SAND2016-12208C

Development of Machine Learning Models for Turbulent Wall Pressure Fluctuations

J. Ling [*] Matthew Barone [†] Warren Davis [‡] K. Chowdhary [§]

Jeffrey Fike [¶]

Sandia National Labs, Albuquerque, NM, 87123, and Livermore, CA, 94450 USA

In many aerospace applications, it is critical to be able to model fluid-structure interactions. In particular, correctly predicting the power spectral density of pressure fluctuations at surfaces can be important for assessing potential resonances and failure modes. Current turbulence modeling methods, such as wall-modeled Large Eddy Simulation and Detached Eddy Simulation, cannot reliably predict these pressure fluctuations for many applications of interest. The focus of this paper is on efforts to use data-driven machine learning methods to learn correction terms for the wall pressure fluctuation spectrum. In particular, the non-locality of the wall pressure fluctuations in a compressible boundary layer is investigated using random forests and neural networks trained and evaluated on Direct Numerical Simulation data.

Nomenclature

$f(x)$	Activation function in neural network
FSI	Fluid Structure Interaction
NN	Neural Network
PS	Power Spectral Density
P	Pressure
RMSE	Root Mean Squared Error
RFR	Random Forest Regressor
w	Weight vector in neuron of neural network
y^+	Distance from the wall in wall units

I. Introduction

Modeling Fluid-Structure Interaction (FSI) systems, such as airframes in crossflow or blade loading in gas turbine engines, requires accurate predictions of the turbulence-induced loads on the structure. Simulation methods typically used for these predictions, such as Large/Detached Eddy Simulations (LES/DES), suffer from high uncertainty due to near wall turbulence models (see, for example, Arunajatesan and Barone[1]). Physically, these loads are generated through highly non-linear and non-local mechanisms, and hence, the origins of errors in their predictions are poorly understood. Direct Numerical Simulations (DNS), which exactly resolve the near wall turbulent flow, can be used to investigate the sources of errors in these turbulence models. However, extracting sensitivity information from the massive amounts of data generated by DNS is an enormous challenge.

Machine learning methods, specifically designed to work on big, high-dimensional data sets, have the potential to transform our ability to quantify and address these uncertainties. Machine learning is a set

[*] Harry S. Truman Fellow, Thermal/Fluids Science and Engineering Department

[†] Principle Member of the Technical Staff, Aerosciences Department, AIAA Associate Fellow

[‡] Principle Member of the Technical Staff, Scalable Analysis and Visualization Department

[§] Extreme Scale Data Science and Analytics Department, AIAA Member

[¶] Post-doctoral Researcher, Aerosciences Department, AIAA Member of data-driven algorithms that can detect patterns in large data sets and build predictive models. These methods have already been applied to turbulence modeling in a number of contexts. Tracey et al.[2] used neural networks to replicate Reynolds Averaged Navier Stokes (RANS) model source terms. Duraisamy et al.[3–5] used Gaussian processes and neural networks to predict turbulence intermittency and a correction term for RANS model source terms. Ling et al.[6–8] used Random Forests and neural networks to predict when RANS Reynolds stress closures would have high uncertainty and to predict corrections for the Reynolds stress anisotropy. They were able to demonstrate significantly reduced error in the Reynolds stresses when using these data-driven models as compared to the linear eddy viscosity model conventionally employed in RANS.

These promising results suggest that these data driven methods could also provide improved models for surface pressure fluctuations. However, the non-locality of pressure fluctuations adds an extra layer of complexity to this problem: it is not clear what the inputs to the machine learning model should be. While Ling et al.[8] and Duraisamy et al.[3] used only local flow variables to predict their RANS corrections, it seems likely that non-local information will be required to improve wall pressure predictions. Furthermore, because of how these flows are typically modeled using DES, the points at the wall typically have the highest model uncertainty. While the free stream flow is modeled using unsteady LES solvers, the near wall regions are modeled using RANS and wall-models, which suffer from higher model form uncertainty. Therefore, the machine learning model will likely have improved performance if it has access to non-local information.

This study explores the question of what information a machine learning model should have access to in order to provide a correction for the wall pressure fluctuation power spectrum. A Direct Numerical Simulation (DNS) has been performed of a compressible flat plate boundary layer at Mach 2. The pressure power spectral densities (PSDs) from this simulation have been extracted at all points in the flow. Deep neural networks and random forest regressors are used to analyze these power spectra to examine the extent to which information about the wall pressure PSD is contained in data at varying distances from the wall. This investigation will provide valuable information about the potential for creating data-driven wall functions for the wall pressure PSD.

Section II will present the computational set-up of the DNS and Section III will explain the machine learning framework and implementation. Sections IV and V will present results and conclusions.

II. Direct Numerical Simulation

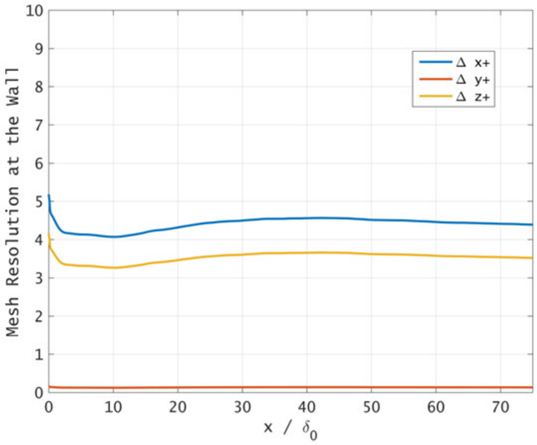

Figure 1. Near-wall mesh resolution for the Mach 2.0 boundary layer DNS.

A DNS of the turbulent boundary layer over a flat plate at free stream Mach number of 2.0 has been performed using the SIGMA-CFD code.[9] SIGMA-CFD emplyos a multi-block, structured grid, finite volume discretization. For DNS, the code solves the compressible Navier-Stokes equations using a low-dissipation, 5th-order upwind biased flux-reconstruction scheme. The time integration scheme is a fourth order accurate explicit Runge Kutta scheme.

For the present DNS, a single-block structured finite volume mesh was used, with dimensions of 2525 × 190 × 210 for a total of 100.7M mesh cells. The bulk flow is in the x-direction, with the solid wall at the plane $y = 0$. A mean turbulent boundary layer profile is imposed at the inflow boundary, with superimposed fluctuations that serve to trip the flow into a turbulent state. The domain extends in the streamwise direction for a distance of $75\delta_0$, where δ_0 is the inlet boundary layer thickness. An absorbing sponge boundary treatment is applied over the region $75\delta_0 < x < 90\delta_0$. The spanwise extent of the domain is $L_z = 6\delta_0$. The specified flow conditions and domain extent allow for a spatially developing turbulent boundary to develop with a range of momentum-thickness Reynolds number of $1075 \leq Re_\theta \leq 1310$.

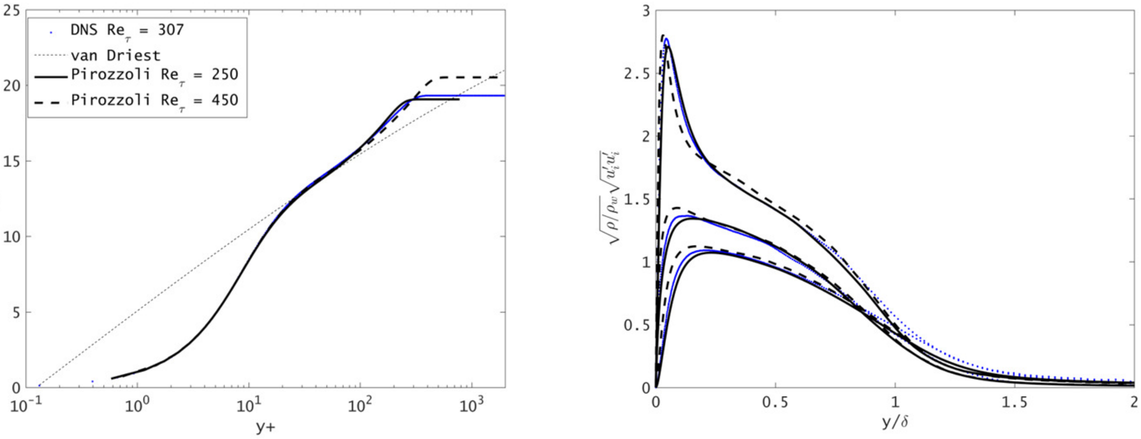

Figure 2. Mean and RMS velocity profiles for the Mach 2.0 boundary layer DNS, compared with previous DNS results from Pirozolli and Bernardini.[10]

After an initial transient period required to establish the boundary layer turbulence, the DNS was run for 1229τ, where τ is a characteristic time equal to the inlet boundary layer thickness δ_0 divided by the free stream velocity U_∞. Figure 1 shows the mesh resolution in terms of mesh spacing in "wall coordinates." The resolution is seen to be very high, with $\Delta x^+ < 5$, $\Delta z^+ < 4$, $\Delta y^+ < 0.2$. However, this is required due to the somewhat dissipative nature of the spatial discretization scheme. Solution verification simulations were performed by running additional simulations on a 200M cell mesh, with grid refinement in the wall-normal direction only, and on a 400M cell mesh, with grid refinement applied in the $x-$ and $z-$ directions only. Insensitivity of mean flow and fluctuation profiles to these refinements demonstrated that 100M cell mesh was adequate to provide a quality DNS database.

Mean and fluctuation velocity statistics were collected from the DNS to verify behavior against previous published results for compressible flat plate boundary layers. Figure 2 compares the mean and RMS velocity profiles for the present DNS at $Re_\tau = 302$ to the results from Pirozolli and Bernardini[10] at $Re_\tau = 250$ and $Re_\tau = 450$. The level of agreement demonstrated in these figures lends further credibility to the current DNS.

III. Machine Learning Methodology

Two different machine learning algorithms are explored in this paper: random forest regressors and neural networks. Both of these algorithms are used for regression. The target output from the machine learning models was the log power spectral density of pressure at a specified point on the wall. The inputs to the machine learning models were the log pressure power spectra at points above the wall. A key goal of this work as to determine what information was most useful in predicting the pressure spectrum at the wall. Therefore, a series of different machine learning models were trained using input power spectra at varying distances from the wall.

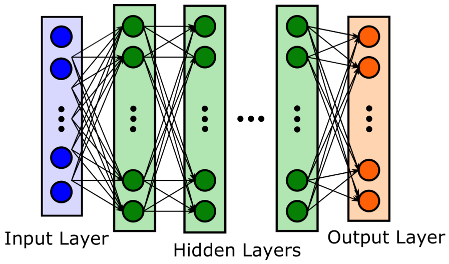

Figure 3. Schematic of feed forward neural network.

A. Random Forest Regressors

Random Forest Regression (RFR) is an algorithm that uses an ensemble of binary decision trees to make a prediction.[11] Each decision tree is trained on a random subset of the training data, selected using bootstrap aggregation, also known as *bagging*. In bagging, data are randomly selected from the training set with replacement. For any new point, the random forest prediction is given as the mean of the predictions of all the trees in the forest.

A key asset of RFRs is that they can not only make predictions, they can also tell the user what features were most important in making those predictions through feature importance metrics. The RFR model is fit to the training data in a greedy fashion: each split is determined based on what will provide the maximal decrease in the variance of the output values for the children nodes. The feature importance of a given feature is the normalized, aggregated reduction in this variance provided by splits on that feature over the entire random forest. This feature importance metric will be used to assess what information is most useful in making predictions of the wall pressure spectra.

B. Neural Networks

Neural Networks (NNs) are a machine learning method in which inputs are transformed through multiple layers of non-linear interactions. Figure 3 shows a schematic of a simple feed-forward multilayer perceptron, the neural network architecture used in this study. In this architecture, the inputs are fed into the Input Layer. This input layer is densely connected to a series of hidden layers. At each node in the hidden layers, the inputs to the node are combined through an activation function to provide an output:

$$y = f(w^T x) \tag{1}$$

In Eq. 1, the output of the node y is given by a non-linear *activation function* f of the dot product of the inputs to the node x with the node weight vector w. Through successive hidden layers, complex representations of the data can be constructed and the strongly non-linear behavior of turbulence can be modeled. The final layer of the neural network has a linear activation function (i.e. $f(x) = x$) and produces the final network prediction.

The process of *training* a NN is analogous to calibrating a Bayesian model or fitting a regression model. In the training process, the weight vectors w of all the neurons in the network are tuned to provide the best fit to the training data. In this case, mini-batch gradient descent was employed for the network training. This training technique performs gradient descent on randomly sampled subsets of the training data enabling the network to more easily step out of local minima in the objective function during training. Back propagation is the process by which weights in each of the hidden layer nodes are successively updated using gradient information starting at the output layer and propagating back through the network.

The NNs used in this work were implemented using the open source Python library Lasagne, which is built on Theano. They used Rectified Linear activation functions and had 7 hidden layers, with 1680 hidden nodes in all. These parameters were chosen as giving the best performance after multiple trials with different numbers of layers and nodes. An early stopping criterion was used to mitigate overfitting. In early stopping, a section of the training data is set aside and not used to fit the neural network weights. After each gradient descent iteration, the performance of the neural network is evaluated on this held-out set and network training is halted when the error on this held-out set plateaus. This widely used technique prevents

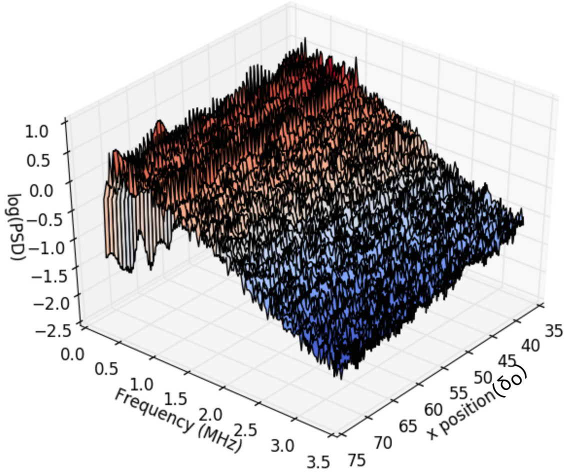

Figure 4. Log pressure PSD at the wall.

the neural network from over-fitting the training data by halting training when the validation error is no longer improving.

IV. Results

A. DNS Pressure Spectra Results

The pressure PSDs were calculated at each point using the fftw C library[12] which performs a discrete PSD. A moving average filter of window size 3 was applied to the resulting discrete PSDs to provide smoothing. This resulted in a discrete PSD of length 112, with a maximum frequency of 3.3 MHz and a frequency resolution of 30 kHz. In all, 1001 wall pressure PSDs were calculated at evenly spaced streamwise intervals, $0.035\delta_0$ apart. These 1001 wall pressure PSDs were used as the training and validation set for the machine learning model.

Figure 4 shows the log pressure PSD at the wall as a function of streamwise position (measured in inlet boundary layer thicknesses, δ_0). Results are shown for $x = 35\delta_0$, at which point the turbulent boundary layer has had sufficient time to develop, to the end of the domain at $x = 70\delta_0$. The data were partitioned into two parts during the machine learning phase: the first 80% of the data from $35\delta_0 < x < 63\delta_0$ were used for training the machine learning models. The last 20% of the data from $63\delta_0 < x < 70\delta_0$ were used for testing the model. This sequential partitioning was used to enable us to detect over-fitting. If the training and validation data had been randomly sampled from the domain, then correlations between neighboring data points would have allowed the neural networks to interpolate and overfit the data.

Figure 5 shows the pressure PSD at $x = 50\delta_0$ at 6 different wall distances: at the wall and at $y^+ = 10, 25, 50, 100$, and 200. As this figure shows, there is a substantial amount of statistical noise on the pressure PSDs. The PSD at $y^+ = 10$ is very similar to the PSD at the wall throughout the frequency range plotted, with strongly correlated peaks. The PSDs at $y^+ = 25$ and $y^+ = 50$ both have substantial correlations to the wall PSD, though their base power level is higher than at the wall for frequencies above 1 MHz. At locations farther from the wall, at $y^+ = 100$ and $y^+ = 200$, the overall power level and peaks of the PSDs are much less strongly related to the wall PSD.

B. Results from Random Forest Regressors

A RFR model was trained to predict each output frequency in the discrete PSD. RFR models were trained using inputs PSDs from five different distances from the wall. In each case, the input PSD was from the

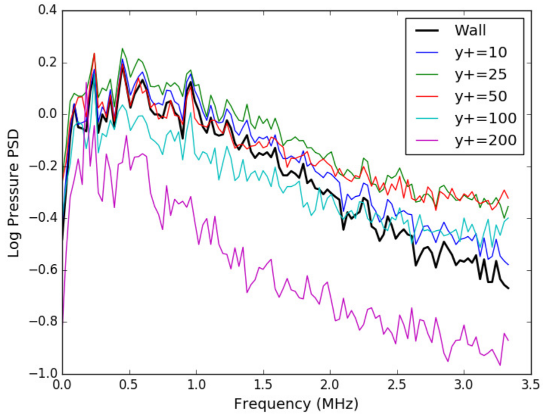

Figure 5. Log pressure PSD at 6 different wall distances.

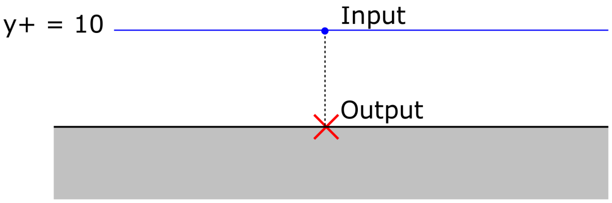

Figure 6. Schematic of inputs and outputs of machine learning models.

point directly above the point at the wall. Therefore, the data set consisted of 1001 pairings of the PSD at a point on the wall with the PSD at the point directly above at a specified distance from the wall. This pairing is shown schematically in Fig. 6. Five different wall distances were investigated for the input PSD: $y^+ = 10, 25, 50, 100$, and 200. These locations were chosen because they include points in the viscous sublayer ($y^+ = 10$), in the buffer layer ($y^+ = 25$), in the log layer ($y^+ = 50, 100$), and beyond the log layer ($y^+ = 200$).

Figure 7 shows the RFR predictions of the wall pressure PSD at $x = 66.5\delta_0$, in the middle in the validation data. This figure also shows the input PSD at that location at each wall distance. As this figure shows, when the input PSD is from $y^+ = 10$, the RFR is able to almost exactly match the wall PSD. This is not surprising given that the PSD at $y^+ = 10$ is already very close to the wall PSD. When greater wall distances are used for the input PSD, the RFR is still able to predict the wall PSD with a high level of accuracy, especially for the lower frequencies in the spectrum. At the higher frequencies, the RFR predictions become less accurate when the input PSDs are from farther distances from the wall.

The relation between RFR accuracy and frequency is shown more clearly in Fig. 8. This figure plots the magnitude of the difference between the RFR predictions of a given PSD frequency and the true wall PSD value at that frequency. This error magnitude was averaged over all streamwise locations in the validation set. It is worth noting that even when predicting a specific output frequency, the RFR has access to the entire input spectrum. The error magnitude has been smoothed using a Savitsky-Golay filter to reduce the noise. Despite the noisiness of this plot, several trends are clear. First, it is clear that the RFR using input PSDs from $y^+ = 10$ has the lowest error at all frequencies. It is also clear that for all wall distances, the error increases at higher frequencies. This effect is most pronounced for the inputs farthest from the wall. This result makes physical sense because the higher-frequency pressure fluctuations are due to smaller eddies that are very close to the wall in the boundary layer. Therefore, less information about these eddies would be available in the PSDs farther from the wall.

Figure 9 shows the root mean squared error (RMSE) of the RFR predictions, averaged over all streamwise locations in the validation set and averaged over all frequencies. This error is compared to the RMSE of the input pressure spectrum versus the wall pressure spectrum. This comparison shows that the RFR is

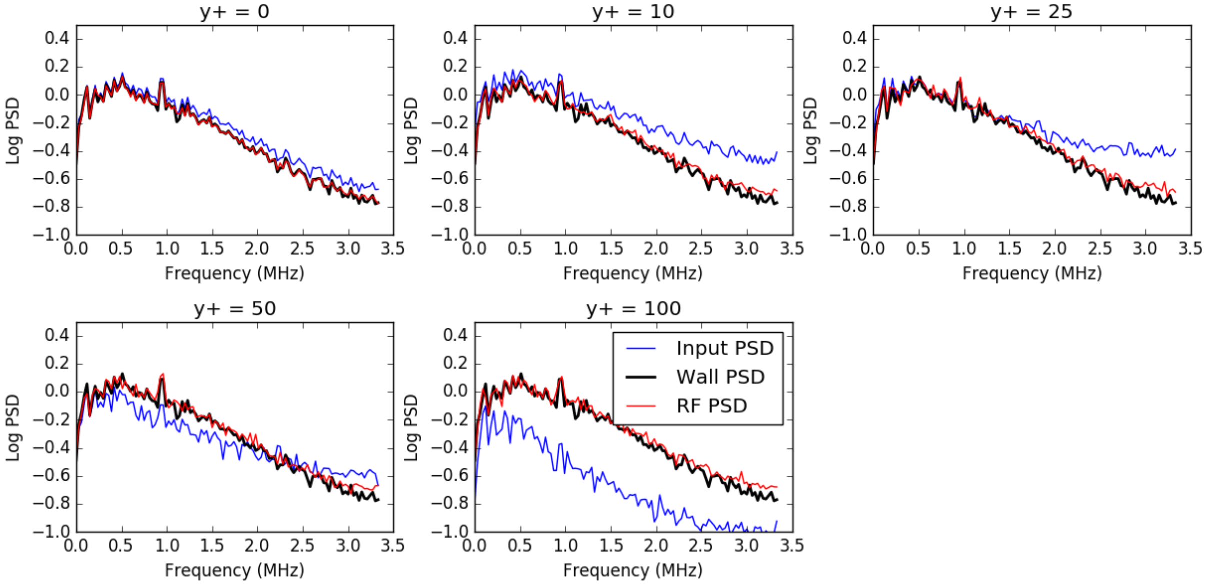

Figure 7. RFR predictions of the wall pressure PSD at $x = 66.5\delta_0$ based on input PSDs from different wall distances. The black line shows the true wall PSD, the red line shows the RFR predicted PSD, and the blue line shows the input PSD.

learning a correction. The RFR prediction is closer to the wall PSD than the input PSD is. In general, the errors of both the input PSD and the RFR PSD increase with increasing distance from the wall. The error at $y^+ = 100$ is slightly lower. While it is not yet fully understood what causes the lower error at this wall distance, it could be related to the fact that this wall distance is in the log layer.

It is also interesting to examine the feature importance of the different input frequencies in predicting the wall PSDs. Figure 10 shows the feature importance metrics for a band of frequencies from 1.26 MHz to 1.86 MHz, in the middle of the resolved frequency range. In this figure, higher feature importance means that that input frequency was important in predicting the wall PSD in the highlighted frequency band. As this figure shows, when the input PSD is from $y^+ = 10$, the feature importance is high in exactly the frequency band we are trying to predict. This result indicates that for $y^+ = 10$, the only frequencies that are needed to predict the wall PSD within a given frequency band are the input frequencies in that same band. For example, in order to predict the wall spectrum at 1 MHz, the RFR only needs the input PSD at that frequency. It is not using information from any of the other frequencies in the spectrum. As the distance from the wall grows, more broadband information is used to predict the PSD within the specified frequency band. Furthermore, at the farther wall distances, the feature importance of the high frequency component of the spectrum is on average lower than the feature importance of the low frequency part of the spectrum. Therefore, while the low frequency portion of the spectrum seems to contain some information about the highlighted frequency band, the high frequency portion of the spectrum does not. It is possible that this result indicates that the strength of the larger, lower frequency eddies farther from the wall has some correspondence to the energy contained in the smaller eddies near the wall.

C. Neural network results

A NN model was also trained to predict the wall pressure PSD, given the pressure PSD at a point directly above the wall point. While RFRs are known to be robust and easy-to-implement, NNs are more complicated to implement but can also capture complex non-linear interactions more naturally. Figure 11 shows the NN predictions of the wall pressure PSD at $x = 66.5\delta_0$, at the same location as Fig. 7 shows the predictions for the RFR. As this figure shows, the NN gives poor predictions of the high frequency power spectra when using input spectra from all wall distances evaluated. This poor performance could be due to a poorly chosen network architecture. NNs, with their thousands of free weights, are highly susceptible to converging to local

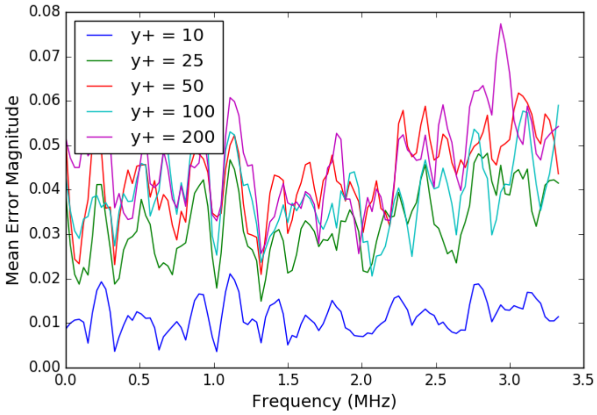

Figure 8. Error in RFR predictions as a function of frequency in the wall PSD.

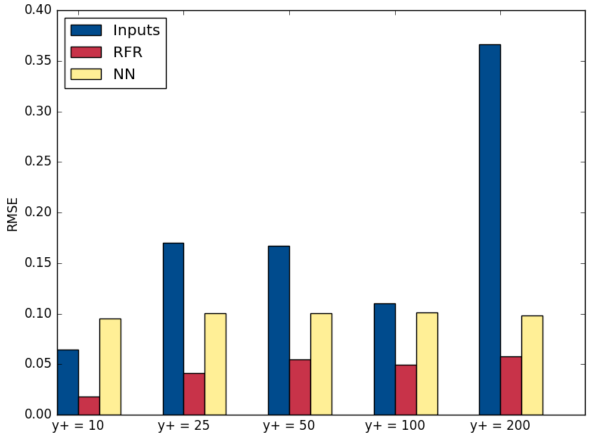

Figure 9. Error averaged over all frequency bands. The blue bars show the root mean squared difference between the input PSD and the wall PSD. The red bars show the root mean squared difference between the RFR predictions and the wall PSD. The yellow bars show the root mean squared difference between the NN predictions and the wall PSD.

minima instead of the global minimum of the objective function. We believe that better NN performance could be achieved in the future by using convolutional neural networks that have fewer free weights and leverage the concept of adjacency in frequency space.

Figure 9 shows the RMSE of the neural network predictions, averaged over the entire test set and over all frequency bands of the wall PSD. This plot shows that overall, the NN gives worse predictions than the RFR. In fact, for the $y^+ = 10$ case, the NN gives worse predictions than just using the input PSD would have. These results highlight the difficulty of training neural networks on relatively small data sets (only 1001 input-output pairs): they are prone to over-fitting and converging to local minima. RFRs, on the other hand, are more robust to over-fitting and easier to implement.

V. Conclusions

A Direct Numerical Simulation of a Mach 2 boundary layer was performed. A machine learning framework was established to determine whether it would be possible to predict the wall pressure spectrum given the pressure spectrum at points directly above the wall. The over-arching objective of this research was to

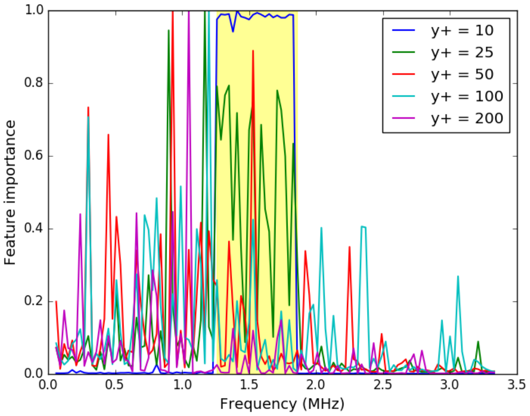

Figure 10. Feature importance of different input frequencies in predicting the wall PSD in the band from 1.26 MHz to 1.86 MHz (this band highlighted in yellow). Results shown for 5 different wall distances for the input PSD.

ascertain whether machine learning could be used to learn a wall function for the pressure spectrum such that the near wall region would not need to be resolved in order to correctly predict the wall pressure PSD. If such a data-driven wall function could be constructed, it could lead to reduced computational cost in fluid-structure interaction simulations. A systematic investigation was undertaken using pressure PSD information at different distances from the wall to determine how much information about the wall PSD was contained in the PSDs at different wall distances.

Two different machine learning algorithms were evaluated: Random Forest Regressors and feed forward Neural Networks. The RFR was able to accurately predict the wall pressure PSD using as input the pressure PSD from $y^+ = 10$ or 25, with under 5% RMSE. Slightly degraded performance at the higher frequency bands was noted when inputs from $y^+ = 50, 100$ or 200 were used, but the RMSE was still under 6%. In general, the accuracy of the wall PSD predictions degraded in the higher frequency bands as the wall distance of the inputs increased. This trend makes sense because the small eddies responsible for the high frequency fluctuations are characteristic of the very near wall region.

The NN predictions were overall less accurate than the RFR predictions. Neural networks are notoriously harder to implement than Random Forests. The authors believe that switching to an architecture more suited to making predictions on frequency spectra data, such as a Convolutional Neural Network architecture, could alleviate training difficulties and lead to better neural network performance in the future.

The strong performance of the RFR models suggest that it would be possible to build an enhanced wall model for the pressure PSD that uses information from the log layer ($50 < y^+ < 100$ in this flow) to predict the pressure PSD at the wall. Such a wall model could have numerous applications in running efficient simulations of fluid-structure interactions. Future work will focus both on investigating a better suited neural network architecture and on validating these random forest results across a wider data-base of flows at multiple Mach numbers.

Acknowledgments

Supported by the Laboratory Directed Research and Development program at Sandia National Laboratories, a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

References

[1] S. Arunajatesan and M. Barone. Towards computational study of flow within cavities with complex geometric features. AIAA 2015-0008, Proceedings of the 53rd Aerospace Sciences Meeting, 2015.

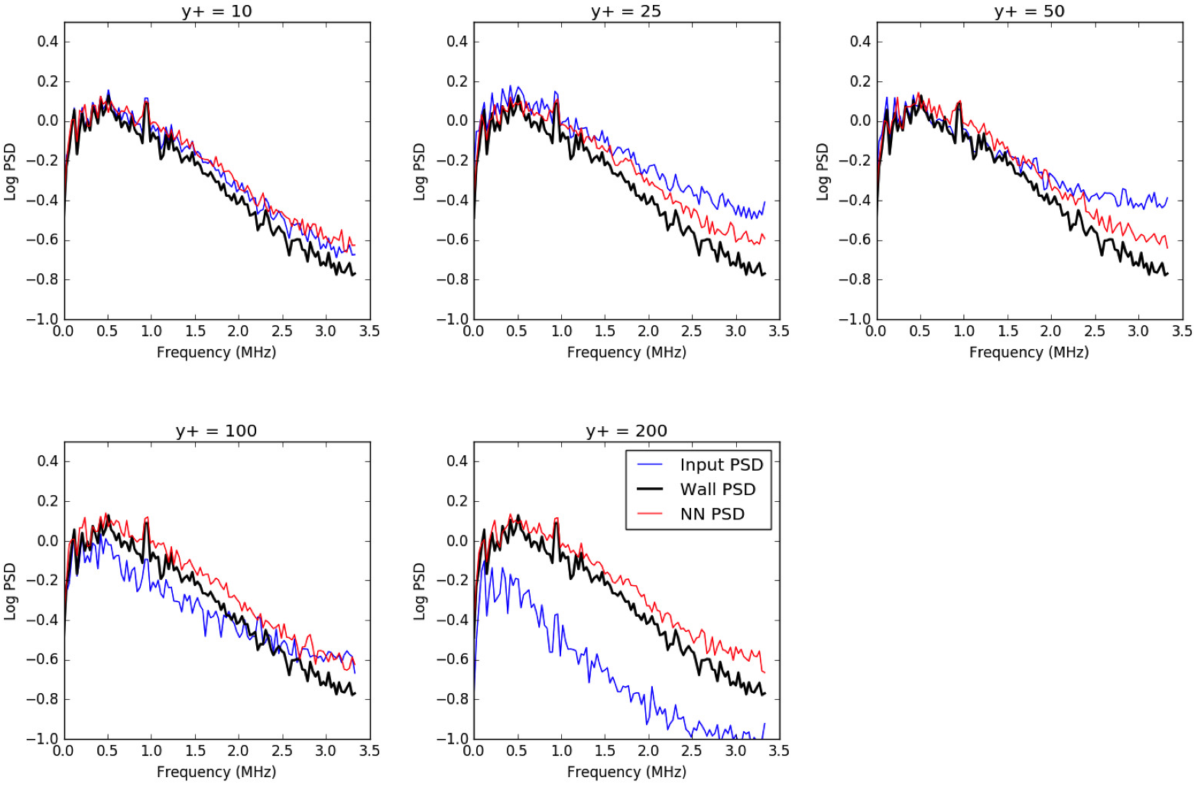

Figure 11. NN predictions of the wall pressure PSD at $x = 66.5\delta_0$ based on input PSDs from different wall distances. The black line shows the true wall PSD, the red line shows the NN predicted PSD, and the blue line shows the input PSD.

[2] B. Tracey, K. Duraisamy, and J.J. Alonso. A machine learning strategy to assist turbulence model development. *AIAA SciTech*, pages 2015–1287, 2015.

[3] K. Duraisamy, Z.J. Shang, and A.P. Singh. New approaches in turbulence and transition modeling using data-driven techniques. *AIAA SciTech*, pages 2015–1284, 2015.

[4] Z.J. Zhang and K. Duraisamy. Machine learning methods for data-driven turbulence modeling. *AIAA Aviation*, pages 2015–2460, 2015.

[5] E. Parish and K. Duraisamy. A paradigm for data-driven predictive modeling using field inversion and machine learning. *Journal of computational physics*, 305:758–774, 2016.

[6] J. Ling and J.A. Templeton. Evaluation of machine learning algorithms for prediction of regions of high RANS uncertainty. *Physics of Fluids*, 27:085103, 2015.

[7] J. Ling, A. Ruiz, G. Lacaze, and J. Oefelein. Uncertainty analysis and data-driven model advances for a jet-in-crossflow. *ASME Turbo Expo 2016*, 2016.

[8] J. Ling, A. Kurzawski, and J. Templeton. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *Journal of Fluid Mechanics*, 807:155–166, 2016.

[9] M. Barone and S. Arunajatesan. Pressure loading within rectangular cavities with and without a captive store. AIAA 2014-1406, Proceedings of the 52nd Aerospace Sciences Meeting, 2014.

[10] S. Pirozolli and M. Bernardini. Turbulence in supersonic boundary layers at moderate Reynolds number. *J. Fluid Mech.*, 688:120–168, 2011.

[11] L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

[12] M. Frigo. A fast fourier transform compiler. *ACM*, 34, 1999.