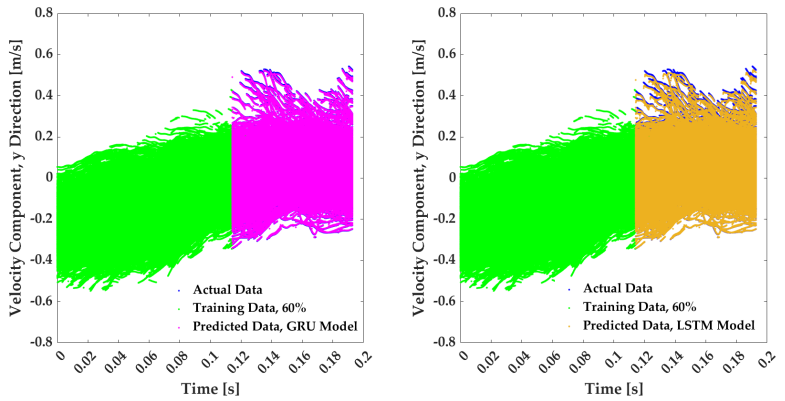


Figure 4.3. Prediction of velocity component in the y direction for a strained turbulent flow with mean strain rate 8 s^{-1}, GRU model on the left-hand side, and LSTM model on the right-hand side. Training data are 60% and test data 40%.

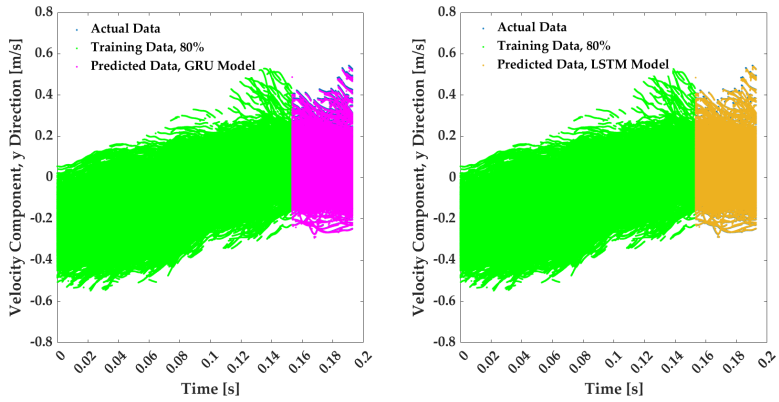


Figure 4.4. Prediction of velocity component in the y direction for a strained turbulent flow with mean strain rate 8 s^{-1}, GRU model on the left-hand side, and LSTM model on the right-hand side. Training data are 80% and test data 20%.

4.4 Deciphering the dynamics of distorted turbulent flows: Lagrangian particle tracking and chaos prediction through transformer-based deep learning models

R. Hassanian, H. Myneni, Á. Helgadóttir, M. Riedel, 'Deciphering the dynamics of distorted turbulent flows: Lagrangian particle tracking and chaos prediction through transformer-based deep learning models', in AIP *Physics of Fluids*, vol. 35, no. 7, 2023, DOI: https://doi.org/10.1063/5.0157897.

*This paper meets the criteria outlined in **TO4** by enhancing an attention mechanism model from the field of fluid dynamics within DL. Utilizing the Transformer model, which represents the cutting-edge architecture in DL, relies on encoder-decoder layers. This pioneering approach within the Lagrangian framework marks a significant advancement.*

This paper proposed a model based on the Transformer model and the definition of the Lagrangian framework relevant to its structure to predict the velocity of a turbulent flow with no known analytical pattern. The model displayed proper prediction capability and performance compared to the previously successful model of the current thesis. Table 4.1 shows the Transformer model performance compared to the LSTM and the GRU models. To apply such models, the HPC resources are required, and for all models, the speedup has been examined.

This paper established groundbreaking knowledge based on the experimental dataset and compared the capability of these three DL models that can be used in a wide range of turbulent flow and fluid dynamics. Despite the fact that all the models show similar results, the Transformer enables a certain interpretability of its inherent learned parameters. Their investigation was not published and is part of future work. Instead, hyperparameter optimization was another track of research performed in the PAPER VI.

Table 4.1. To assess the capability of the Transformer model as a mechanism for attention, a comparison is made between its performance and that of LSTM [43] and GRU [43] from previous studies with similar datasets.

Training ratio	Performance	LSTM	GRU	Transformer
80%	MAE	0.002	0.002	0.002
	R^2 score	0.98	0.98	0.98
	Training time (s)	295	318	301
60%	MAE	0.002	0.002	0.003
	R^2 score	0.98	0.98	0.98
	Training time (s)	214	229	219

4.5 Optimizing Wind Energy Production: Leveraging Deep Learning Models Informed with On-Site Data and Assessing Scalability through HPC

R. Hassanian, A. Shahinfar, Á. Helgadóttir, M. Riedel, 'Optimizing Wind Energy Production: Leveraging Deep Learning Models Informed with On-Site Data and Assessing Scalability through HPC', in *IEEE Hungary Section-Obuda, Acta Polytechnica Hungarica Journal*, vol. 21, no. 9, 2024, DOI: https://doi.org/10.12700/APH.21.9.2024.9.4.

*This paper satisfies the criteria set forth in **TO5** by showcasing the engineering application of fundamental research in developing DL models for turbulent flow. As an illustrative case, these models are applied to forecast wind speed for wind power generation, addressing a significant challenge in power production.*

This study utilizes insights gained from **TO1** through **TO4** to refine DL models for industrial applications. Predicting wind power production is vital, particularly for power producers who must forecast power outputs to meet demand in the grid. These forecasts are typically classified as short-term or long-term predictions.

This paper introduces an innovative approach that leverages historical seasonal data to forecast future periods using a GRU model. Particularly, the model demonstrates strong predictive performance, notably with a dataset featuring 10-minute time steps compared to the traditional 1-hour intervals. In terms of computational resources, the paper recommends employing a local HPC cluster powered by the wind farm itself, thus establishing a sustainable energy source.

The findings highlight the effectiveness of sequential DL in capturing nonlinear and stochastic patterns prevalent in turbulent flow regions, such as wind speed.

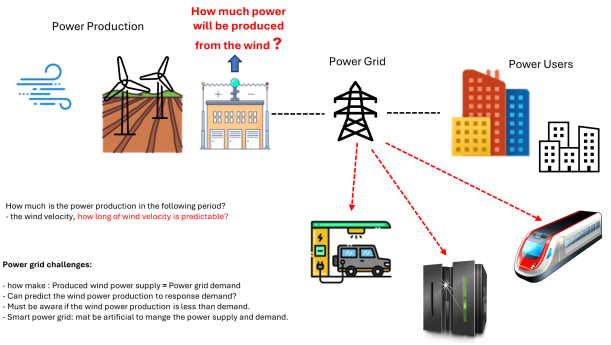


Figure 4.5. A schematic representation of how wind power production prediction could assist the power supplier in managing the response to the power grid demand.

4.6 Turbulent Flow Prediction-Simulation: Strained flow with Initial Isotropic Condition Using a GRU Model Trained by an Experimental Lagrangian Framework, with Emphasis on Hyperparameter Optimization

R. Hassanian, M. Aach, A. Lintermann, Á. Helgadóttir, M. Riedel, 'Turbulent Flow Prediction-Simulation: Strained flow with Initial Isotropic Condition Using a GRU Model Trained by an Experimental Lagrangian Framework, with Emphasis on Hyperparameter Optimization', in MDPI *Fluids*, vol. 9, no. 4, 2024, DOI: https://doi.org/10.3390/fluids9040084.

*This paper meets the criteria outlined in **TO6** by enhancing DL models with spatial and temporal features to analyze their combined impact on target feature predictions. Additionally, it investigates hyperparameter optimization to enhance computational performance and compares it with previous models.*

This paper introduces an enhanced version of the GRU model, incorporating spatial and temporal input data for training, with the target being the velocity of turbulent flow. The study demonstrates that increasing the volume of training data improves prediction accuracy, albeit at the cost of increased computational time. To address this, the research applies hyperparameter tuning, specifically focusing on batch size, to expedite computation.

Table 4.2 illustrates that the improved model, referred to as GRU-h, achieves higher squared R values and more accurate MAE compared to previous models. Additionally, hyperparameter optimization effectively reduces computational time, nearly matching that of earlier models despite utilizing more training data.

Table 4.2. Comparison table of the GRU-h model of the current study that is improved by HPO and trained with larger data and four sequential variable inputs: x, y, V_X, and V_Y. Transformer, LSTM, and GRU, illustrated in the table, are models from previous studies [43, 87], with smaller boundary conditions and two sequential variable inputs V_X and V_Y and without HPO.

Training Proportion	Performance	GRU-h	Transformer	LSTM	GRU
80%	MAE	0.001	0.002	0.001	0.002
	R^2 score	0.99	0.98	0.98	0.98
	Runtime (s)	256	301	295	318

Table 4.3 showcases the results of hyperparameter tuning on the JUWELS-BOOSTER machine, identifying a batch size of 512 as the optimal choice.

In summary, this paper presents an optimized GRU model capable of successfully predicting turbulent flow velocity when incorporating both spatial and temporal features.

Table 4.3. Effect of the size of the batch size on the computing time and the MAE.

Machine Module	GPUs	Batch Size per GPU	Computing Time [s]	MAE
JUWELS-BOOSTER	4	8	14723.30	0.0016698
	4	16	7499.96	0.0015822
	4	32	3757.98	0.0015293
	4	64	1820.90	0.0014718
	4	128	963.49	0.0014551
	4	256	493.07	0.0013771
	4	512	255.93	0.0013613
	4	1024	147.70	0.0014453

5 Conclusions

The objective of this study is to assess the effectiveness of DL models in the field of fluid dynamics, particularly leveraging HPC, which is pivotal for implementing such models. There is a growing interest in employing DL applications in fluid dynamics, especially in tackling complex problems like turbulent flow, which remains unresolved through classical physics. Hence, traditionally, fluid dynamics studies using HPC were thus primarily focussing on numerical methods based on known physical laws, and therefore, this thesis embarks on new studies using cutting-edge data-driven approaches via various DL models. Given the reliability of statistical methods developed over recent decades in turbulent flow studies, this thesis study adopts a hybrid data-driven approach utilizing statistical data analysis. To sum up, the thesis showed that DL models can be successfully applied to fluid flow problems. However, the following paragraphs provide detailed information on the study findings with respect to different TOs.

Initially, experimental data from the author's research was processed for utilization. The experiment produced a broadly observable strained turbulent flow. To study its properties, tracer particles were used to seed the flow, and the LPT technique was employed. Additionally, the flow was seeded separately with inertial particles to investigate the effects of turbulence intensity, gravity, and mean strain rate on the particles. The main findings of this publication are that the newly designed and conducted experiments successfully simulated the flow. The particle behavior within this flow demonstrated the impact of flow distortion on particle dynamics, including velocity root mean square and Reynolds stress. The resulting dataset was analyzed to provide a resource for deep learning model training, addressing the **TO1** requirements, and published.

A pioneering effort was made to establish a framework connecting DL architectures to turbulent flow properties, with a focus on the Lagrangian framework as the foundational definition for flow properties as a function of time. In the Lagrangian framework, the motion of a particle (or point) within the flow is tracked over time, resulting in a time series of flow properties. This thesis specifically focuses on flow velocity. In turbulent flow, the velocity consists of both a mean component and fluctuating components. The experimental data in this thesis capture the total velocity, encompassing both the mean and the fluctuations. The main findings of the publications with respect to **TO2** indicate that the Lagrangian framework in fluid dynamics is effectively interconnected with a sequential deep learning model, making it suitable for use with time series datasets. The **TO2** in novel approach determines turbulent flow properties by linking the Lagrangian framework to sequential DL models.

TO3 aims to embed the established framework of **TO2** and use the data from **TO1** to design a DL model to predict the turbulent flow that has no known pattern. Hence, the study proceeded to successfully develop LSTM and GRU models using HPC resources via varying training and test data ratios. As the publications demonstrate, remarkable results were achieved, demonstrating these models' capability to accurately predict turbulent flow velocity solely based on past velocity components. Since tracer particles were employed in the experiments, certain effects like strain deformations and the influence of gravity were not analytically understood (i.e., using traditional only physics-based approaches and no DL models), making the study's findings significant for the turbulence community by enabling velocity prediction without detailed flow characteristic information.

Recently, Transformer models have become very successful in a wide variety of applications, and thus **TO4** introduced and evaluated the Transformer model for turbulent flow velocity prediction, showcasing superior performance compared to LSTM and GRU models. To the author's knowledge, there have been no similar applications of Transformer models prior to this work. publications in general and the assessments in particular also included evaluating computing resources, which is critical in CFD and DL, in terms of speedup. Hence, achieving **TO4** covers research on the unknown limits of DL models in turbulent flow, mainly the range of Re that current DL models can handle. Also, the research of **TO4** identified several cases in turbulent flow that could benefit from the proposed and validated DL models from this study, which have been published as open access publications to contribute to the turbulent flow research community.

Given the recent momentum in Green Energy in Europe driven by the European Commission, an application in wind energy was explored to illustrate the engineering utility of the proposed approach in **TO5**, specifically in predicting wind power production to meet demand on the power grid. Adapting the model for wind speed prediction enhanced wind power production in wind parks, yielding substantial results. **TO5** was achieved by identifying engineering applications relevant to the thesis objectives that are able to utilize the DL model to predict turbulent flow in diverse contexts of physics and industry, as discussed in other publications listed in this manuscript. Other publications of the thesis have shown results of the application of the proposed DL models in this thesis to various fields, including leading edge erosion, stagnation point, MTF prediction models, wake loss effects for vertical axis wind turbines, and solar energy.

Finally, the PhD study aimed to enhance GRU model performance in turbulent flow prediction by integrating spatial and temporal features based on the Lagrangian framework, which was achieved and published in the context of **TO6**. This improved approach not only increased prediction accuracy but also optimized computing requirements on HPC via HPO. The research of the achieved **TO6** demonstrates that increasing the input variables from the Lagrangian framework enhances the accuracy of turbulent flow predictions. Additionally, to manage the increased computational demand due to the larger training dataset, HPO was effectively employed, resulting in reduced computing time and improved model performance. The findings of the current PhD study have been published in several respected scientific journals, and the supporting data are available from the PhD student and his supervisor upon

reasonable request.

The proposed data-driven approach validated in this study underscores DL models as excellent tools for studying turbulent flow alongside CFD and EFD. Moreover, as data-driven DL models operate on measured data without reliance on theoretical assumptions, they offer a robust capability to forecast turbulent flow properties. This approach's adaptability to measured data, prevalent in various industrial and natural studies, underscores its broad applicability. Hence, the overall findings of this PhD thesis can be re-used in many different application domains.

Future Work

The approach assessed in this study holds promise for examining inertial particles in MTF, a phenomenon lacking analytical understanding. Traditional CFD analyses using large computational application runs on HPC have still not explored the gravitational effects on such particles, highlighting the potential of employing this approach to investigate particle behavior. Data-driven approaches assist the core philosophy of the emerging Hybrid Analysis and Modeling (HAM) paradigm. HAM maximizes the utilization of Physics-Based Model (PBM) and resorts to data-driven models to address unmodeled or unknown physics.

Moreover, there is potential to enhance the scalability of the proposed data-driven model by incorporating larger and more temporally extensive datasets. The author proposes integrating this data-driven model with a physics-based model to explore turbulent flow further, facilitating a deeper analytical understanding of these intricate phenomena. Hence, many findings should be re-used to create so-called Physics-Informed Neural Networks (PINNS) approaches in the future. The idea of these innovative but hard-to-develop DL models is to guide the data-driven learning process with constraints obtained from physics. A promising approach that combines the power of cutting-edge DL models with various established laws of physics. As the thesis TOs are achieved, the author will continue after the PhD to work on these approaches together with the larger physical application community.

Paper I

An Experiment Generates a Specified Mean Strained Rate Turbulent Flow: Dynamics of Particles

R. Hassanian, Á. Helgadóttir, L. Bouhlali, M. Riedel

https://doi.org/10.1063/5.0134306, 2023

Physics of Fluids ARTICLE scitation.org/journal/phf

An experiment generates a specified mean strained rate turbulent flow: Dynamics of particles

Cite as: Phys. Fluids **35**, 015124 (2023); doi: 10.1063/5.0134306
Submitted: 9 November 2022 · Accepted: 14 December 2022 ·
Published Online: 9 January 2023

R. Hassanian,[1,a)] Á. Helgadóttir,[1] L. Bouhlali,[2] and M. Riedel[1,3]

AFFILIATIONS

[1]The Faculty of Industrial Engineering, Mechanical Engineering, and Computer Science, University of Iceland, 102 Reykjavík, Iceland
[2]Reykjavik University, 102 Reykjavik, Iceland
[3]Juelich Supercomputing Centre, 52428 Jülich, Germany

Note: This paper is part of the special topic, Turbulence in Plasmas and Fluids.
[a)]**Author to whom correspondence should be addressed:** seh@hi.is

ABSTRACT

This study aimed to simulate straining turbulent flow empirically, having direct similarities with vast naturally occurring flows and engineering applications. The flow was generated in $100 < Re_\lambda < 500$ and seeded with passive and inertial particles. Lagrangian particle tracking and particle image velocimetry were employed to extract the dynamics of particle statistics and flow features, respectively. The studies for axisymmetric straining turbulent flow reported that the strain rate, flow geometry, and gravity affect particle statistics. To practically investigate mentioned effects in the literature, we present the behavior of both passive and inertial particles from the novel experiment conducted on initially homogeneous turbulence undergoing a sudden axisymmetric expansion. We represent the result with two different mean strains and Reynolds–Taylor microscales. However, this study, in contrast to the previous studies, considers the fields of inertial particles in the presence of gravity. The result discloses that the novel designed and conducted experiments simulated the flow satisfactorily. Then, the particle behavior in such flow showed the effectiveness of the flow distortion on particle dynamics such as velocity root mean square and Reynolds stress. Straining turbulence flow is subject to many industrial applications and physics studies, such as stagnation points, external flow around an airfoil, internal flow in changeable cross section pipe, expansion in the engine mixing chamber, and leading edge erosion. This study's conclusion could apply constructively to these areas.

I. INTRODUCTION

The dynamics of particles in turbulent flow relate to a broad range of natural phenomena and engineering applications.[1] The distribution of pollutants in the atmosphere, the formation of rain in a cloud,[2] and the spread of sediments in oceans and rivers are instances.[3] Many industrial applications are related, including internal engine combustion, particle interaction in a mixing chamber,[4] and leading-edge[5] erosion in compressors and turbines.[6,7] The external flow over the airfoil[8,9] and internal flow in changeable cross-section pipe[10–12] are instances of straining turbulence flow.[13,14] This study's motivation has been from recent works that examined the Lagrangian inertial particle in shear flow[12–14] and boundary layer,[15,16] reporting a profound effect on inertial particle statistics. Later, Lee *et al.*[17] showed that the mean strain rate substantially affected tracer and inertial particle statistics. They reported that for a higher strain rate, acceleration variance increased notably.[17] They also noticed that large-scale motion could affect particle motion, and the heavier inertial particles represent lower acceleration variance than the lighter inertial particles.[17]

Most natural flows combine inertial particles and straining geometries;[18,19] thus, the straining motion consideration is essential. It is well addressed that turbulent flow statistics[1,20] are applicable.[21–23] Due to inertia and interaction, the lighter inertial particle to mean flow has higher acceleration than the tracer particle.[17] When the particle has higher inertia, the ballistic motion will increase the mean flow acceleration.[17] The straining motion appears in the stagnation point region.[17] It has been known that extra fluctuation is derived from mean strain and transmitted to the turbulence.[20,21] Most studies focused on Lagrangian acceleration particle statistics in isotropic turbulence flow, but the recent literature mentioned the vitality of large anisotropic scale Lagrangian dynamics.[17] The small scale may be isotropic, but the large scale is highly anisotropic.[24] Indeed, flow with mean zero flow is generally isotropic, not realistic, and usually close to being

anisotropic.[20] The large-scale structure depends on initial turbulence conditions.[20] Local strain rates in mean flow do not control it although the magnitude of Reynolds stress is related to the immediate history of turbulence which can say the scales have a remote feeling.[20] Statistics of the scales are not a function of local gradients in the mean flow.[20] Moreover, its specified gravity and strain affect the acceleration variance and Stokes number, narrowing the acceleration probability distribution function (p.d.f.s) and skewed with inertia.[17]

The current work designs and suggests a novel procedure to simulate a turbulence flow with a specified strain rate in the laboratory environment. Tracer particles seeded the flow, and the particle image velocimetry (PIV) technique was employed to extract the flow feature. Furthermore, to examine the dynamics of the particles affected by turbulence intensity, mean strain rate, and presence of gravity,[25,26] inertial and tracer particles were seeded the flow in separate implementations, and the Lagrangian particle tracking (LPT) approach was applied to investigate the particle behaviors. The proposed procedure applies to many industrial applications, particularly stagnation point and flow with various geometry. Furthermore, it manifests sight to study the particles in the natural environment and variant flow conditions.

This study presents velocity root mean square (RMS) and normalized Reynolds stress results. In addition to the novel procedure, the examination is conducted in the presence of gravity,[26,27] not considered in the previous numerical studies that have examined the inertial particles' dynamics in the turbulent flow.[23,28] It is well determined that the conclusions achieved in zero gravity should not be applied outside this condition.[27,29] Therefore, it is essential to understand the gravity effects on the dynamics of the particles. Hence, this paper is organized as follows: the applied theory for the dynamics of the particles is presented in Sec. II. The experiment is explained in Sec. III, and the measurement method is explored in Sec. IV. Results and discussion are represented in Sec. V and Sec. VI provides the conclusions.

II. THEORY

In this section, the applied theory in turbulent flow statistics is described. Since the data are extracted from measurement, it is essential to define the characteristics of the flow and particle properties and how the measured data are used to investigate the dynamics of a particle and flow feature.

A. Strained turbulence

The turbulent flow is generated with the action of impeller rotors in the corners of a box turbulence facility. The turbulent flow is then strained in the vertical direction (see Fig. 1) by the motion of flat circular plates, as shown in the sketch. The fluid is seeded with buoyant (tracer) particles and inertial particles with median diameters 8–10 μm and 210–250 μm, respectively. The specific gravity for tracers was 1.1 g/cm^3 (hollow glass) and 2.5 g/cm^3 (solid glass) for inertial particles. The flow field properties are obtained through the particle image velocimetry (PIV), and the Lagrangian particle tracking (LPT) method is employed to study the particles' movement. Equation (1) describes the mean flow field in the facility

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

where $-2S$ is the primary strain rate in the Y-dir; S is the strain rate for the other two directions; and x, y, and z are the particle location. In this work, the flow is considered in two-dimensional (2D); therefore, Z-dir is not addressed.

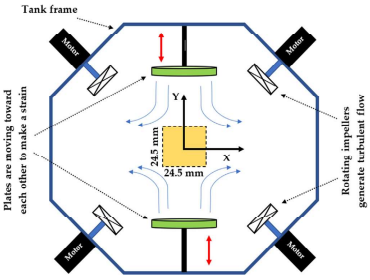


FIG. 1. A sketch of the tank facility and the location of the LPT/PIV measurement area. The viewing window (24.5 × 24.5 mm^2) is located in the center of the water tank. The Y-axis of the coordinate system (CS) points upwards, whereas the X-axis points from left to right. The origin of the CS is located at the center point of the tank.

The straining flow cases in the experiment were created with two mean strain rates, $2S = 4\,\mathrm{s}^{-1}$ and $2S = 8\,\mathrm{s}^{-1}$. Equation (1) is based on the laminar flow; however, we know that velocity fluctuates in the turbulence flow. Further details on the experimental setup are described in Sec. III. Based on the determined coordinate system (CS) in this experiment, the strain rate for both directions has these notations:[1] $dU_y/dy = -2S$ and $dU_x/dx = S$.

B. Turbulent flow characterization

The turbulence flow statistics were used to extract the flow field properties. We analyzed the trajectories of particles in the flow field using the LPT technique. Accordingly, the velocity fluctuations u_x and u_y can be calculated as

$$u_i = U_i - \langle U_i \rangle, \quad i = x, y, \tag{2}$$

where U is the measured (total) velocity, $\langle U \rangle$ is the mean velocity, and the subscript i refers to the component of the velocity, in x, y (or z) direction.

We applied the following equations to obtain the variance of velocity $var(U_i)$, and the velocities RMS $u_{rms,i}$:

$$u_{rms,i} = \sqrt{var(U_i)} = \langle u_i^2 \rangle^{1/2}. \tag{3}$$

Reynolds-stress is a term that stems from momentum transfer by fluctuating velocity field.[1] Reynolds-stresses are the components of a second-order tensor, which is symmetric. The diagonal components $\langle u_i u_i \rangle$ are normal stress, while the off diagonal components $\langle u_i u_j \rangle$ are shear stress (the subscripts i and j denote the components of the velocity).[1] It has been addressed the Reynolds stress gaining from the irrotational field has definitely no effect on the mean velocity field.[1] The normalized Reynolds-stress is calculated for particles in this work to

examine the fluctuation of particles caused by distinct strain rates and turbulence intensity. The definition for the normalized Reynolds-stress is[1]

$$(u_i^2)/2k_0 = (u_i^2)/(u_{0,x}^2 + u_{0,y}^2), \quad i = x, y, \tag{4}$$

where u is the particle velocity in the deformed flow; $u_{0,x}$, $u_{0,y}$ are the velocity; $k_0 = (u_{0,x}^2 + u_{0,y}^2)/2$ is the turbulent kinematic energy of the particle in turbulent flow without strain; and the subscript i denotes the components of the velocity. Since the measurement is carried on in 2D in this work, the velocity has two components in the x and y directions.

The number of points used to compute statistics depends on the type of experiment conducted. The turbulence energy dissipation rate is calculated using a second-order longitudinal velocity structure function $D_{LL}(r)$, assuming inertial subrange $D_{LL}(r) = C_2(\varepsilon r)^{1/2}$ with a universal constant $C_2 = 2.1$,[1] r is the separation and $U_i(x,t)$ is Eulerian velocity field.

$$\begin{aligned} D_{LL}(r) &= D_{ij}(r,x,t) \\ &= \langle [U_i(x+r,t) - U_i(x,t)][U_j(x+r,t) - U_j(x,t)] \rangle. \end{aligned} \tag{5}$$

The mean energy dissipation rate is constant in the inertial subrange, as illustrated in the results. We used the Eulerian autocorrelation function $\rho(L)$ to obtain the Eulerian integral scale l, which can be defined as:[1,30]

$$\rho(L) = \langle u(r_0 + L)u(r_0) \rangle / u^2. \tag{6}$$

The Eulerian integral scale is given by[1,30]

$$l = \int_0^\infty \rho(L) dL. \tag{7}$$

In Eq. (7), the velocity is taken on a grid obtained from the particle image velocimetry (PIV) measurements, and r_0 is taken in both the x and y directions; L is the Eulerian integral variable. Based on Kolmogorov's hypothesis, the length and time scales of the turbulence flow can be calculated using the following equations:

$$\eta = (\nu^3/\varepsilon)^{1/4}, \tag{8}$$

$$\tau_\eta = (\nu/\varepsilon)^{1/2}, \tag{9}$$

where ν is the kinematic viscosity of the fluid, η is Kolmogorov's length scale, τ_η is the Kolmogorov's timescale, and ε is the energy dissipation rate which is evaluated via described second-order longitudinal velocity structure function.

C. Stokes number

In this study, we employed LPT to track the particle in the generated flow. Therefore, the Stokes number must be considered for both particle types. The Stokes number specifies whether a particle introduced to the flow will follow the flow streamline or not. This identification is defined by the following equation:

$$S_t = \tau_p/\tau_\eta, \tag{10}$$

where τ_p is Stokes' relaxation time. Kolmogorov scale is based on the flow quantities before applying the strain, and these amounts are calculated for two-particle types and presented in Sec. IV. Stokes' relaxation time τ_p is, in turn, calculated by the following equation:

$$\tau_p = \rho_p d_p^2/18\mu, \tag{11}$$

where ρ_p is the particle density, d_p is a spherical particle diameter, and μ is the dynamic fluid viscosity that, in this experiment, is water. Both the Stokes' relaxation time and the Kolmogorov timescale are required in Eq. (9). The relaxation time for each particle is calculated based on the particle property from Eq. (10). The Stokes number significantly greater than 1 ($St \gg 1$) describes particles that are unaffected by a fluid velocity change and continue their original trajectory; if ($St \ll 1$), the particle will follow the fluid's local velocity.

III. EXPERIMENT

The flow facility at the Laboratory for Fundamental Turbulence Research (LFTR) at Reykjavik University is shown in Fig. 2, where the experiments were carried out in a zero-mean turbulence box (with the corners cut off to facilitate impellers that force the flow). The tank is specifically designed for studying turbulence (Lagrangian and Eulerian motion at moderate Reynolds numbers). This flow facility produces a nearly stationary homogeneous isotropic turbulence near the center of the tank, where measurements are performed. The water tank ($60 \times 60 \times 60$ cm^3) has 20 mm thick acrylic walls (transparent Plexiglas XT) that enable optical access to the data. The eight corners of the box have a triangular shape, while the top and the bottom are nearly circular. An aluminum frame holds the components of the turbulence box together. The turbulence is generated by eight impellers driven by independently controlled servo motors (Lenz-model: MCS), which are mounted at the cube's eight corners and point to the center of the tank. The rotation rate of each servo motor is adjustable over a range of 100–4500 revolutions per minute (rpm) at a gearing rate of 0.075. The motion-view filtering software that came with these motors was used to monitor and set up the suited speed of each impeller. The degassing system was used to remove bubbles and solid dust from the water before starting the experiment.

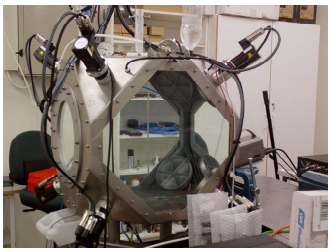


FIG. 2. A photo of the turbulence box taken at the Laboratory for Fundamental Turbulence Research (LFTR) at Reykjavik University.[31,32] Reproduced with permission from Hassanian, M.S. dissertation (Reykjavik University, 2020). Copyright 2020 Author.

Circular plates, shown at the top and bottom (see Fig. 1), generate the strain motions; a linear actuator drove each plate with a piston rod (Parker, model: ETV32M10PA47JMA400A). When they are moved toward the center with a pre-described rate that ensures a nearly constant strain rate in the fluid. For each flow case, variables are investigated with different propeller rotation rates, strain rates, and particle types; a total of 20 piston movements is recorded to collect a sufficient number of particle trajectories. Each video is statistically independent, as the flow is given ample time to recover to near isotropy between different strokes. The recording area is $24.5 \times 24.5\ \text{mm}^2$ and is located in the center of the tank for LPT/PIV measurements (Fig. 1). The flow case in this study depends on the size of the particles and the rotation speed of the impeller in the tank and is applied at 1000 and 1500 rpm. Two circular plates described previously created two mean strain rates: $2S = 4$ and $2S = 8\ \text{s}^{-1}$. The plate driver motor moves according to an exponential profile, and the same as a particle would move by if it were moving along the y-axis toward the center of the coordinate system. The straining turbulence flow was generated for two kinds of particles: a passive and an inertial particle. Each particle has two nominal strain rates and two impeller rotation speeds as the numerical variations of the turbulence factor. The detection system for LPT was set at 10 kHz [10 000 frames per second (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, which is determined in Sec. V; therefore, the properties of the energy dissipation rate range in the flow are solved. For each flow case, the average of the measurements was done over 20 videos in the same instant time, and one averaged data file was created. The average file was used to obtain the flow field properties. The software includes several stages. First, it converts the image to a readable file, processes it to achieve the necessary data, and follows the post-processing stages.

IV. MEASUREMENT TECHNIQUE

A. Particle image velocimetry technique

The particle image velocimetry (PIV) technique is applied and extracts the flow properties before strain deformation. Thus, via a second-order longitudinal velocimetry structure function, the turbulent flow dissipation rate is obtained; therefore, the Kolmogorov timescale is calculated based on Eq. (9) and Stokes number obtained from Eq. (10). In this study, the PIV has been applied independently of the LPT experiment because the number of seeded particles in PIV is much higher. The PIV measurement is applied to extract the flow characteristics. A single high-speed CMOS camera with a 105 mm focal length lens was used to record the PIV images, and it was set at 512×512 pixels resolution. For the illumination of the tracer particles, an Nd-YAF laser (527 nm) was used and synchronized at the same sampling frequency as the camera. The laser was set at internal mode, 14 A Q-switch current, and pulse width of 2.5 μs. The laser beam is expanded to a sheet along the (X, Y)-plane using spherical and cylindrical lenses. The sample area has a size of the frame and thickness equal to the thickness of the laser sheet, which is 2 mm. The PIV configuration 2C-2D is applied (see Fig. 3), which delivers two velocity components in the x and the y directions.[33] In this configuration, one

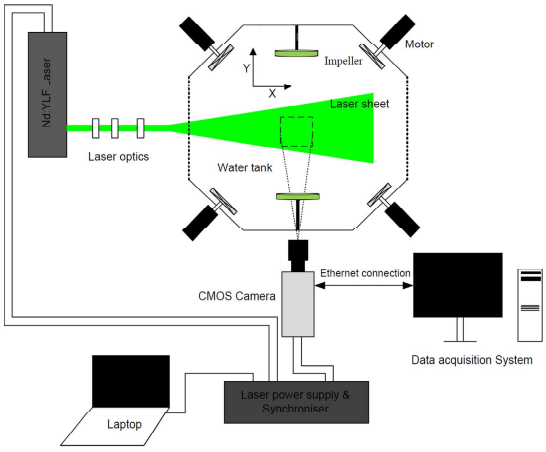


FIG. 3. A view of the experiment set up to record the PIV/LPT frames.

camera is used, and its position is perpendicular to the laser light sheet. The displacement of the particles is estimated using cross correlation and fast-Fourier transformers. For evaluation of recorded images, the digital PIV recording is divided into small subareas called interrogation areas (IA).[33] It is assumed that all particles within one IA have moved homogeneously between two illuminations. In order to obtain a valid velocity measurement, a spot should contain anywhere from seven to ten particle pairs, where one pair refers to a particle imaged at both t and t'.[33] In the literature, it is noted it is advantageous to offset the second IA according to the mean displacement of the tracer particles between the two illuminations,[34] which is performed. Regarding the particle seeding in the flow, ideally, the particle should have the same density as the flow, which is respected in this study by the tracer particle. Another important factor in the measurements' accuracy is the seeding's homogeneity. The applied PIV recording technique is based on the method which provides a single illuminated image for each illumination pulse (multi-frame/single exposure PIV) since the used CMOS camera and laser are internally synchronized at the same frequency rate. In the current study, the PIV recording system is able to employ the double-frame/single-exposure recording technique. The measurement area of size 512 × 512 pixels ($24.5 \times 24.5\,\text{mm}^2$) was divided into interrogation areas (IA) of size 64 × 64 pixels. To compute the structure functions from PIV, the IA has a size of 64 × 64 pixels with an overlap of 50% to decrease the computing time. This yielded 15 × 15 velocity vectors with a spacing grid of 32 pixels. The local displacement vector for the images of the tracer particles of the first and second illumination is determined for each IA utilizing statistical methods (cross correlation). The projection of the vector of the local flow velocity into the plane of the light sheet (two-component velocity vector) is calculated respecting the time delay between the two illuminations and the magnification at imaging. The interrogation process is repeated for all IA's of the PIV recording and was delivered a complete velocity field. The PIV was conducted with similar turbulence intensity to LPT data and has controlled by a servomotor at speeds of 1000 and 1500 rpm without an act of strain. The frame rate set up was 2 kHz [2000 frames per second (fps)], which gives an exposure time (time interval between two successive frames) of 500 μs (0.5 ms). The recorded videos were 20 for each experiment, where each video has 1000 frames; this gives 20 000 instantaneously recorded images. This work applies a fast-Fourier transformer, a strong tool to accelerate the correlation process in digital PIV. The measurement represents the the Taylor microscale Reynolds number[1,21] range is 100–500 in the performed experiment.

B. Lagrangian particle tracking measurements

The Lagrangian particle tracking (LPT) measurements were carried out for two flow cases in this work in the presence of the strain. The camera and laser setup for LPT measurement are as explained in the PIV measurement. LPT uses a lower seeding density but finds individual and longer particle tracks. One camera was used in the LPT system, enabling the reconstruction of particle tracks in two dimensions.

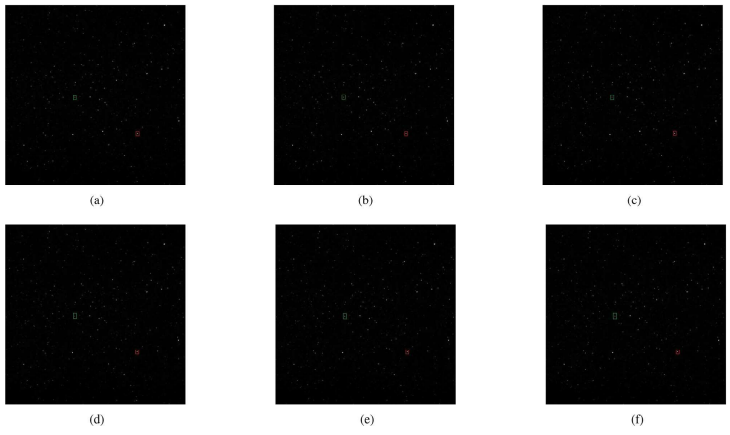


FIG. 4. A representation of six sequential frames of the recording via a high-speed camera in the LPT technique: (a) frame 1 at t s, (b) frame 2 at $t + 0.0001$ s, (c) frame 3 at $t + 0.0002$ s, (d) frame 4 at $t + 0.0003$ s, (e) frame 5 at $t + 0.0004$ s, and (f) frame 6 at $t + 0.0005$ s (t is an arbitrary moment). Each frame has been taken in 0.0001 s. The green and red boxes display the locations of two different particles in each time frame.

These particle tracks are used to calculate the dynamics of particles of initial turbulence, including the Lagrangian statistics. Figure 4 illustrates LPT frames recorded and how particles are seen and could be tracked. The particles' center detection (identification) method used in this LPT technique is called the 2D Gaussian fitting; its principle is based on the idea that a Gaussian can approximate particles' intensity profile, and therefore, the Gaussian function can be used to fit this intensity profile of the particle image. For each particle pixel group, two one-dimensional Gaussian fits[35] are used: one Gaussian will determine the horizontal position of the particle, and the second will determine the vertical position and, hence, solving the two system of equations:

$$I_i = \frac{I_0}{\sqrt{2\pi}\sigma}\exp\left[-\frac{1}{2}\left(\frac{x_i - x_c}{\sigma}\right)^2\right], \tag{12}$$

$$I_i = \frac{I_0}{\sqrt{2\pi}\sigma}\exp\left[-\frac{1}{2}\left(\frac{y_i - y_c}{\sigma}\right)^2\right], \tag{13}$$

for $i = 1, 2, 3$, it leads to particle center coordinates (x_c, y_c). Notation (x_i, y_i) are the coordinates of the local maximum pixel intensity I_2 and the two neighbor's pixel intensities I_1 and I_3, σ is the width of the distribution that is assumed equally in x and y coordinates, and I is the pixel intensity. Notice that the captured images are dynamically thresholded before starting the process of detecting the particles. It is assumed that every local maximum in intensity above a threshold represents a particle. Once particle centers on each image are detected, the next step is reconstructing their 2D tracks from successive images by matching each particle's position in the first image with its corresponding position in the second image.

The four-frame particle tracking technique used in this work, which is called four-frame Best Estimate,[36] is an extension of the four-frame Minimum Change in Acceleration heuristic (4MA);[36] it tracks a particle path through four sequential flow images (see Fig. 5) by a prediction which is based on a given heuristic and cost function. This method proceeds in two steps. First, the center of a search region is calculated using a constraint on velocity or acceleration. Then, all particles within the search region are tested using a cost function that has to be minimized. Let x_i^n denotes the ith position in the nth frame. A tracking algorithm, then, tries to find an x_j^{n+1} for each x_i^n such that x_j is the position of the particle in the frame $[n+1]$ that was at position x_i in the frame $[n]$. The most common is restricting the number of frames over which the cost-tracking function is optimized. The second approximation is to restrict the x_j^{n+1} investigated as possible matches for each x_i^n by imposing a limit on the distance a particle can travel from one frame to the next. Within these approximations, a tracking algorithm is specified by two parameters: the heuristic used to calculate the cost function and the method used to break tracking conflicts. In this process, if more than one track shares the same particle, the incorrect tracks are discarded. The position of the particle in the frame $[n+1]$ is estimated by using the three-frame Minimum Acceleration heuristic (3MA)[36] which is given by

$$x_i^{n+1} = x_i^n + U_i^n \Delta t = 2x_i^n - x_i^{n-1}, \tag{14}$$

where U_i^n is the estimated velocity and Δt is the time elapsed between frames. For each of the particles in the search volume search region: $S1$ in the frame $[n+1]$, a position x_i^{n+2} in the frame $[n+2]$ is estimated to be[36]

$$x_i^{n+2} = x_i^n + U_i^n 2\Delta t + a_i^n \Delta t^2, \tag{15}$$

where a_i^n is the estimated particle acceleration. Particles in a search region: $S2$ around it are investigated. The particle chosen as the best match in the frame $[n+2]$ is the one that has the smallest distance between particles in the frame $[n+2]$ and the second estimated position, and hence, the particle that is linked here in the previous frame $[n+1]$ is chosen as the best match too. The cost function ϕ_{ij}^n applied in this case is given by[36]

$$\phi_{ij}^n = ||x_j^{n+2} - x_i^{n+2}||, \tag{16}$$

where x_i^{n+2} is the estimated position in the frame $[n+2]$. The simplest way to handle conflicts is to give up: when a particle in the frame $[n+1]$ is the best match for multiple particles in the frame [n], the

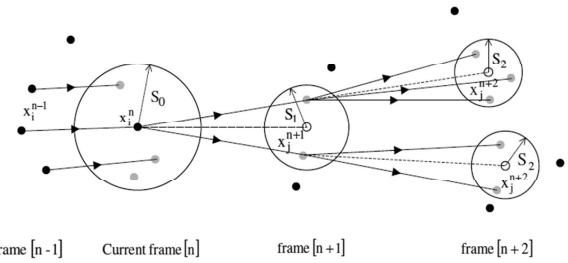


FIG. 5. Schematic diagram of the conventional four-frame PTV tracking algorithm. The black circles are the particles positioned outside of the search region, while the light gray circles are the particles positioned inside the search region, and the white circles are the predicted positions of the particles in the next frame. The arrows that link the frames between them are the possible particle path.

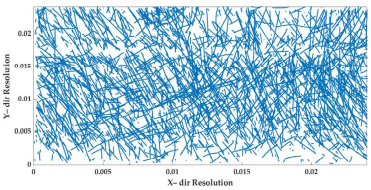


FIG. 6. Real view of the path traveled by the particles, obtained from the video recordings for a data set that included 4000 images (resolution of 512 × 512 pixels), obtained from one of the 20 individual and independent videos observing the same experimental condition.

involved tracks are stopped at the frame $[n]$, and a new track is considered to have begun in the frame $[n+1]$. Note that our LPT code for identifying and tracking the tracer particles is based on the work of Ouelette *et al.*[36]

V. RESULTS AND DISCUSSION

A. Turbulent flow quantities

Figure 6 presents real tracks from a run in an LPT experiment performed in the described facility. Turbulent flow quantities in the experiments are initially extracted to base the subsequent comparison on; this includes the energy dissipation rate, Kolmogorov length and timescale, and the turbulent Reynolds–Taylor microscale. PIV is applied for this purpose to the turbulent flow prior to the application of strain in a nearly stationary homogenous turbulent box flow. Figure 7 shows the compensated longitudinal structure function $D_{LL}(r)$ in terms of the separation r to evaluate the dissipation ϵ. Two cases are displayed based on the two turbulence intensities produced by the two motor speeds of the impellers driving the turbulent flow. In the inertial subrange, the compensated structure functions are nearly constant, allowing us to determine the dissipation rate ε in the flow (we use r between 60η and $l/6$ according to Ref. 1).

Table I summarizes the different cases under investigation, which are controlled by the rotation speed of the impellers in the experimental facility. Once the dissipation and velocity RMS are measured, the remaining quantities are known.

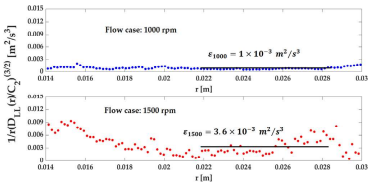


FIG. 7. Compensated second-order longitudinal structure functions $D_{LL}(r)$ via Eq. (4) for the flow cases $Re_{\lambda 1} = 110$; 1000 rpm and $Re_{\lambda 2} = 152$; 1500 rpm. The black lines indicate the expected value of the dissipation rate in each case. ε is the energy dissipation rate, $C_2 = 2.1$ is the universal constant, and r is the separation.

TABLE I. Flow parameters obtained from PIV measurements for passive particles. ε, turbulence energy dissipation rate; ν, kinematic viscosity; l, large eddy length scale (integral scale); u_{rms}, velocity root mean square; η, Kolmogorov length scale; τ_η, Kolmogorov timescale; λ, Taylor microscale; Re_l, Reynolds number based on the length scale of the large eddies present in the flow; Re_λ, Reynolds–Taylor microscale.

Flow quantities	Flow case 1000 rpm	Flow case 1500 rpm
$\varepsilon\,(\mathrm{m}^2/\mathrm{s}^3)$	1.0×10^{-3}	3.6×10^{-3}
$\nu(\mathrm{m}^2/\mathrm{s})$	1×10^{-6}	1×10^{-6}
$l\,(\mathrm{cm})$	5.70	7.40
$u_{rms}\,(\mathrm{m/s})$	0.026	0.047
$\eta = (\nu^3/\varepsilon)^{1/4}\,(\mathrm{mm})$	0.177	0.129
$\tau_\eta = (\nu/\varepsilon)^{1/2}\,(\mathrm{ms})$	31.6	16.6
$\lambda = (15\nu/\varepsilon)^{1/2}\,(\mathrm{mm})$	3.8	3.2
$Re_l = u_{rms}l/\nu$	1485	3478
$Re_\lambda = u_{rms}\lambda/\nu$	110	152

B. Particle characterization

Table II summarizes the properties of the particles seeded in the flow. The Stokes numbers are calculated based on Eq. (9), e.g., the ratio of the particle response and the timescale of the flow presented. It must be noted that the particle Stokes numbers are based on the Kolmogorov timescale of the flow before the straining motion is applied to the turbulent field. Although it is expected that the timescale in the flow changes during the deforming motion, observations are assumed on the Stokes number's initial value. In addition, it is expected that the short duration of the straining motion will not affect the smallest scales of motion in the turbulent flow field to a high degree. The Stokes number for tracer and inertial particles are in the range of 0.006 32–0.018 07 and 0.113–0.307, respectively.

C. Mean strain motion

In this study, the experiment is conducted with a novel approach to simulate a strain flow with a specified mean strain rate. Figure 8 indicates the mean strain measurement for $2S = 4\,\mathrm{s}^{-1}$ for both flow cases. The results show that the turbulence flow reaches the mean

TABLE II. Stokes number for the two types of particles applied in the experiments. Buoyant and Inertial particles with several flow cases.

Particle	Size	Flow case $Re_{\lambda 1} = 110$; 1000 rpm $\tau_\eta = 31.6$ ms	Flow case $Re_{\lambda 2} = 152$; 1500 rpm $\tau_\eta = 16.6ms$
Type	$d_p(\mu\mathrm{m})$	Stokes number	Stokes number
Tracer	(Min.) 8	6.32×10^{-3}	12.04×10^{-3}
	(Max.) 10	9.49×10^{-3}	18.07×10^{-3}
Inertial	(Min.) 210	0.113	0.216
	(Max.) 250	0.161	0.307

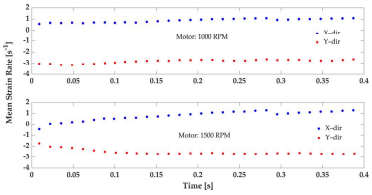


FIG. 8. A presentation of the mean strain rate is measured for the flow during the straining generation for tracer particles. The strain is conducted in Y-dir with $-4\,s^{-1}$ and in X-dir $2\,s^{-1}$. The filled blue circle is for the X-dir, and the filled red square is for Y-dir. The above drawing is for flow case $Re_{\beta 1} = 110$; 1000 rpm, and down curves are for flow case $Re_{\beta 2} = 152$; 1500 rpm.

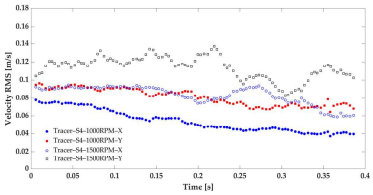


FIG. 10. The velocity RMS for the straining turbulence flow with $2S = 4\,s^{-1}$ and flow cases $Re_{\beta 1} = 110$; 1000 rpm and $Re_{\beta 2} = 152$; 1500 rpm for X and Y components for tracer particles. The filled blue circle, the empty blue circle, the filled red square, and the empty black square represent X-dir; flow case $Re_{\beta 1}$, X-dir, flow case $Re_{\beta 2}$, Y-dir; flow case $Re_{\beta 1}$ and Y-dir, flow case $Re_{\beta 2}$, respectively.

strain rate during strain generation in the main direction y as $-4\,s^{-1}$ and the x direction is $2\,s^{-1}$ based on Eq. (1) for mean strain. Figure 9, in the same manner, dictates the mean strain measurement for $2S = 8\,s^{-1}$ for both flow cases. The mean strain rate for $2S = 4\,s^{-1}$ was achieved approximately with a 15% difference from the theoretical strain rate, but for $2S = 8\,s^{-1}$, the measured mean strain rate reached the theoretical rate accurately. The reason for this distinction is not apparent, but it seems that for a lower strain rate, the tracer particles need more time to attain the theoretical rate because of the large-scale eddies effects and rate of energy transfer. The literature notes that the external motion and energy source moving on the flow, such as strain or shear, could drive instability in the flow.[1,17,20] It could be another reason to see a less achieved strain rate for the case flow with $2S = 4\,s^{-1}$. Results for the strain measurement have been earned based on the tracer particles. These measurements show that the experiment simulated the strain motion with the desired outcomes, matching the strain motion behavior quantitatively and qualitatively.

D. Velocity root mean square (RMS)

Figures 10 and 11 show that for the tracer particles with equal turbulence intensity, the velocity RMS component in the y-direction is more extensive than in the x-direction, as the strain motion in the y-direction has a double mean strain rate than the other direction. However, for tracers with a similar mean strain rate, the flow case with intense turbulence has higher velocity RMS and amplitude fluctuation.

In the literature, it has been noticed that the strain motion could generate extra oscillation and increase the fluctuation.[20,21] Figures 10 and 11 illustrate that tracer particles with the same direction and equal turbulence intensity have gained intensive fluctuation with a higher mean strain rate. Moreover, since all flow cases approximately had the same initial condition before the strain, the results showed the velocity RMS is not only related to the initial condition but also affected by the history of the turbulence.[20]

Figures 12 and 13 display that strain motion has affected the inertial particle, and velocity RMS has fluctuated much more. The inertial particle has more substantial oscillations than tracers for cases with analogous mean strain rate and turbulence intensity. Figures 12 and 13 represent that the inertial particles in the flow with higher turbulence intensity are much more sensitive to the strain motion and have profound effects than lower turbulence flow. This issue could be caused by scale distinction for flow with different turbulence intensities. In the previous studies, it has been noted that the large scale could

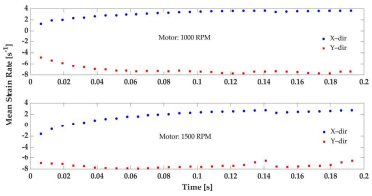


FIG. 9. A presentation of the mean strain rate is measured for the flow during the straining generation for tracer particles. The strain is conducted in Y-dir with $-8\,s^{-1}$ and in X-dir $4\,s^{-1}$. The filled blue circle is for the X-dir, and the filled red square is for Y-dir. The above drawing is for flow case $Re_{\beta 1} = 110$; 1000 rpm, and the down curves are for flow case $Re_{\beta 2} = 152$; 1500 rpm.

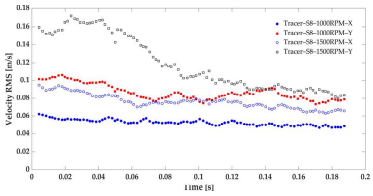


FIG. 11. The velocity RMS for the straining turbulence flow with $2S = 8\,s^{-1}$ and flow cases $Re_{\beta 1} = 110$; 1000 rpm and $Re_{\beta 2} = 152$; 1500 rpm for X and Y components for tracer particles. The filled blue circle, the empty blue circle, the filled red square, and the empty black square represent for X-dir; flow case $Re_{\beta 1}$, X-dir, flow case $Re_{\beta 2}$, Y-dir; flow case $Re_{\beta 1}$ and Y-dir, flow case $Re_{\beta 2}$, respectively.

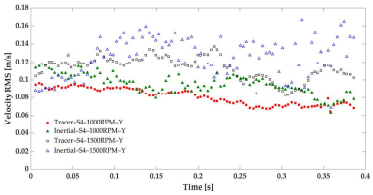


FIG. 12. The velocity RMS for the straining turbulence flow with $2S = 4\,s^{-1}$ and flow cases $Re_{\lambda 1} = 110$; 1000 rpm and $Re_{\lambda 2} = 152$; 1500 rpm for Y component for tracer and inertial particles. The filled red square, the empty black square, the filled green triangle, and the empty blue triangle represent tracer; Y-dir; flow case $Re_{\lambda 1}$, tracer; Y-dir, flow case $Re_{\lambda 2}$, inertial; Y-dir; flow case $Re_{\lambda 1}$ and inertial; Y-dir, flow case $Re_{\lambda 2}$, respectively.

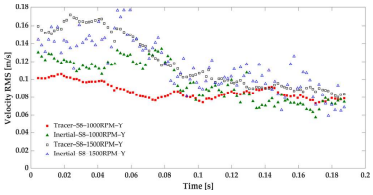


FIG. 13. The velocity RMS for the straining turbulence flow with $2S = 8\,s^{-1}$ and flow cases $Re_{\lambda 1} = 110$; 1000 rpm and $Re_{\lambda 2} = 152$; 1500 rpm for Y component for tracer and inertial particles. The filled red square, the empty black square, the filled green triangle, and the empty blue triangle represent tracer; Y-dir; flow case $Re_{\lambda 1}$, tracer; Y-dir, flow case $Re_{\lambda 2}$, inertial; Y-dir; flow case $Re_{\lambda 1}$ and inertial; Y-dir, flow case $Re_{\lambda 2}$, respectively.

noticeably affect particle motion.[17,20] However, because the inertial particle is denser than the tracer, the gravity effect could be considered the second effective parameter in addition to strain motion.

E. Normalized Reynolds-stress in the strained turbulent flow

As described in the theory section, the Reynolds-stress is an appropriate term to assess the fluctuation effects. Figures 14–17 show the normalized Reynolds-stress and determine that the denser particle gains an intensive effect from the strain motion than the tracer in the same condition. Moreover, the inertial particles with similar straining turbulence flow have less sensitivity to the strain motion in flow with higher turbulence intensity. It results in the same manner for tracer and inertial particles and must be taken as a specific issue regarding the scale size for distinct turbulence flow. However, we must notice that the gravity effect for the inertial particle could increase the strain motion impact. In this practical study, gravity is a parameter not

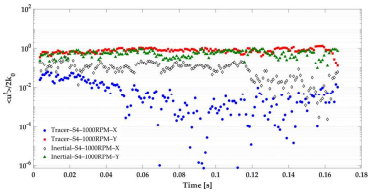


FIG. 14. The normalized Reynolds stress for the straining turbulence flow with $2S = 4\,s^{-1}$ and flow case $Re_{\lambda 1} = 110$; 1000 rpm for Y component for tracer and inertial particles. The red-filled square, the filled blue circle, the filled green triangle, and the empty black diamond represent the normalized Reynolds stress in the Y direction for the tracer, in the X direction for the tracer, the Y direction for the inertial particle, and in the X direction for the inertial particle, respectively.

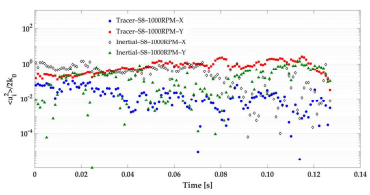


FIG. 15. The normalized Reynolds stress for the straining turbulence flow with $2S = 8\,s^{-1}$ and flow case $Re_{\lambda 1} = 110$; 1000 rpm for Y component for tracer and inertial particles. The red-filled square, the filled blue circle, the filled green triangle, and the empty black diamond represent the normalized Reynolds stress in the Y direction for the tracer, in the X direction for the tracer, the Y direction for the inertial particle, and in the X direction for the inertial particle, respectively.

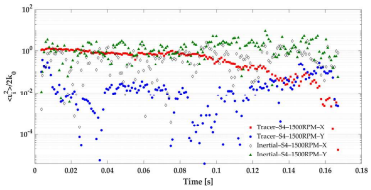


FIG. 16. The normalized Reynolds stress for the straining turbulence flow with $2S = 4\,s^{-1}$ and flow case $Re_{\lambda 2} = 152$; 1500 rpm for Y component for tracer and inertial particles. The red-filled square, the filled blue circle, the filled green triangle, and the empty black diamond represent the normalized Reynolds stress in the X direction for the tracer, in the Y direction for the tracer, the Y direction for the inertial particle, and in the X direction for the inertial particle, respectively.

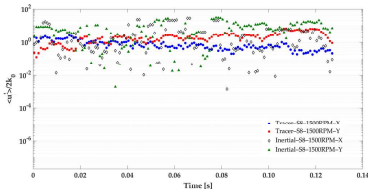


FIG. 17. The normalized Reynolds stress for the straining turbulence flow with $2S = 8\,s^{-1}$ and flow case $Re_{\beta 2} = 152$; 1500 rpm for Y component for tracer and inertial particles. The red-filled square, the filled blue circle, the filled green triangle, and the empty black diamond represent the normalized Reynolds stress in the Y direction for the tracer, in the X direction for the tracer, the Y direction for the inertial particle, and in the X direction for the inertial particle, respectively.

considered in previous numerical works, and the observation in this study appears that gravity affects inertial particles' behavior more than tracer particles.

VI. CONCLUSIONS

First, this study aims to design a novel experimental procedure to generate a straining deformation in turbulence flow, and second, to investigate the Lagrangian dynamics of tracer and inertial particles affected by flow distortion practically in the presence of gravity. The Lagrangian particle tracking method and particle image velocimetry approach were employed to extract particle statistics and flow characterizations from the videos recorded by a high-speed camera, respectively. In the first step, the outcome of strain motion measurements showed that the designed experiment matched the desired state with remarkable results. The measured strain rate intensively corresponds to the theoretical mean strain rate and can be used to create a straining turbulence flow with a different mean strain rate in this approach. The considerable observation is related to the sensitivity of the conducted strain rate. The results depict that the high mean strain rate made its effect faster than the lower mean strain rate on the flow with similar turbulence intensity. This behavior could be related to large-scale interactions and how energy transfers to small scales. In the next step, post-processing on the particle statistics presents us with velocity RMS and Reynolds-stress. The results illustrate that the geometry of the flow or distortion affects the inertial particles more extensively than the tracer particles. The inertial particles in the flow with extreme turbulence intensity have less sensitivity to distortion. It seems the presence of gravity causes distortion effects to be increased on the dynamics of the inertial particles. The results of this study can be listed as follows:

- The strain motion has been affected by both tracer and inertial particles.
- Inertial particles have a more intensive impact from the strain motion than tracer particles.
- For two flow cases with distinct turbulence intensities, there are observations regarding scale size, leading to some effects on particle motion.
- The results illustrate that the strain motion eventually generates extra fluctuation, increasing the fluctuation amplitude increased than without strain.
- The Reynolds-stress results represent that they are affected by the initial condition, strain motion, and history of the turbulence.
- Although the inertial particle is heavier than the tracer, it could be effective by gravity added to the strain motion impact for these types of particles.

ACKNOWLEDGMENTS

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 Program under Grant Agreement Nos. 951733 and 951740, respectively. We acknowledge the support from Rannis, The Icelandic Centre for Research in this work. We thank Professor Ármann Gylfason from Reykjavik University for his technical comments on the experiment conducted at the Laboratory of Fundamental Turbulence Research (LFTR) at Reykjavik University.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Reza Hassanian: Conceptualization (equal); Investigation (equal); Methodology (equal); Software (equal); Validation (equal); Visualization (equal); Writing – original draft (equal). **Ásdís Helgadóttir:** Supervision (equal); Writing – review & editing (equal). **Lahcen Bouhlali:** Formal analysis (equal); Methodology (equal); Software (equal). **Morris Riedel:** Funding acquisition (equal); Supervision (equal); Writing – review & editing (equal).

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

[1]S. B. Pope, *Turbulent Flows* (Cambridge University Press, 2000).
[2]R. A. Shaw, "Particle-turbulence interactions in atmospheric clouds," Annu. Rev. Fluid Mech. **35**, 183–227 (2003).
[3]F. Toschi and E. Bodenschatz, "Lagrangian properties of particles in turbulence," Annu. Rev. Fluid Mech. **41**, 375–404 (2009).
[4]G. Yeoh, J. J. Chen, and C. H. Chen, "Investigation of swirling flows in mixing chambers," Modell. Simul. Eng. **2011**, 259401.
[5]M. H. Arabnejad, A. Amini, M. Farhat, and R. E. Bensow, "Numerical and experimental investigation of shedding mechanisms from leading-edge cavitation," Int. J. Multiphase Flow **119**, 123–143 (2019).
[6]T. Tanuma, *Advances in Steam Turbines for Modern Power Plants* (Woodhead Publishing, 2017).
[7]B. Taherkhani, A. P. Anaraki, J. Kadkhodapour, N. K. Farahani, and H. Tu, "Erosion due to solid particle impact on the turbine blade: Experiment and simulation," J. Failure Anal. Prev. **19**, 1739–1744 (2019).
[8]G. K. Batchelor, *The Theory of Homogeneous Turbulence* (Cambridge University Press, 1982).

[9]J. C. R. Hunt, "A theory of turbulent flow round two-dimensional bluff bodies," J. Fluid Mech. **61**, 625–706 (1973).
[10]Z. Warhaft, "An experimental study of the effect of uniform strain on thermal fluctuations in grid-generated turbulence," J. Fluid Mech. **99**, 545–573 (1980).
[11]J. C. R. Hunt and D. J. Carruthers, "Rapid distortion theory and the 'problems' of turbulence," J. Fluid Mech. **212**, 497–532 (1990).
[12]S. Ayyalasomayajula and Z. Warhaft, "Nonlinear interactions in strained axisymmetric high-Reynolds-number turbulence," J. Fluid Mech. **566**, 273–307 (2006).
[13]J. Chen, C. Meneveau, and J. Katz, "Scale interactions of turbulence subjected to a straining-relaxation-destraining cycle," J. Fluid Mech. **562**, 123–150 (2006).
[14]P. Gualtieri and C. Meneveau, "Direct numerical simulations of turbulence subjected to a straining and destraining cycle," Phys. Fluids **22**, 065104 (2010).
[15]L. J. Baker and F. Coletti, "Particle-fluid-wall interaction of inertial spherical particles in a turbulent boundary layer," J. Fluid Mech. **908**, A39 (2021).
[16]W. Xiao, T. Jin, K. Luo, Q. Dai, and J. Fan, "Eulerian-Lagrangian direct numerical simulation of preferential accumulation of inertial particles in a compressible turbulent boundary layer," J. Fluid Mech. **903**, A19 (2020).
[17]C.-M. Lee, Á. Gylfason, P. Perlekar, and F. Toschi, "Inertial particle acceleration in strained turbulence," J. Fluid Mech. **785**, 31–53 (2015).
[18]A. Klein, "Characteristics of combustor diffusers," Prog. Aerosp. Sci. **31**, 171–271 (1995).
[19]Z. Han and R. D. Reitz, "Turbulence modeling of internal combustion engines using RNG κ-ε models," Combust. Sci. Technol. **106**, 267–295 (1995).
[20]P. A. Davidson, *Turbulence: An Introduction for Scientists and Engineers* (Oxford University Press, 2004).
[21]H. T. John and L. Lumley, *A First Course in Turbulence* (MIT Press, 1972).
[22]R. Hassanian, s Helgadóttir, and M. Riedel, "Deep learning forecasts a strained turbulent flow velocity field in temporal Lagrangian framework: Comparison of LSTM and GRU," Fluids **7**, 344 (2022).
[23]P. J. Ireland, A. D. Bragg, and L. R. Collins, "The effect of Reynolds number on inertial particle dynamics in isotropic turbulence. Part 1. Simulations without gravitational effects," J. Fluid Mech. **796**, 617–658 (2016).
[24]L. Biferale and I. Procaccia, "Anisotropy in turbulent flows and in turbulent transport," Phys. Rep. **414**, 43–164 (2005).
[25]A. J. Petersen, L. Baker, and F. Coletti, "Experimental study of inertial particles clustering and settling in homogeneous turbulence," J. Fluid Mech. **864**, 925–970 (2019).
[26]J. Lee and C. Lee, "The effect of wall-normal gravity on particle-laden near-wall turbulence," J. Fluid Mech. **873**, 475–507 (2019).
[27]L. Brandt and F. Coletti, "Particle-laden turbulence: Progress and perspectives," Annu. Rev. Fluid Mech. **54**, 159–189 (2022).
[28]M. Mehrabadi, J. A. K. Horwitz, S. Subramaniam, and A. Mani, "A direct comparison of particle-resolved and point-particle methods in decaying turbulence," J. Fluid Mech. **850**, 336–369 (2018).
[29]M. Momenifar and A. D. Bragg, "Local analysis of the clustering, velocities, and accelerations of particles settling in turbulence," Phys. Rev. Fluids **5**, 034306 (2020).
[30]H. Xia, N. Francois, H. Punzmann, and M. Shats, "Lagrangian scale of particle dispersion in turbulence," Nat. Commun. **4**, 2013 (2013).
[31]R. Hassanian, "An experimental study of inertial particles in deforming turbulence flow, in context to loitering of blades in wind turbines," M.S. dissertation (Reykjavik University, 2020).
[32]L. Bouhlali, "On the effects of buoyancy on passive particle motions in the convective boundary layer from the Lagrangian viewpoint," M.S. dissertation (Reykjavik University, 2012).
[33]M. R. J. Kompenhans, C. E. Willert, F. Scarano, C. J. Kähler, and S. T. Wereley, *Particle Image Velocimetry* (Springer International Publishing, 2018).
[34]R. J. Adrian and R. D. Keane, "Theory of cross-correlation analysis of PIV images," Appl. Sci. Res. **49**, 191 (1992).
[35]E. A. Cowen, S. G. Monismith, E. A. Cowen, and S. G. Monismith, "A hybrid digital particle tracking velocimetry technique," Exp. Fluids **22**, 199–211 (1997).
[36]N. T. Ouellette, H. Xu, and E. Bodenschatz, "A quantitative study of three-dimensional Lagrangian particle tracking algorithms," Exp. Fluids **40**, 301–313 (2006).

Paper II

The Capability of Recurrent Neural Networks to Predict Turbulence Flow via Spatiotemporal Features

R. Hassanian, M. Riedel, L. Bouhlali

https://doi.org/10.1109/ICCC202255925.2022.9922754, 2022

2022 IEEE 10th Jubilee International Conference on Computational Cybernetics and Cyber-Medical Systems (ICCC 2022) | 978-1-6654-8177-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICCC202255925.2022.9922754

The Capability of Recurrent Neural Networks to Predict Turbulence Flow via Spatiotemporal Features

Reza Hassanian*, Morris Riedel*,†, Lahcen Bouhlali‡
*The Faculty of Industrial Engineering, Mechanical Engineering and Computer Science, University of Iceland, Reykjavik, Iceland
†Juelich Supercomputing Centre, Germany
‡Department of Engineering, School of Technology, Reykjavik University, Reykjavik, Iceland
*seh@hi.is, †morris@hi.is, ‡lahcen09@ru.is

***Abstract*—This study presents a deep learning (DL) neural network hybrid data-driven method that is able to predict turbulence flow velocity field. Recently many studies have reported the application of recurrent neural network (RNN) methods, particularly the Long short-term memory (LSTM) for sequential data. The airflow around the objects and wind speed are the most presented with different hybrid architecture. In some studies, the investigated data set in fluid dynamics were generated via known equations, and they have no random and chaotic behavior. Data series extracted from Computational Fluid Dynamics (CFD) have been used in many cases. This work aimed to determine a method with raw data that could be measured with devices in the airflow, wind tunnel, water flow in the river, wind speed and industry application to process in the DL model and predict the next time steps. This method suggests spatial-temporal data in time series, which matches the Lagrangian framework in fluid dynamics. Gated Recurrent Unit (GRU), the next generation of LSTM, has been employed to create a DL model and forecasting. Time series data source is from turbulence flow has been generated in a laboratory and extracted via 2D Lagrangian Particle Tracking (LPT). This data has been used for the training model and to validate the prediction in the suggested approach. The achievement via this method dictates a significant result and could be developed.**

***Index Terms*—Recurrent Neural Network, Unsteady Flow, Deep Learning**

I. Introduction

Turbulence is observed in the most natural and artificial phenomena [1] [2]. Water in the waterfall, airflow in the wind, smoke from a chimney, and airflow around the objects are examples from the environment [1]. The industry cases are the flow in the engine mixing chamber; two working flows inside the heat exchanger, and airflow around the airplane and car [1] [3] [4] [5]. In large-scale turbulence, solar flare, oceanic and atmospheric flow are other giant emanations that influence our lives [2]. Turbulence flow is chaotic, non-repeatable, and random, and it is well addressed that the statistics aspect of the flow is applicable [1]. On the other hand, Computational Fluid Dynamics (CFD) is a leading traditional numerical approach to dealing with nonlinear fluid dynamics phenomena such as turbulence flow. Direct Numerical Method and Large Eddy Simulation are two capable and accurate methods to resolve the turbulent flow problems. But, from the computational cost, they are costly. High-performance computation is an essential factor for all solutions in Direct Numerical Method and Large Eddy Simulation. Simulation for many types of turbulence problems is almost impossible on the actual scale because of the limitation in the computation. Scientists have efforts to create similar scale problems to natural phenomena. However, we are still far from solving problems with extensive size. In many CFD applications, it is required to validate the solution with empirical data, is an another limitation. These constraints illustrate a reliable tool is necessary to overcome the above-called obstacles. Machine learning (ML) based on Artificial Intelligence has become an important key to encountering nonlinear phenomena. Deep learning (DL) is a capable approach in ML and is able to extract the hidden features from complex and nonlinear dynamic systems [6] [7] [8] [9]. Recurrent neural network (RNN) is a type of neural network especially appropriate for sequential data such as time series [6]. An RNN is a neural network composed of an individual hidden layer with a feedback loop in which the hidden layer output with the current input is returned to the hidden layer [6]. RNN networks define the temporal relationship because of sequential input data, and three weight matrices and two biases characterize it. RNNs can almost not train sequence data with long-range temporal dependencies because the vanishing gradients problem exists [6]. Long short-term memory (LSTM) network was developed and suggested in 1995 [10]. LSTM applies a gating structure to control the transients of the recurrent connectors and can deal with the vanishing gradient issue. Moreover, it is able to model longer temporal dependencies rather than standard RNNs [6]. Recently, LSTM has been employed in many studies in order to model time

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.

series prediction. The interest in this method has also increased in the fluid dynamics area. Vinuesa et al. [6] have used LSTM to predict the turbulence shear flow. Veisi et al. [7] used LSTM hybrid model prediction for unsteady flows. LSTM Potential has been led to hybrid models such as convolutional neural network (CNN)–LSTM, Autoencoders–LSTM, and LSTM/RNN [11]. Gated recurrent unit (GRU) [12] is a variant of LSTM which has fewer parameters than LSTM, and the training rate is faster [11]. In GRU, the forget gate and input gate in LSTM are replaced with only one update gate [11]. GRU is required fewer data to train the model, therefore gaining a similar performance in multiple tasks with less computation [11]. Recently GRU has been employed to forecast wind speed and predicts electricity demand [12] [13] [14]. Most fluid flow studies that were applied ML/DL are composed of data extracted from CFD studies' known equations. On the other hand, many works included preliminary steps to do autoencoder to extract the main features, such as, proper orthogonal decomposition, dynamic mode decomposition, and well-known reduced order methods [7] [15] [16] [17]. In the ML/DL context, there is a capability to determine a training method with raw data from the Lagrangian framework velocity field involving spatial and temporal features. In many applications of industry, research and experiment, it is possible to measure the velocity field directly or indirectly via devices such as constant temperature anemometer, flowmeter (and obtain the velocity), pitot tube, laser doppler anemometry, and light detection and ranging. This study introduces a method to use time series data consisting of velocity components and position in 2D coordinate to train the GRU model and evaluate the prediction in future time. Hence, this paper is organized as follows. The applied theory is presented in Section II. In Section III, the method is introduced. Section IV discusses the result, and the conclusions are presented in Section V.

II. Theory

A. Lagrangian Framework in fluid dynamics

Lagrangian framework is a description of the motion fluid, involves keeping track of the position vector and velocity vector of each point of flow which it is called fluid particle [1] [18]. A fluid particle is a point that moves with the local fluid velocity, therefor it specifies the position at time t of fluid particle [16]. The definition of fluid particle mathematically is [1]:

$$x_i = x_i(t, x_{i,0}), \quad i = 1, 2, 3 \tag{1}$$

$$U_i = U_i(t, x_1(t, x_{1,0}), x_2(t, x_{2,0}), x_3(t, x_{3,0})), \quad i = 1, 2, 3 \tag{2}$$

where (1) and (2) determines the fluid particle position and velocity in 3D coordinates respectively. x is the position, U is the velocity, t is the time and denote i specifies the vector component. Based on the Lagrangian definition, for fluid particle there is a time series data which specify a position and velocity at particular time. Particularly in turbulence flow which has not known equation and it is investigated in statistics, these time series data available and appropriate to use.

B. Gated Recurrent Unit (GRU)

From the DL method, it is well known that RNNs can perform prediction for sequence data via LSTM. GRU [19] is a next-generation determination from LSTM with a bit distinction in the model architecture. Literature reports that GRU is comparable in performance is considerably faster to compute than LSTM and has a streamlined model [20]. GRU cell that is displayed in Fig. 1, is composed of a hidden state, reset gate, and update gate. We can control how much of the previously hidden state might be remembered from the reset gate. On the other hand, via the update gate, we can understand how much of the new hidden state is just a copy of the old hidden state. This architecture in the GRU establishes two significant features: the reset gate captures short-term dependencies in sequences, and the update gate receives long-term dependencies in sequences [19].

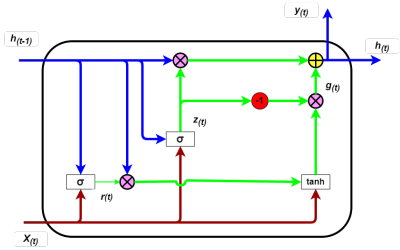


Fig. 1. Gated Recurrent Unit cell; h_{t-1} is hidden state from previous step, X_t is current input, h_t is new hidden state, y_t is output, $r(t)$ is reset gate, $z(t)$ is update gate, $g(t)$ is candidate hidden state, σ is sigmoid function, tanh is hyperbolic tangent function.

III. Methodology

A. Data-set generated in a laboratory facility

This study uses a data-set from experiments conducted for turbulence flow investigation [21] [22]. A special tank is designed to create a turbulence flow via eight impellers in the corners, and servomotors control the impellers' speed. The rotation speed specifies the turbulence intensity of the flow case. The flow was seeded with tracer particles with median diameters 8—10 μm and specific gravity g/cm^3 (hollow glass). The Stokes number for the seeded particle is less than one and meets the tracing requirement. Equation 3 describes the mean flow field $\langle \mathbf{U} \rangle$ in the facility;

$$\langle \mathbf{U} \rangle = (Sx, -2Sy, Sz), \tag{3}$$

where -2S is the primary strain rate in the y-dir, and S is the strain rate for the other two directions, x, y and z are the particle location. In this work the flow considered in 2D,

therefore z-dir is not addressed. The straining flow case in the experiment was created with a mean strain rate $2S = -4s^{-1}$. Equation 3 is based on the laminar flow; however, we know that velocity fluctuates in the turbulence flow. The measurement area is equivalent to 24.5 $\times$ 24.5 mm^2 (512 $\times$ 512 pixels) located in the tank center, and a high-speed camera with 10000 frames per second is employed to record the tracer particle move. Two circular plates were prepared in the tank and located up and down the study area, and they were moved toward each other by an actuator to generate a straining turbulence flow. The data-set included 4000 frames, and the Lagrangian particle tracking technique was employed to process the recorded frames and extract the particle motion feature statistics.

B. Velocity time series data

This study has designed and applied a suggested hybrid model based on time series vector data for velocity. The spatial and temporal data extracted from 2D Lagrangian Particle Tracking. Data is included time, velocity in x and y directions, and position in x and y coordinates. Therefore, we have corresponding velocity and position with a specific time in this time series. In the suggested model, since the velocity is with two components in the x and y direction, we carried on the model for every component individually. Hence, the model predicts the velocity component in both directions and then could be developed in 3D time-series data. The strain motion has been conducted in the y-direction and is considered an effective dominant direction for the flow case behavior; thus, the prediction model in this work is created for this direction.

C. GRU model

The proposed GRU model is created with data series involving two velocity components in x and y directions and two-position coordinates x and y. Every fluid particle at a specified time has a velocity component, and based on the Lagrangian view; they are dependent on the time and position. Both position vectors also function of time and primary position. The input features are on different scales, and then it is essential to scale the features. A function is defined to create time-series data set. The data are split into 80% training and 20% test data set. The GRU model is created with hundred GRU layers and one Dense layer, and the model is optimized with an Adam optimizer. In order to evaluate the model, the mean absolute error (MAE) and coefficient of determination (R^2) are measured.

IV. Result

This work applied a recorded data set from an experiment to create a GRU model. The flow case was a straining turbulence flow with a specified mean strain rate. In order to assess the experiment, the processed data investigated the mean strain rate from the measured data. Since the theoretical mean strain was $2S = -4s^{-1}$ in y-direction based on the laminar flow description, it is well known in the turbulent flow, the velocity fluctuates, and this affects the measured strain rate. Literature also has mentioned that the generated strain in the turbulent flow will create extra fluctuation [2] [5] and could be the reason to measure a lower means strain rate than the theoretical amount. Fig 2 presents the measured mean strain rate in the investigated flow case.

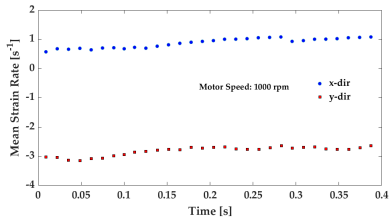


Fig. 2. The mean strain rate of the generated deformed turbulence flow is extracted from measured data in the experiment. The mean strain rate theoretical was $2S = -4s^{-1}$ in y-direction and $S = 2s^{-1}$ in x-direction based on laminar flow.

Fig 3 displays the velocity component in the y-direction measured from the experimental data for the Lagrangian fluid particles. The mean velocity is expected to increase approximately linearly, but the turbulence flow velocity field fluctuates, which makes the velocity behave chaotically and irregularly. The strain motion was generated in the y-direction, which is the dominant motion and has much more oscillations and fluctuations than another direction which makes it difficult to predict the next time step. This study considered this velocity data set in y-direction via the GRU model to create predictions for the next step and evaluate the prediction with actual available recorded data.

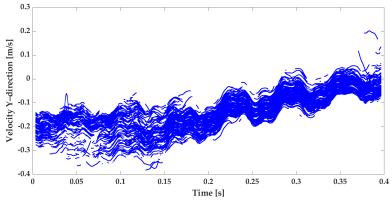


Fig. 3. The velocity component in the y-direction of the tracked particles is extracted from the measured data. The strain motion time is recorded in $0.4s$.

The explained method in the study is based on the capability of DL via GRU, which is able to store long-term dependencies. From the data set 80% used as train data and 20% as test data. Fig. 4 represents the result of this model that has been

used to predict the velocity components in the y-direction. In Fig. 4, the actual data, hidden and covered by train data and predicted data, dictates the suggested model could make remarkable forecasting for a future time. For the conducted

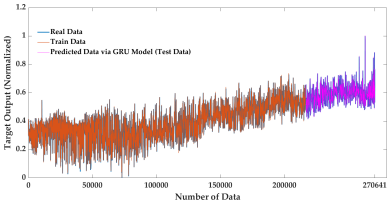


Fig. 4. GRU model for turbulence flow velocity in y direction with spatial temporal features

model, MAE and R^2 were measured equal to 0.002 and 0.98, respectively. These measurements determine that the GRU model can establish a significant prediction for time series with features that have relationships analogous to described data in this work that could be seen in many turbulence flow applications.

V. Conclusion

This work aimed to determine a method to use spatial-temporal features of the Lagrangian framework data in a turbulent flow to create a prediction model based on DL authority. In this view, the velocity functions of the position and time. On the other hand, the position is related to the time and primary place. DL networks for sequential data have been developed in subsets in RNNs such as LSTM and GRU. Turbulence flow is a high dimensional phenomenon, and to use a feature for LSTM/GRU model, it is essential to figure out the main features among the high-dimensional data. This study proposed a GRU model relying on velocity components and the position of the fluid particles and exclusive of high dimensionality. Moreover, GRU can predict a time series with long-term dependencies based on the result presented and the Lagrangian definition for the velocity field, storing long-term dependencies is a crucial factor that led to this significant prediction and matched the actual data in the test. On the other hand, this method creates predictions for every velocity component individually, making it applicable for 2D and 3D fluid flow. The error measurement represented in the evaluation of this method implies the capability of GRU in this kind of application and could be developed for long-term forecasting studies. Since the dominant motion in the considered flow case is a y-direction, we created a prediction model for the velocity in the y-direction. The model can develop and perform in other directions, and it has planned to implement in next works of our research group.

Acknowledgment

We thank Prof. Ármann Gylfason from Reykjavik University for his technical comments on the experiment conducted at the Laboratory of Fundamental Turbulence Research (LFTR) at Reykjavik University. We also acknowledge Prof. Chung-Min Lee from California State University Long Beach for her assistance.

References

[1] S. B. Pope, *Turbulent Flows*. Cambridge University Press, 2000.
[2] P. A. Davidson, *Turbulence: An Introduction for Scientists and Engineers*. Oxford University Press, 2004.
[3] F. Toschi and E. Bodenschatz, "Lagrangian properties of particles in turbulence," *Annual Review of Fluid Mechanics*, vol. 41, no. 1, pp. 375–404, 2009.
[4] G. Yeoh, J. J. Chen, and C. H. Chen, "Investigation of swirling flows in mixing chambers," *Modelling and Simulation in Engineering*, 2011.
[5] C.-M. Lee, A. Gylfason, P. Perlekar, and F. Toschi, "Inertial particle acceleration in strained turbulence," *Journal of Fluid Mechanics*, vol. 785, p. 31–53, 2015.
[6] 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.
[7] 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.
[8] A. Kolkova and M. Navratil, "Demand forecasting in python: Deep learning model based on lstm architecture versus statistical models," *Acta Polytechnica Hungarica*, vol. 18, no. 8, pp. 123–141, 2021.
[9] A. Pejic and P. S. Molcer, "Predictive machine learning approach for complex problem solving process data mining," *Acta Polytechnica Hungarica*, vol. 18, no. 1, pp. 45–63, 2021.
[10] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 11 1997.
[11] C. Gu and H. Li, "Review on deep learning research and applications in wind and wave energy," *Energies*, vol. 15, no. 4, p. 1510, Feb 2022.
[12] Y. Wang, R. Zou, F. Liu, L. Zhang, and Q. Liu, "A review of wind speed and wind power forecasting with deep neural networks," *Applied Energy*, vol. 304, p. 117766, 2021.
[13] K. Nam, S. Hwangbo, and C. Yoo, "A deep learning-based forecasting model for renewable energy scenarios to guide sustainable energy policy: A case study of korea," *Renewable and Sustainable Energy Reviews*, vol. 122, p. 109725, 2020.
[14] T. Kuremoto, S. Kimura, K. Kobayashi, and M. Obayashi, "Time series forecasting using a deep belief network with restricted boltzmann machines," *Neurocomputing*, vol. 137, pp. 47–56, 2014, advanced Intelligent Computing Theories and Methodologies.
[15] J. L. Lumley, "The structure of inhomogeneous turbulent flows," *Atmospheric Turbulence and Radio Wave Propogation*, pp. 166–178, 1967, nauka, Russia.
[16] J. L, Lumley, *Stochastic tools in turbulence*. Elsevier, 1970.
[17] SCHMID and P. J., "Dynamic mode decomposition of numerical and experimental data," *Journal of Fluid Mechanics*, vol. 656, p. 5–28, 2010.
[18] Y. Cengel and J. Cimbala, *Fluid Mechanics Fundamentals and Applications*. McGraw Hill, 2017.
[19] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder–decoder approaches," in *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 103–111.
[20] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *NIPS 2014 Workshop on Deep Learning, December 2014*, 2014.
[21] H. Reza, "An experimental study of inertial particles in deforming turbulence flow, in context to loitering of blades in wind turbines," M. Eng. thesis, Reykjavik University, 2022.
[22] B. L, "On the effects of buoyancy on passive particle motions in the convective boundary layer from the lagrangian viewpoint," M. Eng. thesis, Reykjavik University, 2012.

Paper III

Deep Learning Forecasts a Strained Turbulent Flow Velocity Field in Temporal Lagrangian Framework: Comparison of LSTM and GRU

R. Hassanian, Á. Helgadóttir, M. Riedel

https://doi.org/10.3390/fluids7110344, 2022

fluids

MDPI

Article

Deep Learning Forecasts a Strained Turbulent Flow Velocity Field in Temporal Lagrangian Framework: Comparison of LSTM and GRU

Reza Hassanian [1,2,*], Ásdís Helgadóttir [1,2] and Morris Riedel [1,3]

[1] The Faculty of Industrial Engineering, Mechanical Engineering and Computer Science, University of Iceland, 102 Reykjavík, Iceland
[2] Computational Fluid Dynamics, Simulation and Data Lab, National Competence Centre of Iceland, University of Iceland, 102 Reykjavík, Iceland
[3] Juelich Supercomputing Centre, 52428 Jülich, Germany
* Correspondence: seh@hi.is

Abstract: The subject of this study presents an employed method in deep learning to create a model and predict the following period of turbulent flow velocity. The applied data in this study are extracted datasets from simulated turbulent flow in the laboratory with the Taylor microscale Reynolds numbers in the range of $90 < R_\lambda < 110$. The flow has been seeded with tracer particles. The turbulent intensity of the flow is created and controlled by eight impellers placed in a turbulence facility. The flow deformation has been conducted via two circular flat plates moving toward each other in the center of the tank. The Lagrangian particle-tracking method has been applied to measure the flow features. The data have been processed to extract the flow properties. Since the dataset is sequential, it is used to train long short-term memory and gated recurrent unit model. The parallel computing machine DEEP-DAM module from Juelich supercomputer center has been applied to accelerate the model. The predicted output was assessed and validated by the rest of the data from the experiment for the following period. The results from this approach display accurate prediction outcomes that could be developed further for more extensive data documentation and used to assist in similar applications. The mean average error and R^2 score range from 0.001–0.002 and 0.9839–0.9873, respectively, for both models with two distinct training data ratios. Using GPUs increases the LSTM performance speed more than applications with no GPUs.

Keywords: turbulent flow; Lagrangian framework; unsteady; prediction; deep learning; sequential

Citation: Hassanian, R.; Helgadóttir, Á.; Riedel, M. Deep Learning Forecasts a Strained Turbulent Flow Velocity Field in Temporal Lagrangian Framework: Comparison of LSTM and GRU. *Fluids* **2022**, *7*, 344. https://doi.org/10.3390/fluids7110344

Academic Editors: Mehrdad Massoudi, Hongbin Yan and Wei-Tao Wu

Received: 15 September 2022
Accepted: 1 November 2022
Published: 3 November 2022

1. Introduction

Turbulent flow is a nonlinear and random phenomenon [1–3]. Water flow in a river, waterfall, airflow passing a wind turbine blade, flow in an engine mixing chamber, smoke from a chimney, and two working flows inside the heat exchanger are examples of turbulent flow in natural events and artificial applications [1–5]. The complexity and multiscale features of turbulent flows make the forecasting of the fluid flow a considerable problem. There are many previous works using experiments and/or numerical methods of turbulent flow to investigate and make efforts to forecast flow periods with specified conditions. However, experiments are costly and, for many applications, could not be performed in a laboratory environment. Computational methods based on partial differential equations, i.e., applying a full-order model, are capable of predicting fluid flow accurately but are computationally costly. High-performance computing is, therefore, essential in those computational methods, yet we are still far from having computing capability to solve even moderately sized problems. Thus, there are limitations in computing costs [6].

These restrictions determine a reliable tool is required to overcome the above-mentioned obstacles. Machine learning based on artificial intelligence has become a pivotal approach

to encountering nonlinear events. Deep learning networks (DL) applications have recently been represented as having strong capability to model and forecast phenomena with unknown patterns. DL is able to extract hidden features from complex and nonlinear dynamic systems [7,8]. Recurrent neural networks (RNNs) are neural networks appropriate for sequential datasets, such as time series [7]. An RNN is composed of an individual hidden layer with a feedback loop in which the hidden layer output with the current input is returned to the hidden layer [7]. RNNs define the temporal relationship because of sequential input data and three weight matrices and two biases characterize it. RNNs can almost not train the sequential data with long-range temporal dependencies because a vanishing gradients problem exists [7]. Long short-term memory (LSTM) networks were developed and suggested in 1995 [9], which apply a gating structure to control the recurrent connectors' transients and deal with a vanishing gradient issue. Moreover, it is able to model longer temporal dependencies than standard RNNs [7].

A gated recurrent unit (GRU) is a variant of LSTM, which has fewer parameters than LSTM, and its training rate is faster [10,11]. In GRU, the forget gate and input gate in LSTM are replaced with only one update gate [11]. GRU requires fewer data to train the model, therefore, gaining a similar performance in multiple tasks with less computation [11]. Recently, LSTM has been employed in many studies to model time series prediction. Interest in this method has also increased in the fluid dynamics area. Vinuesa et al. [7] have used LSTM to predict a shear turbulence flow. Veisi et al. [8] used LSTM hybrid model prediction for unsteady flows. LSTM potential has led to hybrid models, such as convolutional neural networks (CNNs)-LSTM, Autoencoders-LSTM, and LSTM/RNN [11]. Bukka et al. [6] applied a hybrid deep learning prediction model based on a reduced-order model for unsteady flow. Duru et al. [12] used CNN to predict transonic flow around the airfoils. GRU has been employed to forecast wind speed and anticipate electricity demands [10,11].

Most fluid flow studies that applied ML/DL are composed of data extracted from CFD studies' known equations. On the other hand, many works include preliminary steps to autoencoders to extract the main features, such as proper orthogonal decomposition, dynamic mode decomposition, and well-known reduced-order methods [8,13–15].

The subject of the current study is a novel approach to present a capability in the DL context to make a training method with raw measured data from the Lagrangian framework velocity field with non-specified pattern and to predict followed fluid flow period. In many applications of industry and experiments, it is possible to measure the velocity fields directly or indirectly via devices, such as a constant temperature anemometer, flowmeter (and obtain the velocity), pitot tube, laser doppler anemometry, and light detection and ranging. This study will introduce an application of an empirical dataset from a laboratory with unknown patterns composed of 2D velocity components and time. LSTM and GRU have been used to create a prediction model for a strained turbulent flow. In order to accelerate the deep models' execution, they were implemented in the DEEP-DAM module from a parallel computing machine at the Juelich supercomputing center. Hence, this paper is organized as follows. The applied theory is introduced in Section 2. In Section 3, the data set from the experiment is explained. Section 4 determines the models used. The results and discussion are provided in Section 5, and the conclusion is presented in Section 6.

2. Theory

2.1. Fluid Flow in Lagrangian Framework

In turbulent flow, it is well-known what statistical aspects of the flow features are applicable [1–3]. A Lagrangian framework is an exploration of fluid motion that keeps track of the velocity vector and displacement vector of each flow point, called a fluid particle [1,16]. A fluid particle is a point that moves with the local fluid velocity, and, therefore, it identifies the position at the time t of a fluid particle [16]. The definition of a fluid particle arithmetically is [1]:

$$x_i = x_i(t, x_{i,0}),\ i = 1, 2, 3 \tag{1}$$

$$U_i = U_i(t, x_1(t, x_{1,0}), x_2(t, x_{2,0}), x_3(t, x_{3,0})) \tag{2}$$

where the fluid particle position and velocity in 3D coordinates is determined by (1) and (2), respectively, x is the position, U is the velocity, t is the time, and i specifies the vector component.

Based on the Lagrangian definition, there is time series data for fluid particles that determine a position and velocity at a specific time. In particular, in turbulent flow, which has no known equation and is investigated using statistics, a sequential dataset could be used from the Lagrangian view for the forecasting model. It is a crucial challenge to be able to have accurate prediction for turbulent flow velocity via an approach that does not need preprocessing to extract hidden features or reduced order methods [6–8,17–19]. Some numerical methods due have drawbacks, such as missing features because of dimension reduction [6–8,17–19]. Based on the above description for the velocity in the Lagrangian framework, in this study we will apply velocity denotation (2) to train LSTM or GRU model via velocity component as an input. Then, the model will predict the velocity component for next period as an output.

This study used the dataset for a strained turbulent flow that has been generated in a laboratory. Turbulence intensity has been created with the action of impeller rotors in the corners of a box turbulence facility. The turbulent flow has been strained in the vertical direction (see Figure 1) by the motion of flat circular plates, as shown in the sketch. Equation (3) defines the mean velocity field of the flow:

$$\langle U \rangle = (\mathrm{Sx}, -2\mathrm{Sy}, \mathrm{Sz}) \tag{3}$$

where 2S is the primary strain rate in the Y-dir, S is the mean strain rate for the other two directions, and x, y and z are the particle location. In this work, the flow was considered in 2D, therefore, Z-dir is not addressed.

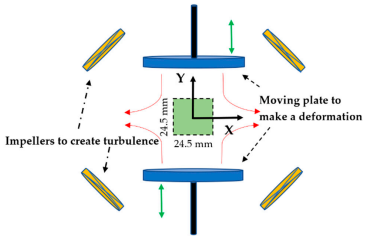


Figure 1. A sketch of the straining mechanism and turbulence generation.

The straining flow case has been created in the experiment with mean strain rate, $2S = 8\ s^{-1}$. Equation (3) is based on laminar flow; however, we know that velocity fluctuates in turbulent flow. Further details on the experimental setup are described in Section 3.

The Stokes number for a seeded particle is calculated to ensure a tracer particle meets the requirements and specifies whether a particle introduced to the flow will follow the flow streamline or not. This identification is defined by Equation (4):

$$\mathrm{St} = \frac{\tau_p}{\tau_\eta} \tag{4}$$

where τ_p is Stokes' relaxation time. Kolmogorov scale (τ_η, defined in Equation (6)) is based on the flow quantities before applying the strain. Stokes' relaxation time τ_p is, in turn, calculated by Equation (5):

$$\tau_p = \frac{\rho_p {d_p}^2}{18\mu} \tag{5}$$

where ρ_p is particle density, d_p is a spherical particle diameter, and μ is the dynamic fluid viscosity that, in the conducted experiment, was water.

The Stokes number significantly greater than 1 ($St \gg 1$) describes particles that are unaffected by a fluid velocity change and continue their original trajectory; if $St \ll 1$, the particle will follow the fluid's local velocity. To extract the flow properties, particle image velocimetry method has been employed and once dissipation rate specified via second-order longitudinal velocity structure function, Kolmogorov scales were calculated by:

$$\tau_\eta = \left(\frac{\nu}{\varepsilon}\right)^{\frac{1}{2}} \tag{6}$$

where ν is kinematic viscosity of the fluid, τ_η is the Kolmogorov's time scale, and ε is dissipation rate evaluated via second-order longitudinal velocity structure function.

The Stokes number for the tracer particles used in the performed experiment are in the range of 0.0063–0.0094.

2.2. LSTM and GRU Architecture

The subject of this study is using the velocity components U_x and U_y, according to definition (2) as sequential input training data for LSTM or GRU model. The model for each component is separated. Even though the velocity is a profound feature in the flow field description and is based on the Lagrangian perspective, it is spatiotemporal; therefore, it carries many flow effects, such as fluctuation, strain, turbulence intensity, and geometry boundary [1–5]. This is the capital concept of this novel proposed approach.

Since the flow velocity is spatial-temporal and is affected via the above-nominated effects, the LSTM or GRU model is trained and learns how to forecast the next period according to received effects. The predicted flow velocity via LSTM or GRU model is validated by test data in the x and y components separately.

Recurrent neural networks are deep network models that can extract sequential data dynamics through recurrent connections. They can be considered as cycles in the network of nodes. Although gradient contraction in recurrent neural networks seems to help exploding gradients, handling vanishing gradients requires a more precise solution [20–22]. One of the first and most successful techniques to solve the vanishing problems was presented in the long short-term memory model [9].

In simple recurrent neural networks, long-term memory is in the form of weights, and the weights change gradually during the training and encode general knowledge about the data. Additionally, these networks have short-term memory, which is in the form of fast transient activations and is continuously transferred from one node to another. In the LSTM model, an intermediate storage type is defined through a memory cell (see Figure 2). A memory cell is a composite unit that consists of simpler nodes and acts through a specific connectivity pattern by imbedding new multiplicative nodes.

Each memory cell is composed of an internal state and several multiplicative gates, which control the data as follows: (i) a given input should affect the internal state (the input gate) or not, (ii) the internal state should drive to 0 (the forget gate), (iii) a given neuron's internal state should be able to impact the cell output (the output gate).

A significant distinction between standard RNNs and LSTM is a hidden state gate determined in LSTM. This state provides an appropriate mechanism for when a hidden state should be updated and when it should be reset. These mechanisms are learned, and they resolve the known concerns from standard RNNs. For example, if the first token has a major significance, it will learn not to update the hidden state after the first perception. In

addition, it will learn to omit incoherent temporary perceptions. Eventually, it will learn to reset the hidden state whenever it is essential.

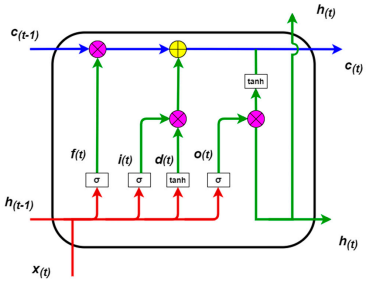


Figure 2. Long short-term memory cell; $h_{(t-1)}$ is hidden state from previous step, $X_{(t)}$ is current input, $h_{(t)}$ is new hidden state, $c_{(t-1)}$ is memory cell internal state, $c_{(t)}$ is new memory cell internal state, $f_{(t)}$ is forget gate, $i_{(t)}$ is input gate, $d_{(t)}$ is input node, $o_{(t)}$ is output gate, σ is sigmoid function, tanh is hyperbolic tangent function; descriptions are based on [9].

GRU is a next-generation determination from LSTM with a bit distinction in the model architecture [23]. Literature reports that GRU is comparable in performance is considerably faster to compute than LSTM and has a streamlined model [17,19,24]. The GRU cell that is displayed in Figure 3 is composed of a hidden state, reset gate, and update gate. We can control how much of the previously hidden state might be remembered from the reset gate.

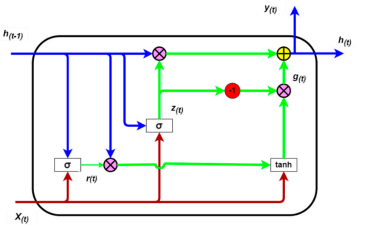


Figure 3. Gated recurrent unit cell; $h_{(t-1)}$ is hidden state from previous step, $X_{(t)}$ is current input, $h_{(t)}$ is new hidden state, $y_{(t)}$ is output, $r_{(t)}$ is reset gate, $z_{(t)}$ is update gate, $g_{(t)}$ is candidate hidden state, σ is sigmoid function, tanh is hyperbolic tangent function [18].

On the other hand, via the update gate, we can understand how much of the new hidden state is just a copy of the old hidden state. This architecture in the GRU establishes two significant features: the reset gate captures short-term dependencies in sequences and the update gate receives long-term dependencies in sequences [23].

3. Experiment and Dataset

3.1. Apparatus and Experiment Setup

The experiment has been performed at the Laboratory of Fundamental Turbulence Research (LFTR) at Reykjavik University and the applied facility is shown in Figure 4. The water tank (60 cm × 60 cm × 60 cm) has 20 mm thick acrylic walls (transparent Plexiglas

XT) that enables optical access to the data. The eight corners of the box have a triangular shape, while the top and the bottom are nearly circular. An aluminum frame holds the components of the turbulence box together. The turbulence has been generated by eight impellers driven by independently controlled servo motors (Lenz-model: MCS), which were mounted at the eight corners of the cube and point to the center of the tank. The rotation rate of each servo motor is adjustable over a range of 100–4500 rpm at a gearing rate of 0.075. For the used dataset in this study the speed is 1000 rpm. The motion-view filtering software that came with these motors was used to monitor and set up the suited speed of each impeller. The degassing system was used to remove bubbles and solid dust from the water before starting the experiment. The tank has been specifically designed for studying turbulence (Lagrangian and Eulerian motion at moderate Reynolds numbers). The flow facility produces a nearly stationary homogeneous isotropic turbulence near the center of the tank, where measurements have been performed.

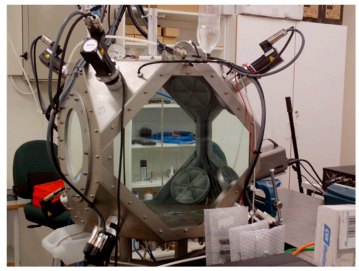


Figure 4. The turbulence flow facility at the Laboratory of Fundamental Turbulence Research, Reykjavik University [18,25,26].

Circular plates, shown at the top and bottom (see Figure 1), generate the strain motions; a linear actuator drove each plate with a piston rod (Parker, model: ETV32M10PA47JMA400A). When they are moved towards the center with a pre-described rate, a nearly constant strain rate is ensured in the fluid. Spherical and hollow glass beads with a median diameter of 8–10 µm and a specific gravity of 1.1 g/cm^3 seed the flow. The recording area is located in the center of the tank with a size of 24.5 × 24.5 mm^2 (see Figure 1). The particle image velocimetry technique is applied and extracts the flow properties before strain deformation. Thus, via a second-order longitudinal velocity structure function, the turbulent flow dissipation rate is obtained; therefore, the Kolmogorov time scale is calculated based on Equation (6) and a Stokes number is obtained from Equation (4). The Taylor microscale Reynolds number is achieved in the range of $90 < R_\lambda < 110$ in the performed experiment. For the dataset used, the strain rate produced by the two above-described circular plates is 8 s^{-1}, and the Lagrangian particle-tracking techniques are applied to inscribe the data. A high-speed camera used as a detection system was set at 10 kHz (10,000 fps) for well-resolved particle velocity 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 resolved. However, in contrast to the previous numerical works, this empirical study considers the velocity field explored by tracer particles in the presence of gravity. Each video has 2000 frames, and to collect sufficient statistical data, the strain motion was repeated 20 times to record 20 videos. All videos together have created 40,000 frames. Each of the videos was statistically independent, as the flow is given a generous time to recover to near isotropy between different strokes.

3.2. Sequential Velocity Dataset

The recorded videos from the experiment are processed via programming based on Lagrangian particle-tracking method and are not subject of this study, and the detail can be seen in [25,26]. Each particle has vector of velocity and displacement during the strain motion. According to denotation (1) and (2) in the Lagrangian view, these statistics could be used to investigate the turbulent flow. As one of the reviewers mentioned, some statistical features can be calculated from extracted data from the experiment [27]. In this study, however, our focus is on velocity because it is a feature that can be measured in many applications, and its prediction model can be helpful. For example, wind speed is a crucial issue in wind energy production; it is, therefore, essential to have a prediction model for speed itself so producers can forecast the power production in the following period, long or short term.

The dataset of this study composed of 2,862,119 tracking points for every vector is as follows:

- Velocity component in y direction;
- Velocity component in x direction;
- Time vector specifies the time t for every tracking point.

These tracking points consist of all particles' velocity vectors achieved via 20 recordings, and every video recording includes several particles. Moreover, it is expected to observe several tracking lines, as it is presented for the velocity in the results in Section 5; every tracking line specifies a particle.

Although the dataset is sequential, we split it into training data and test data for the first model, 80% and 20%, respectively, and for the second model, 60% and 40%, proportionally. Therefore, we assessed and validated the velocity prediction of the following period with the test data for LSTM and GRU models.

4. LSTM and GRU Model Set Up

We coded the models in Python and used the TensorFlow platform [28,29]. The LSTM model is set up with 100 layers and one dense layer, and Adam is specified as an optimizer. The GRU model has also been set up with the same layers and optimizer. The dataset was normalized by MinMaxScaler transformation [30]. The MinMaxScaler is a type of scaler that scales the minimum and maximum values to be 0 and 1, respectively [30]. Since the modeling was implemented on the DEEP-DAM module [31] parallel computing machine, we have applied a distributed strategy application programming interface from the TensorFlow platform abstraction to distribute the training across multiple custom training loops [32]. The strategy has been set up with one to four GPUs on one node. The result of the computing and the models' performance distinction are reported in Section 5.

5. Result and Discussion

5.1. Meseured Turbulent Flow Velocity

As is described in Section 3, from the experiments, based on the Lagrangian-particle tracking, the recorded videos included velocity vectors for particles moving in the flow velocity field. Figures 5 and 6 illustrate the velocity component in the y and x directions, respectively. Since the initial strain rate was generated in the flow in the y direction, as was expected, the velocity component in the y direction has an inclined average velocity relative to the velocity in the x direction. In this study, these extracted data have been used to train LSTM and GRU with a ratio of the training data and assess the prediction with the test data proportion. According to Equation (3), which defines the velocity field for laminar flow, it can be seen in Figures 5 and 6 that the turbulent flow behaves differently than the laminar equation because of velocity fluctuations.

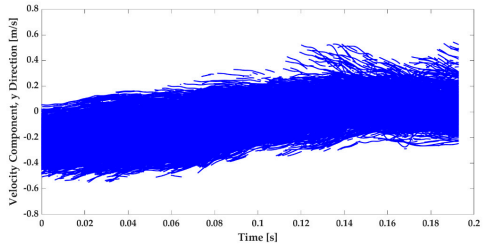


Figure 5. Velocity component in y direction from twenty videos, which involves 40,000 frames extracted via Lagrangian-particle tracking technique.

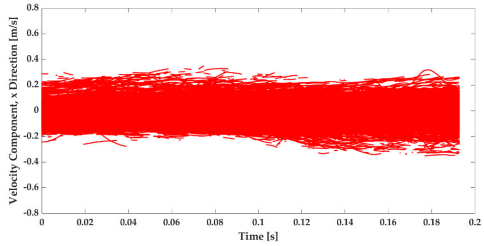


Figure 6. Velocity component in x direction from twenty videos, which involves 40,000 frames extracted via Lagrangian-particle tracking technique.

5.2. Turbulent Flow Velocity Predicition via LSTM and GRU

Figures 7–10 display the velocity prediction via GRU and LSTM model for the y and the x directions. The models trained and assessed with two distinct data ratios. In the first model, 80% of the data have applied as training and 20% rest of the dataset validated the prediction. In second model, 60% of the training data have been applied and 40% used as test data.

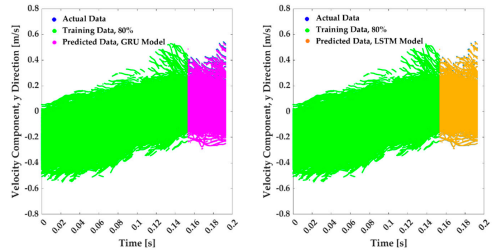


Figure 7. Prediction of velocity component in the y direction for a strained turbulent flow with mean strain rate 8 s^{-1}, GRU model on the left-hand side, and LSTM model on the right-hand side. Training data are 80% and test data 20%.

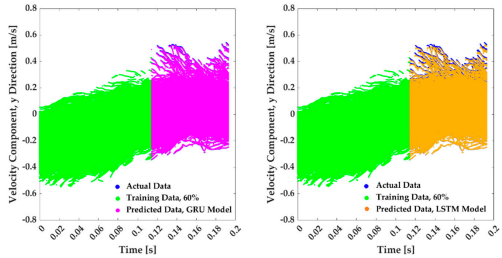


Figure 8. Prediction of velocity component in the y direction for a strained turbulent flow with mean strain rate 8 s^{-1}, GRU model on the left-hand side, and LSTM model on the right-hand side. Training data are 60% and test data 40%.

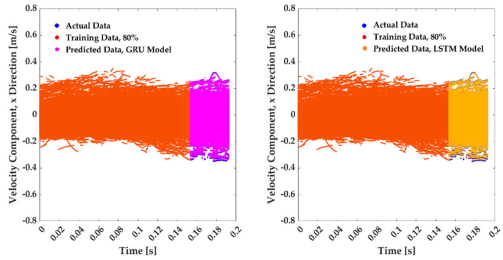


Figure 9. Prediction of velocity component in the x direction for a strained turbulent flow with mean strain rate 8 s^{-1}, GRU model on the left-hand side, and LSTM model on the right-hand side. Training data are 80% and test data 20%.

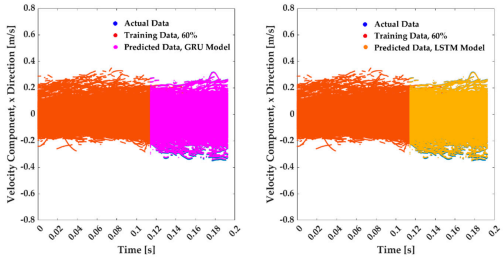


Figure 10. Prediction of velocity component in the x direction for a strained turbulent flow with mean strain rate 8 s^{-1}, GRU model on the left-hand side, and LSTM model on the right hand side. Training data are 60% and test data 40%.

LSTM and GRU models have provided accurate predictions of a strained turbulent flow velocity with no known pattern in theory. We must notice that the period of the experiment and data used in this study are short, but proportionally the prediction model could be used for similar velocity field application.

Figures 7 and 8 represent the velocity in y direction, forecasting 80% training and 60% training. As can be seen, the prediction section is an impressive match for the test data for LSTM and GRU model with two training ratios. The mean average error (MAE) = 0.001–0.002 and R^2 score is in range of 0.983–0.987 for both models.

Figures 9 and 10 illustrate the prediction of the velocity in the x direction with two different training data ratios. The MAE and R^2 for x direction forecasting has the same range and with an outstanding match.

The result of the velocity prediction of the turbulent flow represents the capability of LSTM and GRU models, which can forecast unknown sequential data. The Lagrangian view provides temporal data, and it appears possible to apply this approach in similar turbulent flow with a longer period.

5.3. LSTM and GRU Models Performance

It has been reported in the literature that the GRU model has faster performance than the LSTM model [10,11]. For 80% training, GRU is 8–12% faster than LSTM and for 60% training it is 5–10% faster. However, when the number of GPU increases, LSTM performs modeling faster than GRU, which can be explained by the application of GPUs that provides much more memory for the LSTM. We investigated the performance of LSTM and GRU on DEEP-DAM module on one node with four GPUs. For all models in this study, LSTM executed 7–8% faster than the GRU model. Table 1 shows the result of this evaluation.

Table 1. Evaluation of the LSTM and the GRU model on DEEP-DAM module on one node and several GPUs.

Training Proportion	Computing Module	Performance	LSTM	GRU
80%	1 node, 1 GPU	Scalability	1	1.12
		MAE	0.001	0.002
		R^2 score	0.984	0.984
	1 node, 4 GPUs	Scalability	3.45	3.20
		MAE	0.002	0.002
		R^2 score	0.983	0.983
60%	1 node, 1 GPU	Scalability	1	1.08
		MAE	0.0015	0.0015
		R^2 score	0.985	0.987
	1 node, 4 GPUs	Scalability	3.61	3.36
		MAE	0.002	0.002
		R^2 score	0.985	0.987

6. Conclusions

The subject of this study was using LSTM and GRU models to provide a prediction for distortion turbulent flow performed in a laboratory with specific turbulent intensity and mean strain rate in the primary direction. For two training efforts, the dataset was split into 80% first and secondly 60%. Every ratio of training in the rest of the data was applied for test and prediction validation. LSTM and GRU models were applied and executed on the DEEP-DAM module of parallel computing machine at Juelich supercomputing center. Two different GPU set ups were applied to assess the model's performance.

The result of this study shows that LSTM and GRU models can predict the straining turbulence flow appropriately and match in quality and quantity. The mean average error (MAE) = 0.001–0.002 and R^2 score is in the range of 0.983–0.987 for both models. Without GPU, the GRU model has faster performance than the LSTM and, with less training ratio (60%), can provide prediction with the same performance of training with 80%. Nevertheless, we must notice the period of the dataset used was short, so the forecast was also brief. However, with GPUs set up, LSTM gets faster performance than GRU, which is related to GPUs memory, which strengthens the LSTM memory.

In many applications of fluid dynamics, there is a possibility to collect the velocity field data in the Lagrangian framework in which data are sequential. It seems this advantage of the Lagrangian view could be applied to predict the velocity field via such LSTM and GRU models.

Author Contributions: R.H. contributed with the conceptualization, method, software, data analysis, visualization, and writing—original draft preparation. Á.H. contributed with writing—review and editing. M.R. contributed with resources, writing—review and editing, and supervision. 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) 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.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: We thank Ármann Gylfason from Reykjavik University for his technical comments on the experiment conducted at the Laboratory of Fundamental Turbulence Research (LFTR) at Reykjavik University. We also acknowledge Chung-Min Lee from California State University Long Beach for her assistance. We thank Lahcen Bouhlali from Reykjavik University for his experimental work in data preparation.

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

References

1. Pope, S.B. *Turbulent Flows*; Cambridge University Press: Cambridge, UK, 2000.
2. Tennekes, H.; Lumley, J.L. *A First Course in Turbulence*; The MIT Press: Cambridge, MA, USA, 1972.
3. Davidson, P.A. *Turbulence: An Introduction for Scientists and Engineers*; Oxford University Press: Oxford, UK, 2004.
4. Toschi, F.; Bodenschatz, E. Lagrangian Properties of Particles in Turbulence. *Annu. Rev. Fluid Mech.* **2009**, *41*, 375–404. [CrossRef]
5. Lee, C.; Gylfason, Á.; Perlekar, P.; Toschi, F. Inertial particle acceleration in strained turbulence. *J. Fluid Mech.* **2015**, *785*, 31–53. [CrossRef]
6. Bukka, S.R.; Gupta, R.; Magee, A.R.; Jaiman, R.K. Assessment of unsteady flow predictions using hybrid deep learning based reduced-order models. *Phys. Fluids* **2021**, *33*, 013601. [CrossRef]
7. Srinivasan, P.A.; Guastoni, L.; Azizpour, H.; Schlatter, P.; Vinuesa, R. Predictions of turbulent shear flows using deep neural networks. *Phys. Rev. Fluids* **2019**, *4*, 054603. [CrossRef]
8. Eivazi, H.; Veisi, H.; Naderi, M.H.; Esfahanian, V. Deep neural networks for nonlinear model order reduction of unsteady flows. *Phys. Fluids* **2020**, *32*, 105104. [CrossRef]
9. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1999**, *9*, 1735–1780. [CrossRef] [PubMed]
10. Wang, Y.; Zou, R.; Liu, F.; Zhang, L.; Liu, Q. A review of wind speed and wind power forecasting with deep neural networks. *Appl. Energy* **2021**, *304*, 117766. [CrossRef]
11. Gu, C.; Li, H. Review on deep learning research and applications in wind and wave energy. *Energies* **2022**, *15*, 1510. [CrossRef]
12. Duru, C.; Alemdar, H.; Baran, O.U. A deep learning approach for the transonic flow field predictions around airfoils. *Comput. Fluids* **2022**, *236*, 105312. [CrossRef]
13. Lumely, J.L. The structure of inhomogeneous turbulent flows. In *Atmospheric Turbulence and Radio Wave Propagation*; Publishing House Nauka: Moscow, Russia, 1967.
14. Lumely, J.L. *Stochastic Tools in Turbulence*; Elsevier: Amsterdam, The Netherlands, 1970.
15. Schmid, P.J. Dynamic mode decomposition of numerical and experimental data. *J. Fluid Mech.* **2010**, *656*, 5–28. [CrossRef]
16. Cengel, Y.Y.; Cimbala, J. *Fluid Mechanics Fundamentals and Applications*; McGraw Hill: New York, NY, USA, 2017.

17. Hassanian, R.; Riedel, M.; Helgadottir, A.; Costa, P.; Bouhlali, L. Lagrangian Particle Tracking Data of a Straining Turbulent Flow Assessed Using Machine Learning and Parallel Computing. In Proceedings of the 33rd Parallel Computational Fluid Dynamics (ParCFD) 2022, Alba, Italy, 25–27 May 2022.
18. Hassanian, R.; Riedel, M.; Lahcen, B. The capability of recurrent neural networks to predict turbulence flow via spatiotemporal features. In Proceedings of the IEEE 10th Jubilee International Conference on Computational Cybernetics and Cyber-Medical Systems (ICCC 2022), Reykjavik, Iceland, 6–9 July 2022.
19. Hassanian, R.; Helgadottir, A.; Riedel, M. Parallel computing accelerates sequential deep networks model in turbulent flow forecasting. In Proceedings of the International Conference for High Performance Computing, Networking, Storage, and Analysis, SC22, Dallas, TX, USA, 15–17 November 2022.
20. Elman, J.L. Finding structure in time. *Cogn. Sci.* **1990**, *14*, 179–211. [CrossRef]
21. Bengio, Y.; Simard, P.; Frasconi, P. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Netw.* **1994**, *5*, 157–166. [CrossRef] [PubMed]
22. Hochreiter, S.; Bengio, Y.; Frasconi, P.; Schmidhuber, J. *Gradient Flow in Recurrent Nets the Difficulty of Learning Long-Term Dependencies*; IEEE Press: Piscataway, New Jersey, USA, 2001.
23. Kyunghyun, C.; Bart, V.M.; Dzmitry, B.; Yoshua, B. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. In Proceedings of the SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, Doha, Qatar, 25 October 2014.
24. Junyoung, C.; Caglar, G.; Kyunghyun, C.; Yoshua, B. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning*; Curran Associates, Inc.: Brooklyn, NJ, USA, 2014.
25. Hassanian, R. *An Experimental Study of Inertial Particles in Deforming Turbulence: In Context to Loitering of Blades in Wind Turbines*; Reykjavik University: Reykjavik, Iceland, 2020.
26. Bouhlali, L. *On the Effects of Buoyancy on Passive Particle Motions in the Convective Boundary Layer from the Lagrangian Viewpoint*; Reykjavik University: Reykjavik, Iceland, 2012.
27. Ireland, P.; Bragg, A.; Collins, L. The effect of Reynolds number on inertial particle dynamics in isotropic turbulence. Part 1. Simulations without gravitational effects. *J. Fluid Mech.* **2016**, *796*, 617–658. [CrossRef]
28. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.S.; Davis, A.; Dean, J.; Devin, M.; et al. TensoreFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. *arXiv* **2016**, arXiv:1603.04467.
29. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. TensorFlow: A System for Large-Scale Machine Learning. In Proceedings of the 12th Usenix Symposium on Operating Systems Design and Implementation (OSDI '16), Savannah, GA, USA, 2–4 November 2016.
30. Kramer, O. Scikit-Learn. In *Machine Learning for Evolution Strategies*; Springer: Berlin/Heidelberg, Germany, 2022.
31. Riedel, M.; Sedona, R.; Barakat, C.; Einarsson, P.; Hassanian, R.; Cavallaro, G.; Book, M.; Neukirchen, H.; Lintermann, A. Practice and Experience in using Parallel and Scalable Machine Learning with Heterogenous Modular Supercomputing Architectures. In Proceedings of the 2021 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), Portland, OR, USA, 17–21 June 2021.
32. TensorFlow. *TensorFlow Core Tutorials*; TensorFlow: Mountain View, CA, USA, 2022.

Paper IV

Deciphering the dynamics of distorted turbulent flows: Lagrangian particle tracking and chaos prediction through transformer-based deep learning models

R. Hassanian, H. Myneni, Á. Helgadóttir, M. Riedel

https://doi.org/10.1063/5.0157897, 2023

Physics of Fluids ARTICLE pubs.aip.org/aip/pof

Deciphering the dynamics of distorted turbulent flows: Lagrangian particle tracking and chaos prediction through transformer-based deep learning models

Cite as: Phys. Fluids **35**, 075118 (2023); doi: 10.1063/5.0157897
Submitted: 12 May 2023 · Accepted: 24 June 2023 ·
Published Online: 11 July 2023

R. Hassanian,[1,a)] H. Myneni,[1] Á. Helgadóttir,[1] and M. Riedel[1,2]

AFFILIATIONS

[1]The Faculty of Industrial Engineering, Mechanical Engineering, and Computer Science, University of Iceland, 102 Reykjavík, Iceland
[2]Juelich Supercomputing Centre, Jülich 52428, Germany

Note: This paper is part of the special topic, Turbulence in Plasmas and Fluids.
[a)]**Author to whom correspondence should be addressed:** seh@hi.is

ABSTRACT

Turbulent flow is a complex and vital phenomenon in fluid dynamics, as it is the most common type of flow in both natural and artificial systems. Traditional methods of studying turbulent flow, such as computational fluid dynamics and experiments, have limitations such as high computational costs, experiment costs, and restricted problem scales and sizes. Recently, artificial intelligence has provided a new avenue for examining turbulent flow, which can help improve our understanding of its flow features and physics in various applications. Strained turbulent flow, which occurs in the presence of gravity in situations such as combustion chambers and shear flow, is one such case. This study proposes a novel data-driven transformer model to predict the velocity field of turbulent flow, building on the success of this deep sequential learning technique in areas such as language translation and music. The present study applied this model to experimental work by Hassanian *et al.*, who studied distorted turbulent flow with a specific range of Taylor microscale Reynolds numbers $100 < \mathrm{Re}_\lambda < 120$. The flow underwent a vertical mean strain rate of $8\,s^{-1}$ in the presence of gravity. The Lagrangian particle tracking technique recorded every tracer particle's velocity field and displacement. Using this dataset, the transformer model was trained with different ratios of data and used to predict the velocity of the following period. The model's predictions significantly matched the experimental test data, with a mean absolute error of 0.002–0.003 and an R^2 score of 0.98. Furthermore, the model demonstrated its ability to maintain high predictive performance with less training data, showcasing its potential to predict future turbulent flow velocity with fewer computational resources. To assess the model, it has been compared to the long short-term memory and gated recurrent units model. High-performance computing machines, such as JUWELS-DevelBOOSTER at the Juelich Supercomputing Center, were used to train and run the model for inference.

I. INTRODUCTION

In fluid dynamics and physics, turbulent flow is a complex problem.[1] Turbulent flow is a nonlinear and high-dimensional phenomenon,[2] and it is commonly seen in industrial and natural applications.[3] In addition, the universe is composed of turbulent and unsteady components.[3] Thus, there is a tremendous interest in studying turbulent flow. In this study, experimental data from Hassanian *et al.* work[3] are applied from the Lagrangian particle tracking technique for tracer particle seeded a turbulent deformation flow with a specific mean strain rate and a particular range of Taylor microscale Reynolds numbers. Turbulent flow with deformation can be observed in various scenarios, including leading-edge erosion in compressors and turbines,[5] combustion in internal engines, and particle interaction in mixing chambers.[4] It can also occur in the external flow over an airfoil and in the internal flow of pipes with variable cross sections.[4]

The prior applied solution in turbulent flow has been experimented,[6] and it is still the most robust way.[7] However, designing and conducting experiments is expensive in the case of most natural and industrial flow studies because of their dimensions and scales, leading to constraints and making them even more often impossible to

perform.[1] The most known numerical approach to solving turbulent flow problems is the computational fluid dynamics (CFD) method.[8,9] In CFD, the numerical approaches can be broadly classified into three categories based on the accuracy vs computation time: Reynolds averaged Navier Stokes (RANS), large eddy simulation (LES), and direct numerical simulation (DNS).[10] RANS is applied widely in the industry and provides an average solution, not an exact one.[11] LES solves the problem with better accuracy than RANS, but weaker than DNS, and consequently, DNS proposes the exact answer.[5,11] LES and DNS suffer cost computation, and the computation cost proliferates depending on the problem size.[8] In this term, to implement LES and DNS, it is essential to apply high-performance computing.[9] Despite development in parallel computing, this issue limits these two solver applications. It must be noted in most CFD solutions, validating the results via experiment plays a crucial role.[10] Accordingly, to overcome above-mentioned obstacles, discovering and using an alternative method to provide the possibility to study the turbulent flow widely is an essential matter. Deep learning (DL) has recently been used broadly, proving remarkable capability in fluid dynamics.[12] To analyze turbulent flow in the Lagrangian framework, both spatial and temporal perspectives are crucial to identify flow characteristics in the future period. Among the various deep learning techniques, sequential architectures such as long short-term memory (LSTM) and combinations of convolutional neural networks (CNN) that cover the temporal perspective have proved to be effective models for resolving or predicting turbulent flow issues. It is well addressed that the statistics of turbulent flow are applicable,[1,2] and they are sequence features based on the Lagrangian framework. Obviously, as literature has noted, LSTM variants are an excellent way for sequential datasets.[13] CNN compositions for sequential data require a large dataset to train, and it has several layers, which leads to a lot of computing time.[14]

DL methods involve semi-supervised and unsupervised learning.[14] In semi-supervised learning, the target has no label, and in unsupervised learning, the target pattern must be understood and extracted by the model.[15] The main necessity in deep learning lies in the input data used to train the model. In the realm of fluid dynamics, obtaining an accurate dataset can be achieved through experiments and DNS. Zhou *et al.*[16] applied a surrogate model based on CNN and higher-order dynamic mode decomposition to predict the unsteady fluid force time history for twin tandem cylinders. An unsupervised machine learning Gaussian mixture model for the detection of viscous-dominated and turbulent regions has been proposed by Otmani *et al.*[17] Salehipour *et al.*[18] applied a DL model to discover a generic parameterization of diapycnal mixing. Raissi *et al.*[19] employed a DL model for the prediction of the lift and drag forces on the vortex structure. Kim *et al.*[20] presented an unsupervised learning model that can be trained with unpaired turbulence data for super-resolution reconstruction. Yousif *et al.*[21] proposed a DL method composed of a convolutional auto-encoder and LSTM for generating turbulent inflow conditions. Lee *et al.*[22] determined a data-driven deep learning model to predict unsteady flow over a circular cylinder. Hassanian *et al.*[23] applied LSTM variants to predict a deformed turbulent flow velocity field. Eivazi *et al.*[24] presented a physics-informed neural network application for solving RANS equations. Duru *et al.*[25] represents an application of DL to forecast the transonic flow around airfoils. Bukka *et al.*[26] defined a hybrid DL model to predict unsteady flows. Most fluid flow studies that applied DL use data extracted from CFD computations.[12] Furthermore, most works included preprocessing steps to identify the dominant features, such as proper orthogonal decomposition or dynamic mode decomposition.[12] Recently, LSTM and GRU models are used to predict a turbulent flow with only temporal features.[27,28] Based on above-mentioned terms, it is essential to correspond the DL model to turbulent flow with these features:

- Predicting turbulent flow with minimum training data that are possible to generate for the study case via DNS or experiment.
- The DL model training cost does not grow in relation to the size of the data.
- The DL model is able to extract the dominant feature from available data.
- The DL model performs reliably in a broad range of high Reynolds numbers.

Recently, transformer model from the attention mechanism represents the remarkable capability to simulate and forecast sequential datasets.[29,30] This study aims to apply the transformer model[29] in a novel data-driven approach and assess it based on the above items. A transformer is a DL model based on encoder–decoder layers, and it processes the data through an attention mechanism.[29] It is used widely in language translation.[31] The distinguishing characteristic of the Transformer is its architecture, which completely eliminates the use of recurrence and convolutions.[29] This intrinsic feature enables the transformer model to be highly parallelizable, resulting in significantly reduced training time and computational requirements.[29] Previous recurrent models encounter a sequence of hidden states as a function of previous states and input, and this sequential nature impedes parallelization with training examples. In addition, the attention mechanism allows modeling of dependencies no matter how far apart the inputs or outputs sequence are.[29,31,32]

In this study, the proposed transformer model can be effectively utilized to analyze the strained turbulent flow cases mentioned earlier, offering valuable insights into the physical properties of turbulent flow. This enhanced understanding has wide-ranging applications in both industrial and natural settings. The dataset comprises two components: the velocity of each individual particle and the corresponding time recorded during the experiment. A subset of the dataset is employed to train the transformer model, while the remaining data are used to test the model's predictions and forecast the flow velocity for subsequent periods. Hence, this paper is organized as follows; the theory is described in Sec. II. The methodology and setup are presented in Sec. III. The results and conclusions are represented in Secs. IV and V, respectively.

II. THEORY

This study designs and proposes a transformer model in a data-driven approach using experimental turbulent flow data.[4] In Secs. II A and B, first, the applied turbulent flow theory is described, and then, the transformer model architectures are explored.

A. Turbulent flow and Lagrangian particle tracking

The data used are from turbulent flow conducted in a laboratory, and details can be found in the original work.[4] The flow has a Taylor microscale in the range of $100 < Re_\lambda < 120$. The flow underwent strain deformation with a specific mean rate of $8\,s^{-1}$ in the y direction. Figure 1 displays a sketch of the generated flow. The experiment was

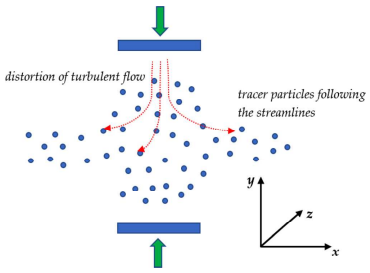


FIG. 1. Sketch of the generated turbulent flow: the particle size is for visualization in the figure, and it is not the actual scale.

conducted in the presence of gravity. The flow was seeded by tracer particles with median diameters 8–10 μm and specific gravity 1.1 g/cm^3 (hollow glass) to extract the flow features. The Lagrangian particle tracking technique based on the work of Ouelette *et al.*[33] was employed to record the tracer particles' movement via the high-speed camera in 2D view. The original work assessed the tracer particle based on Stokes number to ensure that the particle track is following the flow streamlines. Stokes number *St* is an identification when it is $St \ll 1$ for a particle, and it proves the particle pathline corresponding to the flow streamlines. The Stokes number for seeded particles has been reported in the range of 0.00632–0.01807.[4] Thus, the measured data included velocity components in the *x* and *y* directions, location in the *x* and *y* coordinates, and corresponding time *t* for every moment measured. The data were recorded with 10000 frames per second (10 kHz) in the period of 0.2 *s*. From the Lagrangian perspective, the velocity of a fluid particle is defined by[2]

$$x_i = x_i(t, x_{i,0}), \tag{1}$$

$$U_i = U_i(t, x_1(t, x_{1,0}), x_2(t, x_{2,0}), x_3(t, x_{3,0})), \tag{2}$$

where the fluid particle position and velocity in 3D coordinates are determined by notations 1 and 2, respectively, *x* is the position, *U* is the velocity, *t* is the time, and *i* specifies the vector component. In this work, data were measured in 2D; therefore, the 3*rd* dimension has not been addressed.

This study applies a data-driven approach based on the above-described dataset. Since turbulent flow is a complex concept and is a high-dimensional phenomenon, the primary of the DL model application is to discover the dominant feature to be able to predict the following periods of the target segments. In this work, the available data consist of velocity and location in corresponding time. The turbulent flow is affected by the external mean strain rate and has deformed. In addition, the experiment was conducted in the presence of gravity, and its effect is unknown from previous studies.[34] In order to specify the input data, this study proposes a novel approach: the velocity components in sequence feeding the model as input and training the model. This design is based on the concept that velocity in turbulent flow is considerable, and it is attained and carries the most properties of turbulent flow. Moreover, the model gains training for each velocity component individually in the *x* and *y* directions. This configuration allows the model to be used with 3D components and without limits in flow dimensions.

The superiority of this proposed approach is that it employs raw measured velocity datasets without preprocessing to specify the dominant feature or make the dimensional reduction. The turbulent flow velocity could be measured in many industries and natural applications via available devices.

B. Transformer and attention mechanism architecture

The transformer is a DL network composed of encoder–decoder architecture.[35] The input data feed the encoder layers and generate the output via the decoder.[36] The mechanism of this process has several steps. The number of encoder layers must be equal to the decoder layers. In order to specify the input sequence and distance, positional embedding is added to the input vectors. The positional embedded input vectors feed the first encoding layer, and the output of the previous encoder layer provides the successive layers. Every encoder layer is broken down into two sublayers. The first encoder inputs stream through a multi-head attention sublayer. In a multi-head sublayer, all inputs' dependencies are considered to create the weight matrices. The next step is the outputs from the multi-head sublayer stream to the feed-forward sublayer. Between these sublayers, there is an Add&Norm intermediate sublayer, which adds the inputs of the multi-head sublayer to its input and normalizes them. In the feed-forward sublayer, the data are independently applied to each position. Thus in the feed-forward sublayer, data can process in parallel and independently. The outputs from the feed-forward sublayer, in the same manner, should pass the Add&Norm intermediate sublayer. Consequently, the data have been proceeded into an encoder layer and then flowed into the next encoder layer. The number of the encoder layers have no specific and magic numbers,[29] and they must be resolved and determined in the architecture design for every problem. When the first transformer architecture was born,[29] it was designed with only six encoder–decoder layers with notable achievement. This feature of the transformer architecture mentions a remarkable capability to model sequential data problems with fewer layers. Reducing the number of layers in deep learning models could lead to less computing.

When the data cross through all encoder layers, then they will be encoded data flow to decoder layers to embed the outputs. The decoder layer consists of two multi-head and feed-forward sublayers that do the same as the encoder layer and are located after a masked sublayer. The first sublayer in the decoder, as a masked multi-head sublayer, makes a masking layer for the embedded outputs in this way that it only depends on the earlier data. It masks the next sequence of data and avoids their effects. When passed the Add&Norm intermediate sublayer, the output from the masked multi-head flows through the multi-head sublayer. Training the model in the encoding layers creates three weight matrices: Query, Key, and Value. Key and Value matrices from the latter encoder feed the multi-head sublayers of decoder layers directly. However, the multi-head sublayer of every encoder layer gains the Query matrix from the earlier masked sublayer. The following steps in decoder layers are similar to encoder layers. Consequently, the output goes through linear and softmax layers from the last decoder layer. The linear layer is a fully connected

neural network that converts the vector created by the stack of a decoder to a much larger vector called a logit vector. Then, softmax turns the scores from linear vector to probability (all positive, all add up to 1.0), and the cell with the highest probability is chosen as the output for this time step. This study applied a transformer model with four encoder–decoder layers, and its architecture is displayed in Fig. 2. This study uses the sequential velocity vector as input and is processed through a determined transformer encoder–decoder to train and predict the following velocity. To measure the model error of the prediction, this study applied to mean absolute error (MAE) and the R^2 score. These two metric measurements are defined by the following equations:[37]

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|s_{i,a} - s_{i,p}|, \tag{3}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(s_{i,a} - s_{i,p})^2}{\sum_{i=1}^{N}(s_{i,a} - s_m)^2}, \tag{4}$$

where $s_{i,a}$ is the actual measured data, $s_{i,p}$ is the predicted data, s_m is the mean of the actual data, i refer to the corresponding time (or vector array number), and N is the number of the test data.

III. METHODOLOGY

In this section, the design setup of the transformer model is explored, and then, the experimental data from the turbulent flow are described.

A. Transformer model tuning

The model is coded in Python and uses the TensorFlow platform.[38,39] The transformer model is set up with 4 encoder and decoder layers, which shows the attention mechanism ability and distinction of the transformer model from other sequential architecture. This study has tuned the transformer model with the optimum layers number to present the best result. Indeed, it is possible to build a transformer model with more layers. However, the subject of this work applies an optimum model with a minimum layer number. It must be noted that the primary transformer model, when invented, only had six encoder–decoder.[29] Adam is specified as an optimizer.[14] The dataset was normalized by MinMaxScaler transformation.[40] The MinMaxScaler is a type of scaler that scales the minimum and maximum values to be 0 and 1, respectively.[40] Since the modeling was implemented on the DevelBooster module[41] from Juwels parallel computing machine, we have applied a distributed strategy application programing interface from the TensorFlow platform abstraction to distribute the training across multiple custom training loops.[42] The strategy has been set up with four GPUs (Graphics processing units) on one node.

B. Experimental turbulent flow velocity dataset

Each particle has a vector of velocity and displacement during the strain motion. This study proposes a transformer data-driven model for the sequence dataset relying on the velocity. The dataset of this study composed of 2862,119 tracking points for every vector is as follows:

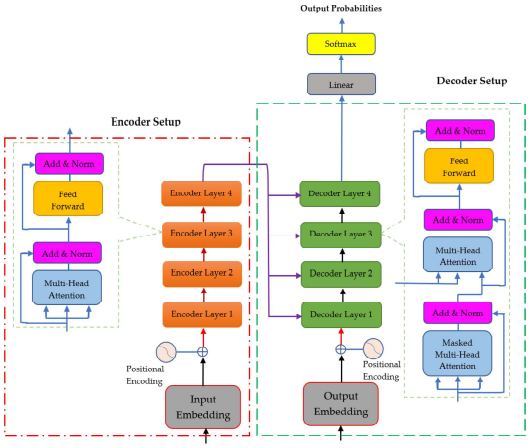


FIG. 2. Transformer model architecture with four encoder–decoder layers.

- Velocity component in the y direction
- Velocity component in the x direction
- Time vector specifies the time t for every tracking point

These tracking points consist of all particles' velocity vectors. Moreover, it is expected to observe several tracking lines, as it is presented for the velocity in the results in Sec. IV; every tracking line specifies a particle. Although the dataset is sequential, it is split into training data and test data for the first model, 80% and 20%, respectively, and for the second model, 60% and 40%, proportionally. Therefore, we assessed the velocity prediction of the following period with the test data for the transformer model.

IV. RESULTS AND DISCUSSION

In this section, the achieved results from the proposed transformer model to forecast the velocity field in a turbulent flow are displayed and discussed. First, the visualization of the measured velocity field will be presented, obtained by the Lagrangian particle tracking.

A. Actual measured velocity

The experimental data from the original work[4] have been applied to the Lagrangian particle tracking techniques. The recorded data involve the velocity vectors in two directions of the x and y. Figures 3 and 4 are presentations of the velocity in the period of the experiment. Accordingly, the turbulent flow underwent a deformation in the y direction, and it is obvious that the velocity fluctuates much more in the y direction than in other directions. It has been noted in the literature that deformation leads to extra fluctuations in a turbulent flow.[1,3] In addition, the velocity in the y orientation gains a slope in the mean velocity, which is caused by the mean strain rate value.[2,4] These two datasets of the velocity fed the transformer model in this study with different training portions, and the observations are reported in the next subsection. For flow with 3D measurement, the third velocity direction will be available and can be performed the same as these two velocity components.

B. Transformer model velocity prediction

The transformer model is trained for each velocity component individually, for the direction of the y once and again the direction of the x. The model has been assessed with two training ratios: start with 80% and then 60% training, and the rest of the data portion has tested

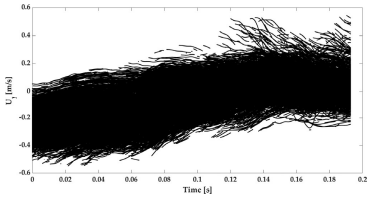


FIG. 3. The measured velocity component in the y direction U_y via Lagrangian particle tracking technique during the deformation.

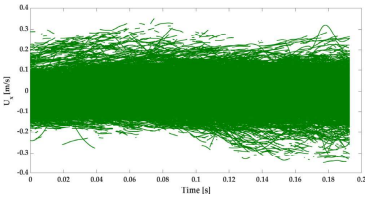


FIG. 4. The measured velocity component in the x direction U_x via Lagrangian particle tracking technique during the deformation.

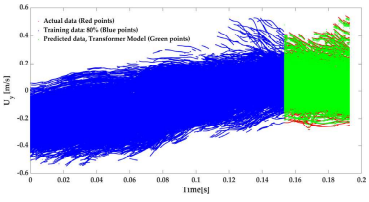


FIG. 5. The prediction of the velocity in the y direction U_y by the proposed transformer model with 80% training data and 20% test data.

the prediction and measured the metric error. Figures 5 and 6 display the transformer model velocity prediction in the y and the x, respectively, which has 80% training data, and 20% are used to test the velocity forecasting. The mean absolute error (MAE) and R^2 score 0.002 and 0.98, respectively. In order to evaluate the transformer model capability with less training, the model again trained with 60% of the data, and 40% are employed for the test. In Figs. 7 and 8, the outcome of the second transformer model training in the y and the x orientation

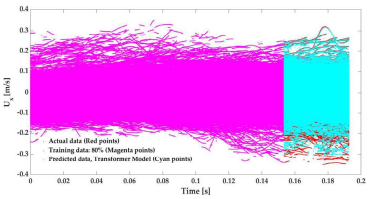


FIG. 6. The prediction of the velocity in the x direction U_x by the proposed transformer model with 80% training data and 20% test data.

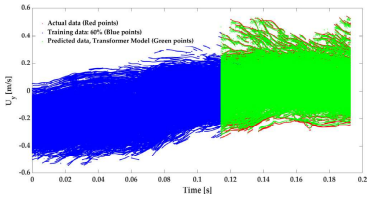


FIG. 7. The prediction of the velocity in the y direction U_y by the proposed transformer model with 60% training data and 40% test data.

is illustrated, respectively. With less training data, the MAE is 0.003, and the R^2 score is 0.98.

It must be noted the applied data from a turbulent flow with identified Taylor microscale Reynolds number range underwent deformation with a specific rate, and the experiment was conducted in the presence of gravity. In this proposed transformer model, training data did not have any direct information regarding the turbulent intensity, strain rate, and gravity effects. This is the superiority of the transformer model that can extract the dominant feature and its dependencies. Moreover, in this model, the study only applied the velocity vector as an inherent flow feature, which carried the flow properties, and can transfer these crucial features to the model and predict the next period of the turbulent flow. The predicted velocity field by the transformer model remarkably matches the actual available data based on the MAE and the R^2 score.

C. Sequential dataset and transformer model

To evaluate the transformer model as an attention mechanism in comparison to well-established sequential deep learning models, its performance is compared to LSTM and GRU models utilized in previous studies with similar datasets, focusing on physical properties and size.[23] Table I presents the results, showcasing that the transformer model achieves comparable mean absolute error (MAE), R^2 scores, and training time in predicting the velocity of strained turbulent flow

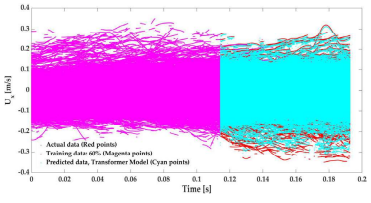


FIG. 8. The prediction of the velocity in the x direction U_x by the proposed transformer model with 60% training data and 40% test data.

TABLE I. To assess the capability of the transformer model as a mechanism for attention, a comparison is made between its performance and that of long short-term memory (LSTM)[23] and gated recurrent units (GRU)[23] from previous studies with similar datasets.

Training ratio	Performance	LSTM	GRU	Transformer
80%	MAE	0.002	0.002	0.002
	R^2 score	0.98	0.98	0.98
	Training time (s)	295	318	301
60%	MAE	0.002	0.002	0.003
	R^2 score	0.98	0.98	0.98
	Training time (s)	214	229	219

when compared to LSTM and GRU models. The LSTM and GRU models in Table I utilized identical training and test ratios as the transformer model employed in this study. LSTM and GRU models have been widely employed as sequential models in numerous studies, making them suitable benchmarks for comparison.

While the use of transformer models in the field of fluid dynamics has been relatively limited compared to the widespread adoption of LSTM and GRU models, Table I demonstrates the competence of the transformer model for such applications. However, further enhancements can be made by leveraging a larger and more diverse dataset specific to fluid dynamics. Moreover, the original work introducing the transformer model[29] highlights its parallelizability and significantly reduced training time compared to traditional models. This inherent characteristic warrants evaluation with a larger-scale dataset to fully exploit its potential benefits in fluid dynamics research.

D. Transformer and sequential models in turbulent flow and physics applications

The primary challenge of the present century lies in enhancing deep learning models, including transformers, to tackle turbulent flow and accurately predict its features. This advancement holds the potential to provide a comprehensive understanding of turbulent flow applications, thereby offering valuable insights and advancements in various domains.

One crucial application is the utilization of wind energy, where the inherent relationship between wind speed and turbulence plays a significant role. Accurate long-term and short-term forecasting of turbulent wind patterns can greatly enhance the reliability and stability of power grids, contributing to the pursuit of sustainable and efficient energy systems.

Another critical area is the study of turbulent flow over airplane wings. Understanding the intricate features and making precise predictions in this domain is essential for addressing the challenge of reducing drag force. Such advancements are instrumental in realizing goals related to green energy and minimizing fuel consumption on a broader scale.

Particle-laden turbulent flow represents an open frontier in fluid dynamics. By harnessing the capabilities of deep learning models like transformers, it becomes feasible to forecast the trajectories of particles in subsequent periods. Additionally, these models can shed light on crucial physical concepts, such as the impact of gravity, which may have been inadequately explored using traditional numerical methods.

In the field of combustion, understanding reactive flows holds great significance for controlling, predicting, and optimizing the conversion processes. Recent attention has been directed toward alternative fuels such as hydrogen, where experimental studies can be complex and costly. Leveraging the capabilities of Transformers and other sequential deep-learning models can lead to remarkable advancements in this area, revolutionizing the exploration and utilization of alternative fuels.

In summary, the integration of deep learning models, particularly transformers, into the study of turbulent flow has the potential to drive substantial progress in understanding complex flow dynamics, unlocking valuable insights, and enabling advancements in a wide range of applications.

E. Limits and enhancing Transformer and sequential models in turbulent flow

The study at hand presents a deep learning-based model utilizing the transformer architecture to forecast the velocity of deformed turbulent flow under specific conditions, encompassing the effects of gravity, defined turbulent intensity, and determined strain rate. Experimental data obtained through the LPT technique were employed as the dataset. However, compared to other deep learning methods such as LSTM, GRU, and CNN, the application of the transformer model in predicting turbulent flow is still an active area of research. It is crucial to understand the capabilities and limitations of this model in the context of turbulent flow, particularly when dealing with high Reynolds numbers characterized by heightened fluctuations. Additionally, the flow characteristics differ depending on whether it is compressible or incompressible. Previous studies applying deep learning techniques to turbulent flow have often focused on specific data or narrow ranges of Reynolds numbers. Consequently, further investigation is necessary to identify the limitations and failure points of these models in prediction tasks. It should be noted that each deep learning model applied to turbulent flow requires specific tuning, and there is no universally applicable setup. Recent developments in deep learning, such as hyperparameter tuning, have underscored its significance in optimizing model performance. Incorporating this technique can aid in identifying the most suitable model design for a given task. Expanding on the proposed model, this study suggests exploring the potential of enhancing deep learning models, including variants of LSTM, Transformer, and CNN, across a wider range of turbulent flow scenarios. Future research endeavors should strive to uncover the working range and performance boundaries of these models in turbulent flow applications.

V. CONCLUSION

This study proposed a novel transformer model from DL approaches in the context of data-driven concepts to predict the velocity field of a turbulent flow with deformation and in the presence of gravity from an experiment. Transformer model architecture is based on encoder-decoder layers and processes the data via an attention mechanism. The transformer model is state-of-the-art for sequential models and is mostly applied in language translation, and it made remarkable development in this area. In the realm of turbulent flow, the application of the transformer model is relatively nascent compared to established methods such as LSTM variants and CNN compositions. However, recent studies in fluid dynamics have demonstrated significant advancements and notable precedents in utilizing the transformer model.[31,43] Long short-term memory and convolutional neural networks are employed in many data-driven and compute-driven fluid dynamics areas.

The application of deep learning models in the field of fluid dynamics, specifically in turbulent flow, has emerged as a crucial area of study. In various fields of fluid dynamics, a combination of analytical, numerical, and artificial intelligence methods has gained traction. Several examples include analyzing flow on airplane wings, reactive flow phenomena, wind speed prediction in wind turbines, studying multiphase flow dynamics, investigating boundary layers, and exploring particle-laden turbulent flow. This integration highlights the increasing recognition of artificial intelligence as a powerful tool alongside traditional methods, empowering researchers to tackle complex fluid dynamics challenges and could open up new avenues for understanding and predicting complex flow phenomena. These models have the potential to uncover a wealth of physical concepts and facilitate accurate flow feature predictions in various domains, contributing to advancements in energy, transportation, and environmental research.

This work used only the velocity components of 2D turbulent flow measurement in a sequence way. It did not feed the other effects, such as turbulence intensity, strain rate, and gravity effect, to the model. This design relies on the concept that velocity carries and transfers the most important flow features. Moreover, the velocity measurement of the turbulent flow by devices is available in many industrial and natural applications; therefore, applying the suggested method to predict the turbulent velocity is convenient. Moreover, the model is independent of the velocity components and trains and predicts each velocity component individually. The transformer model was trained with two portions of training data, 80%, and 60%, respectively, and the rest of the data tested the velocity prediction. The error measurement illustrates the MAE and the R^2 score in 0.002–0.003 and 0.98, respectively, which is a considerable prediction and almost matches the actual data. It is extraordinary that with less training data, the transformer model is able to keep the error and prediction quantity constant and predict a period similar to a larger training ratio. It proves that the capability of the Transformer model is excellent. For future studies, it is suggested to investigate the transformer model with extensive data to evaluate its computation cost. Moreover, other turbulent flow features can be taken into model consideration and prediction. Based on the Transformer architecture and its capability, it could hopefully be useful for a deeper physical understanding of the turbulent phenomenon. The suggested method in this study could be employed broadly.

In light of the proposed model, this study advocates for an extensive exploration of deep learning models, such as various LSTM variants, transformer architectures, and CNNs, to unlock their full potential in a broader spectrum of turbulent flow scenarios. It is imperative for future research efforts to go beyond the current boundaries and unravel the working range and performance limits of these models in turbulent flow applications. By pushing the boundaries of deep learning techniques, we can gain deeper insights into their applicability and efficacy, paving the way for advancements in turbulent flow prediction and analysis. This comprehensive investigation will contribute to a more thorough understanding of the capabilities and limitations of deep learning models, allowing for their optimal utilization in real-world turbulent flow scenarios.

ACKNOWLEDGMENTS

This work was performed in the Center of Excellence (CoE) Research on AI and Simulation-Based Engineering at Exascale (RAISE) and the EuroCC 2 projects receiving funding from EU's Horizon 2020 Research and Innovation Framework Programme and European Digital Innovation Hub Iceland (EDIH-IS) under Grant Agreement Nos. 951733, 101101903, and 101083762, respectively. We thank Mr. Lahcen Bouhlali from Reykjavik University for his experimental work in data preparation.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Reza Hassanian: Conceptualization (equal); Methodology (equal); Resources (equal); Software (equal); Visualization (equal); Writing – original draft (equal). **Hemanadhan Myneni:** Software (equal); Writing – review & editing (equal). **Asdis Helgadottir:** Methodology (equal); Writing – review & editing (equal). **Morris Riedel:** Methodology (equal); Supervision (equal); Writing – review & editing (equal).

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

[1]H. T. John and L. Lumley, *A First Course in Turbulence* (MIT Press, Massachusetts, 1972).
[2]S. B. Pope, *Turbulent Flows* (Cambridge University Press, London, 2000).
[3]P. A. Davidson, *Turbulence: An Introduction for Scientists and Engineers* (Oxford University Press, London, 2004).
[4]R. Hassanian, A. Helgadottir, L. Bouhlali, and M. Riedel, "An experiment generates a specified mean strained rate turbulent flow: Dynamics of particles," Phys. Fluids **35**, 015124 (2023).
[5]R. Hassanian and M. Riedel, "Leading-edge erosion and floating particles: Stagnation point simulation in particle-laden turbulent flow via lagrangian particle tracking," Machines **11**, 566 (2023).
[6]G. I. Taylor, "The interaction between experiment and theory in fluid mechanics," Annu. Rev. Fluid Mech. **6**, 1–17 (1974).
[7]S. Goldstein, "Fluid mechanics in the first half of this century," Annu. Rev. Fluid Mech. **1**, 1–29 (1969).
[8]S. Patankar, *Numerical Heat Transfer and Fluid Flow* (CRC Press, New York, 1980).
[9]J. Anderson, *Computational Fluid Dynamics* (McGraw-Hill Education, New York, 1995).
[10]H. Versteeg and W. Malalasekera, *Introduction to Computational Fluid Dynamics, an: The Finite Volume Method* (Pearson, London, 2007).
[11]T. Kajishima and K. Taira, *Computational Fluid Dynamics: Incompressible Turbulent Flows* (Springer Cham, New York, 2017).
[12]R. Hassanian, M. Riedel, and L. Bouhlali, "The capability of recurrent neural networks to predict turbulence flow via spatiotemporal features," in *2022 IEEE 10th Jubilee International Conference on Computational Cybernetics and Cyber-Medical Systems (ICCC)* (IEEE, 2022), pp. 000335–000338.
[13]S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Comput. **9**, 1735–1780 (1997).
[14]A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems* (O'Reilly Media, Sebastopol, 2019).
[15]I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (The MIT Press, Massachusetts, 2016).
[16]T. Liu, L. Zhou, H. Tang, and H. Zhang, "Mode interpretation and force prediction surrogate model of flow past twin cylinders via machine learning integrated with high-order dynamic mode decomposition," Phys. Fluids **35**(2), 023611 (2023).
[17]K.-E. Otmani, G. Ntoukas, O. A. Mariño, and E. Ferrer, "Towards a robust detection of viscous and turbulent flow regions using unsupervised machine learning," Phys. Fluids **35**(2), 027112 (2023).
[18]H. Salehipour and W. R. Peltier, "Deep learning of mixing by two 'atoms' of stratified turbulence," J. Fluid Mech. **861**, R4 (2019).
[19]M. Raissi, Z. Wang, M. S. Triantafyllou, and G. E. Karniadakis, "Deep learning of vortex-induced vibrations," J. Fluid Mech. **861**, 119–137 (2019).
[20]H. Kim, J. Kim, S. Won, and C. Lee, "Unsupervised deep learning for super-resolution reconstruction of turbulence," J. Fluid Mech. **910**, A29 (2021).
[21]M. Z. Yousif, L. Yu, and H. Lim, "Physics-guided deep learning for generating turbulent inflow conditions," J. Fluid Mech. **936**, A21 (2022).
[22]S. Lee and D. You, "Data-driven prediction of unsteady flow over a circular cylinder using deep learning," J. Fluid Mech. **879**, 217–254 (2019).
[23]R. Hassanian, A. Helgadottir, and M. Riedel, "Deep learning forecasts a strained turbulent flow velocity field in temporal lagrangian framework: Comparison of LSTM and GRU," Fluids **7**, 344 (2022).
[24]H. Eivazi, M. Tahani, P. Schlatter, and R. Vinuesa, "Physics-informed neural networks for solving Reynolds-averaged Navier-Stokes equations," Phys. Fluids **34**, 075117 (2022).
[25]C. Duru, H. Alemdar, and O. U. Baran, "A deep learning approach for the transonic flow field predictions around airfoils," Comput. Fluids **236**, 105312 (2022).
[26]S. R. Bukka, R. Gupta, A. R. Magee, and R. K. Jaiman, "Assessment of unsteady flow predictions using hybrid deep learning based reduced-order models," Phys. Fluids **33**, 013601 (2021).
[27]R. Hassanian, A. Helgadottir, and M. Riedel, "Parallel computing accelerates sequential deep networks model in turbulent flow forecasting," in the *International Conference for High Performance Computing, Networking, Storage, and Analysis, SC22,* Dallas TX, USA, 13-18 November 2022 (IEEE, Piscataway, NJ, 2022); available at https://sc22.supercomputing.org/proceedings/tech_poster/tech_poster_pages/rpost142.html.
[28]R. Hassanian, M. Riedel, A. Helgadottir, P. Costa, L. Bouhlali *et al.*, "Lagrangian particle tracking data of a straining turbulent flow assessed using machine learning and parallel computing," in *33rd International Conference on Parallel Computational Fluid Dynamics*, Alba, Italy, 25–27 May 2022 (ParCFD, 2022); available at https://hdl.handle.net/20.500.11815/3515.
[29]A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. U. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Curran Associates Inc, 2017), pp. 11; available at https://dl.acm.org/doi/10.5555/3295222.3295349.
[30]Q. Xu, Z. Zhuang, Y. Pan, and B. Wen, "Super-resolution reconstruction of turbulent flows with a transformer-based deep learning framework," Phys. Fluids **35**, 055130 (2023).
[31]N. Geneva and N. Zabaras, "Transformers for modeling physical systems," Neural Networks **146**, 272–289 (2022).
[32]A. Hemmasian and A. Barati Farimani, "Reduced-order modeling of fluid flows with transformers," Phys. Fluids **35**, 057126 (2023).
[33]N. T. Ouellette, H. Xu, and E. Bodenschatz, "A quantitative study of three-dimensional lagrangian particle tracking algorithms," Exp. Fluids **40**, 301–313 (2006).
[34]L. Brandt and F. Coletti, "Particle-laden turbulence: Progress and perspectives," Annu. Rev. Fluid Mech. **54**, 159–189 (2022).
[35]R. F. Miotto and W. R. Wolf, "Flow imaging as an alternative to non-intrusive measurements and surrogate models through vision transformers and convolutional neural networks," Phys. Fluids **35**, 045143 (2023).
[36]P. Wu, F. Qiu, W. Feng, F. Fang, and C. Pain, "A non-intrusive reduced order model with transformer neural network and its application," Phys. Fluids **34**, 115130 (2022).
[37]C. Gu and H. Li, "Review on deep learning research and applications in wind and wave energy," Energies **15**, 1510 (2022).
[38]M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," arXiv:1603.04467 (2016).
[39]M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning" in *12th USENIX Symposium on Operating Systems Design*

and Implementation (OSDI 16) (OSDI, 2016), pp. 265–283; available at https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi.
[40]O. Kramer and O. Kramer, "Scikit-learn," in *Machine Learning for Evolution Strategies* (Springer, 2016), pp. 45–53.
[41]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," *Proceedings of the 2021 IEEE International Parallel and Distributed Processing Symposium Workshops* (IPDPSW), Portland OR, USA, 17–21 June 2021 (IPDPSW, 2021).
[42]TensorFlow, *Tensorflow Core Tutorials* (TensorFlow, 2022).
[43]M. Z. Yousif, M. Zhang, L. Yu, R. Vinuesa, and H. Lim, "A transformer-based synthetic-inflow generator for spatially developing turbulent boundary layers," J. Fluid Mech. **957**, A6 (2023).

Paper V

Optimizing Wind Energy Production: Leveraging Deep Learning Models Informed with On-Site Data and Assessing Scalability through HPC

R. Hassanian, A. Shahinfar, Á. Helgadóttir, M. Riedel

https://doi.org/10.12700/APH.21.9.2024.9.4, 2024

Optimizing Wind Energy Production: Leveraging Deep Learning Models Informed with On-Site Data and Assessing Scalability through HPC

Reza Hassanian[1,2,*], Abdollah Shahinfar[1], Ásdís Helgadóttir[1,2], Morris Riedel[1,2,3]

[1] The Faculty of Industrial Engineering, Mechanical Engineering and Computer Science, University of Iceland, Reykjavik, 102, Iceland, seh@hi.is, abs64@hi.is, asdishe@hi.is, morris@hi.is

[2] FreaEnergy, Energy, AI-HPC and CFD Solution, Mýrin, Gróska, Bjargargata 1, 102 Reykjavik, Iceland, info@freaenergy.com

[3] Juelich Supercomputing Centre, 52428 Jülich, Germany

* Correspondence: seh@hi.is

Abstract: This study suggests employing a deep learning model trained on on-site wind speed measurements to enhance predictions for future wind speeds. The model uses a gated recurrent unit (GRU) derived from the long short-term memory (LSTM) variant, and is trained using actual measured wind velocity data collected at both 10-minute and hourly intervals. The approach relies on using same-season data for predicting wind velocity, necessitating regular updates to the model with recent measurements to ensure accurate predictions in a timely manner.

The results from the prediction model, particularly at a 10-minute interval, demonstrate a significant alignment with the actual data during validation. Comparative analysis of the employed model over a two-year span, with a 24-year distinction, indicates its efficiency across different time periods and seasonal conditions, contingent upon frequent updates with recent on-site wind velocity data.

Given the reliance of sequential deep learning models on extensive data for enhanced accuracy, this study emphasizes the importance of employing high-performance computing (HPC). As a recommendation, the study proposes equipping the wind farm or wind farm cluster with an HPC machine powered by the wind farm itself, thereby transforming it into a sustainable green energy resource for the HPC application. The recommended approach in this work is enforcing the smart power grid to respond to the power demand that is connected to predictable wind farm production.

Keywords: Deep Learning; Wind Energy; Wind Turbine; Smart Grid; Renewable Energy Prediction; High-performance computing

1 Introduction

1.1 Wind Energy Resource

The paramount global challenge is climate change, and each nation bears the responsibility and capacity to invest in renewable energy as a means to mitigate the emission of greenhouse gases [1, 2].

In recent years, the remarkable expansion of wind energy has emerged as a noteworthy development in the worldwide energy scenario [3]. Wind energy currently stands as the swiftest-growing form of renewable energy, boasting a cumulative installed capacity of 763 GW in 2020—a substantial increase from the modest 24 GW recorded in 2000 [4, 5]. This extraordinary growth can be attributed to technological advancements, cost reductions, and favorable policies that encourage the shift from fossil fuels to renewable sources.

In recent times, advancements in artificial intelligence (AI) have enhanced the prediction and management of power generation in wind energy [6]. Wind power presents numerous advantages, positioning it as a compelling alternative to conventional energy sources. Unlike fossil fuels, wind energy is renewable and environmentally friendly, as it does not emit harmful greenhouse gases or pollutants. The modular and scalable nature of wind turbines makes them suitable for a diverse range of applications, spanning from large-scale utility projects to small-scale residential systems. Additionally, wind energy stands out as a dependable and cost-effective electricity source, with the leveled cost of wind energy experiencing a significant decline over the past decade [3].

Forecasts suggest that the global capacity for wind energy will achieve 2,110 GW by the year 2030, constituting roughly 20% of the world's electricity generation [7]. This upward trajectory is propelled by various factors, including the rising demand for clean energy, supportive policies, and technological advancements that contribute to the enhanced efficiency and cost-effectiveness of wind turbines [7].

1.2 Wind Turbine

Wind energy production involves converting kinetic energy from moving wind into electrical power. There are two main types of wind turbines: horizontal axis wind turbines (HAWTs) and vertical axis wind turbines (VAWTs) [1]. The efficiency of the HAWTs is much larger than that of VAWTs; however, both of these types have advantages and disadvantages [8, 9]. The power potential of a wind turbine is proportional to the cubic power of the wind velocity [10]. Wind speed has a turbulence behavior and diverse fluctuations [11]. Moreover, the

power potential of the wind turbine is proportional to the density of air. As a result, cold air has a higher wind power potential than warm air [10]. These nonlinear and random features of the wind makes its forecasting a crucial issue for wind power producers.

Power production from wind farms depends on the wind velocity. Furthermore, it is crucial issue that the producer be aware of the kind of farm production to respond to the demand for electricity on the power grid. Moreover, the smart grid technology is a function of the smart components that supply the power grid. In fact, if the wind farm has the capability to predict the wind speed in the short and long-term, it has forecasting for electricity production [3, 6]. This leads to smart wind farm production and enforcement of the smart power grid [12]. Figure 1 displays how wind power production prediction could assist the power supplier in managing the response to the power grid demand.

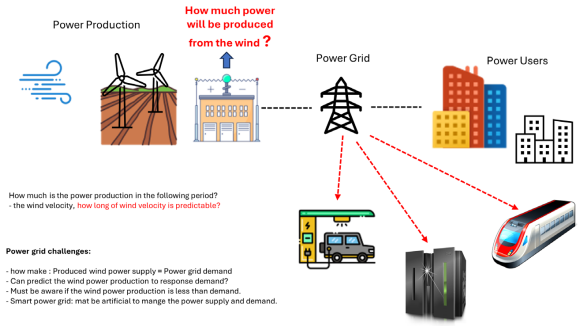


Figure 1
A schematic representation of how wind power production prediction could assist the power supplier in managing the response to the power grid demand

1.3 Deep Learning and Wind Farm

The deep learning model, based on the sequential models, displayed the successful capability to predict the nonlinear phenomenon [13]. In order to optimize the accuracy of the DL model for the wind velocity, using the appropriate period and size of the data is essential. Additionally, depending on the wind park location, the wind speed has a different pattern for monthly, seasonal, bi-annual, and annual datasets. Based on the author's experiences in Nordic countries like Iceland, the wind speed in the winter is extremely higher than in the summer. Because of this

difference, the previous study demonstrated a DL model for summer that should not be used for winter prediction [6]. Thus, it is essential to have an online and updated DL model for a wind park. This leads to updating the DL model with measured data from many years ago to a few minutes before.

1.4 Literature Review

In recent years, the DL model for wind velocity forecasting was developed with different DL layers architecture [6, 14]. The majority of the available studies focused on short-term prediction [6, 15]. The dataset used to train the DL model consists of 5-10 minutes and 1-2 hours [6]. The measured data in the previous studies from onshore wind farms [6, 16]. The literature displays 1-6 hour prediction with different DL models. However, there is no universal model to be used globally, and they are specified for a particular site location where trained data has been measured [6, 17].

Looking at the above-mentioned aspects of the proposed DL model for wind speed prediction leads to a novel approach and perspective proposal. Since the wind farm's location, air temperature, month, season, and year of the measured data impact the prediction [3, 6], it dictates an essential local DL model design for each specified wind farm, and the model training must be updated per hour or daily.

The present study proposes a DL model for wind velocity prediction that is updated with training data depending on effective factors such as hour, daily, cold, and warm air and season. The result of the study is a remarkable capability that can cause long-term prediction in addition to short-term forecasting. Hence, this paper is organized as follows. The applied methodology is presented in Section 2. In Section 3, the result and discussion are presented and at the end the conclusion is presented.

2 Methodology

2.1 Measured Wind Velocity

This study applies on-site measured wind velocity data from the Vedurstofan (the Metrological Office) of Iceland. The data involves a time step of 10 minutes for specific years and an hour time step for other years. Figure 2 displays December 1995 to February 1996 and December of 1996 to February 1997.

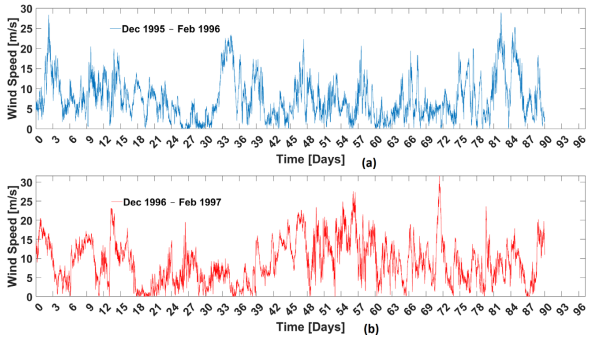


Figure 2
Presentation of the measured wind velocity for two different time periods in Iceland. (a) December 1995 to February 1996, (b) December 1996 to February 1997

These are the same period of time for two years. It can be seen that the wind velocity does not have a similar pattern to be able to use the previous year's data and simulate the next year.

Moreover, two different time periods (seasons) can be seen in Figures 3 and 4. These presentations reveal how owning a distinct pattern is from September 1996 to November 1996 to December 1996 to February 1997. In Figure 4, the same period of September 2021 to February 2022 is displayed. The illustration of these two figures uncovers that the wind speed has nonlinear and random features, and there is no known equation or pattern to use the previous wind velocity of the earlier time to simulate the next time.

As pointed out in the introduction, in recent years, DL networks have been employed to predict a sequential nonlinear dataset, such as wind speed, which has turbulence behavior in the fluid dynamics area. However, the models depend on the specific site location and measured data. The present study suggests using online and recent data to train each wind park's DL model to overcome this defect. To make this application possible in the actual wind farm, it is essential to connect the measured data online to the DL model and update the training in a short time. Additionally, this study would emphasize the fact that using training data from the same period of time will be much more efficient. For example, the speed data from the summer train in the DL model may not be sufficient to predict the wind speed in the winter and needs to be merged with data from winter time. This concept is used in the current study.

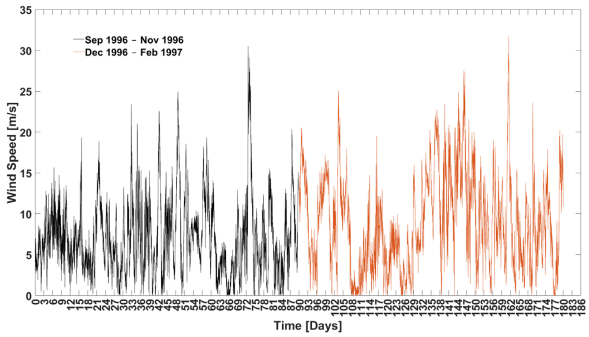


Figure 3

On-site measured wind velocity for two different seasons in Iceland, black color curve, September 1996 to November 1996 and red color curve, December 1996 to February 1997

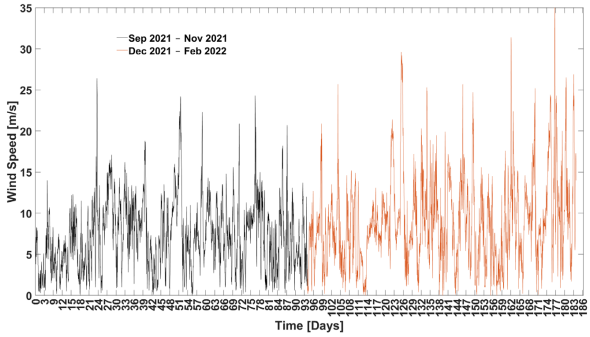


Figure 4

On-site measured wind velocity for two different seasons in Iceland, black color curve, September 2021 to November 2021, and red color curve, December 2021 to February 2022

2.2 Deep Learning Models

Among available DL models for sequential data, LSTM for sequential nonlinear and random datasets displayed successful application. Additionally, the Transformer as an up-to-date DL model from the attention mechanism provided appropriate prediction for the random sequential dataset. The current study employs a gated recurrent units (GRUs) model trained with on-site measured wind velocity and forecasts the wind speed for the following period of time. Based on the literature, GRU is a variant of LSTM and has a simpler architecture. It is reported that GRU has the same efficiency as LSTM with less data.

The model has been assessed with two datasets, one from 1996 with a time step of 10 minutes and the second dataset from 2021 with an hour time step. For each model, 60% of the data was used for training, and 40% of the rest of the data was employed for testing the model prediction. The present study shows that reducing the training data ratio to lower than 60% will reduce the model prediction accuracy. The mean absolute error and squared R (R^2) are measured as metrics for the models. Figure 5, a diagram shows the required dataset for DL model training with HPC resources, and the target is a prediction of the wind speed.

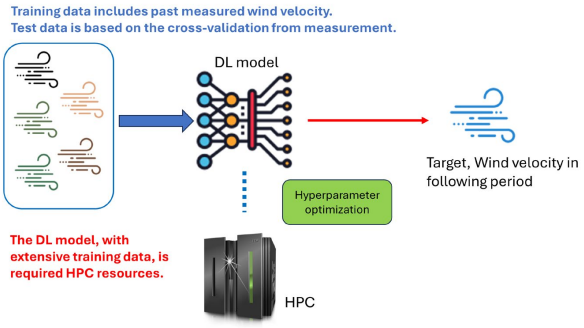


Figure 5
The DL model is trained and tested with measured wind speed. Training data are 60% of the measured data, and 40% of the rest of the data are used to test the model prediction

2.3 High-Performance Computing in Wind Farm

The sequential model of DL will lead to an accurate model with a larger amount of training data. The extensive training data and the DL architecture make it essential to use high-performance computing (HPC). As discussed earlier in this study, the suggestion is to use an online DL model training with up-to-date measured data at the wind farm site.

Having access to HPC to train a DL model with extensive data that is related to scalability is a crucial issue.

However, since the wind frame produces power, it will be an option for each wind farm to own its HPC system or install an HPC system for cluster wind farms that share the computing between them; this will make the HPC supported with green energy, which is a remarkable achievement since many of the HPC clusters using traditional and fossil fuel resources.

3 Result and Discussion

This section presents and discusses the result of the proposed approach, which is composed of the on-site measured data and GRU model.

Figure 6 shows the GRU model result that used measured wind speed data with a period of September 1996 to November 1996 with time step 10 minutes. To train the GRU model, 60% of the data is used to train the GRU model, and 40% to validate the model prediction. The metric evaluation shows MAE 0.019 and R^2 is 0.97. this model used data with short time steps.

Figure 7 illustrates the wind velocity prediction result of the GRU model that trained with actual wind speed from in-site measurement with a period of September 2021 to November 2022 with time step an hour. In this model, 60% of the data is used for training and 40% as validation. The prediction of the model has MAE 0.059 and R^2 is 0.71. This model used data with longer time steps.

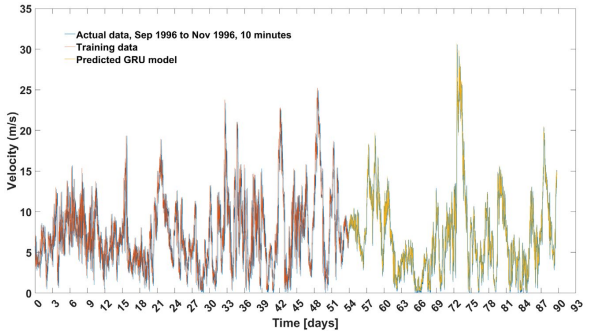


Figure 6

Presentation of GRU model prediction that is trained with on-site measured wind velocity with time step 10 minutes with a period of September 1996 to November 1996, with 60% training ratio and 40% validation. Blue colors the actual measured data, red colors the training ratio, and yellow colors the prediction of the model.

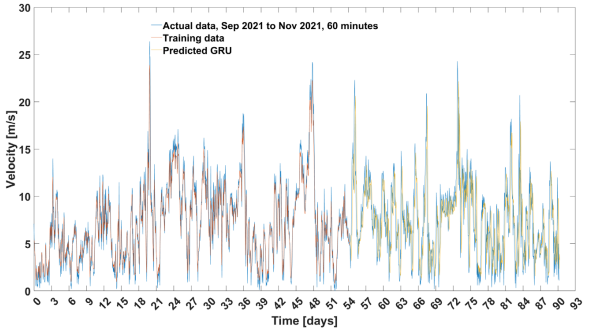


Figure 7

Representation of GRU model prediction that is trained with on-site measured wind velocity with time step an hour with a period of September 2021 to November 2022, with 60% training ratio and 40% validation. Blue colors the actual measured data, red colors the training ratio, and yellow colors the prediction of the model.

The illustrated results in Figures 6 and 7 show a remarkable observation. The GRU model that trained with shorter time steps is much more accurate. In contrast, to the model trained with an hour time step, shorter time step data caused a 36% increase in the R^2 and a 67% decrease in the MAE. Therefore, measuring in-site and speed with short time steps makes the prediction model more efficient and accurate.

Additionally, the current study used one GRU model with a distinct period of time from 24 years ago (1996 and 2021). For each model, the training data was updated, and the model resulted in an appropriate wind speed prediction. These remarkable achievements show that the training update significantly affects the model's accuracy. It could be taken into account that the season of the data for wind speed training should match the target wind speed time.

Conclusions

The current study proposes an approach to using time series data of in-site measured wind speed to predict the wind velocity in the following period of time with the application of deep learning capability. A GRU model from the LSTM variant was designed and trained with a specific ratio of the measured data, and its prediction was validated by the actual data.

The superiority of the present work suggests the use of updated data to predict wind velocity. Furthermore, the study used data from the same season (winter or summer) to train and predict the wind velocity. The study results uncovered that the shorter time step, 10 minutes, makes the model extremely accurate than the model trained with a longer time step, an hour (60 minutes).

The present study recommends using a DL model as software in wind frames that are trained with updated measured wind speed via online connection and updated training to be able to have short and long-term predictions with desirable accuracy. This application makes it possible for the wind energy producer to have a period of wind velocity and wind energy production, and this capability leads to an efficient smart power grid to respond to the power demand. It is planned to evaluate the wind speed prediction via updated data with a Transformer as an attention mechanism and compare it to LSTM variants.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgement

This work was performed in the Center of Excellence (CoE) Research on AI and Simulation-Based Engineering at Exascale (RAISE) and the EuroCC 2 projects receiving funding from EU's Horizon 2020 Research and Innovation Framework Programme and European Digital Innovation Hub Iceland (EDIH-IS) under grant agreement no. 951733, no. 101101903 and 101083762, respectively.

Conflicts of Interest

The authors declare no conflicts of interest.

References

[1] N. R. Deevela, T. C. Kandpal, and B. Singh, "A review of renewable energy based power supply options for telecom towers," Environment, Development and Sustainability, Vol. 2023, 2023

[2] U. P. United Nations Climate Change. (2015) The paris agreement. [Online] Available: https://unfccc.int/process-and-meetings/the parisagreement

[3] R. Hassanian, Ásdís Helgadóttir, and M. Riedel, "Iceland wind farm assessment case study and development: An empirical data from wind and wind turbine," Cleaner Energy Systems, Vol. 4, p. 100058, 2023

[4] I. E. A. (IEA). (2021) Global energy review 2021 [Online] Available: https://www.iea.org/reports/global-energy-review-2021

[5] G. W. E. Council. (2021) Global wind report 2021 [Online] Available: https://gwec.net/wp-content/uploads/2021/03/GWEC-Global-Wind-Report-2021.pdf

[6] C. Gu and H. Li, "Review on deep learning research and applications in wind and wave energy," Energies, Vol. 15, No. 4, 2022

[7] U. Bub, A. Picot, and H. Krcmar, "The future of telecommunications," Business Information Systems Engineering, Vol. 3, No. 5, pp. 265-267, 2011

[8] R. Hassanian, Helgadóttir, and M. Riedel, "A wake loss model asymmetry induced by the circulation of a vertical axis wind turbine," in 2023 International Conference on Future Energy Solutions (FES), 2023, pp. 1-5

[9] R. Hassanian, H. Myneni, Helgadóttir, and M. Riedel, "Vertical axis wind turbine powers telecom towers: Green and clean configuration," in 2023 6th International Conference on Electrical Engineering and Green Energy (CEEGE), 2023, pp. 114-118

[10] M. Kano˘glu, Y. A. Çengel, and J. M. Cimbala, Fundamentals and Applications of Renewable Energy. New York: McGraw-Hill Education, 2020

[11] R. Hassanian, H. Myneni, Helgadóttir, and M. Riedel, "Deciphering the dynamics of distorted turbulent flows: Lagrangian particle tracking and chaos prediction through transformer-based deep learning models," Physics of Fluids, Vol. 35, No. 7, p. 075118, 07 2023

[12] Muhammad Kamran, Fundamentals of Smart Grid Systems. Cambridge: Academic Press, 2022

[13] R. Hassanian, Helgadóttir, and M. Riedel, “Deep learning forecasts a strained turbulent flow velocity field in temporal lagrangian framework: Comparison of lstm and gru,” Fluids, Vol. 7, No. 11, 2022

[14] Erlong Zhao, Shaolong Sun, Shouyang Wang, New developments in wind energy forecasting with artificial intelligence and big data: a scientometric insight, Data Science and Management, Vol. 5, Issue 2, 2022, pp. 84-95

[15] Chen Xiaojiao, Zhang Xiuqing, Dong Mi, Huang Liansheng, Guo Yan, He Shiying, "Deep Learning-Based Prediction of Wind Power for Multi-turbines in a Wind Farm", Frontiers in Energy Research, Vol. 9, 2021

[16] Paula, M.; Casaca, W.; Colnago, M.; da Silva, J. R.; Oliveira, K.; Dias, M. A.; Negri, R. Predicting Energy Generation in Large Wind Farms: A Data-Driven Study with Open Data and Machine Learning. Inventions 2023, 8, 126

[17] Puri, V., Kumar, N. Wind energy forecasting using artificial neural network in himalayan region. Model. Earth Syst. Environ. 8, 59-68 (2022) https://doi.org/10.1007/s40808-020-01070-8

Paper VI

Turbulent Flow Prediction-Simulation: Strained flow with Initial Isotropic Condition Using a GRU Model Trained by an Experimental Lagrangian Framework, with Emphasis on Hyperparameter Optimization

R. Hassanian, M. Aach, A. Lintermann, Á. Helgadóttir, M. Riedel

https://doi.org/10.3390/fluids9040084, 2024

fluids

MDPI

Article

Turbulent Flow Prediction-Simulation: Strained Flow with Initial Isotropic Condition Using a GRU Model Trained by an Experimental Lagrangian Framework, with Emphasis on Hyperparameter Optimization

Reza Hassanian [1,*,†], Marcel Aach [1,2,†], Andreas Lintermann [2,†], Ásdís Helgadóttir [1,†] and Morris Riedel [1,2,†]

[1] The Faculty of Industrial Engineering, Mechanical Engineering and Computer Science, University of Iceland, 102 Reykjavik, Iceland; asdishe@hi.is (Á.H.); morris@hi.is (M.R.)
[2] Juelich Supercomputing Centre, 52428 Jülich, Germany
* Correspondence: seh@hi.is
† These authors contributed equally to this work.

Abstract: This study presents a novel approach to using a gated recurrent unit (GRU) model, a deep neural network, to predict turbulent flows in a Lagrangian framework. The emerging velocity field is predicted based on experimental data from a strained turbulent flow, which was initially a nearly homogeneous isotropic turbulent flow at the measurement area. The distorted turbulent flow has a Taylor microscale REYNOLDS number in the range of $100 < Re_\lambda < 152$ before creating the strain and is strained with a mean strain rate of 4 s^{-1} in the Y direction. The measurement is conducted in the presence of gravity consequent to the actual condition, an effect that is usually neglected and has not been investigated in most numerical studies. A Lagrangian particle tracking technique is used to extract the flow characterizations. It is used to assess the capability of the GRU model to forecast the unknown turbulent flow pattern affected by distortion and gravity using spatiotemporal input data. Using the flow track's location (spatial) and time (temporal) highlights the model's superiority. The suggested approach provides the possibility to predict the emerging pattern of the strained turbulent flow properties observed in many natural and artificial phenomena. In order to optimize the consumed computing, hyperparameter optimization (HPO) is used to improve the GRU model performance by 14–20%. Model training and inference run on the high-performance computing (HPC) JUWELS-BOOSTER and DEEP-DAM systems at the Jülich Supercomputing Centre, and the code speed-up on these machines is measured. The proposed model produces accurate predictions for turbulent flows in the Lagrangian view with a mean absolute error (MAE) of 0.001 and an R^2 score of 0.993.

Keywords: turbulent flow; prediction; deep learning; simulation; high-performance computing

check for updates

Citation: Hassanian, R.; Aach, M.; Lintermann, A.; Helgadóttir, Á.; Riedel, M. Turbulent Flow Prediction-Simulation: Strained Flow with Initial Isotropic Condition Using a GRU Model Trained by an Experimental Lagrangian Framework, with Emphasis on Hyperparameter Optimization. *Fluids* **2024**, *9*, 84. https://doi.org/10.3390/fluids9040084

Academic Editors: Martin Skote and Rob Poole

Received: 5 February 2024
Revised: 6 March 2024
Accepted: 28 March 2024
Published: 1 April 2024

1. Introduction

Turbulent flow is a high-dimensional and nonlinear phenomenon [1]. It can be found in many artificial and natural applications, and it is therefore of great interest to study its features [1–3]. All turbulent flows have random characteristics, rendering deterministic approaches impossible to apply. Therefore, existing analyses rely on statistical methods addressing the energy cascade theory [1,2]. The use of computational fluid dynamic (CFD) methods is a convenient approach for simulating turbulent flows, mainly via direct numerical simulation (DNS) and large eddy simulation (LES) [1]. Although LES is less accurate than DNS, both methods require extensive computing [4] on high-performance computing (HPC) systems. Solving Reynolds-averaged Navier Stokes (RANS) equations is a cheap method used widely in the industry, though it does not provide results on the level of accuracy of LES or DNS. The size and scalability of HPC systems is continuously growing, allowing for more and more fine-grained simulations. However, current numerical

methods are far from being able to compute every CFD problem, especially those featuring highly complex and detailed flow structures [4,5]. Furthermore, in many CFD applications, a validation of the solution via empirical data is essential, which is another disadvantage [4]. Experiments are frequently used to study the turbulent flow. However, due to their scale and size limitations, they can only be applied to particular and size-limited problems [1,3,6,7]. These constraints underpin the demands for a reliable tool to overcome the obstacles mentioned above and analyze turbulent flows in a broader range of scales [4]. Several methods extract the dominant features via the reduced-order model (ROM). Proper orthogonal decomposition (POD), dynamical mode decomposition (DMD), and Koopman analyses are some of the well-known techniques to yield ROM [8]. Moreover, dimensionality reduction, feature extraction, super-resolution, applying ROM, turbulence closure, shape optimization, and flow control are some of the crucial tasks in CFD [9].

In many areas, deep learning (DL) models have demonstrated an extensive capability to extract hidden features from nonlinear events and create predictions [8,9]. The applicability of DL models has also been studied in fluid dynamics [4]. Recent studies show that with DL, model-free predictions of spatiotemporal dynamical systems, particularly for high-dimensional, dynamical systems [8], are possible. Recurrent neural networks (RNN) are neural networks composed of an individual hidden layer with a feedback loop, in which the hidden layer output and the current input are turned to the hidden layer [9]. They are well-suited for sequential datasets [9]. They determine a temporal relationship, as they learn from sequential input data and are characterized by featuring three weight metrics and two biases. However, RNN cannot learn long-range temporal dependencies from sequential data due to the vanishing gradient problem [9]. The long short-term memory (LSTM) model was developed in 1995 [10]. It features a gating structure to control the recurrent connector transients. In contrast to RNN, vanishing gradients are avoided. It is therefore a proper tool to model longer temporal dependencies [9]. Gated recurrent unit (GRU) models are variants of LSTM models that work with fewer parameters [11,12]. Besides, in GRU architectures, the forget and input gates of LSTM are altered only with one update gate. In the literature, it has been shown that GRU models can be trained faster while still achieving results similar to LSTM, even with fewer training data [12]. Duru et al. [13] apply DL to predict the transonic flow around airfoils. Srinivasan et al. [9] use Multilayer Perceptron (MLP) and DL networks to predict a turbulent shear flow from equations known from a Moehlis model [14]. LSTM' susceptibility has led to hybrid models such as autoencoders-LSTM, LSTM/RNN, and Convolutional Neural Network (CNN)-LSTM [12]. Eivazi et al. [8] present a DL application for the nonlinear model reduction in unsteady flows. The review of Gu, Chengcheng, and Li, Hua [12] reports on an LSTM network being applied to predict the wind speed, which has turbulent behavior. Bukka et al. [5] employ a hybrid, deeply reduced model to predict unsteady flows. Most fluid flow studies that applied DL use data extracted from CFD computations [4,9]. Furthermore, most works include pre-processing steps to identify the dominant features, such as POD or DMD [4]. Recently, Hassanian et al. [15] used LSTM and GRU models to predict a turbulent flow with only temporal features. Moreover, the Transformer model, as an up-to-date DL technique, displays successful capabilities to simulate and forecast emerging unknown patterns of turbulent flow [16].

This study proposes an innovative idea, using a GRU model to predict turbulent flows with spatial-temporal data based on raw data from flow measurements in an experiment of strained turbulent flow [17]. The Lagrangian particle tracking (LPT) technique is applied to extract 2D (two components of each property, such as velocity) from the 3D experiment (consisting of all components of each property) of the strained turbulent flow. As the turbulent flow manifests as a three-dimensional phenomenon, employing experimental data yields a dataset containing authentic and comprehensive turbulence characteristics. The data contain information on the time t, location x and y, and the velocity components in the X and the Y directions. The Lagrangian framework is defined by particle traces in a spatiotemporal way [6,7]. A particle in the flow with a specific velocity and position at

each particular time t is followed [1,6]. This way, the particle's velocity over time can be represented as a time series [4], which is a function of the particle's location. Relying on this concept, a GRU model can be trained with the spatiotemporal data and predict the velocity. The velocity time series in fluid dynamics have been recorded in several biological and industrial applications via special devices [4] and can be used in combination with the suggested model. Since the turbulent flow is a nonlinear problem and there is no deterministic approach to solve or forecast the emerging period of its feature, the suggested method in the present study provides a transparent window to study turbulent flow.

In prior research on turbulent flow employing deep learning models, a hybrid approach incorporating Proper Orthogonal Decomposition (POD), Reduced Order Modeling (ROM) [18], and deep learning techniques was employed to address nonlinear parametrized Partial Differential Equations (PDEs) [19,20]. The superiority of this proposed method is that it eliminates the steps of extracting the dominant data and the necessary pre-processing steps before the application of DL, and directly provides predictions of the future flow through DL models. This advantage renders the model adaptable for training with raw measurement data, eliminating the need for processing, such as ROM or POD. Its novelty in applying training data for a DL model is based on spatio-temporal attributes. In sequential DL models such as LSTM and GRU, the training data are times series and, therefore, temporal. The current study employs the spatial attributes of the turbulent flow since, in the Lagrangian framework, the location is a function of the time. Furthermore, the preeminence of the present study is utilizing the GRU model to be trained with measured property, forecasting it in the following period without training, and informing the model with flow characteristics such as the Reynolds number, Stokes number, length, or time scale. In many industries and applications, fluid flow properties such as velocity, flow rate, vorticity, and acceleration can be measured with technical devices. This consistency helps the proposed approach to be broadly utilized. The experimental dataset used in the present study stems from a strained turbulence flow in the presence of gravity and tracking tracer particles. However, the prediction model only relies on the velocity and location time series, and the training does not include parameters such as particle size, turbulence intensity, gravity, and strain rate. The parallel computing machines JUWELS-BOOSTER and DEEP-DAM [21] from the Jülich Supercomputer Centre are used to accelerate the GRU model training process. Hence, this manuscript is organized as follows. The applied methodology is introduced in Section 2. Subsequently, the results and discussion are provided in Section 3. Finally, conclusions are drawn in Section 4.

2. Methodology

This section represents the theory of the LPT, which is used to employ a dataset from the experiment in this study. Furthermore, the dataset details have been explained. Thus, the employed GRU model and its setup for training and prediction have been demonstrated.

2.1. The Lagrangian Framework and Fluid Particles

In a Lagrangian framework, individual fluid particles' position and velocity vectors are tracked [1,6]. A fluid particle is a point that streams with the local flow velocity; thus, it identifies the velocity and position at time t. The arithmetic definition of a fluid particle is [1,4]:

$$U_i = U_i(t, x_1(t, x_{1,0}), x_2(t, x_{2,0}), x_3(t, x_{3,0})), \tag{1}$$

where the velocity U is determined in 3D coordinates, x is the position vector, t is the time, and i specifies the vector components in the X, the Y, and the Z directions. Notation (1) defines the particle velocity in sequential time series and is frequently used in turbulent flow statistics, where no universal velocity function is available. $x_{i,0}$ ascertains the initial condition of the particle in the i direction. Figure 1 displays a sketch of the strained turbulent flow.

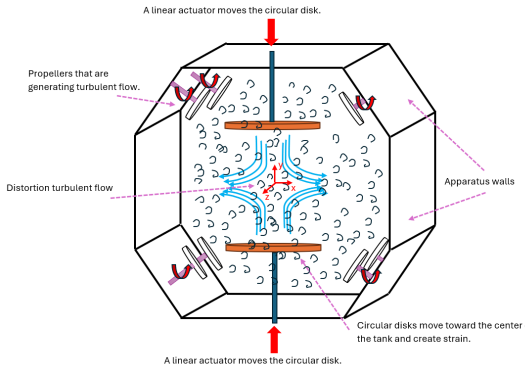


Figure 1. A sketch displays the strain acting on the turbulent flow. The turbulent flow at the measurement area, located at the center of the tank, is a nearly stationary homogeneous isotropic turbulence flow initially and before the distortion. The measured data are used in the current study to train a GRU model.

2.2. Experimental Data

The experiment was conducted within a water tank featuring eight impellers strategically positioned at the corners of a cube and directed toward the tank's center, as displayed in Figure 1. These impellers rotated at specific speeds falling within the range of $100 < Re_\lambda < 152$, effectively simulating the turbulent flow before creating the strain deformation. The resulting flow in the central region of the tank, where measurements were taken, exhibited a nearly stationary homogeneous isotropic turbulence [22]. The tank, with dimensions of 60 cm $\times$ 60 cm $\times$ 60 cm, had transparent Plexiglas for XT walls that were 20 mm thick, allowing optical access to the data. An aluminum frame held the components of the turbulence box in place. The fluid in the tank was seeded with tracer particles with median diameters of 8–10 µm. Tracer particles had a specific gravity of 1.1 g/cm^3 (hollow glass). Two circular flat disks positioned vertically in the center of the tank moved towards each other, generating a specified mean strain rate. The experiment involved a mean strain rate, primarily in the y-direction of -4 s^{-1}. The measurement area, situated in the center of the tank, had dimensions of 24.5 $\times$ 24.5 mm^2.

The Lagrangian particle tracking (LPT) technique was employed to monitor and extract the dynamic features of the particles. Lagrangian Particle Tracking (LPT) [23–25] is a non-intrusive optical methodology that is widely utilized in experimental fluid dynamics. This technique involves capturing images of particles suspended in a fluid and subsequently tracking the movement of individual particles within a small interrogation window. In the context of two-dimensional LPT, the flow field is observed within a thin plane illuminated by a laser sheet, allowing for the measurement of particle motion within that specific slice of the flow. Introducing low-density particles into the flow of interest allows each particle to be individually tracked across multiple frames.

In this particular experiment, a single camera was utilized to reconstruct particle tracks in two dimensions, providing valuable insights into the initial turbulence and Lagrangian statistics of the turbulent flow. The construction of particle tracks in 2D-LPT involves two primary tasks. Firstly, the images captured by the camera undergo processing to determine the two-dimensional positions of the particles within the camera's image space. Secondly, a tracking algorithm, based on the principle of the 4-frame best estimate pioneered by

Ouelette et al. [25], is applied to establish the paths followed by the particles over time using a sequence of images.

A solitary high-speed CMOS camera equipped with a 105 mm focal length lens was employed to capture LPT images, set at a resolution of 512 × 512 pixels. The detection system operated at 10 kHz, equivalent to 10,000 frames per second (fps), ensuring well-resolved particle velocity and acceleration statistics. This exceptionally high temporal resolution (0.1–0.2 ms) is significantly smaller than the Kolmogorov time τ_λ (16.6–31.6 ms) of the smallest eddies in the flow, allowing for the resolution of dissipation range properties. It is reported that the STOKES number (relaxation time over the Kolmogorov scale) for the tracer particles is in the range of 0.0063–0.0094 [17]. For illuminating the tracer particles; an Nd-YAF laser (527 nm) was utilized, synchronized at the same sampling frequency as the camera. The laser operated in an internal mode, with a 14 A Q-switch current and a pulse width of 2.5 µs. To ensure accurate statistics of the particle-laden turbulent flow, the recording process was iterated 20 times for each flow case. It is important to note that the present study uses a dataset to train a GRU model originated from the LPT measurement based on Ouelette et al. [25] and Hassanian et al. [17]. The original work [17] details the experiments and their measurements.

2.3. Sequential Velocity Dataset

The velocity dataset is extracted from the LPT experiment described in the previous section, following the procedure of Hassanian et al. [17]. The dataset is composed of 6,225,457 tracking points for every vector, as follows:

- Velocity component in the Y direction, V_Y;
- Velocity component in the X direction, V_X;
- Location in the x coordinate;
- Location in the y coordinate;
- The time vector specifies the time t for every tracking point.

These tracking points comprise velocity and location vectors, attained via 20 recordings to provide sufficient statistical data. As expected, the tracking yields several tracking lines, as illustrated in the result section, and every tracking line specifies the fate of a single particle. This study employed different ratios of the training dataset to determine the optimal model with accurate predictions for the strained turbulent flow. To measure the performance of the forecasting model, the data are split into 80% training data and 20% test data. The prediction quality of the model is evaluated on the unseen test data. The model is trained in a way that individually predicts the velocity in the X direction and the Y direction. This design makes the model applicable to higher-dimensional data. For instance, if there are data with a third component in the Z direction, this model can forecast the corresponding velocity component in a separate training run. It should be noted that the dataset in this study underwent strain deformation in the Y direction, which is the dominant orientation in this flow; therefore, it is expected to see more fluctuation in this direction [3,17].

2.4. Gated Recurrent Unit Model

The study relies on the concept that the flow properties in the Lagrangian frameworks are carried by the velocity, which is a function of time and location. Therefore, the input data from the 2D measurement involves the location in the x and the y coordinates in addition to velocity components in both orientations. The current study trained a DL model on these data to assess the ability to forecast flow fields, because the concept of sequentiality is an inherent feature in the Lagrangian framework. The DL model thereby takes into account all historical impacts. Despite the mean strain rate, turbulence intensity, geometry of the boundary condition as an effectiveness parameter [26,27], and gravity as a presence effect [17], they are not part of the model input. The only inputs to train the model are locations and the velocity. The target is the velocity in the future. A GRU is based on the LSTM model with slight changes in the architecture [28]. The literature reports that a

GRU is faster to compute than an LSTM and has a streamlined model [11,12,29]. A GRU cell, which is displayed in Figure 2, is composed of a hidden state h_{t-1}, a reset gate r_t, and an update gate z_t. The reset gate controls how much of the previously hidden state is remembered. Via the update gate, it can be quantified how much of the new hidden state h_t is just a copy of the old hidden state. This architecture establishes two significant features: the reset gate captures short-term dependencies and the update gate models' long-term dependencies in sequences [28].

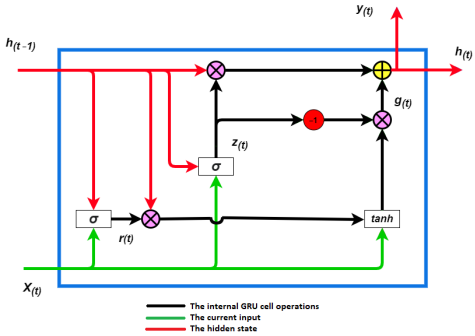


Figure 2. Architecture of a GRU model: $h_{(t-1)}$ is the hidden state from the previous step, $X_{(t)}$ is the current input, $h_{(t)}$ is a new hidden state, $y_{(t)}$ is the output, $r_{(t)}$ is the reset gate, $z_{(t)}$ is the update gate, $g_{(t)}$ is the candidate hidden state, σ is the sigmoid function, and *tanh* is the hyperbolic tangent function [15].

2.5. Forecasting Model Set Up and Parallel Computing

The models are coded in Python with the TensorFlow library [30,31]. The GRU model is set up with 100 layers and one dense layer, and Adam is specified as an optimizer [15]. The dataset was normalized by the MinMaxScaler transformation [32], scaling the minimum and maximum values to be 0 and 1. In the GRU model, *kernel_initializer* is $glorot\_uniform$, and the learning rate is 0.001. Since the model training runs on the JUWELS-BOOSTER [33] and DEEP-DAM [21] machines, a distribution strategy from the TensorFlow interface to distribute the training across multiple GPU with custom training loops is applied [34]. The training has been set up to use 1 to 4 GPU on one node. The result of the computing and the models' performance distinction are reported in Section 3.

3. Results

The current study makes use of a dataset from an LPT measurement, which provides spatial and temporal information. The visualization of the velocity that is measured in the X and Y directions is obtained to observe the flow turbulency behavior. The velocity in a specific direction at location x and y is used as input training data with a ratio of 80%. The velocity prediction was evaluated with the rest of the data (20%). The trained model performs the forecast for both velocities individually. In this section, the results and discussion are presented.

3.1. Measured Turbulent Flow Velocity

The subject of this study is to employ the dataset from the experiment in the training of the GRU model and to analyze its training and predictive performance. The data extracted from the experiments contain the velocities of tracer particles in the Lagrangian

framework [17]. Figures 3 and 4 illustrate the measured velocity component in the X and Y directions, respectively.

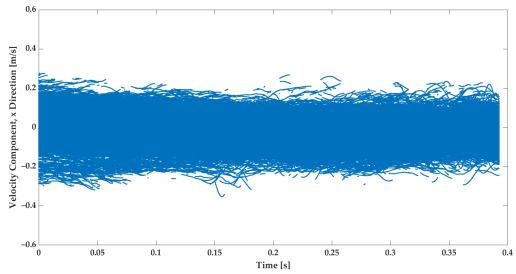


Figure 3. The measured velocity in the X direction from 20 videos for strained turbulent flow. The experiments have been repeated in analogous conditions.

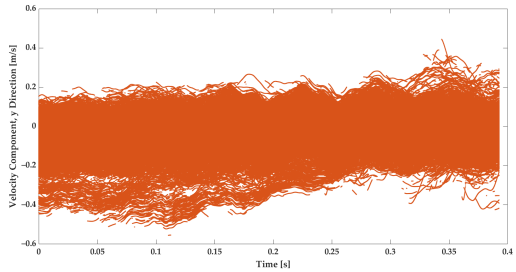


Figure 4. The measured velocity in the Y direction from 20 videos for strained turbulent flow. The experiments have been repeated in analogous conditions.

The velocity measurements in the X and Y directions both show fluctuations. Comparing Figures 3 and 4 reveals that in the Y direction, the turbulence is more intense. This is due to the fact that the strain direction mainly points to this orientation [17]. That is, the velocity in the Y direction has a gradient that is caused by the strain. It is, therefore, much more visible than the velocity in the X direction. The literature emphasizes that the strain could lead to extra fluctuations [2,3,17]. Besides the strain and turbulence intensity, the geometry boundary influences the flow velocity [3].

3.2. Predicted Velocity and GRU Model Evaluation

Figures 5 and 6 illustrate that 80% of the velocity time series are used to train the GRU model in this study. The rest of the data (20%) are applied as test data to assess the predicted velocity via the GRU model.

The model provides considerably accurate velocity forecasting. The MAE and the R^2 score metrics are applied to evaluate the model; with 80% training data, the MAE and R^2 scores are 0.001 and 0.993, respectively. It must be noted that the actual data in Figures 5 and 6 are in the filled blue circles and are because of the high level of the predic-

tions covered by the prediction.To evaluate the designed GRU model, its performance is compared to model applications from previous studies that used LSTM, GRU, and Transformer models trained only with temporal features. The comparison is displayed in Table 1. In the present study, the dataset included 6,225,457 tracking points and four sequential variables composed of x, y, V_X and V_Y to predict the V_X and V_Y in the following periods. The model of this work is tuned for performance in terms of the runtime and accuracy with HPO, evaluating different batch sizes, $BS = [8, 16, 32, 64, 128, 256, 512, 1024]$. The accuracy of the model, trained with the optimal batch size found, is specified by GRU-h in Table 1. From the previous study of the author's research group, LSTM, GRU, and Transformer models have been applied with 2,862,119 tracking points, with two sequential variable inputs (temporal feature) composed of V_X and V_Y to predict the V_X and V_Y [15,16]. Table 1 shows that the GRU-h model of this study is 20% faster than the GRU model with a smaller dataset, and it is 14% and 15% faster than the LSTM and Transformer models, respectively. Since the dataset in this study is approximately 220% larger, with twice the size of input features, the modification and hyperparameter tuning made it faster, around 14–20%, which is a remarkable speed up for extensive data that could be employed in this model. Moreover, the GRU-h led to slightly more accurate predictions with an R^2 equal to 0.99 and an MAE of 0.001; see Table 1.

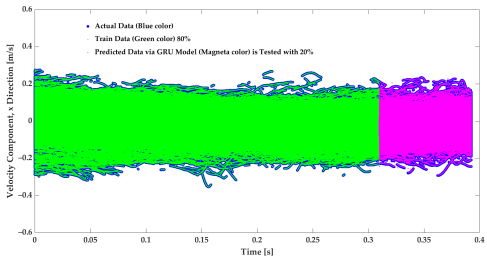


Figure 5. Velocity prediction of the velocity in the X direction from the GRU model. The model is trained on 80% of the data, while the remaining 20% is used for testing. The filled blue circles are actual data, the green points are train data, and the magenta points are GRU-predicted data.

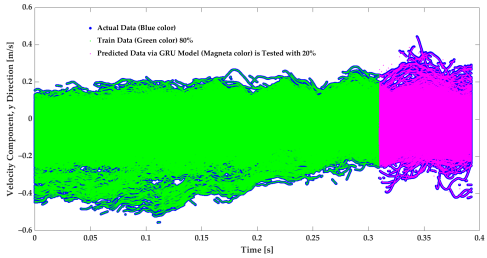


Figure 6. Velocity prediction of the velocity in the Y direction from the GRU model. The model is trained on 80% of the data, while the remaining 20% is used for testing. The filled blue circles are actual data, the green points are trained data, and the magenta points are GRU-predicted data.

Table 1. Comparison table of the GRU-h model of the current study that is improved by HPO and trained with larger data and four sequential variable inputs: x, y, V_X, and V_Y. Transformer, LSTM, and GRU, illustrated in the table, are models from previous studies [15,16], with smaller boundary conditions and two sequential variable inputs V_X and V_Y and without HPO.

Training Proportion	Performance	GRU-h	Transformer	LSTM	GRU
80%	MAE	0.001	0.002	0.001	0.002
	R^2 score	0.99	0.98	0.98	0.98
	Runtime (s)	256	301	295	318

3.3. Parallel Computing Assessment

It is reported that GRU is faster and produces similar prediction results as LSTM with fewer data [4,11,12,15]. In this study, 6,225,457 tracking points are available just from the 0.4 s long period of the experiment. To cope with the amount of data, the GRU is trained on parallel computing architectures, and its speed-up is examined. The training of the GRU is performed on two machines, i.e., on the DEEP-DAM and JUWELS-BOOSTER machines. On DEEP-DAM, the training is performed on a single node using one GPU. The corresponding training time using this setup is 5802.60 s, serving as a baseline. By varying the number of GPU on the JUWELS-BOOSTER, it is possible to measure the speed-up gained by the additional GPU. Here, strong scaling is the metric of choice, as the amount of work stays constant no matter how many processors are used [35]. The goal of parallelizing the computation is to reduce the time to solution. As is obvious from the data in Table 2 and Figure 7, the speed-up of the model increased with 1.59, 2.13, and 2.57 for using 2, 3, and 4 GPU, respectively.

Table 2. Parallel computing machine scalability to train the GRU model with GPU.

Machine Module	Node	GPUs	Computing Time [s]	Speedup
JUWELS-BOOSTER	1	1	5801.20	1
	1	2	3640.31	1.59
	1	3	2719.36	2.13
	1	4	2252.52	2.57
DEEP-DAM	1	1	5802.60	1

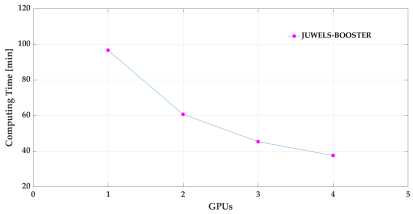


Figure 7. Computing time on the JUWELS-BOOSTER on one node assessed with one to four GPU for the GRU training model.

In addition to the MAE, the HPO process for optimizing the batch size also affects the runtime of the training, which is reported in Table 3. As the batch size per GPU increases with a factor of 2, the total training runtime reduces approximately with the same factor. This indicates that the GPU are not fully utilized with small batch sizes, and for the computational efficiency, the training should be conducted with larger batch sizes.

The lowest MAE is observed for a batch size of 512, which is an indication that this batch size is the optimal trade-off between speed-up and accuracy.

Table 3. Effect of the size of the batch size on the computing time and the MAE.

Machine Module	GPUs	Batch Size per GPU	Computing Time [s]	MAE
JUWELS-BOOSTER	4	8	14723.30	0.0016698
	4	16	7499.96	0.0015822
	4	32	3757.98	0.0015293
	4	64	1820.90	0.0014718
	4	128	963.49	0.0014551
	4	256	493.07	0.0013771
	4	512	255.93	0.0013613
	4	1024	147.70	0.0014453

4. Summary and Conclusions

This study employed empirical data from strained turbulence flow experiments conducted in a laboratory setup to create a velocity prediction model. The simulated turbulent flow has a Taylor microscale REYNOLDS number in the range of $100 < Re_\lambda < 152$. The turbulent flow at the measurement area was a nearly stationary homogeneous isotropic before the deformation. Tracer particles with a median diameter of 8–10 µm and a specific gravity of 1.1 g/cm^3 were seeded in the flow. The mean strain rate in the Y direction is generated to be 4 s^{-1}, and the LPT technique is applied to record the flow features. Based on the Lagrangian perspective, the extracted velocity and location dataset has been used to train a GRU model for flow predictions. The strained turbulent flow is a type of shear flow that can be observed in many applications, such as the external flow over an airfoil and internal flow within a variable cross-section pipe, internal combustion in engines, particle interactions in mixing chambers, erosion at the leading edges, dispersion of pollutants in the atmosphere, formation of rain within clouds, and dispersion of sediments in oceans and rivers [17].

A GRU network is a version of the LSTM network that can perform training faster and with fewer data. As has been noted in the literature, the turbulence intensity, boundary geometry, and strain rate affect the flow velocity. Moreover, this experiment was performed in the presence of gravity, which was not investigated in previous numerical studies on deformed turbulent flow, and its effect remains unknown. This study relies on the concept that the velocity as a function of the locations and sequential feature of the flow carries all relevant information affecting the above-mentioned factors. Therefore, in the training of the GRU, the model is evaluated to observe how it is capable of learning how the historical effect of all parameters will impact the following period, since DL can extract hidden features. Each velocity component and location are measured by LPT in sequence form, and the locations x, y, and velocity components in the corresponding directions are applied as input data to train the GRU model. Based on the training, the GRU predicts the velocity component individually in the following period. In this study, 80% of the data was used as training data, and the remaining 20% of the data were employed to test the prediction and validate it.

The predictions from the GRU model are considerably accurate, as the MAE and R^2 score are 0.001 and 0.993, respectively. The suggested approach leads to predicting turbulence flow in many applications. However, it is essential to evaluate the model with extensive data and long-term predictions, as well as apply different boundary conditions and vary the REYNOLDS number range to observe the limit of the projections. The current model has been compared to previous DL models with a similar application. The results in Table 1 show that the proposed model, with 220% larger data and two times more input variables, has a faster performance of 14–20% than similar model applications of LSTM, GRU, and Transformer because of the HPO. This performance is a remarkable achievement, particularly when applying the model to a more extensive dataset. Besides the accurate predictions generated by this model, the model was executed on the parallel machines

JULES-BOOSTER and DEEP-DAM at the Jülich Supercomputer Centre to investigate the training's speed-up. The performance on one node and one to four GPU has been examined in JUWELS-BOOSTER. The results show the speed-up to increase in two GPU. With four GPU, the model trains 2.57 faster than the metric measurement with a single GPU. To further enhance this model, its performance with respect to the prediction accuracy and scalability will be examined extensively using more data. Furthermore, the impact of the hyperparameters in this model will be investigated to accelerate the model under the constraint of keeping the accuracy at suitable conditions.

Author Contributions: R.H. contributed to the conceptualization, method, software, data analysis, and writing—original draft preparation. M.A. contributed to the software, methodology, and model analysis. A.L. contributed to the review and editing. Á.H. contributed to the review and editing. M.R. contributed to the writing, review editing, and supervision. All authors have read and agreed to the published version of the manuscript.

Funding: The research leading to these results has been conducted in the Center of Excellence (CoE) Research on AI and Simulation-Based Engineering at Exascale (RAISE), the EuroCC two projects receiving funding from EU's Horizon 2020 Research and Innovation Framework Programme, and European Digital Innovation Hub Iceland (EDIH-IS) under the grant agreement no. 951733, no. 101101903, and no. 101083762, respectively.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: We thank Ármann Gylfason from Reykjavik University for his technical comments on the experimental data and Lahcen Bouhlali from Reykjavik University for his experimental work in data preparation. The authors thank the technical support of the FreaEnergy team (Energy, AI, HPC, and CFD solutions), at Mýrin located in the Gróska -innovation and business growth center- in Reykjavik.

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

Acronyms

CFD	Computational Fluid Dynamics
CNN	Convolutional Neural Network
CPU	Central Processing Unit
DL	Deep Learning
DMD	Dynamical Mode Decomposition
DNS	Direct Numerical Simulation
GPU	Graphics Processing Unit
GRU	Gated Recurrent Unit
HPC	High-Performance Computing
HPO	Hyperparameter Optimization
LES	Large Eddy Simulation
LPT	Lagrangian Particle Tracking
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
ML	Machine Learning
MLP	Multilayer Perceptron
MPI	Message Passing Interface
POD	Proper Orthogonal Decomposition
RANS	Reynolds-Averaged Navier Stokes
RANS	Reynolds-Averaged Navier Stokes
RNN	Recurrent Neural Network
ROM	Reduced-Order Model

References

1. Pope, S.B. *Turbulent Flows*; Cambridge University Press: London, UK, 2000.
2. John L. Lumley, H.T. *A First Course in Turbulence*; MIT Press: Cambridge, MA, USA, 1972.
3. Davidson, P.A. *Turbulence: An Introduction for Scientists and Engineers*; Oxford University Press: London, UK, 2004.
4. Hassanian, R.; Riedel, M.; Bouhlali, L. The Capability of Recurrent Neural Networks to Predict Turbulence Flow via Spatiotemporal Features. In Proceedings of the 2022 IEEE 10th Jubilee International Conference on Computational Cybernetics and Cyber-Medical Systems (ICCC), Reykjavík, Iceland, 6–9 July 2022; pp. 000335–000338. [CrossRef]
5. Bukka, S.R.; Gupta, R.; Magee, A.R.; Jaiman, R.K. Assessment of unsteady flow predictions using hybrid deep learning based reduced-order models. *Phys. Fluids* **2021**, *33*, 013601. [CrossRef]
6. Cengel, Y.; Cimbala, J. *Fluid Mechanics Fundamentals and Applications*; McGraw Hill: New York, NY, USA, 2013.
7. White, F. *Fluid Mechanics*; McGraw Hill: New York, NY, USA, 2015.
8. Eivazi, H.; Veisi, H.; Naderi, M.H.; Esfahanian, V. Deep neural networks for nonlinear model order reduction of unsteady flows. *Phys. Fluids* **2020**, *32*, 105104. [CrossRef]
9. Srinivasan, P.A.; Guastoni, L.; Azizpour, H.; Schlatter, P.; Vinuesa, R. Predictions of turbulent shear flows using deep neural networks. *Phys. Rev. Fluids* **2019**, *4*, 054603. [CrossRef]
10. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
11. Wang, Y.; Zou, R.; Liu, F.; Zhang, L.; Liu, Q. A review of wind speed and wind power forecasting with deep neural networks. *Appl. Energy* **2021**, *304*, 117766. [CrossRef]
12. Gu, C.; Li, H. Review on Deep Learning Research and Applications in Wind and Wave Energy. *Energies* **2022**, *15*, 1510. [CrossRef]
13. Duru, C.; Alemdar, H.; Baran, O.U. A deep learning approach for the transonic flow field predictions around airfoils. *Comput. Fluids* **2022**, *236*, 105312. [CrossRef]
14. Moehlis, J.; Faisst, H.; Eckhardt, B. A low-dimensional model for turbulent shear flows. *New J. Phys.* **2004**, *6*, 56. [CrossRef]
15. Hassanian, R.; Helgadottir, A.; Riedel, M. Deep Learning Forecasts a Strained Turbulent Flow Velocity Field in Temporal Lagrangian Framework: Comparison of LSTM and GRU. *Fluids* **2022**, *7*, 344. [CrossRef]
16. Hassanian, R.; Myneni, H.; Helgadottir, A.; Riedel, M. Deciphering the dynamics of distorted turbulent flows: Lagrangian particle tracking and chaos prediction through transformer-based deep learning models. *Phys. Fluids* **2023**, *35*, 075118. [CrossRef]
17. Hassanian, R.; Helgadottir, A.; Bouhlali, L.; Riedel, M. An experiment generates a specified mean strained rate turbulent flow: Dynamics of particles. *Phys. Fluids* **2023**, *35*, 015124. [CrossRef]
18. Pant, P.; Doshi, R.; Bahl, P.; Barati Farimani, A. Deep learning for reduced order modelling and efficient temporal evolution of fluid simulations. *Phys. Fluids* **2021**, *33*, 107101. [CrossRef]
19. Fresca, S.; Manzoni, A. POD-DL-ROM: Enhancing deep learning-based reduced order models for nonlinear parametrized PDEs by proper orthogonal decomposition. *Comput. Methods Appl. Mech. Eng.* **2022**, *388*, 114181. [CrossRef]
20. Papapicco, D.; Demo, N.; Girfoglio, M.; Stabile, G.; Rozza, G. The Neural Network shifted-proper orthogonal decomposition: A machine learning approach for non-linear reduction of hyperbolic equations. *Comput. Methods Appl. Mech. Eng.* **2022**, *392*, 114687. [CrossRef]
21. Riedel, M.; Sedona, R.; Barakat, C.; Einarsson, P.; Hassanian, R.; Cavallaro, G.; Book, M.; Neukirchen, H.; Lintermann, A. Practice and Experience in using Parallel and Scalable Machine Learning with Heterogenous Modular Supercomputing Architectures. In Proceedings of the 2021 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), Portland, OR, USA, 17–21 June 2021; pp. 76–85. [CrossRef]
22. Hassanian, R.; Riedel, M. Leading-Edge Erosion and Floating Particles: Stagnation Point Simulation in Particle-Laden Turbulent Flow via Lagrangian Particle Tracking. *Machines* **2023**, *11*, 566. [CrossRef]
23. Cowen, E.A.; Monismith, S.G. A hybrid digital particle tracking velocimetry technique. *Exp. Fluids* **1997**, *22*, 199–211. [CrossRef]
24. Hassanian, R. An Experimental Study of Inertial Particles in Deforming Turbulence Flow, in Context to Loitering of Blades in Wind Turbines. Master's Thesis, Reykjavik University, Reykjavik, Iceland 2020.
25. Ouellette, N.T.; Xu, H.; Bodenschatz, E. A quantitative study of three-dimensional Lagrangian particle tracking algorithms. *Exp. Fluids* **2006**, *40*, 301–313. [CrossRef]
26. Lee, C.M.; Gylfason, A.; Perlekar, P.; Toschi, F. Inertial particle acceleration in strained turbulence. *J. Fluid Mech.* **2015**, *785*, 31–53. [CrossRef]
27. Ayyalasomayajula, S.; Warhaft, Z. Nonlinear interactions in strained axisymmetric high-Reynolds-number turbulence. *J. Fluid Mech.* **2006**, *566*, 273–307. [CrossRef]
28. Cho, K.; van Merriënboer, B.; Bahdanau, D.; Bengio, Y. On the Properties of Neural Machine Translation: Encoder–Decoder Approaches. *arXiv* **2014**, arXiv:1409.1259.
29. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv* **2014**, arXiv:1412.3555.
30. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.S.; Davis, A.; Dean, J.; Devin, M.; et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv* **2016**, arXiv:1603.04467.
31. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation*; USENIX Association: Berkeley, CA, USA, 2016.

32. Kramer, O. Scikit-learn. *Mach. Learn. Evol. Strateg.* **2016**, *20*, 45–53.
33. Alvarez, D. JUWELS Cluster and Booster: Exascale Pathfinder with Modular Supercomputing Architecture at Juelich Supercomputing Centre. *J. Large-Scale Res. Facil. JLSRF* **2021**, *7*, A183. [CrossRef]
34. TensorFlow. *TensorFlow Core Tutorials*; TensorFlow: Mountain View, CA, USA, 2022.
35. Hager, G.; Wellein, G. *Introduction to High Performance Computing for Scientists and Engineers*; Chapman & Hall/CRC Computational Science: London, UK, 2010.

References

[1] Stephen B. Pope. *Turbulent Flows*. London: Cambridge University Press, 2000. ISBN: 9780521598866.

[2] Frank White. *Fluid Mechanics*. New York: McGraw Hill, 2015.

[3] Yunus Cengel and John Cimbala. *Fluid Mechanics Fundamentals and Applications*. New York: McGraw Hill, 2013. ISBN: 9780073380322.

[4] Henk Tennekes John L. Lumley. *A First Course in Turbulence*. Massachusetts: MIT Press, 1972.

[5] H. Versteeg and W. Malalasekera. *Introduction to Computational Fluid Dynamics, An: The Finite Volume Method*. New York: Pearson, 2007.

[6] Suhas Patankar. *Numerical Heat Transfer and Fluid Flow*. London: CRC Press, 1980.

[7] Sandeep Reddy Bukka, Rachit Gupta, Allan Ross Magee, and Rajeev Kumar Jaiman. "Assessment of unsteady flow predictions using hybrid deep learning based reduced-order models." In: *Physics of Fluids* 33.1 (Jan. 2021), p. 013601. ISSN: 1070-6631. DOI: 10.1063/5.0030137.

[8] Reza Hassanian, Morris Riedel, and Lahcen Bouhlali. "The Capability of Recurrent Neural Networks to Predict Turbulence Flow via Spatiotemporal Features." In: *2022 IEEE 10th Jubilee International Conference on Computational Cybernetics and Cyber-Medical Systems (ICCC)*. 2022, pp. 000335–000338. DOI: 10.1109/ICCC202255925.2022.9922754.

[9] Lewis Fry Richardson and William Napier Shaw. "The supply of energy from and to atmospheric eddies." In: *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character* 97.686 (1920), pp. 354–373. DOI: 10.1098/rspa.1920.0039.

[10] Geoffrey Ingram Taylor. "Statistical theory of turbulenc." In: *Proceedings of the Royal Society of London. Series A - Mathematical and Physical Sciences* 151.873 (1935), pp. 421–444. DOI: 10.1098/rspa.1935.0158.

[11] Andrei Nikolaevich Kolmogorov, V. Levin, Julian Charles Roland Hunt, Owen Martin Phillips, and David Williams. "The local structure of turbulence in incompressible viscous fluid for very large Reynolds numbers." In: *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences* 434.1890 (1991), pp. 9–13. DOI: 10.1098/rspa.1991.0075.

[12] Christian Küchler, Gregory P. Bewley, and Eberhard Bodenschatz. "Universal Velocity Statistics in Decaying Turbulence." In: *Phys. Rev. Lett.* 131 (2 July 2023), p. 024001. DOI: 10.1103/PhysRevLett.131.024001.

[13] Markus Raffel, Christian E. Willert, Steve T. Wereley, and Jurgen Kompenhans. *Particle Image Velocimetry: A Practical Guide*. Springer, 2007.

[14] Pedro Costa, Francesco Picano, Luca Brandt, and Wim-Paul Breugem. "Universal Scaling Laws for Dense Particle Suspensions in Turbulent Wall-Bounded Flows." In: *Phys. Rev. Lett.* 117 (13 Sept. 2016), p. 134501. DOI: 10.1103/PhysRevLett.117.134501.

[15] Amir Hossein Shiravi, Mojtaba Shafiee, Mohammad Firoozzadeh, Hadis Bostani, and Maryam Bozorgmehrian. "Experimental study on convective heat transfer and entropy generation of carbon black nanofluid turbulent flow in a helical coiled heat exchanger." In: *Journal of Thermal Analysis and Calorimetry* 145.2 (2021), pp. 597–607.

[16] Margherita Dotti, Rasmus K. Schlander, Preben Buchhave, and Clara M. Velte. "Experimental investigation of the turbulent cascade development by injection of single large-scale Fourier modes." In: *Experiments in Fluids* 61 (2020), p. 214. DOI: https://doi.org/10.1007/s00348-020-03041-2.

[17] Luca Brandt and Filippo Coletti. "Particle-Laden Turbulence: Progress and Perspectives." In: *Annual Review of Fluid Mechanics* 54.1 (2022), pp. 159–189. DOI: 10.1146/annurev-fluid-030121-021103.

[18] Hamidreza Eivazi, Hadi Veisi, Mohammad Hossein Naderi, and Vahid Esfahanian. "Deep neural networks for nonlinear model order reduction of unsteady flows." In: *Physics of Fluids* 32.10 (Oct. 2020), p. 105104. ISSN: 1070-6631. DOI: 10.1063/5.0020526.

[19] Peder J. Olesen Azur Hodžić and Clara M. Velte. "On the discrepancies between POD and Fourier modes on aperiodic domains." In: *Journal of Engineering Mathematics* 145 (2024), pp. 10–145. DOI: https://doi.org/10.1007/s10665-024-10340-8.

[20] P. A. Srinivasan, L. Guastoni, H. Azizpour, P. Schlatter, and R. Vinuesa. "Predictions of turbulent shear flows using deep neural networks." In: *Phys. Rev. Fluids* 4 (5 May 2019), p. 054603. DOI: 10.1103/PhysRevFluids.4.054603.

[21] Marcel Aach, Eray Inanc, Rakesh Sarma, Morris Riedel, and Andreas Lintermann. "Large scale performance analysis of distributed deep learning frameworks for convolutional neural networks." In: *Journal of Big Data* 10 (2023), p. 96. ISSN: 2196-1115. DOI: https://doi.org/10.1186/s40537-023-00765-w.

[22] Forschungszentrum Jülich GmbH. *DEEP system*. 2024. URL: https://www.fz-juelich.de/en/ias/jsc/systems/prototype-systems/deep_system.

[23] Forschungszentrum Jülich GmbH. *Jülich Wizard for European Leadership Science (JUWELS)*. 2024. URL: https://www.fz-juelich.de/en/ias/jsc/systems/supercomputers/juwels.

[24] E. Suarez, N. Eicker, and T. Lippert. *Contemporary High-Performance Computing From Petascale toward Exascale, Volume 3: Chapter 9- Modular Supercomputing Architecture: from Idea to Production*. Chapman Hall/CRC Computational Science, 2019.

[25] Raymond A. Shaw. "Particle-turbulence Interactions in Atmospheric Clouds." In: *Annual Review of Fluid Mechanics* 35.1 (2003), pp. 183–227. DOI: 10.1146/annurev.fluid.35.101101.161125.

[26] Federico Toschi and Eberhard Bodenschatz. "Lagrangian Properties of Particles in Turbulence." In: *Annual Review of Fluid Mechanics* 41.1 (2009), pp. 375–404. DOI: 10.1146/annurev.fluid.010908.165210.

[27] Guan Yeoh, Jyh Jian Chen, and Chun Huei Chen. "Investigation of Swirling Flows in Mixing Chambers." In: *Modelling and Simulation in Engineering* (2011). DOI: https://doi.org/10.1155/2011/259401.

[28] Mohammad Hossein Arabnejad, Ali Amini, Mohamed Farhat, and Rickard E. Bensow. "Numerical and experimental investigation of shedding mechanisms from leading-edge cavitation." In: *International Journal of Multiphase Flow* 119 (2019), pp. 123–143. ISSN: 0301-9322. DOI: https://doi.org/10.1016/j.ijmultiphaseflow.2019.06.010.

[29] Tadashi Tanuma. *Advances in Steam Turbines for Modern Power Plants*. Woodhead Publishing, 2017. ISBN: 9780081003251.

[30] Bahman Taherkhani, Ali Pourkamali Anaraki, Javad Kadkhodapour, Nahid Kangarani Farahani, and Haoyun Tu. "Erosion Due to Solid Particle Impact on the Turbine Blade: Experiment and Simulation." In: *Journal of Failure Analysis and Prevention* 19 (2019), pp. 1739–1744.

[31] G. K. Batchelor. *The Theory of Homogeneous Turbulence*. Cambridge University Press, 1982. ISBN: 9780521041171.

[32] J. C. R. Hunt. "A theory of turbulent flow round two-dimensional bluff bodies." In: *Journal of Fluid Mechanics* 61.4 (1973), pp. 625–706. DOI: 10.1017/S0022112073000893.

[33] Z. Warhaft. "An experimental study of the effect of uniform strain on thermal fluctuations in grid-generated turbulence." In: *Journal of Fluid Mechanics* 99.3 (1980), pp. 545–573. DOI: 10.1017/S0022112080000766.

[34] J. C. R. Hunt and D. J. Carruthers. "Rapid distortion theory and the 'problems' of turbulence." In: *Journal of Fluid Mechanics* 212 (1990), pp. 497–532. DOI: 10.1017/S0022112090002075.

[35] Jun Chen, Charles Meneveau, and Joseph Katz. "Scale interactions of turbulence subjected to a straining–relaxation–destraining cycle." In: *Journal of Fluid Mechanics* 562 (2006), pp. 123–150.

[36] P. Gualtieri and C. Meneveau. "Direct numerical simulations of turbulence subjected to a straining and destraining cycle." In: *Physics of Fluids* 22.6 (2010), p. 065104. DOI: 10.1063/1.3453709.

[37] Martin O. L. Hansen. *Aerodynamics of Wind Turbines*. Routledge, 2015. ISBN: 9781138775077.

[38] Mahdi Abkar. "Theoretical Modeling of Vertical-Axis Wind Turbine Wakes." In: *Energies* 12.1 (Dec. 2018), p. 10. ISSN: 1996-1073. DOI: 10.3390/en12010010.

[39] Iqbal H. Sarker. "Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions." In: *SN Computer Science* 2 (6 Sept. 2016), p. 420. DOI: https://doi.org/10.1007/s42979-021-00815-1.

[40] Danilo P. Mandic and Jonathon A. Chambers. *Recurrent Neural Networks for Prediction: Learning Algorithms, Architectures and Stability*. Hoboken: Wiley, 2001.

[41] Dupond S. "A thorough review on the current advance of neural network structures." In: *Annu Rev Control*. 14 (2019), pp. 200–2030.

[42] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory." In: *Neural Computation* 9.8 (Nov. 1997), pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.

[43] Reza Hassanian, Ásdís Helgadóttir, and Morris Riedel. "Deep Learning Forecasts a Strained Turbulent Flow Velocity Field in Temporal Lagrangian Framework: Comparison of LSTM and GRU." In: *Fluids* 7.11 (2022). ISSN: 2311-5521. DOI: 10.3390/fluids7110344.

[44] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation." In: *CoRR* abs/1406.1078 (2014).

[45] Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling." In: *CoRR* (2014).

[46] Nicole Gruber and Alfred Jockisch. "Are GRU Cells More Specific and LSTM Cells More Sensitive in Motive Classification of Text?" In: *Frontiers in Artificial Intelligence* 3 (2020). DOI: 10.3389/frai.2020.00040.

[47] Renato F. Miotto and William R. Wolf. "Flow imaging as an alternative to non-intrusive measurements and surrogate models through vision transformers and convolutional neural networks." In: *Physics of Fluids* 35.4 (Apr. 2023). DOI: 10.1063/5.0144700.

[48] Pin Wu, Feng Qiu, Weibing Feng, Fangxing Fang, and Christopher Pain. "A non-intrusive reduced order model with transformer neural network and its application." In: *Physics of Fluids* 34.11 (Nov. 2022). DOI: 10.1063/5.0123185.

[49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, et al. *Attention is All you Need*. Ed. by I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, et al. 2017. URL: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[50] Pablo García-risueño and Pablo E. Ibáñez. "A Review of High Performance Computing Foundations for Scientists." In: *International Journal of Modern Physics C* 23.07 (2012), p. 1230001. DOI: 10.1142/S0129183112300011.

[51] Jülich Supercomputing Centre. "JUWELS Cluster and Booster: Exascale Pathfinder with Modular Supercomputing Architecture at Jülich Supercomputing Centre." In: *Journal of large-scale research facilities JLSRF* 7 (Oct. 2021), A183. DOI: http://dx.doi.org/10.17815/jlsrf-7-183.

[52] Tal Ben-Nun and Torsten Hoefler. "Demystifying Parallel and Distributed Deep Learning: An In-Depth Concurrency Analysis." In: (2018). arXiv: 1802.09941 [cs.LG].

[53] Joaquin Vanschoren Frank Hutter Lars Kotthoff. *Automated Machine Learning*. Cham: Springer, 2019. ISBN: 978-3-030-05317-8.

[54] Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola. *Dive into Deep Learning*. https://D2L.ai. Cambridge University Press, 2023.

[55] R. Hassanian, A. Helgadottir, L. Bouhlali, and M. Riedel. "An experiment generates a specified mean strained rate turbulent flow: Dynamics of particles." In: *Physics of Fluids* 35.1 (Jan. 2023), p. 015124. ISSN: 1070-6631. DOI: 10.1063/5.0134306.

[56] C.-M. Lee, Á. Gylfason, P. Perlekar, and F. Toschi. "Inertial particle acceleration in strained turbulence." In: *Journal of Fluid Mechanics* 785 (2015), pp. 31–53. DOI: 10.1017/jfm.2015.579.

[57] P. A. Davidson. *Turbulence: An Introduction for Scientists and Engineers*. London: Oxford University Press, 2004. ISBN: 9780198529491.

[58] Andrew Pollard, Luciano Castillo, Luminita Danaila, and Mark Glauser. *Whither Turbulence and Big Data in the 21st Century?* Springer Cham, 2016.

[59] Eric Perlman, Randal Burns, Yi Li, and Charles Meneveau. "Data exploration of turbulence simulations using a database cluster." In: *Proceedings of the 2007 ACM/IEEE Conference on Supercomputing*. SC '07. Reno, Nevada: Association for Computing Machinery, 2007. ISBN: 9781595937643. DOI: 10.1145/1362622.1362654.

[60] Xiaohua Wu and Parviz Moin. "A direct numerical simulation study on the mean velocity characteristics in turbulent pipe flow." In: *Journal of Fluid Mechanics* 608 (2008), pp. 81–112. DOI: 10.1017/S0022112008002085.

[61] Steven L. Brunton, Bernd R. Noack, and Petros Koumoutsakos. "Machine Learning for Fluid Mechanics." In: *Annual Review of Fluid Mechanics* 52.Volume 52, 2020 (2020), pp. 477–508. ISSN: 1545-4479. DOI: https://doi.org/10.1146/annurev-fluid-010719-060214.

[62] B. Siddani, S. Balachandar, Jiazhong Zhou, and Shankar Subramaniam. "Investigating the influence of particle distribution on force and torque statistics using hierarchical machine learning." In: *AIChE Journal* 70.5 (2024), e18339. DOI: https://doi.org/10.1002/aic.18339.

[63] Andrea Beck, David Flad, and Claus-Dieter Munz. "Deep neural networks for data-driven LES closure models." In: *Journal of Computational Physics* 398 (2019), p. 108910. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2019.108910.

[64] A H Shiravi, A S Mujumdar, and G J Kubes. "Numerical Study of Heat Transfer and Fluid Flow in Multiple turbulent Impinging Jets." In: *Drying Technology* 13.5-7 (1995), pp. 1359–1375. DOI: 10.1080/07373939508917027.

[65] Marius Kurz, Philipp Offenhäuser, Dominic Viola, Oleksandr Shcherbakov, Michael Resch, and Andrea Beck. "Deep reinforcement learning for computational fluid dynamics on HPC systems." In: *Journal of Computational Science* 65 (2022), p. 101884. ISSN: 1877-7503. DOI: https://doi.org/10.1016/j.jocs.2022.101884.

[66] S. Wiewel, M. Becher, and N. Thuerey. "Latent Space Physics: Towards Learning the Temporal Evolution of Fluid Flow." In: *Computer Graphics Forum* 38.2 (2019), pp. 71–82. DOI: https://doi.org/10.1111/cgf.13620.

[67] Sindre Stenen Blakseth, Adil Rasheed, Trond Kvamsdal, and Omer San. "Deep neural network enabled corrective source term approach to hybrid analysis and modeling." In: *Neural Networks* 146 (2022), pp. 181–199. ISSN: 0893-6080. DOI: https://doi.org/10.1016/j.neunet.2021.11.021. URL: https://www.sciencedirect.com/science/article/pii/S0893608021004494.

[68] K. Choudhary, K.A. Krishnaprasad, S. Pandey, N. Zgheib, J.S. Salinas, et al. "Effectiveness of RANS in predicting indoor airborne viral transmission: A critical evaluation against LES." In: *Computers Fluids* 256 (2023), p. 105845. ISSN: 0045-7930. DOI: https://doi.org/10.1016/j.compfluid.2023.105845.

[69] S.M. Hosseini, R. Vinuesa, P. Schlatter, A. Hanifi, and D.S. Henningson. "Direct numerical simulation of the flow around a wing section at moderate Reynolds number." In: *International Journal of Heat and Fluid Flow* 61 (2016). SI TSFP9 special issue, pp. 117–128. ISSN: 0142-727X. DOI: https://doi.org/10.1016/j.ijheatfluidflow.2016.02.001.

[70] Paul Fischer, Stefan Kerkemeier, Ananias Tomboulides, Misun Min, Elia Merzari, and Aleksandr Obabko. *Nek5000*. 2008. URL: Webpage:%20https://nek5000.mcs.anl.gov/.

[71] S.J. Lawson, M. Woodgate, R. Steijl, and G.N. Barakos. "High performance computing for challenging problems in computational fluid dynamics." In: *Progress in Aerospace Sciences* 52 (2012). Applied Computational Aerodynamics and High Performance Computing in the UK, pp. 19–29. ISSN: 0376-0421. DOI: https://doi.org/10.1016/j.paerosci.2012.03.004.

[72] *The aerodynamics of ship superstructures*. Nov. 1991.

[73] School of Physics and Astronomy at the University of Edinburgh. *High-End Computing Terascale Resource: UK National Supercomputing Service*. 2014. URL: http://www.hector.ac.uk/.

[74] Forschungszentrum Jülich GmbH. *Best Practice mini-guide "JUROPA"*. 2013. URL: https://prace-ri.eu/wp-content/uploads/Best-Practice-Guide_JUROPA.pdf.

[75] Cihat Duru, Hande Alemdar, and Ozgur Ugras Baran. "A deep learning approach for the transonic flow field predictions around airfoils." In: *Computers & Fluids* 236 (2022), p. 105312. ISSN: 0045-7930. DOI: https://doi.org/10.1016/j.compfluid.2022.105312.

[76] Jeff Moehlis, Holger Faisst, and Bruno Eckhardt. "A low-dimensional model for turbulent shear flows." In: *New Journal of Physics* 6.1 (May 2004), p. 56. DOI: 10.1088/1367-2630/6/1/056.

[77] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. "Quantized Neural Networks: Training Neural Networks with Low Precision Weights and Activations." In: *Journal of Machine Learning Research* 18.187 (2018), pp. 1–30. URL: http://jmlr.org/papers/v18/16-456.html.

[78] J. R. Herring. "Subgrid Scale Modeling — An Introduction and Overview." In: *Turbulent Shear Flows I*. Ed. by Franz Durst, Brian E. Launder, Frank W. Schmidt, and James H. Whitelaw. Berlin, Heidelberg: Springer Berlin Heidelberg, 1979, pp. 347–352.

[79] Chengcheng Gu and Hua Li. "Review on Deep Learning Research and Applications in Wind and Wave Energy." In: *Energies* 15.4 (2022). ISSN: 1996-1073. DOI: 10.3390/en15041510.

[80] International Energy Agency. *World Energy Outlook 2023*. 2024. URL: https://www.iea.org/reports/world-energy-outlook-2023.

[81] Reza Hassanian, Asdis Helgadottir, Marcel Aach, Andreas Lintermann, and Morris Riedel. "A proposed hybrid two-stage DL-HPC method for windspeed forecasting: using the first average forecast output for long-term forecasting." In: Proceedings of the IACM Computational Fluids Conference (CFC2023), Cannes (France), 25 Apr 2023 - 28 Apr 2023. Apr. 25, 2023. URL: https://juser.fz-juelich.de/record/1007354.

[82] Global Wind Energy Council. *https://gwec.net/globalwindreport2023/*. 2024. URL: https://gwec.net/globalwindreport2023/.

[83] Yun Wang, Runmin Zou, Fang Liu, Lingjun Zhang, and Qianyi Liu. "A review of wind speed and wind power forecasting with deep neural networks." In: *Applied Energy* 304 (2021), p. 117766. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2021.117766.

[84] Bedassa R Cheneka, Simon J Watson, and Sukanta Basu. "The impact of weather patterns on offshore wind power production." In: *Journal of Physics: Conference Series* 1618.6 (Sept. 2020), p. 062032. DOI: 10.1088/1742-6596/1618/6/062032.

[85] Naveed Akhtar, Beate Geyer, and Corinna Schrum. "Impacts of accelerating deployment of offshore windfarms on near-surface climate." In: *Scientific Reports* 12.1 (Oct. 2022), p. 18307. DOI: 10.1038/s41598-022-22868-9.

[86] A. Kolios, M. Richmond, S. Koukoura, and B. Yeter. "Effect of weather forecast uncertainty on offshore wind farm availability assessment." In: *Ocean Engineering* 285 (2023), p. 115265. ISSN: 0029-8018. DOI: https://doi.org/10.1016/j.oceaneng.2023.115265.

[87] R. Hassanian, H. Myneni, Á. Helgadóttir, and M. Riedel. "Deciphering the dynamics of distorted turbulent flows: Lagrangian particle tracking and chaos prediction through transformer-based deep learning models." In: *Physics of Fluids* 35.7 (July 2023), p. 075118. ISSN: 1070-6631. DOI: 10.1063/5.0157897.

Appendices

A Appendix

Appendix A documents the experiments' details and how the turbulent flow characteristics were measured and calculated.

1. Strain rate measurement

The experiment was performed two times with tracers employing the Particle Image Velocimetry (PIV) technique:

I. The turbulence flow without strain: homogeneous isotropic turbulent flow (HIT)

where, the equation shows one component of the flow velocity.

$$V_{HIT} = V + v' \tag{A.1}$$

where, V_{HIT} is measured velocity, V is mean velocity and v' is fluctuation.

II. Strained turbulent flow (S-HIT): the velocity generated by strain added to the total velocity via an experiment with specific strain deformation $2S$ in the y-direction:

$$V_{S-HIT} = V + v' + (-2Sy) \tag{A.2}$$

here V_{S-HIT} is the total measured velocity, and y is the location in the y-direction The measurement extracted the velocity of two experiments separately. To calculate the generated strain, only must subtract the two velocities:

$$V_{strain} = V_{S-HIT} - V_{HIT} \tag{A.3}$$

$$V_{strain} = (V + v' + (-2Sy)) - (V + v') = -2Sy \tag{A.4}$$

where, V_{strain} the velocity generated because of the straining. So, to calculate the strain rate:

$$S_{rate} = \frac{dV_{strain}}{dy} \tag{A.5}$$

here S_{rate} mean strain rate. The generated strain was 4 s^{-1} and 8 s^{-1} in the y-direction; the strain rate measured fluctuated close to the value.

Since $U_{strain} = (Sx, -2Sy, Sz)$ is a for laminar flow, it does not expect to measure the same mean strain value since the generated strain in turbulent flow added to the velocity fluctuations as it well addressed in the literature (P. A. Davidson, 2004).

2- Circular disk move: When they are moved toward the center with a predescribed rate that ensures a nearly constant strain rate in the fluid (Paper I, page 4, paragraph 1.). Since the distance between the disk was vertically constant for experiments, for two mean strain rates, the speed was regulated to generate the specified strain rate.

3. Time duration and dimensionless parameter:

To compare the different experiments with dimensionless parameters, the $S \times t$ is defined. $S \times t$= 1.6, so when the strain rate is 4 s^{-1}, the time duration t is 0.4 s and when the strain rate is 8 s^{-1}, the time duration t is 0.2 s.

4. Calculating Re_λ

According to Table 1 (Paper I), first, it must calculate the dissipation rate ε (Stephen B. Pope, 2000). It requires the use of a second-order velocity structure function (Stephen B. Pope, 2000):

$$D_{ij} = [V_i(y^{(2)}) - V_i(y^{(1)})][V_j(y^{(2)}) - V_j(y^{(1)})] \tag{A.6}$$

$$D_{11} = D_{LL} \tag{A.7}$$

$$D_{22} = D_{33} = D_{NN} \tag{A.8}$$

$$D_{12} = D_{13} = D_{23} = 0 \tag{A.9}$$

here D_{ij} is second-order velocity structure function. D_{LL} and D_{NN} are longitudinal and transverse structure functions, respectively. i and j specify the vector component. $V(y)$ is the velocity. y is the location. The equation can be used (Stephen B. Pope, 2000):

$$D_{11}/(\varepsilon r)^{\frac{2}{3}} = C_2 \tag{A.10}$$

$$D_{22}/(\varepsilon r)^{\frac{2}{3}} = D_{33}/(\varepsilon r)^{\frac{2}{3}} = \frac{4}{3}C_2 \tag{A.11}$$

where ε is the mean dissipation rate, $r = y^{(2)} - y^{(1)}$, and $C_2 = 2$ is the universal constant (Sadoughi eta al, 1994; Stephen B. Pope, 2000).

Eulerian auto-correlation function is used to calculate the integral scale and measure the length scale (Stephen B. Pope, 2000):

$$\rho(l) = u(r_0 + L)u(r_0)/u^2 \tag{A.12}$$

$$L = \int_0^\infty \rho(l)\, dl \tag{A.13}$$

where $\rho(l)$ is the integral scale, u is the velocity, r is location, r_0 refer to origin location that assumed, l is the variable of integration and L is length scale.

$$u_i = U - U_{mean} \tag{A.14}$$

here, u_i is the velocity fluctuation, U is the total velocity, U_{mean} is the mean velocity.

$$u_{rms} = \sqrt{u_i^2} \tag{A.15}$$

where u_{rms} is the root mean square velocity.

$$\eta = (\frac{\nu^3}{\varepsilon})^{\frac{1}{4}} \tag{A.16}$$

where η is Kolmogorov length scale and ν is the kinematic viscosity.

$$\tau_\eta = (\frac{\nu}{\varepsilon})^{\frac{1}{2}} \tag{A.17}$$

here τ_η is Kolmogorov time scale.

$$\lambda = u_{rms}(\frac{15\nu}{\varepsilon})^{\frac{1}{2}} \tag{A.18}$$

where λ is Taylor mircoscale.

$$Re_L = \frac{u_{rms}L}{\nu} \tag{A.19}$$

here Re_L is the flow Reynolds number.

$$Re_\lambda = \frac{u_{rms}\lambda}{\nu} \tag{A.20}$$

where Re_λ is the Taylor microscale Reynolds number.