Development of machine learning techniques to enhance turbulence models

O Razizadeh\(^1\) and S N Yakovenko\(^2\)

\(^1\)Novosibirsk State University, Novosibirsk 630090, Russia
\(^2\)Khristianovich Institute of Theoretical and Applied Mechanics, Siberian Branch of Russian Academy of Sciences, Novosibirsk 630090, Russia

E-mail: s.yakovenko@mail.ru

Abstract. The implementation of the machine learning methods of convolutional neural networks to enhance RANS closure models is presented. The RANS models are not universal and accurate, however they are computationally affordable. Finding a way to improve the model predictability will be an advantage. For this, machine learning algorithms based on available high-fidelity data sets for canonical flow cases obtained from DNS and measurements can be helpful. The application of these algorithms for a fully-developed turbulent channel flows with periodic hills, in a square duct and for other cases is considered.

1. Introduction

There are several well-known methods to predict turbulent flows, including DNS, LES, RANS, hybrid RANS-LES (HRL) tools, each having their own advantages and crucial shortcomings. In spite of variety of the techniques proposed, the need for development of a computationally efficient tool to simulate flows in various industrial and environmental problems is still open. It turns out that DNS and LES approaches are not able to capture high-Reynolds-number cases with near-wall layers at present and in the foreseeable future, whereas steady and unsteady RANS models give insufficient and/or inaccurate information. The HRL methods are still at the initial development stage and remain conceptually complicated; their improvement continues by means of extensive tests to strengthen confidence in results and to delineate applicability range [1].

Moreover, heat and mass transfer problems need special attention. For instance, inaccurate prediction of near-wall layers may give incorrect wall heat fluxes, i.e. elaboration of techniques to numerically solve mass and energy (heat) conservation equations is of considerable interest, too. The issue of accurate comparison of a numerical solution of hybrid or LES methods with measured data merits consideration as well: one has to add both resolved explicitly and modeled contributions, although this is delicate with certain models. The issue of solution convergence with grid refinement in HRL, LES and DNS is crucial too, in contrast to RANS.

In surveys on numerical simulations of turbulent flows (e.g. [2, 3]), it is noted that RANS models, due to the limitations and high costs of eddy-resolving approaches, will remain a popular tool in the near future for scientific and engineering computations of fluid flows. It is expected [2] that along with an increase in use of HRL methods, using RANS closures will still be the norm in 2030. On the other hand, their poor predictability and non-universality can be depressing: a model calibrated on several tests after transition to another class of problems becomes unsatisfactory. In the last 20 years, there
have been almost no studies to improve the RANS models. For instance, suitable modifications of the advanced Reynolds stress model based on physical considerations can improve the description of flows with flat walls and a backward-facing step (see e.g. [4]); however, even for a simple-geometry flow in a plane channel, deviations from the DNS [5] and measurement data remain significant.

Since 2013, a new direction has appeared [6], based on the use of machine learning (ML) methods in addition to physical considerations to develop enhanced turbulence models and to calibrate them on big data arrays of high-fidelity methods (high-resolution LES, DNS, measurements). ML strategies allow constructing mappings between the data of high-fidelity methods and approximate models, effectively minimizing deviations between them and finding the optimal form of model corrections. Such a direction, which can be regarded as a breakthrough, has been mentioned in the recent reviews [6-8] and has become possible due to the recent achievements in high-performance computing, which allow obtaining high-fidelity databases for the characteristics of canonical flows, and in big data processing algorithms, based on the use of artificial intelligence.

In recent published works, popular one- and two-parameter turbulence models were taken as the baseline ones, and a large set of canonical test flows (flows in channels with flat and wavy walls, flow around obstacles, wakes, jets) were considered to train enhanced models. It should also be mentioned that the ML methods are used not only for the RANS models but also for the LES ones, e.g. in channel flow simulations [9]. There are very few works on the ML application to the development of turbulence models for flows complicated by the presence of heat/mass transfer and two phases [10].

In the framework of the ML methods applied to enhance the RANS models, four main versions of algorithms with different architecture developed after 2016 can be distinguished. The first one corresponds to a new architecture based on tensor basis neural networks (TBNN) [11] with embedded invariance properties to determine the coefficients of the functional dependence between the Reynolds stress tensor and the finite set of functions of the mean strain and rotation rate tensors. Based on the TBNN method, the tensor basis decision tree (TBDT) and random forest (TBRF) methods have been developed [12]. In the second algorithm [13], for the same purpose, symbolic regression and gene expression algorithms were used to correct explicit algebraic models of turbulent stresses. In the third approach [14], the field inversion and machine learning (FIML) strategy was proposed, where the model is improved when calibrated on the basis of high-fidelity data in two stages: the first one solves the inverse problem of determining the discrepancy function, and the second uses this function to obtain a correction to a baseline turbulence model using the ML-algorithms. The FIML approach was further developed using neural networks [15] and allowing, for instance, improving predictions of a flow around the airfoil. For the fourth architecture version, the convolutional neural network (CNN) has been used in preliminary tests for the square duct flow case [12].

In the present study, to improve predictions of the Reynolds stress anisotropy (RSA) tensor

\[ b_i = \frac{\langle u_i u_j^t \rangle}{2k} - \frac{1}{3} \delta_{ij} \]  

(computed from the Reynolds stress (RS) tensor \( \langle u_i u_j^t \rangle \) and the turbulent kinetic energy \( k = 0.5 \langle u_i u_i^t \rangle \)) by the widely used k-ω turbulence model for flows with periodic hills [16] and in the square duct [17], two different CNN architectures following [18, 19] are implemented. The CNN is a type of neural networks that has been used mostly in computer vision problems such as face recognition, driverless cars and so on. Depending on the task, such a technique is usually applied for classification, and sometimes for regression. The classification task means that the output is discrete and regression means that the output is continuous. The CNN architectures used here include the “mean squared error” loss function, or the “squared hinge” loss function appearing as one of layers.

2. Methodology

Initially, a baseline RANS model is applied to obtain low-fidelity data which are used for subsequent training and improving the model by means of the ML methods. For such a training, high-fidelity data as a target solution are taken from available datasets. Each mesh cell of the computational grid is used as input for the CNN architectures, so it is considered as one pixel as in the previous studies [11, 12].
Note that in the ML algorithm [12], where only the CNN was used, some noise appeared in the components of $b_{ij}$, whereas the implementation of the TBRF or the TBNN methods for a square duct flow strongly overestimates the component $b_{23}$ which is the most sensitive to the model predictability among other $b_{ij}$ components for such a flow. One can also see for this case [12] that $b_{23} = 0$ for the baseline RANS model, and further efforts are needed to improve the ML+RANS algorithms. For the deep neural network applied here, it is possible to examine whether noise can be reduced without using the Gaussian filter added in [12]. As noted above, architectures with a mean squared error or with a squared hinge loss are implemented here, using the Tensorflow and the Keras libraries.

The following methodology is used. First, mesh nodes are extracted from the LES or the DNS data and then imported into the ANSYS ICEM meshing software. Second, a mesh is created inside the ICEM and exported to OpenFOAM. Third, the results of OpenFOAM computations with adjusting boundary conditions are obtained by running the simpleFoam solver to get the steady-state solution of the $k$-$\omega$ model, and convergence to the solution, which is independent of numerical aspects is checked. For the ML stage, the latter results are fed into CNN architectures (together with the high fidelity data of benchmark LES or DNS solutions to get the discrepancy of the low-fidelity RANS data from these solutions) in order to see how the model can improve the predictions of $b_{ij}$. Before feeding the neural network with the obtained RANS data, the interpolation for these data values is done to make a consistent comparison between the RANS and DNS/LES datasets at the same mesh points.

![Figure 1. A flow with periodic hills: a scheme of mesh and characteristics in the LES [16].](image)
3. Test cases

In the present study, as the first test case, the developed turbulent flow in the two-dimensional channel with periodic hills mounted on the bottom at Re = $U_b H/\nu = 10\,595$ [16] is considered. The scheme of mesh and spatial distributions of the mean velocity vectors and magnitudes, the turbulent kinetic energy, the shear and normal components of the RS and the RSA tensors with normalization by the hill height $H$ and the bulk velocity $U_b$ (i.e. averaged over the flow cross-section) are presented in Figure 1 for the LES data [16].

The second test case to be taken is a square duct flow at Re$_B = U_b D/\nu = 10\,320$, Re$_\tau = u_\tau D/\nu = 600$ [17] (Figure 2), where $D$ is the duct width, $U_b$ is the bulk axial velocity, $u_\tau$ is the mean friction velocity. Such a flow was also used in the previous ML+RANS studies [11, 12], because it has a relatively simple geometry, whereas all six components of the RSA tensor are non-zero.

Figure 2. A flow in the square duct: characteristics in the DNS [17].
It should be noted that the two-parameter $(k-\omega, k-\varepsilon)$ turbulence models with the linear isotropic-eddy-viscosity Boussinesq hypothesis for the RS tensor yield zero values of four components ($b_{11}, b_{22}, b_{33}, b_{23}$) of the RSA tensor and no secondary vortex motions in the duct cross-section [11, 12]. Thus, it is a challenge for an enhanced ML+RANS model to correctly predict the fine flow features.

Next, the numerical simulations by the baseline RANS model are performed for the above-noted flow cases, with calculation of the components of the RSA tensor $b_{ij}$ (Figures 3, 4).

**Figure 3.** A flow with periodic hills: a scheme of mesh and main characteristics in the present RANS computations (values are non-dimensionalized using the bulk velocity $U_b$), where $\langle w'w' \rangle = 2k/3$ since $W = 0$, therefore $b_{11} + b_{22} = b_{33} = 0$. 


Figure 4. A flow in the square duct: the RANS computation (values are non-dimensionalized using \(U_b\)) where \(\langle u' u' \rangle = \langle v' v' \rangle = \langle w' w' \rangle = 2k/3\) and \(V = W = \langle v' w' \rangle = 0\), therefore \(b_{11} = b_{22} = b_{33} = b_{23} = 0\).

4. Results

The model based on the CNN architectures is trained first on the LES data [16] and then is used to predict a dataset for the periodic hills case initially simulated by the baseline RANS model.

Table 1 shows the details of the CNN architecture applied in the present studies. In this architecture, the total number of parameters is 333 795, the number of trainable parameters is also 333 795, and there are no non-trainable parameters. For the first version (the CNN with the mean square error loss function) the accuracy is 0.6395 after 500 epochs. For another architecture (the CNN with the squared hinge loss function) a slightly lower accuracy of 0.6035 arises after 500 epochs.

Table 1. Details of the CNN architecture implementation for the periodic hills flow case.

| Layer (type)       | Output shape     | Number of parameters |
|--------------------|------------------|----------------------|
| Conv2d_1 (Conv2D) | (None, 131, 150, 16) | 1 216                |
| Conv2d_2 (Conv2D) | (None, 131, 150, 32) | 41 504               |
| Conv2d_3 (Conv2D) | (None, 131, 150, 64) | 247 872              |
| Conv2d_4 (Conv2D) | (None, 131, 150, 3)  | 43 203               |

Next, the second flow case (the square duct) is considered, with the DNS dataset [17] used for training the same CNN architectures, after numerical simulations by the baseline RANS \(k-\omega\) model initially taken for prediction. The details of the CNN architecture for the second flow case, where numbers of trainable (333 795) and non-trainable (0) parameters were the same, are shown in Table 2. The accuracy for the model with the mean square loss is 0.5150 after 500 epochs, whereas for the second architecture with the squared hinge loss, the accuracy is 0.3715 after 500 epochs. Therefore, the first architecture with the mean squared error loss function gives better accuracy for both flow cases: the square duct and the periodic hills.
Table 2. Details of the CNN architecture implementation for the square duct flow case.

| Layer (type)   | Output shape | Number of parameters |
|----------------|--------------|----------------------|
| Conv2d_1 (Conv2D) | (None, 49, 49, 16) | 1 216 |
| Conv2d_2 (Conv2D) | (None, 49, 49, 16) | 41 504 |
| Conv2d_3 (Conv2D) | (None, 49, 49, 64) | 247 872 |
| Conv2d_4 (Conv2D) | (None, 49, 49, 3)  | 43 203 |

To illustrate the application of the CNN method for a turbulent flow with periodic hills placed on the channel bottom, the components of the Reynolds-stress anisotropy tensor \( b_{22} \) and \( b_{12} \) have been first obtained and plotted in the present studies (Figure 5). The corresponding results for the RSA components \( b_{12} \) and \( b_{23} \) in a square duct flow are given next (Figure 6).

Figure 5. A flow with periodic hills: contours of the RSA tensor components obtained using the CNN.

Figure 6. A flow in the square duct: contours of the RSA tensor components obtained using the CNN.
One can see that the RANS simulations give considerable deviations of the values of $b_{ij}$ (Figures 3, 4) from those of [11, 12], e.g. zero values of $b_{11}$, $b_{22}$, $b_{33}$, $b_{23}$ as noted above, which is qualitatively incorrect as is seen in Figure 2. On the other hand, the CNN yields the same behavior and levels of $b_{ij}$ as in the LES [11] and the DNS [12] (Figures 5 and 6), although there is some noise in contour plots as is mentioned above for the square duct flow computations by the CNN and those reported in [12].

It should be noted that the second version of the CNN architecture (with the squared hinge loss function) gives physically invalid results, with values of $b_{ij}$ larger than unity, that are not shown here.

5. Conclusion

The results for the implementation of the convolutional neural network to enhance the RANS model performance using the high-fidelity data of the DNS or the LES are presented. It is shown that predictions for the Reynolds-stress anisotropy tensor components are improved after implementation of the CNN with the mean square error loss function in comparison with that of the baseline RANS model for flows in channels of different geometry.

The remaining challenge is to build a neural network that would be flexible due to different architectures and could enhance the RANS model performance for different flow cases. For turbulence modelers and the CFD community, it is important to have a unified model (to be implemented into a CFD solver) based on some explicit modifications in governing equations like in [13, 14].

The outcome obtained can be helpful for further development of the machine learning tools for the computational fluid dynamics issues to improve the efficiency and fidelity of turbulence models, and to represent the complex physics of fluid flows.

Acknowledgements

The study was partly supported by Russian Foundation for Basic Research (Project No. 17-01-00332) and is performed within the framework of the Program of Fundamental Scientific Research of the state academies of sciences in 2013-2020 (Project No. AAAA-A17-117030610128-8).

References

[1] Frohlich J and von Terzi D 2018 Prog. Aerosp. Sci. 44 349
[2] Slotnick J, Khodadoust A, Alonso J, Darmofal D, Gropp W, Lurie E and Mavriplis D 2014 NASA Technical Report (NASA/CR-2014-21878)
[3] Durbin P 2018 Annu. Rev. Fluid Mech. 50 77
[4] Yakovenko S N and Chang K C 2019 AIAA J. 57 279
[5] Lee M and Moser R D 2015 J. Fluid Mech. 774 395
[6] Duraisamy K, Iaccarino G and Xiao H 2019 Annu. Rev. Fluid Mech. 51 357
[7] Tyacke J, Vadlamani N R, Trojak W, Watson R, Ma Y and Tucker P G 2019 Prog. Aerosp. Sci. 110 100554
[8] Brunton S L, Noack B R and Koumoutsakos P 2020 Annu. Rev. Fluid Mech. 52 477
[9] Gawahara M and Hattori Y 2017 Phys. Rev. Fluids 2 054604
[10] Chang C W 2018 Ph.D. Thesis (North Carolina State University)
[11] Ling J, Kurzawski A and Templeton J 2016 J. Fluid Mech. 807 155
[12] Kaandorp M 2018 M.Sc. Thesis (Delft University of Technology)
[13] Weatheritt J and Sandberg R D 2016 J. Comput. Phys. 325 22
[14] Parish E J, Duraisamy K 2016 J. Comput. Phys. 305 758
[15] Holland J R 2019 Ph.D. Thesis (University of Maryland, College Park)
[16] Breuer M, Peller N, Rapp C and Manhart M 2009 Comput. Fluids 38 433
[17] Huser A, Biringen S 1993 J. Fluid Mech. 257 65
[18] Tang Y 2013 ICML 2013 Challenges in Representation Learning Workshop
[19] Huynh P, Nguyen V and Do T 2018 Modern Approaches for Intelligent Information and Database Systems (Studies in Computational Intelligence vol 769) ed by A Sieminski, A Kozierkiewicz, M Nunez and Q T Ha (Springer) p 233