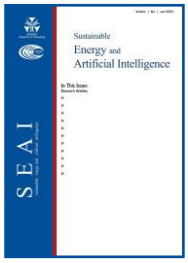
Hamedan University of Technology
Iranian Association of Electrical and Electronics Engineers

Sustainable Energy and Artificial Intelligence

Online ISSN: 3060-8015

Journal Homepage: www.enai.hut.ac.ir



A Numerical Investigation of Subgrid-Scale Stresses in Turbulent Channel Flow Using Artificial Intelligence Techniques

Mohammadreza Azarshab, Zeinab Pouransari*

Department of Mechanical Engineering, Iran University of Science and Technology, Tehran, Iran
** Corresponding Author: pouransari@iust.ac.ir*

Article Info

Article type:
Original Article

Article history:
Received 2025-10-01;
Revised 2025-11-14;
Accepted 2025-11-24.

How to cite this article:
Azarshab, M. R. and Pouransari, Z. (2025). A Numerical Investigation of Subgrid-Scale Stresses in Turbulent cChannel Flow Using Artificial Intelligence Techniques. *Sustainable Energy and Artificial Intelligence*, 2(1), 37-45.
DOI: 10.61882/seai.2510-1035

Abstract

This study presents a data-driven approach for modeling subgrid-scale (SGS) stresses in turbulent channel flow using a fully connected neural network (FCNN), also known as a multilayer perceptron (MLP). Our data-driven closure model is based on localized learning, and the FCNN architecture has a point-to-point mapping framework. The MLP-based SGS model is trained using direct numerical simulation (DNS) data of a turbulent channel flow at the bulk Reynolds number $Re = 4400$, corresponding to the friction Reynolds number $Re_\tau = 180$. Performance of the FCCN model is assessed using *a priori* analysis in comparison with the filtered DNS (fDNS) data of the turbulent channel flow. Resolved flow statistics, including filtered velocity gradients and wall distance, are employed as input parameters for training the FCNN. To evaluate the significance of the feature inputs, we utilize random forest regression, which reveals that y^+ is a critical factor in predicting the output, which can reduce the compactional costs. In an *a priori* test, the model achieved a correlation coefficient exceeding 95% for the components of the SGS stress tensor, which have non-zero mean values. Finally, potential strategies for improving prediction accuracy and overall performance are explored. The findings demonstrate that the proposed FCNN can accurately reproduce key turbulence characteristics, offering a promising step toward efficient, AI-based turbulence modeling.

***Keywords*:** Turbulent Channel Flow, FCNN, SGS- Modeling, *A Priori* Analysis, Random Forest Regression.

1. Introduction

Simulations of turbulent flows aim to model the complex and chaotic behavior of fluid motion, often using computational fluid dynamics techniques, to understand and predict phenomena such as turbulence, heat transfer, and pressure distribution, which are critical in various engineering applications [1]. The DNS of complex fluid flows, often found in engineering and geophysical applications, is computationally impractical because it requires capturing a wide range of spatial and temporal scales. Large eddy simulation (LES) and Reynolds-averaged Navier-Stokes (RANS) modeling are two widely used approaches that offer accurate predictions by considering the interactions between the smaller unresolved scales and the larger resolved ones. Developing these models, known as the turbulence closure problem, has been a persistent challenge in fluid mechanics [2]. In LES, the Navier-Stokes equations are filtered to separate large and small scales, which leads to simpler equations that are quicker to solve than the full Navier-Stokes

equations. The interaction between the resolved and unresolved scales is captured by adding SGS modeling [3],The main role of the SGS model is to represent the energy dissipation, which describes how energy moves from the larger scales to the smaller scales. This dissipation is modeled using an eddy viscosity parameter based on the grid-resolved data[4]. A wide range of classical SGS models has been proposed. For instance, the Smagorinsky model [5] utilizes a global constant, known as the Smagorinsky coefficient, to represent the mean dissipation of energy. However, numerous studies have shown that a single value for the Smagorinsky coefficient is not suitable for capturing a wide range of flow phenomena [6]. The limitations of the static Smagorinsky model can be addressed by employing the dynamic Smagorinsky model (DSM) introduced by Germano et al. [2]. Several variations of Germano's DSM have been developed to address the limitations of the standard DSM. These include a localized version to resolve mathematical inconsistencies [7], a Lagrangian version [8], and a DSM with an added corrector step [9].

Over the past decade, the vast amount of data generated from experiments and high-fidelity simulations has enabled the widespread application of machine learning (ML) algorithms in fluid mechanics. ML techniques are now employed in various areas such as flow control, flow optimization, reduced-order modeling, flow reconstruction, super-resolution, and flow cleansing [10]. One of the earliest uses of deep learning turbulence was demonstrated by Milano and Koumoutsakos [11], who utilized artificial neural networks to reconstruct near-wall turbulence, significantly improving the prediction accuracy of velocity fields. Since then, numerous ML approaches have been proposed, including shallow decoders for flow reconstruction [12], convolutional neural networks for super-resolution of turbulent flows [13], and deep convolutional autoencoders for nonlinear model order reduction.

ML algorithms have also been effectively applied to the turbulence closure problem, alongside reduced-order modeling and chaotic dynamical systems, helping to reduce the time of simulation in RANS and LES models. For instance, Maulik and San [14], as well as Maulik et al. [15], employed multi-grid-point data from filtered flow fields as inputs to predict single-grid-point deconvolved vorticity and stream functions in turbulence. Similarly, Zhou et al. [16] proposed a novel approach for SGS stress closure by using the filter size and the velocity gradient tensor as input features of an ML algorithm. Sarghini et al.[17] employed LES data to predict the coefficients of the Smagorinsky term within a Smagorinsky mixed model. Similarly, Gamahara and Hattori [18] showed that includingy^+along with the filtered strain rate tensor improved neural network (NN) predictions of non-diagonal SGS stress components in turbulent channel flow. Park and Choi [19] utilized a large dataset of instantaneous flow fields and a multi-grid point approach for NN training without incorporating y^+as an input variable. While the *a priori* tests showed relatively low correlation coefficients for the predicted variables, *a posteriori* tests revealed that the NN model outperformed traditional SGS models, including the DSM and the scale-similarity model. These previous studies have demonstrated the potential of NN-based models for turbulence modeling; however, they primarily relied on the filtered strain rate tensor as input. The motivation for this study is to highlight the importance of feature inputs that have a significant influence on learning SGS stresses and improving prediction accuracy. To achieve this objective, we propose a FCNN specifically designed for a $3D$ turbulent channel flow at a single grid point [20]. The FCNN employs filtered velocity gradients and wall distance as input features. The present work model differs from previous studies in several important aspects, such as the inclusion of wall distance as an input feature, using localized training, and the use of single-point mapping. Additionally, we focus solely on *a priori* testing to assess the model's predictive performance. The rest of this paper is organized as follows:

In Section 1, we introduce the problem definition, which is divided into three parts. First, we describe the turbulent channel flow. Next, we discuss the numerical method used. Finally, we explain the architecture of the FCNN and its setup. In Section 2, we present the results and discussion, organized into three parts. We begin by discussing the training process and loss validation. Then, we analyze the importance of the input features using the random forest regression algorithm. Lastly, we explain the predicted SGS stresses.

2. Problem Definition

2-1. Turbulent Channel Flow

A DNS of turbulent channel flow at the friction Reynolds number $Re_\tau = 180$ was performed to construct the input-output datasets required for training the FCNN [4]. We used a single time snapshot for *a priori* training/assessment; therefore, temporal resolution does not apply to our analysis. The friction Reynolds number and the

bulk Reynolds number are defined as $Re_\tau = u_\tau \delta/\vartheta$ and $Re_b = U_b(2\delta)/\nu$, where u_τ is the friction velocity, δ is the half of the channel width, U_b represents the bulk velocity, and ν denotes the kinematic viscosity. As depicted in Figure 1, the computational domain spans $L_x = 2\pi\delta$, $L_y = 2\delta$, $L_z = \pi\delta$, with x, y and z denoting streamwise, wall-normal, and spanwise directions, respectively. Periodic boundary conditions are imposed in the streamwise and spanwise directions, whereas no-slip and boundary conditions are enforced at the channel walls ($y/\delta = \pm 1$). Spatial discretization is carried out using Fourier series in the homogeneous directions and Chebyshev polynomial expansions in the wall-normal direction, with aliasing errors mitigated through spectral filtering to ensure numerical accuracy and solution stability. The computational grid consists of $N_x = 128, N_y = 129\ and\ N_z = 128$ points. The corresponding grid spacings, expressed in wall units, are given in equation (1),

$$\Delta x^+ = \frac{u_\tau \Delta x}{\vartheta}, \Delta y^+ = \frac{u_\tau \Delta y}{\vartheta}, \Delta z^+ = \frac{u_\tau \Delta z}{\vartheta}. \quad (1)$$

Here $\Delta x, \Delta y$ and Δz represent the grid spacings in the x, y and z directions, respectively. The friction velocity u_τ is defined as $u_\tau = \sqrt{\frac{\tau_{wall}}{\rho}}$, where τ_{wall} is the wall shear stress and ρ is the fluid density.

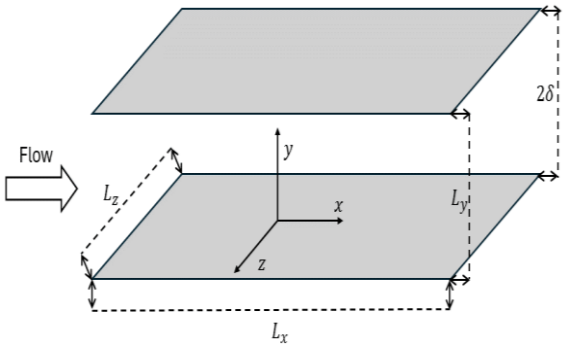


Fig. 1. A schematic representation of the computational domain.

2-2. Numerical method

2-2-1. Governing equations

The Navier–Stokes are given as:

$$\frac{\partial u_i}{\partial x_i} = 0 \quad (2)$$

$$\frac{\partial u_i}{\partial t} + \frac{\partial u_i u_j}{\partial x_j} = -\frac{1}{\rho}\frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_j \partial x_j} \quad (3)$$

Here,x_i represents the cartesian coordinates, $u_i$denotes the velocity components, t is time, p is pressure, ρ is density, and ν is the kinematic viscosity of the fluid.

2-2-2. LES and Filtering Method

LES is a technique for turbulent flow simulations by directly capturing large-scale structures and modeling the influence of smaller unresolved scales. This is accomplished using filter operations that divide the flow field into resolved large-scale features and SGS components. The filtered Navier–Stokes equations for LES are derived by applying a top-hat filter shows by$(\bar{\cdot})$to the original equations.

$$\frac{\partial \bar{u}_i}{\partial x_i} = 0 \quad (4)$$

$$\frac{\partial \bar{u}_i}{\partial t} + \frac{\partial \bar{u}_i \partial \bar{u}_j}{\partial x_j} = -\frac{1}{\rho}\frac{\partial \bar{p}}{\partial x_i} + \nu \frac{\partial^2 \bar{u}_i}{\partial x_j \partial x_j} - \frac{\partial \tau_{ij}}{\partial x_j} \quad (5)$$

Here, $\bar{u}_i$ represents the filtered velocity components, $\bar{p}$ is the filtered pressure, and $\tau_{ij} = \overline{u_i u_j} - \bar{u}_i \bar{u}_j$ represents the SGS stress tensor.

Prior to computing the sub-grid scale components from DNS data using Eq. (6), a filtering operation must be conducted. Within the framework of LES, this filtering is conventionally implemented through a convolution of the target function with a prescribed filter kernel.

$$\tilde{f} = \int_D f(\zeta) G_{\bar{\Delta}}(x - \zeta) d\zeta \quad (6)$$

The convolution-based filtering is performed over the domain D utilizing the top-hat kernel $G_{\bar{\Delta}}$[21], with the corresponding filter width given by $\bar{\Delta} = 4\Delta$.

The flow variables are processed using a top-filter, which is implemented through a one-dimensional top-hat filter formulation.

$$f_i = \frac{1}{2n}\left(f_{i-n/2} + 2\sum_{i-n/2+1}^{i+n/2-1} f_i + f_{i+n/2} \right) \quad (7)$$

For further information regarding the filtering technique, refer to [22]. Here f_i represents a function of interest at a discrete spatial location. In this study, the filter-to-grid size ratio is defined as $n = \bar{\Delta}/\Delta$. Filtering is performed in the horizontal $x - z$ plane, which is a standard approach in LES of turbulent channel flows with non-uniform grid spacing in the wall-normal direction [4]. Filtering in the wall-normal direction using the top-hat filter employed in this work is deliberately avoided, as it would violate the continuity equation due to the non-uniform grid distribution in that direction.

For this investigation, the filtering scale ratio n is set to 4 in both the streamwise x and spanwise z directions. The filtering procedure is applied to a single flow variable from the DNS[20]. The impact of the size on the SGS terms is studied through an

analysis of the turbulent kinetic energy, which indicates that the filtering approach is consistence with the anticipated results. However, as the filter size increases, the accuracy tends to decrease. A larger Δ means more energy goes unresolved, making the SGS terms larger and more nonlocal, leading to higher errors in the overall learning task.

2-2-3. Architecture of the artificial neural network

This study utilizes FCNN, also known as MLP, to predict the six components of the SGS stress tensor, denoted as τ_{ij}. The network is trained using filtered input data, which are the gradient velocities, obtained from fDNS fields, and another input data is y^+. These fields are produced by applying top-hat filters to the instantaneous DNS data of turbulent channel flow [23]. All pre-processing and numerical operations, including DNS filtering with the top-hat filter, are performed in MATLAB. The processed datasets are then exported and used to train/evaluate the neural networks in Python with an open-source library named TensorFlow and Keras. The FCNN, implemented using the Tensorflow library and Keras API, serves as a nonlinear approximation model for predicting the SGS stress tensor. In contrast to the method proposed by Park and Choi [19], which used flow data from multiple grid points and 200 instantaneous flow fields, this study uses flow variables from a single grid point within a single instantaneous flow field as input data. To improve the model's generalization and learning capability, the dataset from one flow field is randomly divided, with 80% allocated for training and the remaining 20% for testing. The architecture of the FCNN, as depicted in Figure 3, consists of an input layer with 10 neurons, two hidden layers, each containing 150 neurons, and an output layer with 6 neurons. The hidden layers employ Rectified Linear Unit (ReLU) activation functions, while the Adam optimizer is utilized for efficient model training. The details of the hyperparameter tuning with grid search are summarized in Table 1. Hyperparameter optimization was performed to identify the most suitable configuration, as shown in Table 2. The model was subsequently trained with an extended epoch limit of 4000, incorporating early stopping to mitigate overfitting. Convergence of the loss was achieved within approximately 306 epochs. The learning rate was set at 0.001, as suggested in previous studies [19]. The details of the configuration used to construct the FCNN are summarized in Table 2. The inputs provided to the FCNN model are outlined below:

$$X_{1-10} = [\frac{\overline{\partial u_i}}{\partial x_j}, y^+] \quad (8)$$

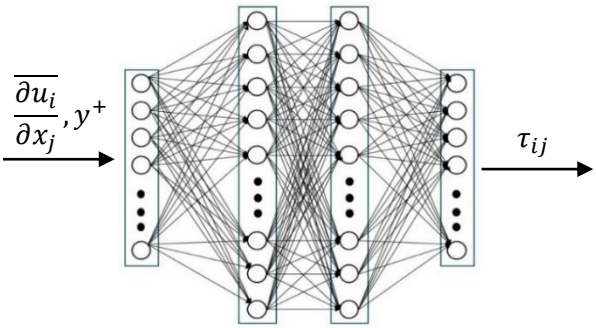


Fig. 3. The structure of the FCNN with {10,150,150,6} neurons for predicting SGS stress tensor.

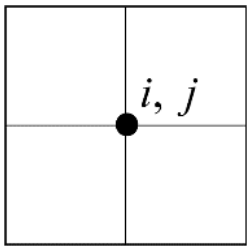


Fig. 4. The FCNN-based point-to-point mapping for recognizing grid structures.

Table1. The hyperparameter tuning with grid search.

Parameter	value
Number of layers	$2, 3, 4$
Number of neurons for per layer	$64, 128, 150, 256$
Learning rate	0.001
Activation function	$ReLU, ELU, GELU$
Optimizer function	$Adam$
Batch normalization	$True$
$K-fold$	$True - n = 5$
Batch size	$32, 64, 128$
Early stopping	$patience = 30$

Table 2. The best configuration for the training.

Layers	Neurons per hidden layer	Hidden layer	Batch size	Epochs
2	150	2	64	4000

Table 3. The setup for FCNN model.

Model	Input	Learning rate	Batch Size	Output
FCNN	$\frac{\partial \bar{u}_i}{\partial x_j}, y^+$	0.001	64	τ_{ij}

To ensure reliable performance for the evaluation and to reduce the risk of overfitting to a specific partition of the data, $K-fold$ cross-validation was employed with K set to 5 [24]. This technique enhances the robustness and

generalizability of the model by ensuring that every data point is used for both training and validation, mitigating the risk of bias due to data partitioning.

During the training process, the input was normalized by using $X' = X_{min} - \mu/\sigma$ in order to achieve a zero mean and unit variance, which enhances the training performance. Where X' denotes the normalized input data, X_{min} refers to the original input data, μ indicates the mean of the original input data, and σ represents its standard deviation.

The transition from $(n-1)_{th}$ layer to n_{th} layer is described mathematically. The two hidden layers of the NN model are defined by the following expressions.

$$X_i^{(n)} = \sigma\left(\sum_{i=1} X_i^{(n-1)} W_i^{(n)} + B_i^{(n)}\right) \quad (9)$$

The parameters $W_i^{(n)}$and $B_i^{(n)}$ were optimized using backpropagation with the Adam algorithm. The output layer was linear. The activation function in the hidden layer $\sigma(a)$ was the rectified linear unit, where for positive values a is given, and for negative values the function holds zero values.

In FCNN, weights are initialized with small random values to avoid redundant computations and enhance learning, while biases are set to zero to simplify the initial state. Hidden layers use the ReLU activation function, which is computationally efficient and prevents the vanishing gradient problem, facilitating effective training of deep networks. The output layer employs a linear activation function, ideal for predicting continuous values like the components of the SGS stress tensor. This combination ensures efficient training and accurate regression performance.

The weights and biases of each layer are progressively adjusted and refined throughout the learning process. The loss function is defined using mean squared error (MSE), as described below:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \quad (10)$$

The variables Y_i , $\widehat{Y_i}$ and n represent the true values, predicted values, and the total number of data points, respectively.

3. Results and Discussion

3-1. Training and Loss Validation

In this section, we discussed the evaluation and performance of the model, as well as its robustness. To assess the model's accuracy and robustness, various metrics can be used, including the R^2-score MSE and Mean Absolute Error (MAE) [25]. These metrics are calculated and presented in the table 5.

Table 4. The performance of the model.

Model	MAE	MSE	R^2 score
FCNN	8.306×10^{-5}	2.98×10^{-8}	86.72%

The R^2 score of 86.72% indicates a good fit and suggests the model captures most of the variability in the data. However, some of the variance remains unexplained, possibly due to factors not included in the model, such as omitted variables, noise, or the complexity of the input features. The MAE indicates that, on average, the predicted values deviate very little from the actual values, which reflects strong model performance. The MSE, though harder to interpret directly, suggests that the model's predictions are highly accurate with very small squared errors. Together, the low values of MAE and MSE reinforce the conclusion that the model is making precise predictions with minimal error.

When training an ML model, plotting the training loss and validation loss against the number of epochs is a common practice to monitor the model's performance, as it allows you to identify issues like overfitting and underfitting by observing how the model learns on both the training data and a separate validation set throughout the training process [26], as it is shown in the Fig. 4.

Fig. 4 shows the training and validation losses for 306 epochs, illustrating the model's learning and generalization. The training loss (black line) steadily decreases, indicating the model is learning well from the training data. The validation loss (green line) initially decreases alongside the training loss, suggesting good generalization to unseen data. Between epochs 50 and 100, both losses stabilize, showing that the model has likely converged to optimal performance. The small gap between the training and validation losses indicates that the model is neither overfitting nor underfitting, suggesting good regularization. While slight fluctuations in the validation loss are normal, they may be due to variations in the dataset. Overall, the trends suggest a well-tuned model, capable of generalizing well to real-world tasks without significant overfitting or underfitting.

3-2. Feature Importance

Feature importance in machine learning refers to techniques used to evaluate the relative contribution of each feature in a dataset to a model's predictions [27]. In the context of random forests, it highlights the most influential features driving the model's decisions. This concept is

applied to analyze the importance of filtered velocity gradient components and wall distance in predicting target variables related to turbulent channel flow using random forest regression, providing insights into the key factors influencing the predictions. A random forest regression with 50 estimators was trained on the dataset using the following features: Velocity gradient components $\overline{\partial u_i}/\partial x_j$ non-dimensional wall distance y^+. We have added the wall distance y^+ as an input, allowing the model to learn spatial dependences. Among the input features analyzed, the wall-normal distance in wall units y^+ exhibited the highest relative importance, with a value of 0.799234, as shown in Fig. 5. This indicates that proximity to the wall is the dominant factor influencing the prediction of SGS stresses. This observation is consistent with the physical behavior of wall-bounded turbulent flows, where turbulence characteristics vary significantly with distance from the wall, especially within the viscous sublayer and buffer region.

The feature importance analysis was also leveraged to guide feature selection, intending to reduce the input dimensionality to the FCNN. By identifying and retaining only the most influential features such as y^+ the model complexity and computational cost can be significantly reduced without sacrificing predictive accuracy. This streamlined input space facilitates more efficient training and inference, which is especially advantageous in large-scale simulations or real-time applications. It is important to note that the present study does not claim that introducing y^+alone reduces computational cost. Rather, the Random Forest feature-importance analysis consistently identified y^+as one of the most informative variables, which motivated retraining the model with a reduced set of input features. This reduction is expected to lower computational cost during *a posteriori* analyses in future investigations.

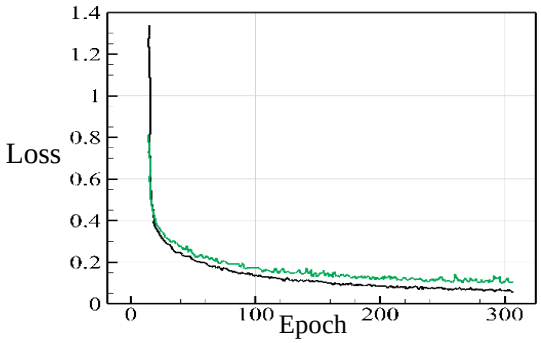


Fig. 5. Training loss (black line) and validation loss (green line) curves for the current FCNN.

Table 5. Input feature importance for the FCNN.

Feature	Importance	Feature	Importance
y^+	0.799234	$\frac{\partial \bar{v}}{\partial x}$	0.014346
$\frac{\partial \bar{u}}{\partial y}$	0.067819	$\frac{\partial \bar{v}}{\partial z}$	0.011124
$\frac{\partial \bar{u}}{\partial z}$	0.036562	$\frac{\partial \bar{w}}{\partial y}$	0.011009
$\frac{\partial \bar{u}}{\partial x}$	0.021885	$\frac{\partial \bar{v}}{\partial y}$	0.010426
$\frac{\partial \bar{w}}{\partial x}$	0.01723	$\frac{\partial \bar{w}}{\partial z}$	0.010369

3-3. SGS stresses predictions

The performance of the FCNN model is assessed based on its ability to accurately predict the unclosed SGS terms compared to their actual values.

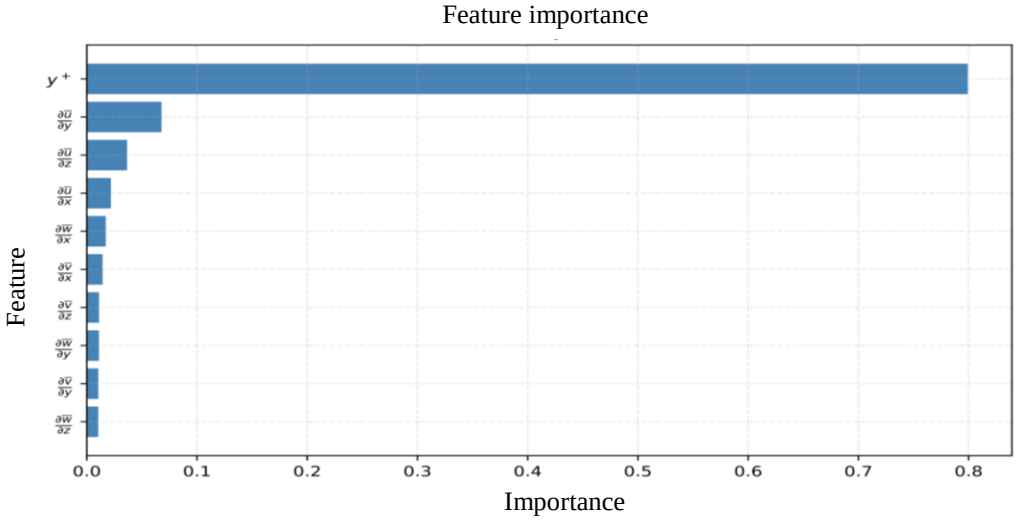


Fig. 6. Feature importances result from applying the random forest regression.

This evaluation uses the correlation coefficient as a metric, which quantifies how closely the predictions of FCNN align with the true unclosed SGS terms [28]. The correlation coefficient is defined as follows:

$$CC(Y) = \sum_{i=1}^{N}(Y_i^{pred} - \bar{Y}_i^{pred})(Y_i^{true} - \bar{Y}_i^{true}) \Bigg/ \left(\sqrt{\sum_{i=1}^{N}(Y_i^{pred} - \bar{Y}_i^{pred})^2} \sqrt{\sum_{i=1}^{N}(Y_i^{true} - \bar{Y}_i^{true})^2} \right) \quad (11)$$

Here, Y_i^{pred} represents the SGS unclosed term predicted by the FCNN, Y_i^{true} denotes the actual SGS unclosed term, and $N = N_x * N_y * N_z$ corresponds to the total number of grid points.
In this section, we discuss the reliability and robustness of these predictions, with the correlation coefficient serving as the primary measure of evaluation. The correlation coefficients were computed for the six components of SGS stress, and the corresponding values are presented in the Table 6.

The components $\tau_{11}, \tau_{12}, \tau_{22}, \tau_{33}$ exhibit high correlations, exceeding 95%, indicating that the model's predictions closely match the actual data. In contrast, the remaining components, τ_{13}, τ_{23} show slightly lower correlations of 86% and 87%, respectively, when compared to the actual SGS stress values. This suggests that the model encounters greater difficulty in predicting these stress components relative to the others. In Fig. 6, the SGS stresses from the fDNS and FCNN models are compared in the plots.

Table 6. SGS stress correlation coefficients.

Model	τ_{11}	τ_{12}	τ_{13}	τ_{22}	τ_{23}	τ_{23}
FCNN	0.969	0.967	0.860	0.961	0.871	0.955

Fig. 7 compares the $\langle\tau_{ij}\rangle/u_\tau^2$ vs y^+ in turbulent channel flow, as predicted by the FCNN model and obtained from fDNS data. Fig 7. (a) represents the $\langle\tau_{11}\rangle/u_\tau^2$. The profile rises sharply in the near-wall region, reaching a peak of about 2.2 at $y^+ = 15$ before decaying toward the channel centerline. The FCNN accurately reproduces both the peak magnitude and location, with only minor deviations near the maximum. Fig. 7(b) shows the $\langle\tau_{12}\rangle/u_\tau^2$. The distribution increases linearly within the viscous sublayer, peaks about 0.26 at $y^+ = 25$, and follows fDNS trends closely, with slight underprediction in the logarithmic region $30 < y^+ < 70$. Fig 7(c) denotes the $\langle\tau_{22}\rangle/u_\tau^2$. The stress grows steadily to 0.34 at $y^+ = 40$ and then decays toward the channel centerline. The FCNN provides an excellent match to fDNS across the entire range, accurately capturing both magnitude and peak location. Fig. 7(d) depicts the $\langle\tau_{33}\rangle/u_\tau^2$. The profile starts from zero at the wall, peaks around 0.40 at $y^+ = 50$, and decreases gradually toward the outer region. The model closely matches fDNS across all y^+, with negligible underprediction at the peak. Overall, the FCNN model demonstrates strong predictive capability for all SGS stress components except τ_{13}, τ_{23} for accurately capturing their magnitudes, peak locations, and wall-normal variations. These results highlight the model's ability to represent turbulence anisotropy and momentum transfer mechanisms in wall-bounded flows. A similar performance at higher Reynolds numbers is anticipated, which needs to be examined in further studies.

4. Conclusions

In this study, we utilized a data-driven $3D$ turbulent channel flow for modeling SGS stresses by using an FCNN model. The performance of the model was assessed through *a priori* analysis to determine its effectiveness. The training dataset for the FCNN was leveraged using DNS at the specified friction Reynolds number $Re_\tau = 180$. To obtain fDNS, a top-hat filter with a size of $n = 4$ was applied in the x and z directions due to the uniform grid. The FCNN demonstrated strong overall performance, though some challenges were observed when predicting certain complex SGS components such as τ_{13} and τ_{23} with 0.86 and 0.871 performance respectively. The $R^2 - score = 86.72\%$ performance shows that the model captures most of the variability in the data. Future work could explore the use of advanced neural network architectures to enhance prediction accuracy and better address these complexities. By analyzing feature importances using random forest regression, it is possible to reduce the number of inputs, thereby lowering the computational cost for future exploration and research. Additionally, applying and comparing different filtering methods could further refine the predictions and provide valuable insights. The proposed model, though trained exclusively at $Re_\tau = 180$, exhibits strong generalization capabilities. Its scale-aware design, based on inner scaling of inputs and targets, allows for effective adaptation to other Reynolds numbers while maintaining comparable accuracy for similar flow conditions.

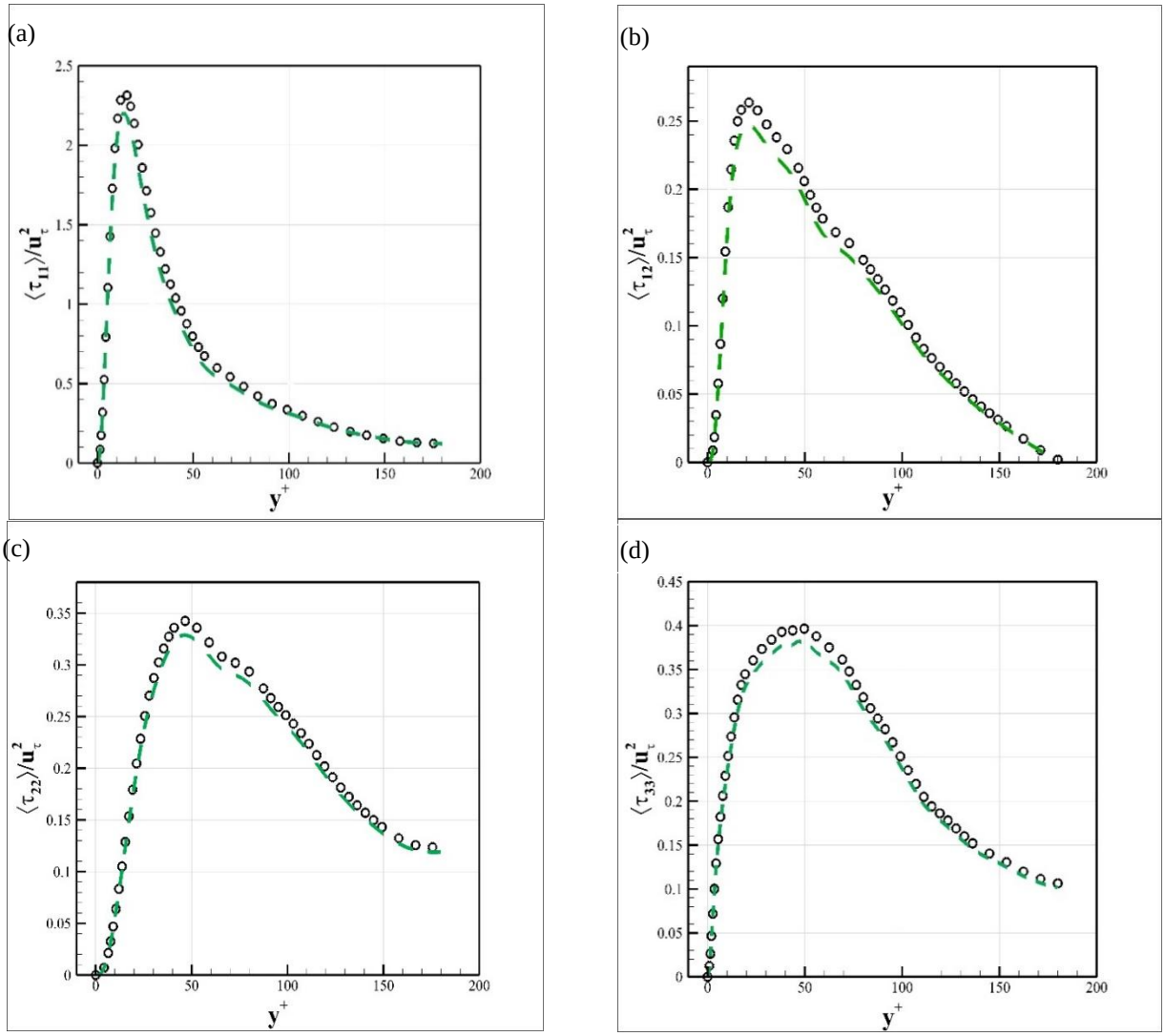


Fig. 7. Comparison of SGS stresses, $\tau_{11}, \tau_{12}, \tau_{22}, \tau_{33}$, averaged in the streamwise and spanwise directions, between fDNS(circle line) and FCNN (dashed line) results.

References

[1] Frisch, U., & Kolmogorov, A. N. (1995). *Turbulence: the legacy of AN Kolmogorov*. Cambridge university press.

[2] Germano, M., Piomelli, U., Moin, P., & Cabot, W. H. (1991). A dynamic subgrid-scale eddy viscosity model. *Physics of fluids a: Fluid dynamics*, *3*(7), 1760-1765.

[3] Rasam, A., Pouransari, Z., Vervisch, L., & Johansson, A. V. (2016). Assessment of subgrid-scale stress statistics in non-premixed turbulent wall-jet flames. *Journal of Turbulence*, *17*(5), 471-490.

[4] Pouransari, Z., & Rasam, A. (2023). Non-equilibrium extension of the explicit algebraic subgrid-scale stress model with application to turbulent channel flow at low Reynolds numbers. *Physics of Fluids*, *35*(3).

[5] Smagorinsky, J. (1963). General circulation experiments with the primitive equations: I. The basic experiment. *Monthly weather review*, *91*(3), 99-164.

[6] Deardorff, J. W. (1970). A numerical study of three-dimensional turbulent channel flow at large Reynolds numbers. *Journal of Fluid Mechanics*, *41*(2), 453-480.

[7] Ghosal, S., Lund, T. S., Moin, P., & Akselvoll, K. (1995). A dynamic localization model for large-eddy simulation of turbulent flows. *Journal of fluid mechanics*, *286*, 229-255.

[8] Meneveau, C., Lund, T. S., & Cabot, W. H. (1996). A Lagrangian dynamic subgrid-scale model of turbulence. *Journal of fluid mechanics*, *319*, 353-385.

[9] Park, N., & Mahesh, K. (2009). Reduction of the Germano-identity error in the dynamic Smagorinsky model. *Physics of Fluids*, *21*(6).

[10] Brunton, S. L., Noack, B. R., & Koumoutsakos, P. (2020). Machine learning for fluid mechanics. *Annual review of fluid mechanics*, *52*(1), 477-508.

[11] Milano, M., & Koumoutsakos, P. (2002). Neural network modeling for near wall turbulent flow. *Journal of Computational Physics*, *182*(1), 1-26.

[12] Erichson, N. B., Mathelin, L., Yao, Z., Brunton, S. L., Mahoney, M. W., & Kutz, J. N. (2020).

Shallow neural networks for fluid flow reconstruction with limited sensors. *Proceedings of the Royal Society A*, *476*(2238), 20200097.
[13] Fukami, K., Fukagata, K., & Taira, K. (2019). Super-resolution reconstruction of turbulent flows with machine learning. *Journal of Fluid Mechanics*, *870*, 106-120.
[14] Maulik, R., & San, O. (2017). A neural network approach for the blind deconvolution of turbulent flows. *Journal of Fluid Mechanics*, *831*, 151-181.
[15] Maulik, R., San, O., Rasheed, A., & Vedula, P. (2019). Subgrid modelling for two-dimensional turbulence using neural networks. *Journal of Fluid Mechanics*, *858*, 122-144.
[16] Zhou, Z., He, G., Wang, S., & Jin, G. (2019). Subgrid-scale model for large-eddy simulation of isotropic turbulent flows using an artificial neural network. *Computers & Fluids*, *195*, 104319.
[17] Sarghini, F., de Felice, G., & Santini, S. (2003). Neural networks based subgrid scale modeling in large eddy simulations. *Computers & fluids*, *32*(1), 97-108.
[18] Gamahara, M., & Hattori, Y. (2017). Searching for turbulence models by artificial neural network. *Physical Review Fluids*, *2*(5), 054604.
[19] Park, J., & Choi, H. (2021). Toward neural-network-based large eddy simulation: Application to turbulent channel flow. *Journal of Fluid Mechanics*, *914*, A16.
[20] Pouransari, Z. (2023). Passive scalar small-scale anisotropy and mixing characteristics in magnetohydrodynamic turbulent channel flow. *Physics of Fluids*, *35*(10).
[21] Cant, S. (2001). SB Pope, Turbulent Flows, Cambridge University Press, Cambridge, UK, 2000, 771 pp. *Combustion and Flame*, *125*(4), 1361-1362.
[22] Sagaut, P., & Grohens, R. (1999). Discrete filters for large eddy simulation. *International Journal for Numerical Methods in Fluids*, *31*(8), 1195-1220.
[23] Majd, B. K., & Pouransari, Z. (2024). Reynolds number effect on the Lagrangian evolution of hairpin vortices in turbulent channel flow. *Journal of Turbulence*, *25*(7), 207-224.
[24] Sanhueza, R. D., Smit, S. H., Peeters, J. W., & Pecnik, R. (2023). Machine learning for RANS turbulence modeling of variable property flows. *Computers & Fluids*, *255*, 105835.
[25] Chicco, D., Warrens, M. J., & Jurman, G. (2021). The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *Peerj computer science*, *7*, e623.
[26] Montesinos López, O. A., Montesinos López, A., & Crossa, J. (2022). Overfitting, model tuning, and evaluation of prediction performance. In *Multivariate statistical machine learning methods for genomic prediction* (pp. 109-139). Cham: Springer International Publishing.
[27] Mandler, H., & Weigand, B. (2023). Feature importance in neural networks as a means of interpretation for data-driven turbulence models. *Computers & Fluids*, *265*, 105993.
[28] Rasam, A., & Najarian, A. (2024). Modeling the subgrid-scale kinetic energy in a turbulent channel flow using artificial neural network. *AUT Journal of Mechanical Engineering*, *8*(1), 19-30.

Biography

Mohammadreza Azarshab is a Master's student in Aerospace Engineering at Iran University of Science and Technology. His academic focus lies in the field of aerodynamics, with a particular interest in turbulence modeling and the application of artificial intelligence in fluid dynamics.

Dr. Zeinab Pouransari is a faculty member in the Department of Mechanical Engineering at Iran University of Science and Technology. She received her Ph.D. in Mechanical Engineering, with a specialization in turbulence, from the Royal Institute of Technology (KTH) in Stockholm, Sweden. She has performed research at several institutions, including the Technical University of Eindhoven in the Netherlands, Chalmers University of Technology in Gothenburg, Sweden, and INSA Rouen in France.