3D flow field reconstruction using ANN as a surrogate model of the fluid flow within rectangular wall-bounded enclosures

M. Popovac[1*], D.P. Gupta[2] and T. B. Gohil[3]

[1]*AIT-Austrian Institute of Technology, Center for Transport Technologies, Vienna, Austria*

[2]*Independent researcher*

[3]*Visvesvaraya National Institute of Technology, Department of Mechanical Engineering, Nagpur, India*

[*]*Corresponding author: mirza.popovac@ait.ac.at*

Keywords: UNET architecture, passenger comfort, vehicle cabin flow, OpenFoam

Abstract. This paper presents the development of a surrogate model for predicting wall-bounded flow within a simplified enclosure with multiple inlets and outlets. The objective is to create a framework capable of accurately and efficiently predicting the flow distribution based on boundary conditions, such as inlet velocity and temperature. The proposed method reconstructs the three-dimensional velocity and temperature fields of the fluid flow from two-dimensional distributions of these quantities extracted from cutplanes within the solution domain. The approach employs an artificial neural network (ANN) based on the UNET architecture, detailing the optimization parameters and the preparation of input and output channels from the dataset. The dataset was generated using OpenFOAM simulations with a non-isothermal incompressible turbulent flow solver. The imposed parameters include inlet velocity, the location of the inlet opening, and the temperatures at the inlets and walls, while the predicted output comprises the three-dimensional velocity and temperature distributions within the enclosure. The developed surrogate model demonstrated satisfactory accuracy on the generated dataset, highlighting its potential for efficient flow prediction in similar configurations.

1. Introduction

The energy efficiency of thermal conditioning systems in electric vehicles (EVs) plays a critical role in their development. Unlike traditional vehicles with internal combustion engines (ICs), which utilize engine-generated thermal energy, EVs depend solely on battery power for heating and cooling. This dependence directly affects battery costs and significantly influences driving range [1]. Efficient thermal management is essential to maintain passenger comfort without reducing the vehicle's range [2].

Predicting the flow field distribution within the cabin is a key requirement for optimizing indoor thermal comfort. Computational fluid dynamics (CFD) provides detailed insights into flow field distributions but is computationally intensive and time-consuming, rendering it unsuitable for real-time applications [3]. Artificial neural networks (ANNs) offer a viable alternative as surrogate models for fluid flow prediction, capable of learning complex flow field relationships and delivering rapid predictions post-training. The design of the ANN architecture, including the selection of layers and activation functions, is critical to achieving accuracy and efficiency [4]. Existing studies demonstrate deep learning frameworks for predicting Reynolds-averaged Navier-Stokes (RANS) fluid flows around airfoils [5], general shapes [6], and compressible flows [7]. However, these works are primarily limited to two-dimensional solutions.

This study investigates the application of the UNET architecture [6] for three-dimensional flow field reconstruction within wall-bounded enclosures. The proposed approach aims to combine the accuracy of CFD with the speed required for real-time applications, enhancing EV thermal management systems and improving energy efficiency and driving range.

2. CFD solution

For the CFD simulations, the open-source CFD library suite OpenFOAM [8] was employed. The geometry for the numerical analysis resembles a simplified truck cabin configuration, as illustrated in

Figure 1. A box-shaped domain was created using the *blockMesh* utility with uniformly distributed cells in all three directions. To impose appropriate boundary conditions, different patches were extracted from the default outer surfaces using *topoSet* and *createPatch* utilities. These patches included the cabin walls (pink in Figure 1), windows (cyan), roof (blue), and floor (yellow). Air inlets were created at the left and right sides (grey and red), while the air outlet was positioned in the middle of the front side (green).

The constant fluid properties of air have been assumed: the density ρ=1 kg/m3, the heat capacity c_p=1000 J/kg.K, the dynamic viscosity μ=100 mPa.s and the Prandtl number Pr=0.7. For the buoyancy effects the Boussinesq assumption was used, with the expansion coefficient β=0.003.

Boundary conditions were assigned as follows:

- Inlet: Dirichlet boundary condition for velocity, temperature, and turbulent quantities, with Neumann for pressure.
- Outlet: Dirichlet boundary condition for pressure, with Neumann for velocity, temperature, and turbulent quantities.
- Walls: Treated as no-slip boundaries. Temperature boundary conditions varied: fixed value for windows, adiabatic (zeroGradient) for side cabin walls, and heat flux at the roof.

A series of simulations was performed by varying:

1. Inlet velocity and temperature boundary conditions,
2. Position of the inlet openings, and
3. Temperature boundary conditions on the walls.

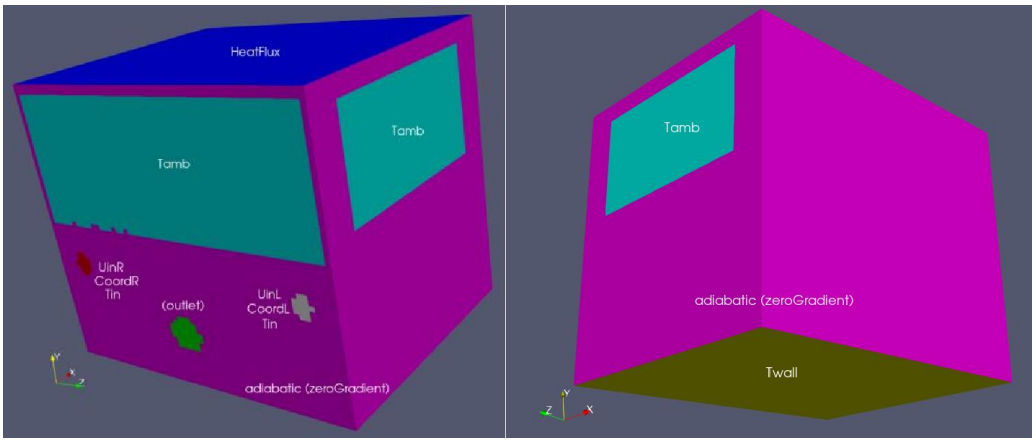

Figure 1: Geometry and boundary conditions.

2.1. Numerical setup

Steady-state simulations were conducted using *buoyantSimpleFoam*, an incompressible Newtonian fluid flow solver based on the SIMPLE pressure-velocity coupling algorithm. This solver addressed equations for momentum and energy conservation, as well as turbulent quantities based on the Reynolds-Averaged Navier-Stokes (RANS) formulation. While the study accounted for both temperature and velocity distributions, only the velocity field results (and corresponding equations) are presented here for brevity:

$$\frac{\partial U_i}{\partial t} + U_j \frac{\partial U_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial P}{\partial x_i} + \frac{\partial}{\partial x_j}\left[\underbrace{(\nu + \nu_t)}_{\nu_{eff}}\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right)\right]$$

$$\frac{\partial \varepsilon}{\partial t} + U_j \frac{\partial \varepsilon}{\partial x_j} = \frac{C_{\varepsilon 1}\mathcal{P} - C_{\varepsilon 2}\varepsilon}{\mathcal{T}} + \frac{\partial}{\partial x_j}\left[\left(\nu + \frac{\nu_t}{\sigma_\varepsilon}\right)\frac{\partial \varepsilon}{\partial x_j}\right]$$

$$\frac{\partial k}{\partial t} + U_j \frac{\partial k}{\partial x_j} = \mathcal{P} - \varepsilon + \frac{\partial}{\partial x_j}\left[\left(\nu + \frac{\nu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right]$$

2.2. CFD Verification and Model Testing

Prior to training the ANN model, the CFD results underwent a verification process to ensure the reliability and consistency of the numerical setup. This process comprised two steps: a grid independence study and turbulence model testing.

Grid Independence Study

The grid independence study involved systematically refining the mesh by uniformly increasing the number of cells in all three directions. Velocity and temperature profiles along the spanwise centerline of the domain were evaluated for each refinement level (Figure 2). Beyond 60 divisions, the profiles exhibited negligible variations and converged to a single line. Consequently, a mesh size with 60 divisions was considered sufficient to achieve grid-independent results.

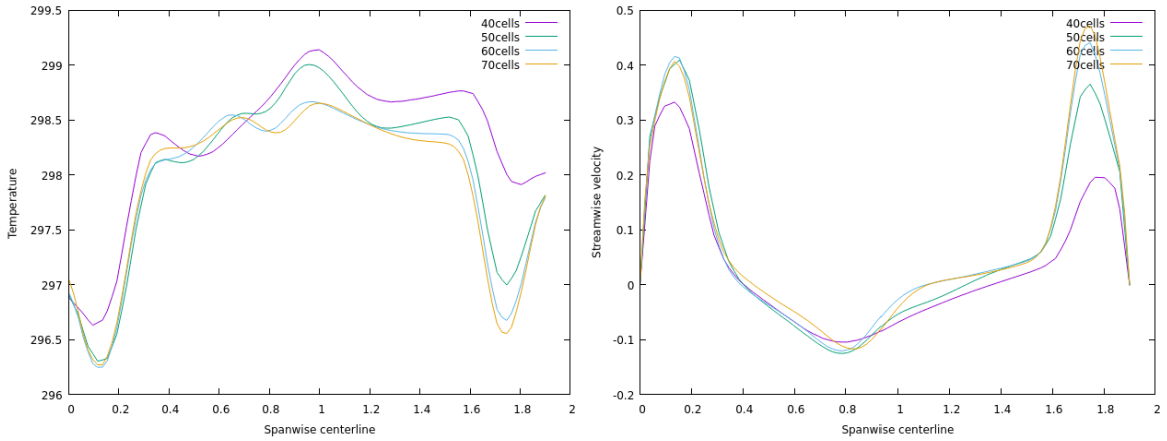

Figure 2: comparison of the flow characteristics for different grid refinement levels.

Turbulence Model Testing

To assess the impact of turbulence modeling on the CFD predictions, simulations were performed using two turbulence models: the realizable k−ε model and the k−ω SST model. As shown in Figure 3, the results from both models displayed only minor differences, confirming the robustness and consistency of the CFD setup.

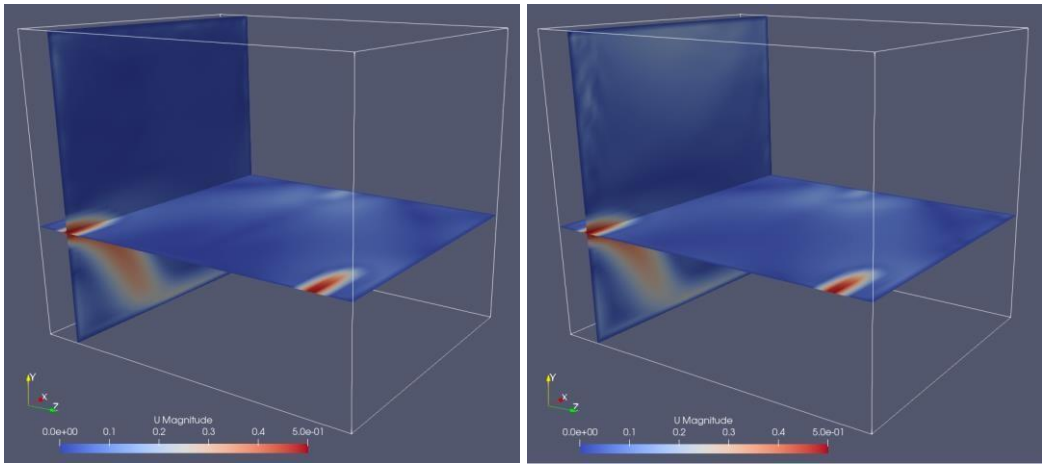

Figure 3: SDF comparison of the flow distribution for different turbulence models (realizable k-ε left, k-ω SST right).

2.3. Flow characteristics

The velocity distributions, as illustrated in Figure 4 for randomly selected flow realizations with varying setup parameters, exhibit the typical flow pattern within the cabin. The jet stream entering through the

air inlet bends upon reaching the back wall of the cabin, forming an outflow stream directed toward the outlet opening at the center of the front wall. The position of the inlet opening, particularly its height (Yin), significantly influences the bending height of the incoming stream within the cabin.

Despite the simplicity of the geometry, the resulting flow patterns highlight the presence of complex three-dimensional flow features. The temperature distribution within the cabin is governed by the interaction between the temperature of the incoming jet and the boundary conditions imposed on the walls. Consequently, the cabin is either heated or cooled as the flow stream progresses, depending on the temperature difference between the inflowing air and the wall temperatures.

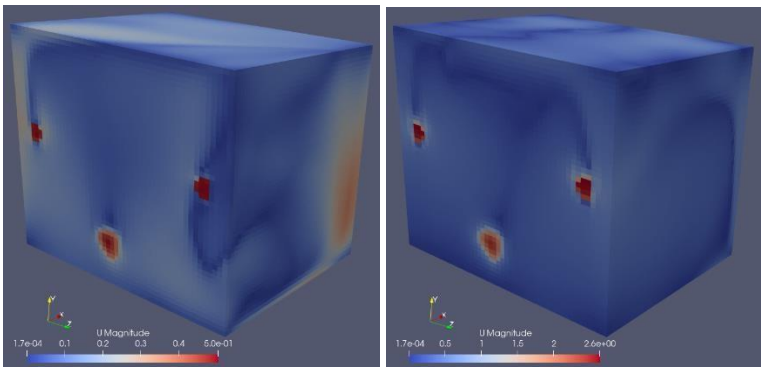

Figure 4: Fluid flow characteristics within the cabin from random simulations.

3. Surrogate model

The dataset was generated using an automated script that systematically varied the parameters of the numerical setup. A total of 729 numerical setups were simulated, with a random selection of flow realizations depicted in Figure 4. Data for the surrogate model were extracted from defined cutplanes within the 3D flow domain.

The surrogate model was developed using an artificial neural network (ANN) based on the UNET architecture. This model reconstructs the 3D flow field from the predictions made on the extracted cutplanes. For the ANN development, the implementation utilized Python 3 along with TensorFlow 2 libraries [9, 10].

3.1. Input channels

The surrogate model utilizes two sets of input data: the user-defined parameters and the Signed Distance Function (SDF). The user-defined parameters specify the numerical setup, including the position of the inlet openings (Yin), inlet velocity (Uin), inlet temperature (Tin), ambient temperature surrounding the cabin (Tamb), bottom wall temperature (Tbot), and heat flux applied to the top wall (hflux). These parameters, summarized in Table 1 and corresponding to the configuration in Figure 1, are arranged to represent their medium values and their upper and lower limits.

Table 1. The setup parameters.

	U_in	Y_in	T_in	T_amb	T_bot	h_flux
High	0.5	0.3	275	268	265	-20
Medium	1.5	0.6	295	288	290	20
Low	2.5	0.9	305	308	310	100

The Signed Distance Function (SDF) serves as the second input. It is a volume field representing the signed distance between a selected surface and the center of each computational cell. In OpenFOAM, the SDF can be computed using the *checkMesh* utility with the *-writeFields "(wallDistance*)" option (it requires the wallDist subdictionary in *fvSchemes* to define the method for wall distance calculation). For the present study, the SDF was calculated for seven specific patches: the left and right inlets, the central

outlet, the windows, the construction walls, and the top and bottom walls of the cabin (Figure 5). The SDF provides valuable geometric information about the flow domain and plays a crucial role in the surrogate model's ability to reconstruct the flow field.

3.2. Output channels

The output channels correspond to the fluid flow simulation results extracted from the defined cutplanes. In this study, nine cutplanes were positioned equidistantly along each axis. To better capture the near-wall effects, two additional cutplanes were placed near the front and back walls of the cabin along each axis. This configuration resulted in a total of (9+2)×3=33 cutplanes, where flow data were provided for each simulation setup.

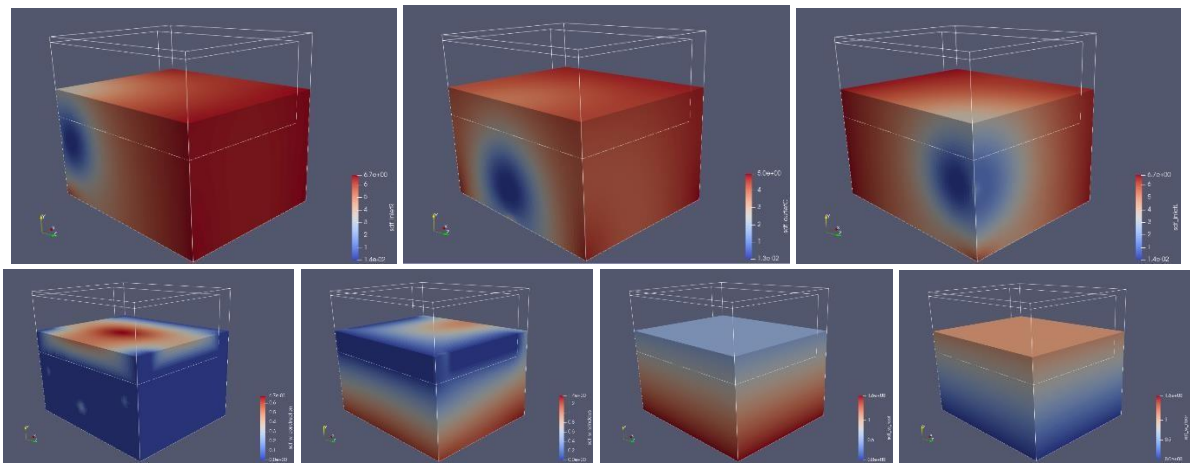

Figure 5: SDF distribution for the selected boundaries.

4. Model predictions

After training the model on the dataset, it was employed to predict the velocity along the specified cutlines. The model achieved an overall accuracy of 98%, demonstrating its strong ability to capture flow features for most cases. However, the performance varied slightly across different scenarios, with some cases achieving a higher accuracy of 99% (Figure 6) while a few others yielded somewhat lower accuracy of 95% (Figure 7). A detailed analysis would be required to identify the factors contributing to these discrepancies.

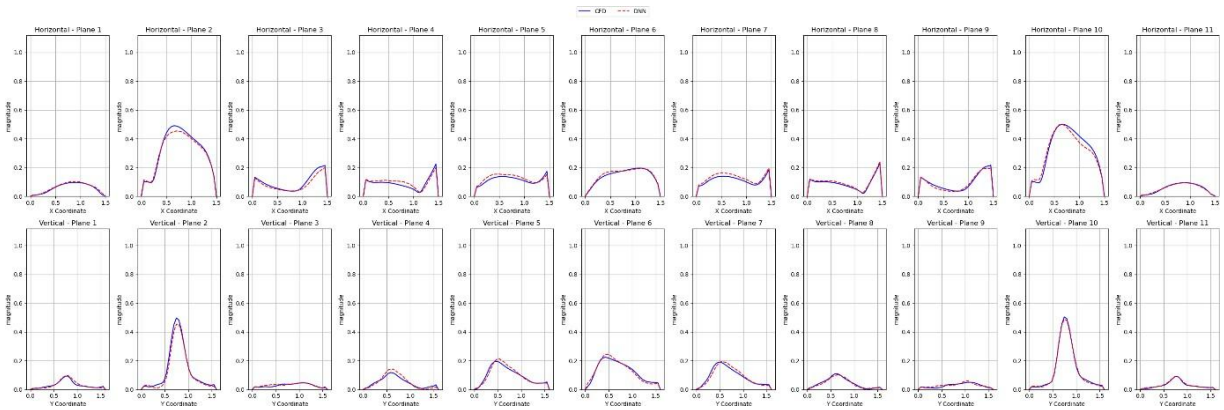

Figure 6: CFD (blue) and ANN (red) predictions along existing cutlines, for a case with high accuracy.

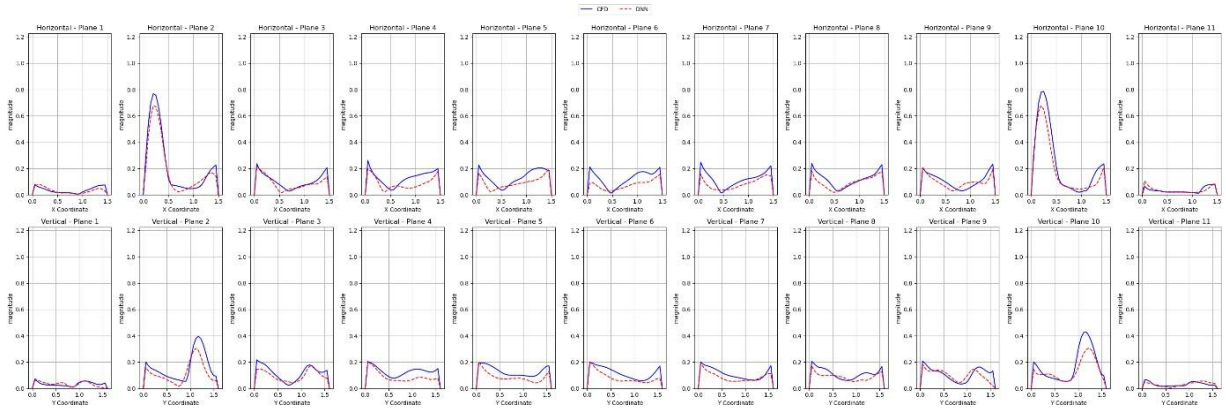

Figure 7: CFD (blue) and ANN (red) predictions along existing cutlines, for a case with low accuracy.

Once the predictions for the planes corresponding to the training dataset were evaluated, the model was further tested to reconstruct flow characteristics in arbitrarily located planes within the defined domain. The comparison shows (vertical cutplanes 1/6, 1/3 and 1/2 of the width, from left to right in Figure 8) that the main flow characteristics obtained by the CFD simulations are mainly recovered by the ANN predictions. This is particularly true for the regions with strong flow features (near the inlet, Figure 8 right) while in the quiescent regions with lower velocities the agreement between the ANN-predicted flow characteristics and the actual CFD simulations is weaker (in the outer corner, Figure 8 left). This highlights an important feature that the model is able to extrapolate flow characteristics accurately across diverse spatial locations, however one needs to be mindful of the generalization issue.

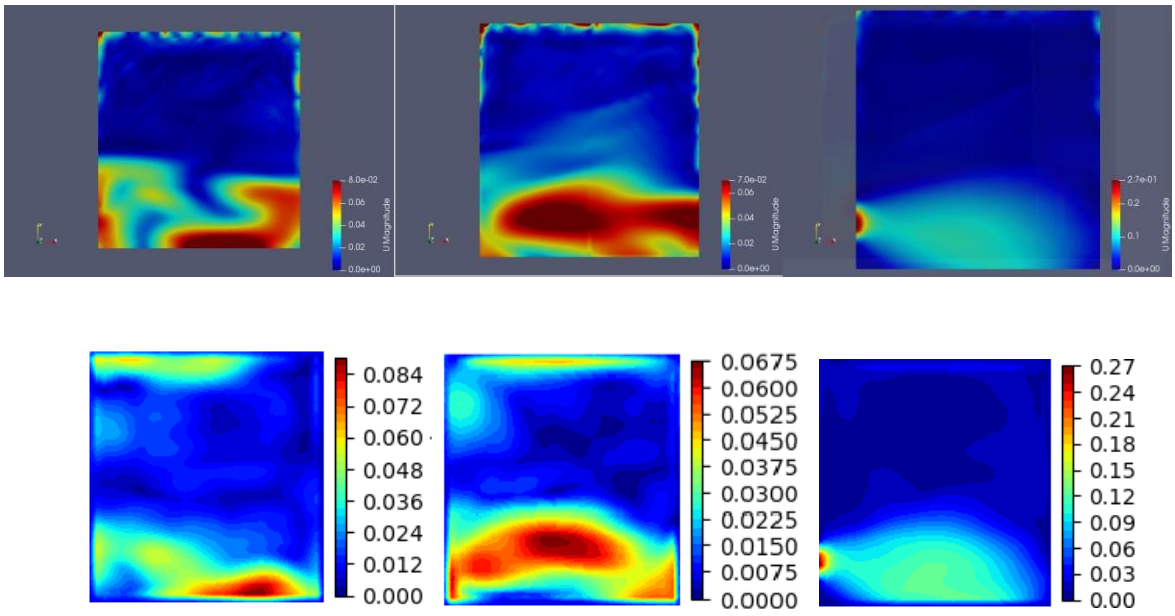

Figure 8: CFD (top) and ANN (bottom) predictions over arbitrary planes (both high and low accuracy case).

5. Conclusions

This study demonstrated the effective use of Computational Fluid Dynamics (CFD) simulations to generate a dataset for analysing flow characteristics by systematically varying boundary conditions, such as the inlet location and velocity. The workflow involved extracting cut-plane data to enable a detailed investigation of flow features.

An Artificial Neural Network (ANN) model based on the UNET architecture was developed as a surrogate for the analysed flow problem. While the complete study included surrogate modelling for

both temperature and velocity distributions, only the results related to the velocity field were presented for brevity. This approach efficiently represented complex flow dynamics, showcasing the potential of machine learning techniques in fluid dynamics applications.

The integration of CFD data with ANN modelling offers a robust framework for predicting flow behaviour under varying conditions, emphasizing the applicability of data-driven methods in addressing complex fluid flow problems.

Acknowledgments
This work was preformed within the project EMPOWER, which received funding from the European Union's HORIZON EUROPE research and innovation programme under grant agreement No. 101096028. The authors also acknowledge the support of VNIT for providing the computational resources required to execute the AI project and for hosting the "AI-Powered CFD" workshop, which significantly contributed to the development of this work.

References

[1] Lajunen, A. (2017). Energy Efficiency and Performance of Cabin Thermal Management in Electric Vehicles. SAE Technical Paper 2017-01-0192. https://doi.org/10.4271/2017-01-0192.
[2] Drage, P., Hinteregger, M., Zotter, G., Šimek, M. (2019). Cabin Conditioning for Electric Vehicles. ATZ worldwide 121, 44-49 . https://doi.org/10.1007/s38311-018-0209-2
[3] Aboosaidi, F., Warfield, M.,Choudhury, D. (1991). Computational Fluid Dynamics Applications in Airplane Cabin Ventilation System Design. SAE Technical Paper 911992, https://doi.org/10.4271/911992.
[4] Liu, G., Li, R., Zhou, X., Sun, T., Zhang, Y. (2023). Reconstruction and fast prediction of a 3D flow field based on a variational autoencoder. arXiv:2304.12598.
[5] Thuerey, N., Weissenow, K., Prantl, L., Hu, X. (2020). Deep learning methods for Reynolds-averaged Navier--Stokes simulations of airfoil flows. AIAA Journal, 58, 25—36.
[6] Ribeiro, M.D., Rehman, A., Ahmed, S., Dengel, A. (2021). DeepCFD: Efficient Steady-State Laminar Flow Approximation with Deep Convolutional Neural Networks. arXiv:2004.08826v3.
[7] Chen, L.W., Thuerey, N. (2023). Towards high-accuracy deep learning inference of compressible flows over aerofoils. Computers and Fluids, 250, 105707. https://doi.org/10.1016/j.compfluid.2022.105707.
[8] OpenCFD Ltd, (2012) OpenFOAM v2112, https://www.openfoam.com/documentation/guides/latest/doc/guide-applications-solvers-heat-transfer-buoyantSimpleFoam.html, 2012, [Online; accessed 30.10.2024].
[9] TensorFlow Team. (2020). TensorFlow 2.0: An end-to-end open-source platform for machine learning. Retrieved from https://www.tensorflow.org/
[10] Keras Team. (2020). Keras: The Python Deep Learning library. Retrieved from https://keras.io/