Modeling the subgrid-scale kinetic energy in a turbulent channel flow using artificial neural network

Amin Rasam ⓘ *, Ali Najarian

Faculty of Mechanical and Energy Engineering, Shahid Beheshti University, Tehran, Iran

ABSTRACT: Modeling the subgrid-scale energy, K_{SGS}, has important applications in large-eddy simulation, including the lattice Boltzmann method and formulation of advanced subgrid-scale models. In this study, a deep neural network is specifically developed to predict K_{SGS} for large-eddy simulation of turbulent channel flow. To produce the training data for the neural network, a direct numerical simulation of turbulent channel flow at the friction Reynolds number $\mathrm{Re}_\tau = 381$ is performed using an existing highly accurate pseudo-spectral method. The impact of the neural network configuration on its predictions is studied by examining the mean, probability density function and K_{SGS} skewness of K_{SGS}. Moreover, the correlation with the filtered data, its relative and root mean square error are also examined using a priori analysis. Appreciable improvements in the predictions were observed with increasing the number of neurons in the hidden layer, up to 64. Increasing the number of hidden layers to two and three showed small improvements in the predictions. The performance of the neural network is also compared with a dynamic subgrid-scale model. The comparison reveals that the neural network predictions reach correlation coefficients higher than 90% with the filtered direct numerical simulation data, whereas the dynamic subgrid-scale model predictions only reach up to about 50%. Also, a closer agreement was observed with the filtered data for the neural network predictions of K_{SGS}, compared with the dynamic subgrid-scale model.

1- Introduction

Turbulent flows are abundant in industrial and natural flows. They often occur at high Reynolds numbers and comprise a wide range of scales. The largest scales contain energy with length scales proportional to the flow scale, ℓ, and the smallest scales are those of the Kolmogorov, with size η, which dissipate kinetic energy to heat. The ratio ℓ / η is in the order of $\mathrm{Re}^{-3/4}$, where Re is the flow Reynolds number [1]. This shows that the size of the small scales reduces as Re increases. Hence, scale-resolving numerical simulations of turbulent flows require a substantial number of grid points at high Reynolds numbers. Numerical simulations of turbulent flows can be categorized by the extent of turbulence modeling involved. A direct numerical simulation (DNS) resolves all flow scales even as small as the Kolmogorov ones. Hence, it often involves a large number of grid points and is not computationally feasible for almost all industrial flow cases [2]. In contrast, modeling is performed for all turbulence scales in Reynolds-averaged Navier-Stokes (RANS) simulations. Hence, they require substantially fewer grid points and are computationally feasible for numerical simulation of turbulent industrial flows. But they do not

provide instantaneous details of the flow and their accuracy is limited by the performance of the RANS model. Large-eddy simulation (LES) is an intermediary approach, where large scales are resolved and the effect of the small unresolved scales, called subgrid scales, is modeled via a subgrid-scale (SGS) model. Reducing the grid requirements, compared with the DNS, is obtained in LES.

By filtering the Navier-Stokes equations LES equations are obtained, which for incompressible flows are expressed in non-dimensional form as [3]

$$\frac{\partial \tilde{u}_i}{\partial t} + \tilde{u}_j \frac{\partial \tilde{u}_i}{\partial x_j} = -\frac{\partial \tilde{p}}{\partial x_i} + \frac{1}{\mathrm{Re}} \nabla^2 \tilde{u}_i - \frac{\partial \tau_{ij}}{\partial x_j} \qquad (1)$$

Here, summation over the repeated indices is implied, $\tilde{u}_i$ is the filtered velocity vector, x_j is the coordinate system, t denotes time, p is the pressure, and $\tau_{ij} = \widetilde{u_i u_j} - \tilde{u}_i \tilde{u}_j$ is the SGS stress tensor, which needs to be modeled [3]. The SGS kinetic energy, K_{SGS}, is half the trace of the τ_{ij}, defined as

*Corresponding author's email: a_rasam@sbu.ac.ir

$$K_{SGS} = \frac{1}{2}\left(\widetilde{u_i u_i} - \tilde{u}_i \tilde{u}_i\right) \tag{2}$$

It is an unknown scalar quantity in LES, which needs to be modeled. When only the deviatoric part of τ_{ij} is modeled, K_{SGS} is lumped in the pressure term in Eq. (1). In other cases, where the whole τ_{ij} is modeled, such as the explicit algebraic SGS model, K_{SGS} needs to be modeled [4]. Another application is LES using the lattice Boltzmann method, where K_{SGS} cannot be added to the pressure term and needs to be modeled [5]. Machine learning (ML) techniques can be used to model K_{SGS}.

In fluid mechanics, ML capabilities have been explored to find data features and correlations [6]. Numerous studies on ML applications for turbulence modeling have also been conducted, including RANS modeling [7] and SGS models for LES applications [8]. In LES, an artificial neural network (ANN) has been previously used for the prediction of τ_{ij} from the resolved velocity-gradient tensor, $\partial \tilde{u}_i / \partial x_j$. Data of two commonly used test cases for training the ANN, are from DNS of homogeneous isotropic turbulence (HIT) and fully developed turbulent channel flow (TCF). Zhou *et al.* used DNS data of HIT to train an ANN to predict the τ_{ij} tensor. They evaluated the influence of filter size, as an input parameter, on the ANN predictions [9]. Frezat *et al.* included physical invariance in their ANN for the prediction of the τ_{ij} tensor for the HIT [9].

Gamahara et al. [8] Train an ANN with filtered DNS data to predict τ_{ij}. They employed the TCF and examined the sensitivity of the correlations of ANN predictions with filtered DNS data to the choice of input parameters. Wang *et al.* carried out a similar analysis where input parameters were the filtered velocity as well as its derivatives [10]. The test case employed by Wang *et al.* was the homogeneous isotropic turbulence. Qingjia *et al.* used a DNN for the prediction of τ_{ij} for compressible TCF at $Re = 3000$ and Mach number $Ma = 0.3$. The input variables for the ANN were $\partial \tilde{u}_i / \partial x_j$, and the ANN predictions reached correlations of up to 0.91 and a relative error of less than 0.43 [11]. Xu *et al.* also used an ANN to model τ_{ij} using a nonlinear algebraic SGS model. Their model predictions had a higher correlation coefficient with filtered DNS data than the dynamic Smagorinsky model in TCF [12].

In the current study, a fully connected ANN is developed to model K_{SGS}, a quantity that has not been directly computed with machine learning algorithms in previous research literature. The motivation for modeling this quantity is LES using the lattice Boltzmann method, where K_{SGS} should be modeled and cannot appear as a modified pressure. The architecture of the employed ANN is similar to the one used by Rasam and Shirazi (2024) [13], where the SGS scalar flux was modeled. The impacts that the neuron number and hidden layers have on the ANN performance are examined using different statistics. Specifically, a good prediction of the probability density function of k_{SGS} is introduced as

an indicator for choosing the optimum number of the ANN elements. This approach has not been used in earlier works for the prediction of other turbulent quantities. Also, to the best of our knowledge, the skewness of K_{SGS} used in the current study, which is a higher-order moment, has not been used in previous assessments of ANN predictions. The training data is obtained by performing a DNS of TCF and subsequent filtering of the data. The performed DNS is at a higher Re than the one performed by Rasam and Shirazi (2024) [13], where the SGS scalar, which is a totally different turbulence quantity, is modeled via the ANN. The ANN predictions are compared with a commonly used K_{SGS} model, which employs a dynamic procedure, for the case of fully developed TCF, which has not been reported in the earlier literature.

The proceeding sections of the paper are organized as follows. The numerical method, grid size, and simulation are explained in section 2. In section 3, filtering operation and computation of K_{SGS} from the DNS data is outlined. The employed dynamic K_{SGS} model is explained in section 4 and the ANN structure is explained in section 5. The presentation of the results is given in section 6. The conclusions are presented in section 7.

2- Numerical Method, Grid size and Simulation

A DNS of fully developed TCF is performed with the pseudo-spectral numerical method. LES quantities are obtained by filtering the DNS data. The code, called SIMSON [14], has a similar structure to the one employed in reference [15]. It has been utilized for DNS and LES of fully developed TCF in earlier works [16-18]. The code solves the following transport equations, in non-dimensional form, for the wall-normal component of velocity (v) and also vorticity (ω) are solved.

$$\frac{\partial}{\partial t}\nabla^2 v = h_v + \frac{1}{Re}\nabla^4 v, \qquad \frac{\partial \omega}{\partial t} = h_\omega + \frac{1}{Re}\nabla^2 \omega \tag{3}$$

Here, h_v and h_ω are given as

$$h_v = \left(\frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_3^2}\right)H_2 - \frac{\partial}{\partial y}\left(\frac{\partial H_1}{\partial x_1} + \frac{\partial H_3}{\partial x_3}\right),$$

$$h_\omega = \frac{\partial H_1}{\partial x_3} - \frac{\partial H_3}{\partial x_1}, \quad H_i = u_j \frac{\partial u_i}{\partial x_j}, \quad i = 1, 2, 3. \tag{4}$$

The $Re = u_b \delta / v$ is based on the bulk velocity, u_b, kinematic viscosity, v, and channel half height, δ. The numerical domain is given in Fig. 1.

The employed spatial discretization of Eqs. (3) and (4) in the x and z directions use Fourier modes, whereas Chebyshev polynomials are used in the y direction. The 3/2 rule [19] is used for the removal of aliasing errors. The integration in time for the nonlinear terms is via a third-order Runge-Kutta scheme, while the nonlinear terms use the
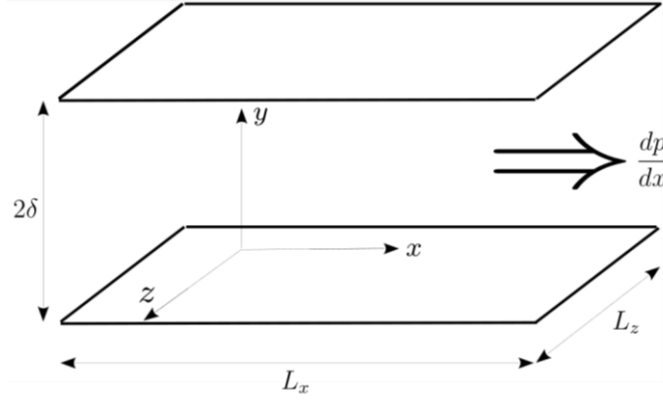
Fig. 1. Schematic of the computational domain with a pressure gradient, dp/dx.

Crank-Nicolson method. Further information about the code can be found in reference [14]. The computational time to perform the DNS was about 35 seconds (on average) for each time step on a core i7-13700 intel CPU using 8 cores. The total computational time needed to reach a fully developed turbulent flow from a laminar solution and getting statistical convergence was about 20 days.

The numerical domain extents in the x, y and z directions are $L_x = 2\pi\delta$, $L_y = 2\delta$ and $L_z = \pi\delta$, respectively, which are shown in Fig. 1. The number of grid points is $N_x = 256, N_y = 257$ and $N_z = 256$, which lead to grid spacings in wall units as:

$$\Delta x^+ = \frac{u_\tau \Delta x}{\upsilon} = 9.35,$$

$$\Delta y^+ = \frac{u_\tau \Delta y}{\upsilon} = 0.029 \sim 4.67, \qquad (5)$$

$$\Delta z^+ = \frac{u_\tau \Delta z}{\upsilon} = 4.68.$$

Here, Δx, Δy, and Δz represent the grid spacings in the x, y, and z directions, respectively. The friction velocity is defined using the wall shear stress τ_{wall} and fluid density ρ as $u_\tau = \sqrt{\tau_{wall}/\rho}$. The channel has periodic boundary conditions in the x and z directions and the fluid satisfies the no-slip condition at the walls. Simulations are carried out with a constant bulk Reynolds number, $Re_b = 6666$, resulting in the friction Reynolds number $Re_\tau = u_\tau \delta / \upsilon \approx 381$.

In the proceeding, $\langle . \rangle$ means statistical averaging in the x and z directions of homogeneity. Turbulent statistics followed by a $^+$ sign, are non-dimensionalized with u_τ and υ, and are expressed in wall units. Ten velocity fields with 168,427,520 data points are used to compute flow statistics, which gave reasonably converged statistics.

A comparison of the mean streamwise velocity, $\langle u \rangle^+$,

and root-mean-square (RMS) of the velocity components in wall units in the streamwise, u_{rms}^+, wall-normal, v_{rms}^+, and spanwise, w_{rms}^+, directions, between the current and reference DNS predictions [20] at a similar Re_τ is illustrated in Fig. 2. The two DNS predictions are very close. The minor differences in the RMS predictions at the vicinity of the channel center are expected to disappear if more data is used to compute the statistics.

3- Filtering Operation and SGS Kinetic Energy

To compute k_{SGS} from the DNS data, using Eq. (2), filtering needs to be performed. Filtering in LES for an arbitrary function f is carried out via the convolution [3]

$$\tilde{f} = \int_D f(p) G_{\tilde{\Delta}}(x-p) dp \qquad (6)$$

Here, integration is over a domain D using the filter kernel of the top-hat filter [1], $G_{\tilde{\Delta}}$, with the filter size $\tilde{\Delta} = 4\Delta$.

To compute k_{SGS}, the velocity, u_i, and the velocity product, $u_i u_i$ are filtered to obtain $\tilde{u}_i$ and $\widetilde{u_i u_i}$, respectively, and subsequently, k_{SGS} is computed using Eq. (2). Filtering the discrete function f_i in one dimension using the top-hat filter results in the following discrete formula

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

where, $n = \hat{\Delta}/\tilde{\Delta} = 4$ is the ratio of filter to grid size. Due to the non-uniform grid distribution in y, filtering is performed only in the x and z directions, which is a common practice in LES. The non-uniform grid distribution prohibits filtering in y direction since it violates the continuity equation [21-24]. Hence, computation of k_{SGS} involves the application of Eq. (7) to both u_i and $u_i u_i$.
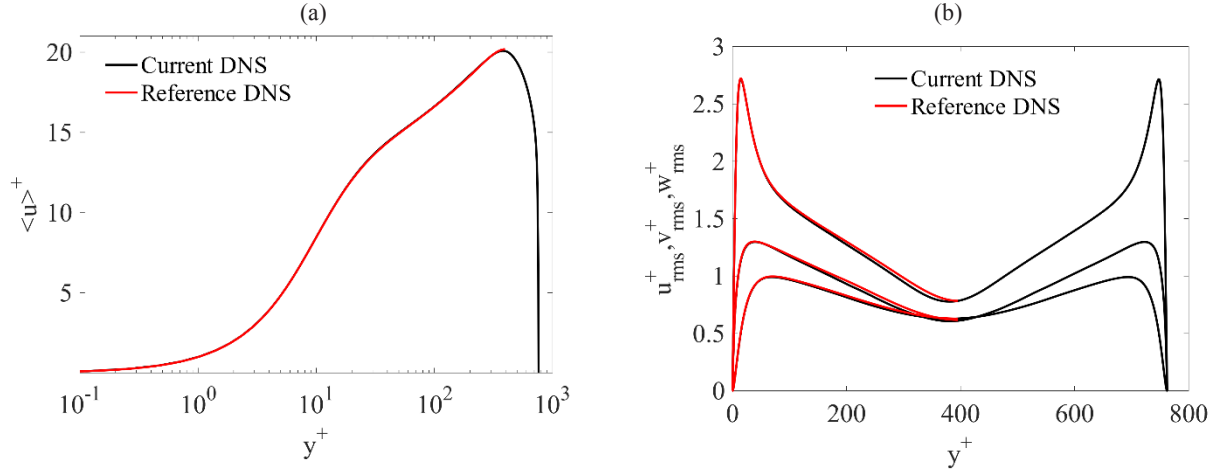
Fig. 2. A comparison of the mean streamwise velocity, $\langle u \rangle^+$ (a) and RMS of the streamwise, wall-normal, and spanwise velocity, u^+_{rms}, v^+_{rms}, w^+_{rms} and , respectively, (b) between the reference and current DNS [20] predictions.

4- The Dynamic Subgrid-scale Kinetic Energy Model

The commonly employed dynamic SGS model (DSM) [4] for the k_{SGS} uses $\widetilde{\Delta}$, and the inverse time scale, $\left|\widetilde{S_{ij}}\right|$, to model k_{SGS} as

$$k_{SGS} = c\widetilde{\Delta}^2 \left|\widetilde{S_{ij}}\right|^2, \quad \left|\widetilde{S_{ij}}\right| = \sqrt{2\widetilde{S_{ij}}\widetilde{S_{ij}}} \tag{8}$$

where $\widetilde{S_{ij}}$ is the resolved strain-rate tensor given as

$$\widetilde{S_{ij}} = \frac{1}{2}\left(\frac{\partial \widetilde{u}_i}{\partial x_j} + \frac{\partial \widetilde{u}_j}{\partial x_i}\right). \tag{9}$$

The c coefficient in Eq.(8) is obtained via the so-called dynamic procedure [25], outlined in the following. First, a second filter denoted by $\widehat{.}$, called the test filter, is introduced. It has a filter width, $\widehat{\Delta}$ equal to two times $\widetilde{\Delta}$. Second, the turbulent kinetic energy due to scales larger than $\widetilde{\Delta}$ and smaller than $\widehat{\Delta}$, denoted by L, is computed as

$$L = \frac{1}{2}\left(\widehat{\widetilde{u_i\widetilde{u}_i}} - \widehat{\widetilde{u}}_i\widehat{\widetilde{u}}_i\right) \tag{10}$$

The same quantity can be modeled using Eq. (8), denoted by cM, as

$$L = c\widehat{\Delta}^2 \left|\widehat{\widetilde{S}}_{ij}\right|^2 - \widehat{c\widetilde{\Delta}^2 \left|\widetilde{S}_{ij}\right|^2} = \left(\widehat{\Delta}^2 \left|\widehat{\widetilde{S}}_{ij}\right|^2 - \widehat{\widetilde{\Delta}^2 \left|\widetilde{S}_{ij}\right|^2}\right) = cM \tag{11}$$

Given that L is a known quantity, Eq. (11) can be solved to compute the model coefficient c as

$$c = \frac{\langle LM \rangle}{\langle M^2 \rangle} \tag{12}$$

where averaging of L and M is performed in the homogeneous directions to reduce the intermittency of the coefficient. Moreover, the numerator and denominator of Eq. (12) are multiplied by M to eliminate the possibility of zero values in the denominator during the computations.

5- The ANN-based Subgrid-scale Modeling

An ANN is developed to predict k_{SGS}. The structure of the current ANN is shown in Fig. 3. The ANN is fully connected to form a deep neural network (DNN) with input, output, and hidden layers. The Keras API and the Tensor Flow are employed [26].

The Rectified Linear Unit (ReLU) activation function [27] is used for the hidden layers. The output layer has a linear activation function. The training of the DNN uses the backpropagation and Adam optimization algorithms [28], where weights and biases are iteratively rectified in the learning process. The following mean absolute error (MAE) expression is used as the loss function:
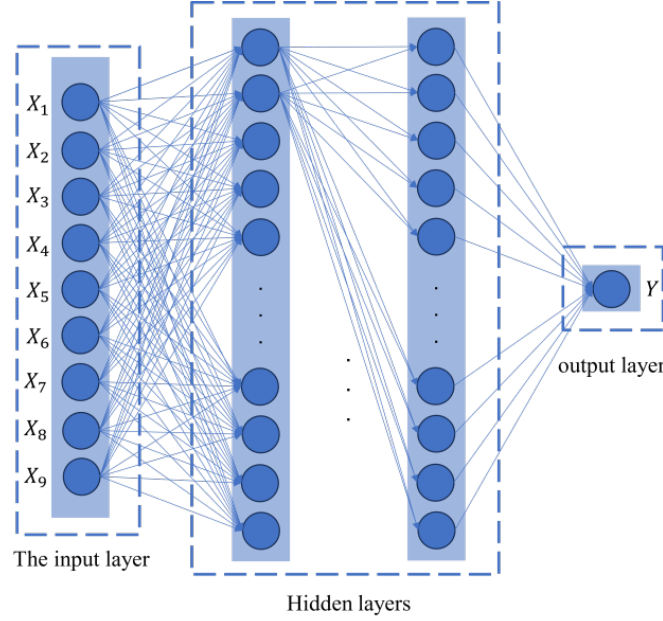
Fig. 3. Schematic of the employed DNN showing the input, output, and hidden layers

$$MAE = \left(\frac{1}{m}\right)\sum_{i=1}^{m}\left|\left(Y_i - \widehat{Y_i}\right)\right|, \qquad (13)$$

where m denotes the number of data points, Y_i and $\widehat{Y_i}$ are the output and exact output values, respectively. The training of the DNN can be performed using either filtered DNS [29] or LES [30] data. Here, filtered DNS data is used to train the DNN for the ten cases considered, resulting in specific sets of weights and biases for each case. The DNN configuration for each case is specified in Table 1.

The filtered dataset is randomly split to obtain training and testing sets, consisting of 80% and 20% of the dataset, respectively. The training dataset is further normalized such that it has a zero mean and unit variance, which improves the convergence rate for the training [9]. The computational time needed to perform the training was about 27 seconds (on average) for each epoch in case of DNN-6 on a core i7-13700 intel CPU and the training was performed for 500 epochs.

The DNN has nine inputs, X_{1-9}, which are the filtered velocity-gradient tensor components, $\partial\widetilde{u}_i / \partial x_j$, given as

$$X_{1-9} = \left\{\frac{\partial\widetilde{u}_1}{\partial x_1},\frac{\partial\widetilde{u}_1}{\partial x_2},\frac{\partial\widetilde{u}_1}{\partial x_3},\frac{\partial\widetilde{u}_2}{\partial x_1},\frac{\partial\widetilde{u}_2}{\partial x_2},\frac{\partial\widetilde{u}_2}{\partial x_3},\frac{\partial\widetilde{u}_3}{\partial x_1},\frac{\partial\widetilde{u}_3}{\partial x_2},\frac{\partial\widetilde{u}_3}{\partial x_3},\right\} \quad (14)$$

This selection of inputs is based on the DSM parameters, which have also been used for modeling of τ_{ij} in previous studies, see e.g. [21].

A filter size ratio $n = \widetilde{\Delta} / \Delta$, the grid size ratio of LES to DNS, of 4 is chosen for this investigation. The grid scale is computed as $\Delta = \sqrt[3]{\Delta x\,\Delta y\,\Delta z}$. Based on Eq. (5), the LES grid sizes in wall units are $\Delta x^+ = 37.4$ and $\Delta z^+ = 18.72$. It has to be pointed out that the DNN performance depends on n used for the training data. A low n is not suitable for LES applications. On the other hand, the correlation coefficient, CC (see Eq.15) decreases with increasing n. The value of n chosen for the current analysis is based on the value suggested in ref. [13]. The value of N used for training of the DNN also has an impact on the DNN performance. The filter type can also affect the DNN predictions. The employed choice of filter is based on the previous investigation of the author [13].

6- Results and Discussions

A flow chart of the computational procedure is given in Fig. 4. In this section, the performance of the DNN is assessed with respect to the filtered DNS data and compared with the DSM predictions. It has to be pointed out that the predictions

To analyze the precision of DNN-predicted K_{SGS}, hereafter denoted as K_{SGS}^{DNN}, the correlation coefficient CC between K_{SGS}^{DNN} and K_{SGS}, computed from the filtered DNS data, denoted hereafter as K_{SGS}^{DNS} is computed using their fluctuations by the following expression [10]

Table 1. Configuration of the DNN, the correlation coefficient, CC, between and N is the number of data points used for training of the DNN, RMS is the root-mean-square error of the DNN prediction with respect to the filtered DNS data, is the filter size ratio, is the number of training data points, and NL is the number of hidden layers.

Cases	NL	CC %	RMS %	ERR %	Activation function	Number of neurons	n	N
DNN-1		90.81	50.03	8.02		8		
DNN-2		95.25	35.90	6.34		16		
DNN-3		96.30	30.30	4.81		32		
DNN-4	1	96.60	30.19	4.74		64		
DNN-5		96.67	29.65	4.32		96		16,842,752
DNN-6		96.73	28.55	4.30	ReLu	128	4	
DNN-7	2	96.75	28.30	3.49		128		
DNN-8	3	96.79	28.25	3.36		128		
DNN-9	2	96.25	32.70	4.53		128		1,020,033
DNN-10	2	96.39	31.50	4.10		128		4,080,132

$$CC = \frac{\left\langle \left(K_{SGS}^{DNS} - \left\langle K_{SGS}^{DNS}\right\rangle\right)\left(K_{SGS}^{DNN} - \left\langle K_{SGS}^{DNN}\right\rangle\right)\right\rangle}{\left[\left\langle\left(K_{SGS}^{DNS} - \left\langle K_{SGS}^{DNS}\right\rangle\right)^2\right\rangle\right]^{0.5}\left[\left\langle\left(K_{SGS}^{DNN} - \left\langle K_{SGS}^{DNN}\right\rangle\right)^2\right\rangle\right]^{0.5}} \qquad (15)$$

CC varies between 0 and 1, as with any other correlation coefficient. A higher CC indicates a K_{SGS}^{DNN} that more closely follows the K_{SGS}^{DNS}. The predicted CC values for all cases are given in percent values in Table 1. It should be pointed out that the CC values depend on y and that the values reported in Table 1 are integrated values over the y direction between the two walls.

6- 1- Effect of the Number of Neurons on the DNN Predictions

The impact that n has on the DNN predictions is studied using cases DNN-1, DNN-2, DNN-3, DNN-4, DNN-5, and DNN-6, where 8, 16, 32, 64, 96, and 128 neurons, respectively, have been employed in one hidden layer. In all these cases, only one hidden layer is considered in the ANN.

The predicted CC values for cases DNN-1 to DNN-6 vary between 90.81% in the DNN-1 case up to 96.73% in the DNN-6 case. Two types of errors are also used for the evaluation

of the DNN predictions, namely the relative absolute error (ERR), or the relative absolute difference between K_{SGS}^{DNN} and K_{SGS}^{DNS}, and the root-mean-square (RMS) error, defined as the relative squared difference between K_{SGS}^{DNN} and K_{SGS}^{DNS}. Both errors are averaged in the x and z directions and are integrated over the y direction and are given in percent values in Table 1. The ERR predictions improve with increasing the number of neurons, nn, from 8.02% in case DNN-1 to 4.30% in case DNN-6. At the same time, the RMS predictions improve with increasing nn from 50.03% in case DNN-1 to 28.55% in case DNN-6.

The DNN predictions of the mean SGS kinetic energy in wall units, K_{SGS}^{+}, are compared with the filtered DNS data in Fig. 5(a-e). It is observed that increasing nn from 8 in the DNN-1 case to 128 in the DNN-6 case improves the DNN prediction of the K_{SGS}^{+}, especially at the vicinity of the wall, i.e., $y/\delta < 0.25$, whereas the DNN-4 and DNN-5 predictions are similar. The results for the DNN-6 are also identical to the DNN-5 and are not shown here.

The DNN predictions of the probability density function (PDF) of K_{SGS}^{+} are also given in Fig. 5. (f-j) for cases DNN-1 to DNN-5. The PDF only has the positive tail, since
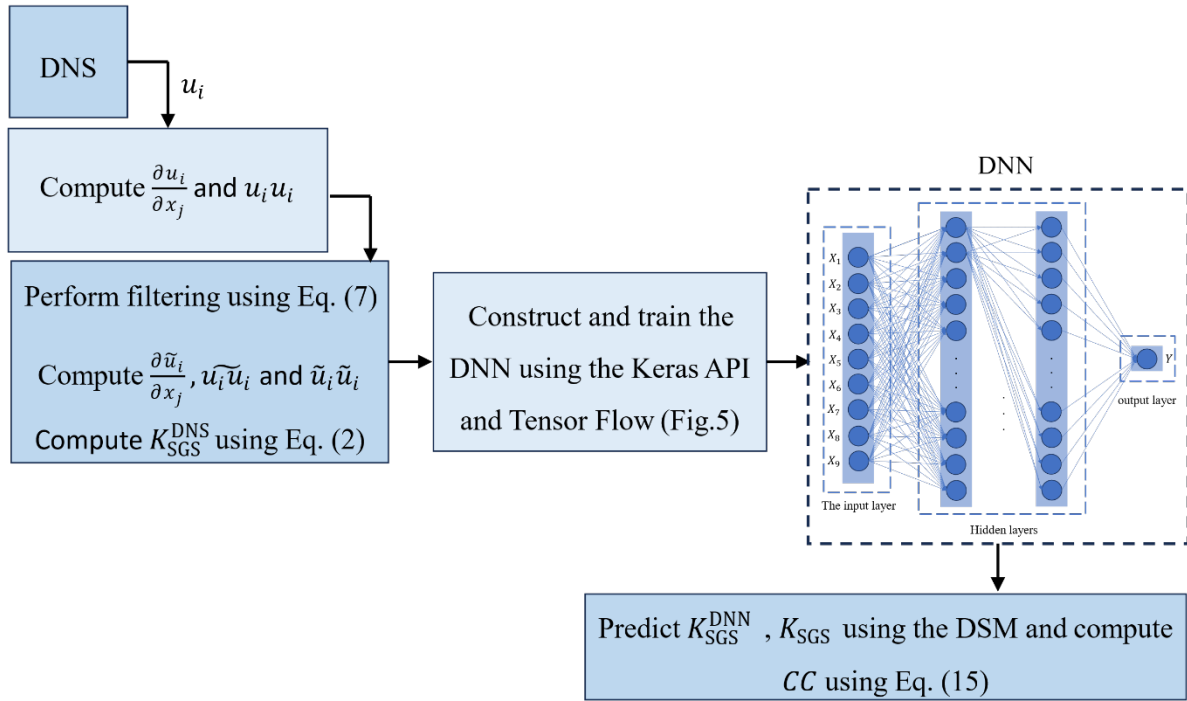
Fig. 4. Flow chart of the computational procedure

K_{SGS}^+ is always positive. It can be observed that the DNN predictions of the PDF follow the filtered DNS predictions $K_{SGS}^+ < 2$ in all cases. It is interesting to observe that a number of neurons in the hidden layers lower than 64, see Fig. 5 (f-h), is not sufficient for accurate prediction of the PDF for higher values of K_{SGS}^+, due to the nonlinear nature of turbulence. Improvements in the predictions are observed up to 64 neurons in the case DNN-4, while DNN-4, DNN-5, and DNN-6 predictions of the PDF are similar. It is worth noting that the discontinuous behavior of the tail of the PDF plot is due to the fact that no data has fallen in that range of the PDF.

To further study the impact of the *nn* on the DNN predictions, the skewness of the predicted K_{SGS} is computed. The skewness, an indicator of the PDF asymmetry, is a higher-order statistic, defined as

$$S_K = \frac{\left\langle \left(K_{SGS} - \langle K_{SGS} \rangle \right)^3 \right\rangle}{\left(K_{SGS}^{RMS} \right)^3}, \tag{16}$$

where, the RMS of K_{SGS} is used in the denominator. The S_K is presented in Fig. 5 (k-o). The PDF was given previously at a particular wall distance, whereas the skewness is given as a function of y. The skewness has three peaks, one close

to each wall and one close to the center of the channel. It is observed that S_K is under-predicted across the entire channel height in the DNN-1 case.

Comparing DNN-1 to DNN-6 predictions, it can be observed that S_K prediction is improved by increasing *nn* from 8 in case DNN-1 up to 32 in case DNN-3. Further increasing *nn* does not significantly improve the S_K predictions.

6- 2- Effect of the Number of Hidden Layers and Training Data on the DNN Predictions

Increasing the number of hidden layers, NL, in cases DNN-7 and DNN-8, decreases the ERR from 4.30% in case DNN-6 with one hidden layer to 3.49% in case DNN-7 with two hidden layers and 3.36% with three hidden layers in case DNN-8, see Table 1. A similar behavior is observed for the RMS, where it decreases from 28.55% in the DNN-6 case to 28.30% in the DNN-7 case and 28.25% in the DNN-8 case. The correlation coefficient, *CC* increases from 96.73% in case DNN-6 to 96.75% and 96.79% with increasing NL to two and three in cases DNN-7 and DNN-8, respectively.

Fig. 7(a-c) shows a comparison of the K_{SGS}^+, predicted by the DNN with different NL in cases DNN-6, DNN-7, and DNN-8. One can observe small improvements in the K_{SGS}^+, especially the peak value, with increasing NL to two in case DNN-7. However, the predictions in cases DNN-7 and DNN-
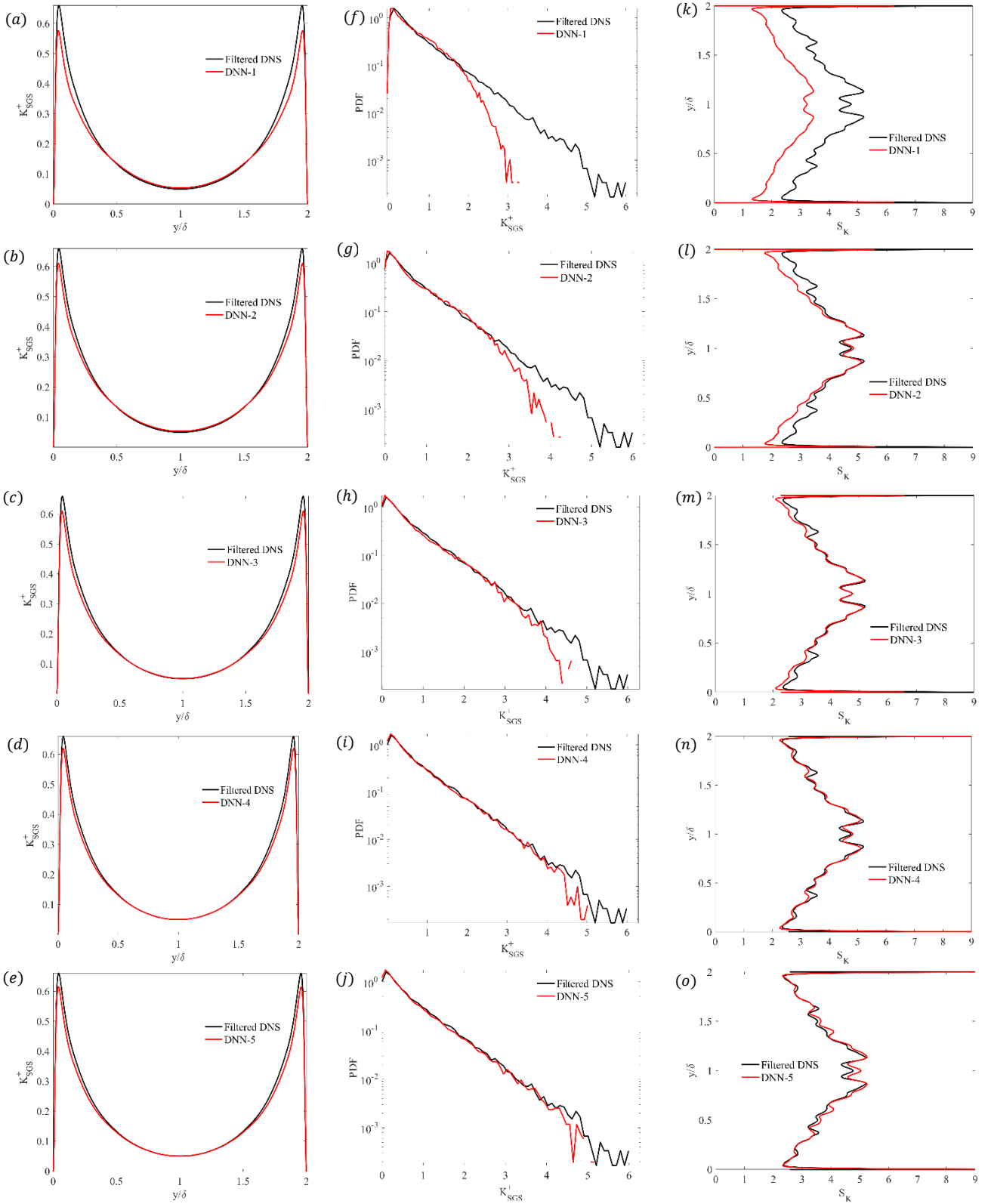
Fig. 5. Comparison between the DNN predictions and filtered DNS data of the mean SGS kinetic energy in wall units, K_{SGS}^+, (a-e), the PDF of K_{SGS}^+ (f-i), and its skewness, S_K (k-o).
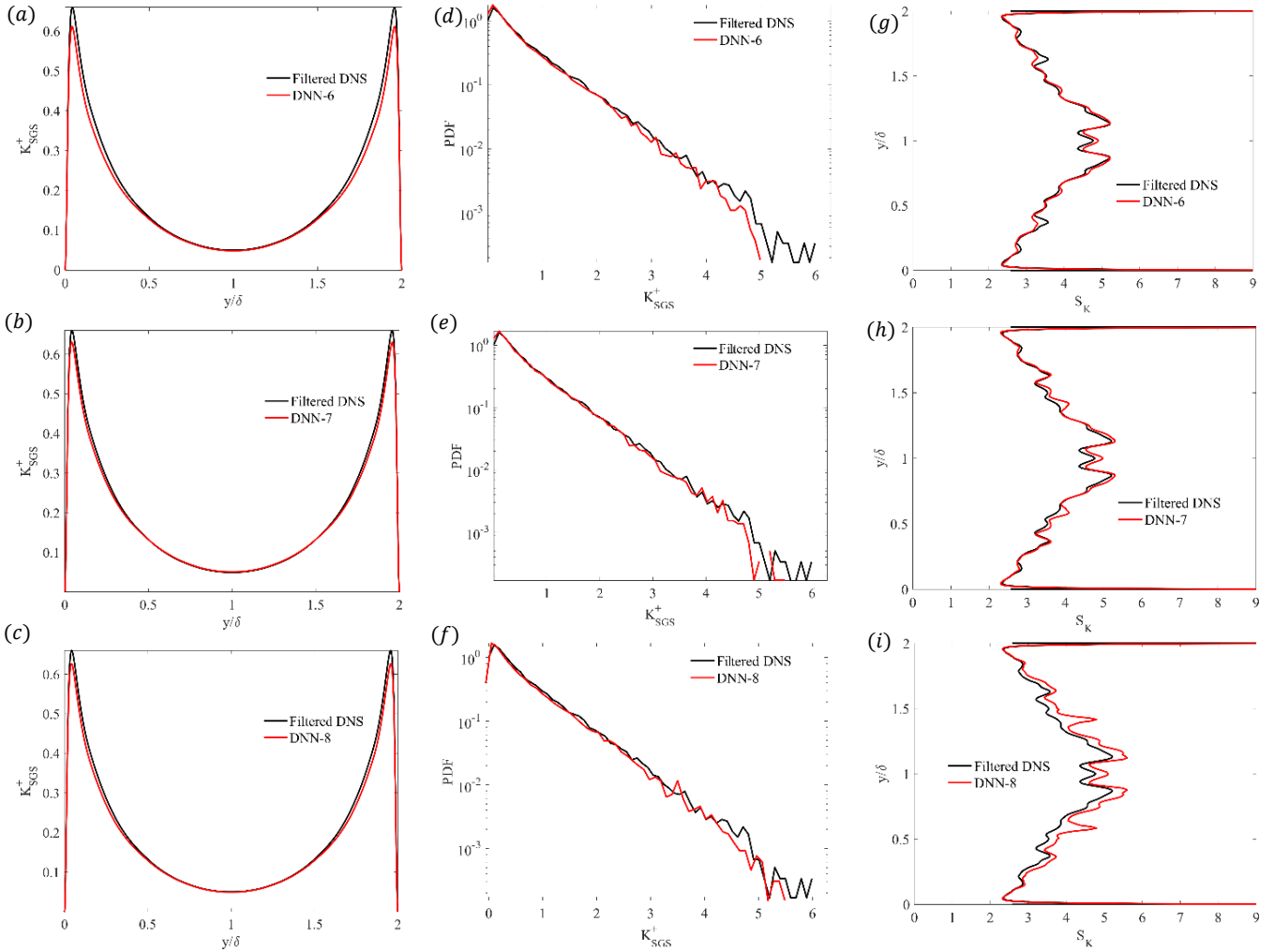
Fig. 6. Comparison between the DNN predictions and filtered DNS data of the mean SGS kinetic energy in wall units, K_{SGS}^+, (a-c), the PDF of K_{SGS}^+ (d-f) and its skewness, S_K, (g-i).

8 are almost similar.

The number of employed training data points on the performance of the DNN can also be examined by comparing the performance of the cases DNN-7, DNN-9, and DNN-10, where, 1,020,033, 4,080,132 and 16,842,752 data points, respectively, have been employed for training of the DNN. It is expected that training the DNN over a larger data set would result in increased accuracy and better correlation between the DNN and filtered DNS predictions. This is due to the fact that the DNN is presented with more diverse turbulence events, which provides the opportunity for a better learning process. In fact, it can be observed that CC increases with the increase N, while RMS and ERR both decrease with increasing N. However, one has to note that the accuracy of

the DNN predictions does not increase linearly with increasing N. Hence, it is decided to stop with 16,842,752 number of training data for the purpose of the current investigation.

The PDF and S_K of K_{SGS}^+ are also given in Fig. 6(d-f) and Fig. 6(g-i), respectively, for cases DNN-6, DNN-7 and DNN-8. It can be seen that the differences in the predictions are minor for these statistics with small differences in the tail of the PDF.

Given that the number of computations of the DNN model for K_{SGS} increases with increasing NL and considering the findings presented in this section, it can be deduced that the DNN-7 case can be chosen as the optimum model. Hence, the rest of the results in the upcoming section are only given for the DNN-7 case.

(a)

(b)
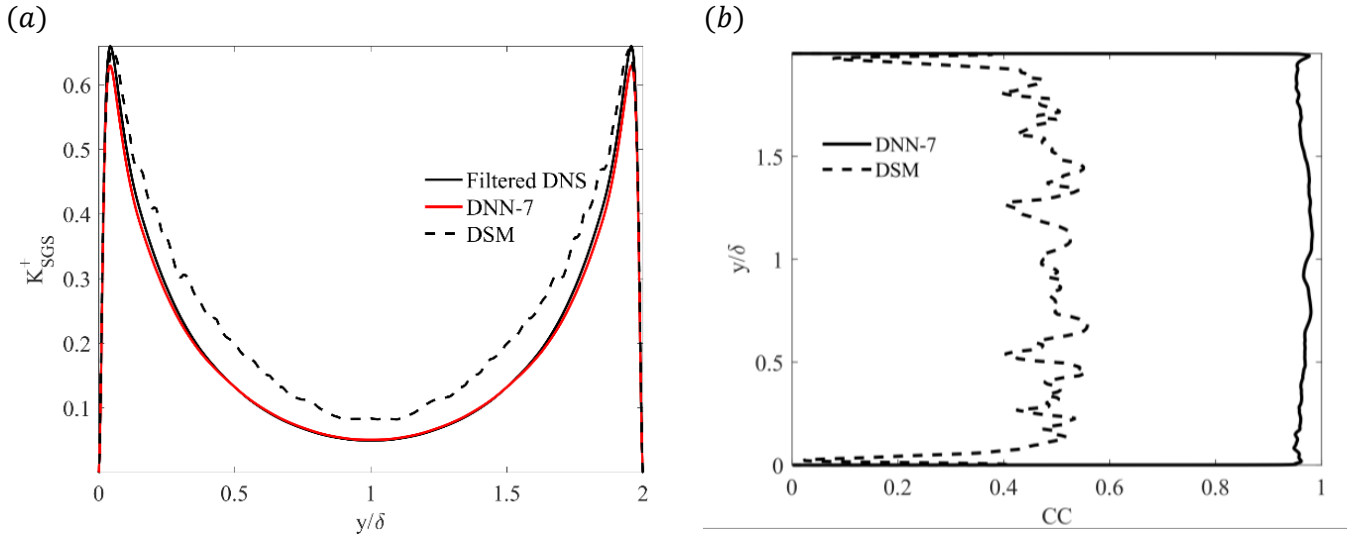
Fig. 7. Comparison of the mean SGS kinetic energy in wall units, K_{SGS}^+ , between the DNN-7 and DSM predictions and the filtered DNS data (a) and comparison of the correlation coefficient, with respect to the normalized wall distance, y/δ , between the DNN-7 and DSM predictions (b).

6- 3- Comparison of the DNN and DSM Predictions

The DSM is often used to model the K_{SGS}. It uses the dynamic procedure, as outlined in section 4. Here, the performance of the DNN for case DNN-7 is compared to that of the DSM in the prediction of the K_{SGS}. Fig. 7(a) presents the K_{SGS}^+ predicted by the DNN-7, the DSM, and the filtered DNS data.

The K_{SGS}^+ has a peak at about $y^+ = 12$ in the buffer layer, which corresponds to the peak location of the turbulent kinetic energy. The location of this peak is well-predicted by the DNN, but its magnitude is slightly under-predicted, compared with the filtered DNS data. On the contrary, the DSM prediction of the peak location is slightly further away from the wall, compared with the filtered DNS data and DNN-7 prediction, but its peak value is closer to the filtered DNS data than the DNN-7 prediction. It is also observed that, in contrast to the DNN-7 case and the filtered DNS data, the DSM prediction is intermittent across the whole channel height. Overall, the DNN shows very promising predictions of K_{SGS} and its prediction lies closer to the filtered DNS data across the whole channel height, compared with the commonly used DSM.

The correlation coefficient, CC is also given in Fig. 7(b). It is observed that the DSM has a much lower CC, compared with the DNN-7, reaching values of only up to 0.5, whereas the DNN-7 prediction reaches values even higher than 0.9 away from the walls. The DSM also has a very low CC in the important near-wall region, $y/\delta < 0.1$, compared with the DNN-7 case. The low CC for the DSM shows that the K_{SGS} is not sufficiently correlated with the $\tilde{S}_{ij}$ tensor. On the

contrary, the DNN uses the velocity-gradient tensor as input, allowing for a better correlation with K_{SGS}^{DNS}.

To comprehend the discussion, the PDF of K_{SGS}^+ is given in Fig. 8(a) for the DNN-7 case, DSM, and filtered DNS data at a wall distance of about $y^+ = 12$. It is observed that the DNN-7 prediction closely follows the filtered DNS data, whereas the DSM does not. The DSM prediction is much lower than the filtered DNS for $K_{SGS}^+ > 2$, in contrast to the DNN-7 prediction. The smaller PDF values with the DSM, indicate a lower number of events with high K_{SGS}^+ values.

Finally, the S_K is given in Fig. 8(b) for the DNN-7 case, DSM, and the filtered DNS data. It is observed that the DSM predicts larger S_K values, compared with the DNN-7 case and the filtered DNS data. It neither correctly predicts the near-wall peak nor the one in the center of the channel. Its prediction is very intermittent, even in the mean value, compared with the DNN-7 case and the filtered DNS data. The fluctuating predictions of the DSM for S_K are due to the fluctuating behavior of the dynamic coefficient.

7- Concluding Remarks

A deep neural network (DNN) was developed, trained and its performance to predict the SGS turbulent kinetic energy, K_{SGS}, for large-eddy simulation (LES) of turbulent channel flow was analyzed. The required training data for the DNN was obtained via a direct numerical simulation (DNS), performed at the friction Reynolds number $Re_\tau = 381$ using a pseudo-spectral method. Filtering was performed over the DNS database, with over 160 million data points, to obtain

(a)

(b)
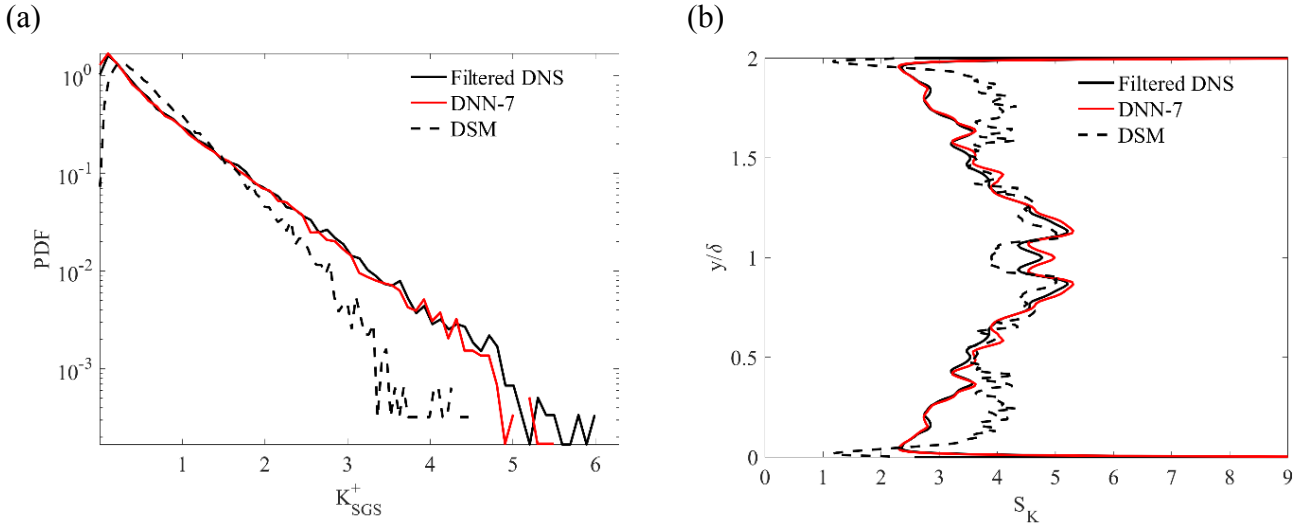
Fig. 8. Comparison of the probability density function, PDF, of the SGS kinetic energy in wall units, K_{SGS}^+, at $y^+ \approx 12$ (a) and the skewness of the SGS kinetic energy, S_K, (b) between the DNN-7 case and DSM predictions with filtered DNS data.

LES data.

The developed DNN successfully predicted the K_{SGS} with correlation coefficients with the filtered DNS data, CC, reaching up to 96.79%. It was observed that increasing the number of neurons, from 8 to 128, increased the accuracy of the K_{SGS}, its probability density function (PDF), and skewness, S_K predictions. The improvements where in the magnitude of the near-wall peak of K_{SGS}, the tail of the PDF, and the magnitude of the S_K. The effect of the NL in the predictions of the DNN was studied using 1, 2, and 3 hidden layers. It was observed that increasing NL to more than two did not markedly improve the predictions of the DNN.

The predictions of the DNN were compared with those of a commonly used dynamic SGS model (DSM). The DNN outperformed the DSM in the prediction of K_{SGS} its PDF and S_K with less intermittent predictions in better agreement with the filtered DNS data. The DNN prediction also showed much higher CC values, compared with the DSM.

The K_{SGS} presented in this paper is intended to be used for LES of wall-bounded flows using the lattice Boltzmann method (LBM). Hence, *a posteriori* assessment of the performance of the current model in LES using the LBM would be an interesting future research. It has to be pointed out that the current ANN is trained using filtered DNS data of turbulent channel flow at $\text{Re}_\tau = 381$. Hence, the accuracy of the current model needs to be evaluated for other test cases and Reynolds numbers in future analysis.

References

[1] S.B. Pope, Turbulent flows, Cambridge university press, 2001.

[2] S.A. Orszag, Analytical theories of turbulence, Journal of Fluid Mechanics, 41(2) (1970) 363-386.

[3] P. Sagaut, Large eddy simulation for incompressible flows: an introduction, Springer Science & Business Media, 2005.

[4] L. Marstorp, G. Brethouwer, O. Grundestam, A.V. Johansson, Explicit algebraic subgrid stress models with application to rotating channel flow, Journal of Fluid Mechanics, 639 (2009) 403-432.

[5] Y. Kuwata, K. Suga, Wall-modeled large eddy simulation of turbulent heat transfer by the lattice Boltzmann method, Journal of Computational Physics, 433 (2021) 110186.

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

[7] K. Duraisamy, G. Iaccarino, H. Xiao, Turbulence modeling in the age of data, Annual review of fluid mechanics, 51 (2019) 357-377.

[8] M. Gamahara, Y. Hattori, Searching for turbulence models by artificial neural network, Physical Review Fluids, 2(5) (2017) 054604.

[9] H. Frezat, G. Balarac, J. Le Sommer, R. Fablet, R. Lguensat, Physical invariance in neural networks for subgrid-scale scalar flux modeling, Physical Review Fluids, 6(2) (2021) 024607.

[10] Z. Wang, K. Luo, D. Li, J. Tan, J. Fan, Investigations of data-driven closure for subgrid-scale stress in large-eddy simulation, Physics of Fluids, 30(12) (2018) 125101.

[11] Q. Meng, Z. Jiang, J. Wang, Artificial neural network-based subgrid-scale models for LES of compressible turbulent channel flow, Theoretical and Applied Mechanics Letters, 13(100399) (2023) 1-12.

[12] D. Xu, J. Wang, C. Yu, S. Chen, Artificial-neural-network-based nonlinear algebraic models for large-eddy simulation of compressible wall-bounded turbulence, Journal of Fluid Mechanics, 960 (2023) A4.

[13] A. Rasam, M. Shirazi, Subgrid-scale flux modeling of a passive scalar in turbulent channel flow using artificial neural network, Iranian Journal of Mechanical Engineering Transactions of the ISME, 24(2) (2024) 157-172.

[14] M. Chevalier, P. Schlatter, A. Lundbladh, D.S. Henningson, SIMSON: A pseudo-spectral solver for incompressible boundary layer flows, KTH, 2007.

[15] J. Kim, P. Moin, R. Moser, Turbulence statistics in fully developed channel flow at low Reynolds number, Journal of fluid mechanics, 177 (1987) 133-166.

[16] A. Rasam, G. Brethouwer, A. Johansson, A stochastic extension of the explicit algebraic subgrid-scale models, Physics of fluids, 26(5) (2014) 055113.

[17] A. Rasam, G. Brethouwer, A.V. Johansson, An explicit algebraic model for the subgrid-scale passive scalar flux, Journal of Fluid Mechanics, 721 (2013) 541-577.

[18] A. Rasam, G. Brethouwer, P. Schlatter, Q. Li, A.V. Johansson, Effects of modelling, resolution and anisotropy of subgrid-scales on large eddy simulations of channel flow, Journal of turbulence, (12) (2011) N10.

[19] M.Y. Hussaini, T.A. Zang, Spectral methods in fluid dynamics, Annual review of fluid mechanics, 19(1) (1987) 339-367.

[20] R.D. Moser, J. Kim, N.N. Mansour, Direct numerical simulation of turbulent channel flow up to Re τ= 590, Physics of fluids, 11(4) (1999) 943-945.

[21] J. Park, H. Choi, Toward neural-network-based large eddy simulation: Application to turbulent channel flow, Journal of Fluid Mechanics, 914 (2021) A16.

[22] P.C. Di Leoni, T.A. Zaki, G. Karniadakis, C. Meneveau, Two-point stress–strain-rate correlation structure and non-local eddy viscosity in turbulent flows, Journal of Fluid Mechanics, 914 (2021) A6.

[23] N. Park, S. Lee, J. Lee, H. Choi, A dynamic subgrid-scale eddy viscosity model with a global model coefficient, Physics of Fluids, 18(12) (2006) 125109.

[24] S. Völker, R.D. Moser, P. Venugopal, Optimal large eddy simulation of turbulent channel flow based on direct numerical simulation statistical data, Physics of Fluids, 14(10) (2002) 3675-3691.

[25] M. Germano, U. Piomelli, P. Moin, W.H. Cabot, A dynamic subgrid-scale eddy viscosity model, Physics of Fluids, 3 (1991) 1760-1765.

[26] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, TensorFlow: a system for Large-Scale machine learning, 12th USENIX symposium on operating systems design and implementation (OSDI 16), (2016) 265-283.

[27] S. Haykin, Neural networks and learning machines, 3/E, Pearson Education India, 2009.

[28] [28] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv, 1412.6980 (2014) 1-15.

[29] M. Kang, Y. Jeon, D. You, Neural-network-based mixed subgrid-scale model for turbulent flow, Journal of Fluid Mechanics, 962 (2023) A38.

[30] F. Sarghini, G. de Felice, S. Santini, Neural networks based subgrid scale modeling in large eddy simulations, Computers & fluids, 32(1) (2003) 97-108.