A Machine Learning Approach to Jet-Surface Interaction Noise Modeling

Cliff Brown *, Jonny Dowdall †, Brian Whiteaker ‡, Lauren McIntyre §, Chris Miller ¶
NASA Glenn Research Center, Cleveland, OH, 44135, USA

This paper investigates using machine learning to rapidly develop empirical models suitable for system-level aircraft noise studies. In particular, machine learning is used to train a neural network to predict the noise spectra produced by a round jet near a surface over a range of surface lengths, surface standoff distances, jet Mach numbers, and observer angles. These spectra include two sources, jet-mixing noise and jet-surface interaction (JSI) noise, with different scale factors as well as surface shielding and reflection effects to create a multi-dimensional problem. A second model is then trained using data from three rectangular nozzles to include nozzle aspect ratio in the spectral prediction. The training and validation data are from an extensive jet-surface interaction noise database acquired at the NASA Glenn Research Center's Aero-Acoustic Propulsion Laboratory. Although the number of training and validation points is small compared a typical machine learning application, the results of this investigation show that this approach is viable if the underlying data are well behaved.

Nomenclature

MSE	mean square error for one configuration, dB
$\overline{MSE}$	mean square error averaged over all configurations, dB
Θ	observation angle relative to upstream jet axis (polar or yaw angle)
x_E	surface length, inches
h_E	surface standoff distance from nozzle lipline, inches
P_a	ambient pressure, psi
NPR	nozzle pressure ratio
U_j	jet exit velocity, ft/s
c_a	jet exit velocity, ft/s
M_a	jet acoustic Mach number, $M_a = U_j/c_a$
T_sR	jet static temperature ratio
T_tR	jet total temperature ratio
RH	ambient relative humidity, %
β_{SSS}	jet potential core length

I. Introduction

MODERN aircraft design seeks to more tightly couple the engines to the airframe in search of greater efficiency or reduced sonic boom. However, one side effect of these designs is that the high-speed engine exhaust is very close to the airframe surfaces. This arrangement can lead to jet-surface interaction (JSI) noise, and its related effects, that can be beneficial or detrimental to the overall aircraft noise levels created

*Research Scientist, Acoustics Branch, 21000 Brookpark Rd., AIAA Member.
†Intern, Scientific Computing Visualization Branch, 21000 Brookpark Rd.
‡Intern, Scientific Computing Visualization Branch, 21000 Brookpark Rd.
§Computational Scientist, Scientific Computing and Visualization Team, 21000 Brookpark Rd.
¶Research Scientist, Acoustics Branch, 21000 Brookpark Rd. AIAA Member

by the aircraft in flight. Therefore, it is important to include the JSI noise considerations in the initial design optimization for any new aircraft.

Jet-surface interaction noise can be separated into two primary sources and two primary effects. The first JSI noise source is the surface "scrubbing" noise that is created when the high-speed exhaust flows along the surface as a wall-bounded flow. The second JSI source is the trailing-edge noise that is created when the high-speed exhaust passes over (or near[8]) the trailing-edge of the surface and becomes a free-shear flow. The two primary effects of having surface near the jet exhaust are noise shielding and noise reflection. Noise shielding occurs when the surface blocks the jet-mixing noise from the observer when noise reflection happens when the surface reflects the jet-mixing noise back at the observers; these effects are characterized respectively as a reduction or amplification at the high-frequency end of the spectra. Finally, there may be secondary effects depending on the particular geometry. For example, flow data has shown that the peak turbulent kinetic energy, the source of the jet-mixing noise, may be amplified downstream of the surface.[8] These combined sources and effects are a big challenge when attempting to a model the system-level (jet and airframe) noise. These JSI noise sources and effects are in addition to the jet-mixing noise that is created when the high-speed exhaust mixes with the ambient surrounding air. Combined, these sources and effects form a complex system that can be difficult to predict and/or model. This is particularly true when these predictions must run as part of a system optimization study where it is critical to minimize computational time. In these cases, an empirical model based on experimental data or detailed computational results is often a viable option to capture the general trends within the time constrained environment.

The complexity of an empirical model generally depends on the number of independent variables and the scatter of the dependent variable. For example, a nicely behaved 1-dimensional dataset might be well represented by linear or a low order polynomial while a good model for a dataset of nearly random samples might impossible to find. In a dataset with several independent variables, it can be difficult to find the inter-relationships between the variables even if the data are fairly well behaved. In general, our ability to "see" relationships in more than three or four dimensions is poor and, therefore, we tend to treat (or remove) the the variables one at a time, adding uncertainty to the model for each reduced dimension. In some cases, the underlying physics can be used to simplify the data, for example through non-dimensionalization of the variables, but the modeling process can still be a lot of time-consuming trial and error. One option in these more complex cases is to apply machine learning techniques, like neural networks, to search for relationships between variables that are not obvious upon first examination of the data.

A computer-based neural network shares many similarities with biological networks. The computer-based neural network consists of a series of interconnected nodes (neurons) in one or more layers that connect the input data to the output data (i.e. prediction, Figure 1). Each node has a value, or weight that is "learned" from the underlying dataset and applied based on the rules for each node type. For example, a threshold logic unit (TLU) would output zero or one[a]. If the input value it receives is less than or greater than some threshold value. The process of determining these weights is call machine learning: the neural network takes a set of related input/output data and determines the node weights that best relate them using a process that minimizes the error between output of the neural network and the actual dataset. The structure of the neural network is largely up to the programmer but it is typical to start with a number of nodes between the number of input variables and output values and to select a number of layers based on the expected non-linearity of the system. In general, the neural network is trained on some random subset of the data (training set) and the remaining subset is used to evaluate the result (validation set).

The possible states that can result from a machine learning trial are shown in Figure 2. The ideal state is a good agreement between the training and validation data which indicates that the model is capturing the general behavior of the data without over-fitting. On the opposite side of this state, the model does a poor job capturing both the training and validation data; this indicates that the neural network may not have the right structure or that an important input parameter is missing. Between these two states is a third where the neural network captures the training data but does not represent the validation data; here the model is over-fitting the training data.[b] Thus, the dataset must be large enough train the neural network while leaving enough points in reserve to evaluate the result.[c]

[a]Or the TLU would multiply its predetermined weight by the input value before sending it out to the other nodes. In this way exceeding the threshold value could also "turn off" the node if the weight is zero.

[b]The one-dimensional analogy would be using an n^{th}-order polynomial to represent n data points. It will hit each point exactly but will likely have large over- or under-shoots between these points that do not match reality.

[c]A fourth state, where there is a poor fit between the training data but a good fit to the validation data, is theoretically possible but is very rare in practice.

This paper will consider using a neural network to model the noise created by a jet near a surface. The neural network model will be trained using experimental database, previously acquired at the NASA Glenn Research Center, where nozzle aspect ratio, jet velocity, surface length, and surface standoff distance are varied. The goals of this work are (1) to create a usable model for system-level aircraft exhaust noise prediction and (2) to understand advantages and limitations of the machine learning approach to empirical modeling and how in might be used in other situations where quality empirical models are still needed.

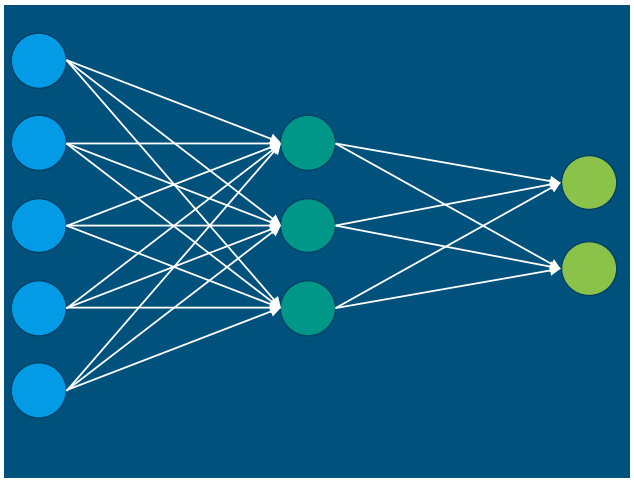


Figure 1: Schematic of a neural network with inputs on the left (blue), one hidden layer (green) and outputs on the right (light green).

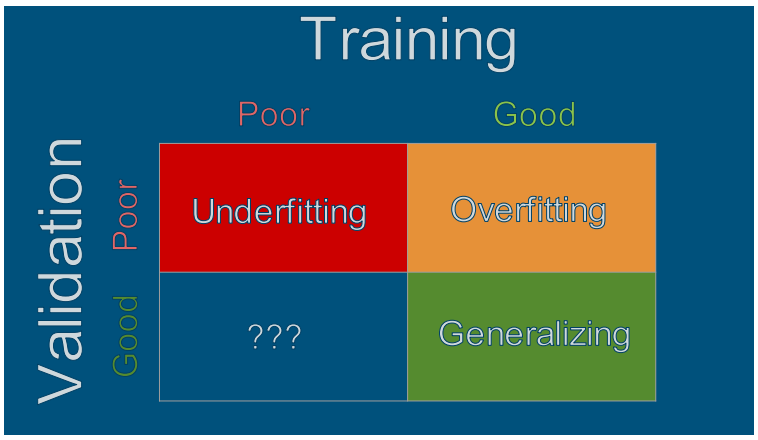


Figure 2: Outcome states from a machine learning trial.

II. Training and Validation Data

The experimental database used to train and validate the neural network was acquired during several tests at the NASA Glenn Research Center's Aero-Acoustic Propulsion Laboratory between 2012 and 2015. These experiments used a simplified jet-surface system: a single-stream round or rectangular nozzle near a flat semi-infinite surface (Figure 3). The jet exit conditions (setpoints) used during these tests are shown in Table 1. The surface length (x_E) and standoff distance (h_E) were varied systematically (Table 2) with data acquired for both the shielded and reflected observer at each surface location. In addition, data for each combination of setpoint and surface position were acquired using a round and three rectangular nozzles (2:1, 4:1, and 8:1 aspect ratios), all with an equivalent diameter $D_e \approx 2$ inches.

The acoustic data were acquired using 24 microphones placed on a 150-inch radius arc at 5° intervals from 50° to 160°. The data were sampled at 200 kHz (90 kHz Nyquist filter) and processed to power spectral

density (PSD) using a standard fast Fourier transform routine. The background noise (measured before each test run) was then subtracted on a frequency-by-frequency basis and the data were corrected for the frequency response of each microphone individually using the calibration provided by the manufacturer. The data were then transformed to a lossless condition by removing the atmospheric attenuation and scaled to a $100D_e$-radius arc assuming spherical spreading of sound. Finally, the narrowband data were converted to 1/12-octave PSD to capture the overall spectral shape but reduce the number of spectral points before starting the machine learning process.

Additional details on the facility can be found in reference 9 and more details on the JSI noise data set can be found in references 1–4

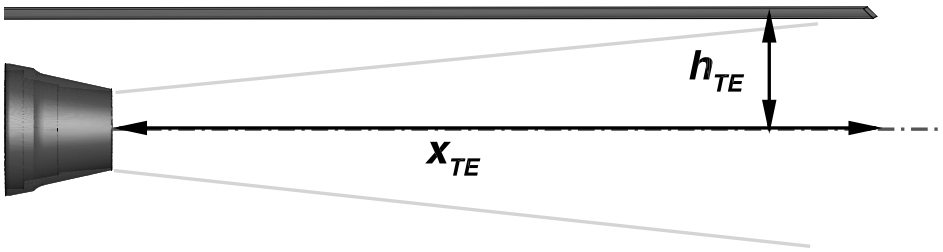


Figure 3: Schematic showing the jet-surface experimental setup.

Setpoint	Nozzle Pressure Ratio	$T_{T,R}$	M_a	Mass Flow (kg/s)
3	1.197	1.0	0.5	0.44
5	1.436	1.0	0.7	0.65
7	1.860	1.0	0.9	0.91
23	1.103	1.81	0.5	0.24
32	1.079	2.43	0.5	0.19
27	1.360	1.91	0.9	0.43
36	1.273	2.43	0.9	0.33
46	1.227	2.75	0.9	0.28

Table 1: Jet exhaust flow conditions in the JSI noise database. Note that only the unheated jet conditions (setpoints 3,5, and 7) will be used for these models.

III. Neural Network and Machine Learning

The first goal of this work was to determine if a machine learning approach to modeling JSI noise is feasible. A typical machine learning trial, like image classification, will have millions of data points. In contrast, the JSI noise database only contains about 1700 physical configurations (nozzle, surface position, setpoint) or approximately 41,000 spectra when all observer locations are considered. Previous experience working with this dataset indicates that these data are, in general, fairly well behaved[5,6] and, therefore, is still considered a candidate for machine learning despite the relatively small number of points.

The JSI database is sparse in nozzle aspect ratio which leads significant spectral changes between data points and, particularly, between the round and 2:1 rectangular nozzle. Therefore, separate models were developed for the round nozzle and for the three rectangular nozzles. The decision illustrates one trade-off in the model development process: modeling only the round nozzle significantly reduces the number of data points but the data has a more consistent behavior and each trial/iteration takes less time to complete. In this case, this approach proved worthwhile so the separate models are described below.

h_E (*inches*)			x_E(*inches*)				
	1.3	2.7	4	8	12	16	20
0.0	3,5,7	3,5,7	*ALL*	*ALL*	*ALL*	*ALL*	*ALL*
0.1	3,5,7	3,5,7	3,5,7	3,5,7	3,5,7	3,5,7	3,5,7
0.2	3,5,7	3,5,7	3,5,7	3,5,7	3,5,7	3,5,7	3,5,7
0.3	3,5,7	3,5,7	3,5,7	3,5,7	3,5,7	3,5,7	3,5,7
0.5	3,5,7	3,5,7	*ALL*	*ALL*	*ALL*	*ALL*	*ALL*
0.7	3,5,7	3,5,7	3,5,7	3,5,7	3,5,7	3,5,7	3,5,7
1.0	3,5,7	3,5,7	*ALL*	*ALL*	*ALL*	*ALL*	*ALL*
1.4	3,5,7	3,5,7	3,5,7	3,5,7	3,5,7	3,5,7	3,5,7
1.5			*ALL*	*ALL*	*ALL*	*ALL*	*ALL*
1.9	3,5,7	3,5,7	3,5,7	3,5,7	3,5,7	3,5,7	3,5,7
2.0			*ALL*	*ALL*	*ALL*	*ALL*	*ALL*
2.5	3,5,7	3,5,7	*ALL*	*ALL*	*ALL*	*ALL*	*ALL*
3.0			*ALL*	*ALL*	*ALL*	*ALL*	*ALL*
3.2	3,5,7	3,5,7	3,5,7	3,5,7	3,5,7	3,5,7	3,5,7
4.0	3,5,7	3,5,7	*ALL*	*ALL*	*ALL*	*ALL*	*ALL*
5.0	3,5,7	3,5,7	*ALL*	*ALL*	*ALL*	*ALL*	*ALL*

Table 2: Surface lengths (x_E) and radial standoff distances (h_E) used to generate the empirical model. The numbers in each box represent the setpoint tested at that surface location (*ALL* indicates that all setpoints in Table 1).

A. Round Nozzle

A neural network was created using Python 3.6 and Keras/TensorFlow to perform the initial trials. The original input architecture was selected by using hidden layers that cascade from the input size to the output size in powers of two. With 15 input features and 78 output values (one for each 1/12-octave frequency), the internal hidden layers were defined with 16, 32, and 64 nodes. This architecture proved satisfactory to model the experimental data and, therefore, the effect of network architecture on training performance and generalization was left for future work. The 15 input features were: nozzle aspect ratio, nozzle area, surface length (x_E), surface standoff (h_E), acoustic Mach number (M_a), gas dynamic Mach number (M_j), nozzle pressure ratio (NPR), nozzle total temperature ratio (T_tR), nozzle static temperature ratio (T_sR), ambient temperature (T_a), ambient pressure (P_a), ambient relative humidity (RH), jet potential core length (β_{SSS}), observer angle (Θ), and a flag indicating if the observer was on the shielded or reflected side of the surface; the input variables were dimensional to allow the machine learning program the largest latitude to explore relationships between them. This model was trained on a randomly selected subset of the data and the remaining data were used to validate the results.

Several of the variables in the JSI noise dataset either do not vary (e.g. aspect ratio for the round nozzle, T_tR), are combinations of other variables (e.g. $\beta_{SSS} = f(M_a, T_sR)$), or have already been used in the preprocessing of the data (e.g. RH[d]). Furthermore, the noise spectra will have a stronger relationship to some variables than others. The machine learning process assigns weights to individual nodes based on the input value and its contribution to the output so that summing the weights associated with an input feature can serve as a (somewhat crude) approximation of the relative importance of that feature to the prediction. The weakest predictive feature can then be removed and the model retrained in a iterative process to find the smallest number of input features that still gives a "good enough" prediction. In this case, the original 15 input features were reduced to 5: x_E, h_E, M_a, Θ, and the shielded/reflected observer flag.

The idea of a "good enough" prediction is that there is a trade-off between model complexity and accuracy. In this case, using all 15 features resulted in a model with an average mean square error ($\overline{MSE}$[e])

[d]Since data were acquired in an outdoor facility, standard processing was used to convert to account for variations in atmospheric losses; atmospheric humidity is the dominant variable in this transformation

[e]Mean square error (MSE) is the mean square error computed across all spectral frequencies at one configuration. Average

of approximately 0.42 dB while using only five most weighted features produced a model with $\overline{MSE} = 0.55$ dB. Since the experimental uncertainty in the underlying dataset is approximately ±0.5 dB, a prediction from five feature model is essentially as accurate as one from the 15 feature model. Thus, the comparisons shown in this section will use the 5 feature model. These results also suggest that number of hidden layer and/or node in the neural network might be reduced but this was again left for future investigation.

Figures 4 and 5 compare predicted and measured spectra for the shielded and reflected observers respectively at four observer angles ($\Theta = 60°$,90°,120°,150°). The spectra shown correspond to the corners of the configuration space (i.e. shortest and longest x_E, minimum and maximum h_E, minimum M_a). The model generally performs well for the shielded configuration. The largest mean square errors are found with the $x_E = 12$ surface at upstream observer angles data where the JSI noise source is trending down from its peak amplitude (which occurs at $\Theta \approx 90°$) and the jet-mixing noise shielding is increasing. In this region, the spectra becomes more sensitive to the experimental setup (particularly the thickness of the surface support system) and signal-to-noise ratio is low which disrupts the trends the model is trying to fit at across the observer angles farther downstream. Similarly, the maximum mean square error (MSE) in the reflected configuration occurs for the $x_E = 12$ surface close to the jet ($h_E = 0$). However, it is not obvious why the model performance is worse in this region and, in general, produces worse predictions on the reflected side than the shielded side.

Jet-mixing noise is commonly characterized as a distribution of incoherent sources so that, theoretically, perfect reflection would result in a 3 dB increase to the reflected observer. One approach to modeling this problem is to separate the shielded and reflected configurations into different models and limit the reflection to 3 dB (e.g. the approach in [6]). Here, the machine learning method gets a random mix of all configurations and is allowed to minimize the error using a given node structure. So while it is easy for us to look at the boolean shield/reflect flag and assume that the machine learning algorithm will use that to create two models internally, there is no good way to figure out how the algorithm will model the data. As a result, it is then hard to determine conclusively why the prediction is farther off for some configurations than others. One possible explanation is that the reflected observer $x_E = 12$ configurations are under-represented in the randomly selected training set.

The average mean square error ($\overline{MSE}$) is one way to look at model performance. In this model, $\overline{MSE} = 0.55$ dB was computed using only the randomly selected training data during the machine learning process. This is on the right side of the possible outcomes box shown in Figure 2. Ideally, the model would perform as well when compared to the validation data and the final state would be in the lower right box of Figure 2. However, $\overline{MSE} \approx 1.5$ dB when computed for $\Theta = 60°$,90°,120°,150° using all configurations indicating that the overall model state is in the upper right corner (i.e. good fit to training data, poor fit to validation data). Figure 5 showed that the model performs poorly for the reflected observer with the $x_E = 12$; in fact, removing these configurations from the validation set gives $\overline{MSE} \approx 0.5$ dB for all remaining configurations ($\overline{MSE} \approx 0.75$ dB for all shielded observers and and $\overline{MSE} \approx 0.25$ dB for all remaining reflected observers). Thus, it appears that the reflected $x_E = 12$ configurations are vastly under-represented in the randomly selected training set.

Tables 3 and 4 show the $\overline{MSE}$ for each surface length and surface standoff respectively. Table 3 shows that the $\overline{MSE}$ increases for both shielded and reflected observers as x_E increases and reiterates that the $\overline{MSE}$ for the $x_E = 12$ reflected observer is out of line with the other configurations. The $\overline{MSE}$ as a function of h_E generally decreases as a h_E increases (Table 4). Furthermore, if the $x_E = 12$ configurations are not included in the $\overline{MSE}$, the model performs on the order of 0.1 dB better for the shield observers but dramatically better for the reflected observer, consistent with the analysis thus far.

B. Rectangular Nozzle

Machine learning trials were run for round and rectangular nozzles in one model and for the rectangular nozzles only in a second trial. The round nozzle was assumed to be the same as a 1:1 rectangular nozzle for when it was included. Whether this assumption or something else in the process, the MSE when the round nozzle was included was significantly higher than the ±0.5 dB experimental uncertainty and models where where the rectangular nozzle data were modeled separate from the round nozzle data. Therefore, the model and results in this section use the model that excludes the round nozzle.

Like the model for the round nozzle, a neural network was created using Python 3.6 and Keras/TensorFlow

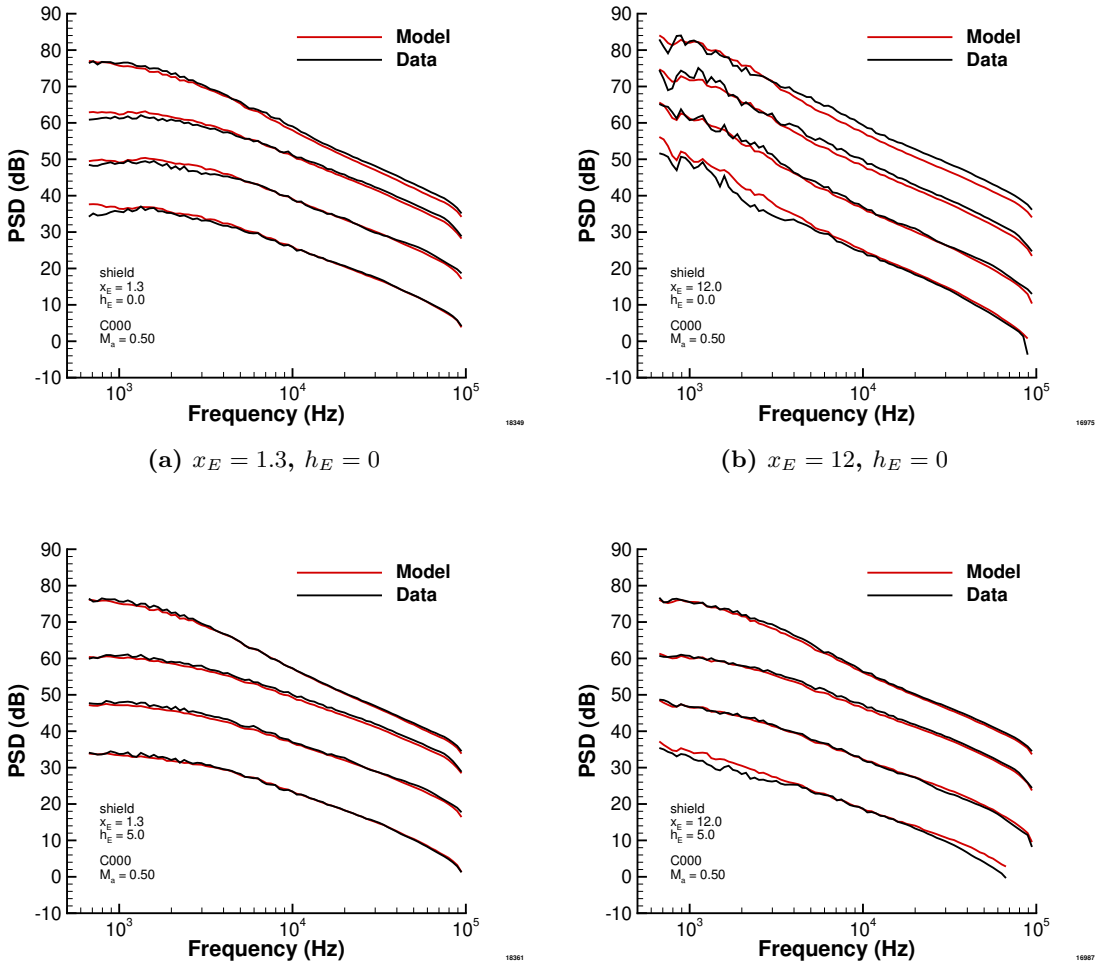
mean square error ($\overline{MSE}$) is the average MSE across all configurations in the training and/or validation dataset.

x_E	Shielded	Reflected
1.3	0.305	0.316
2.7	0.446	0.331
4	0.673	0.410
8	1.119	0.563
12	1.266	5.535

Table 3: Average mean square errors for each surface length using the round nozzle.

	All x_E		$x_E < 12$	
h_E	Shielded	Reflected	Shielded	Reflected
0	1.379	4.365	1.217	0.604
0.1	1.020	3.847	0.931	0.544
0.2	0.898	3.525	0.858	0.616
0.3	0.917	3.257	0.915	0.811
0.5	0.847	2.609	0.779	0.604
0.7	0.709	2.197	0.568	0.458
1	0.621	2.037	0.429	0.378
1.4	0.619	1.985	0.369	0.337
1.9	0.646	1.682	0.372	0.278
2.5	0.565	1.213	0.369	0.223
3.2	0.447	0.808	0.360	0.194
4	0.451	0.648	0.406	0.183
5	0.527	0.614	0.489	0.180

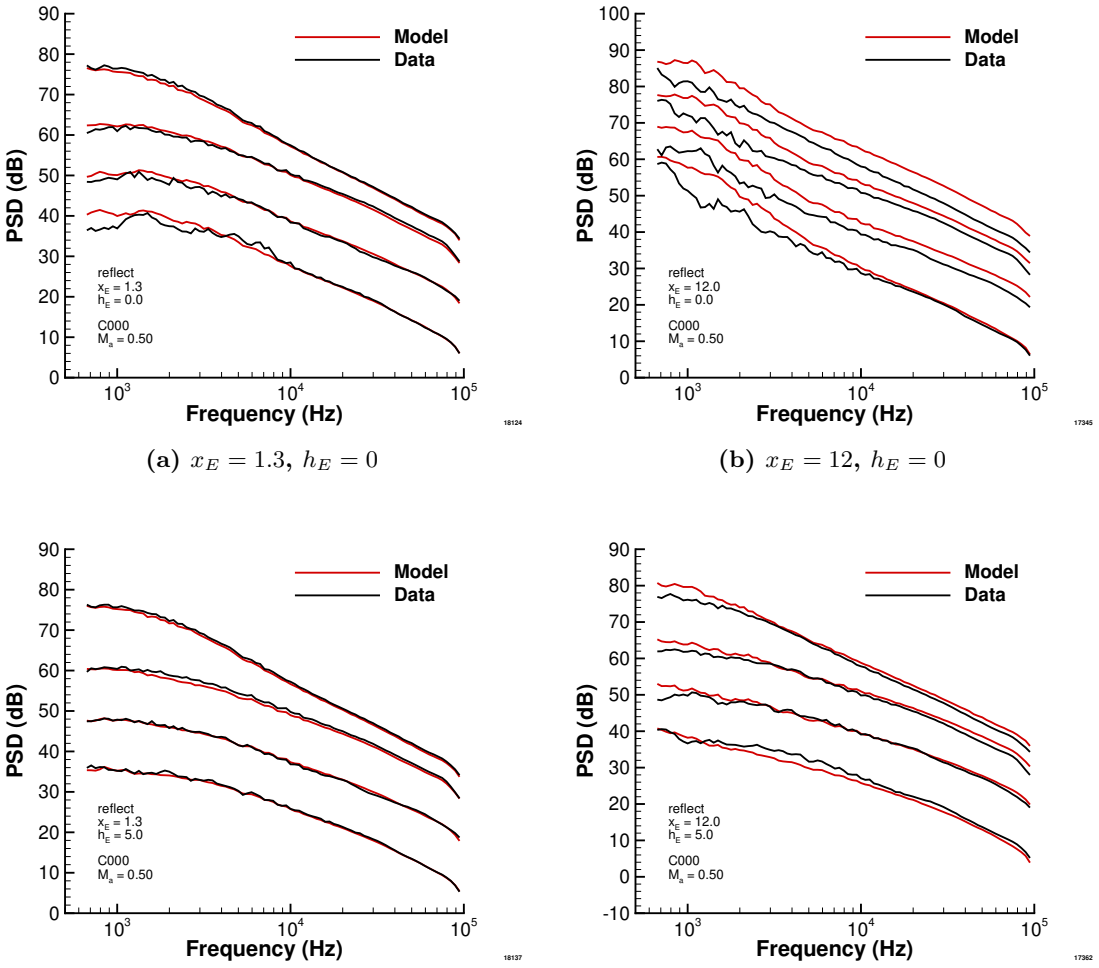
Table 4: Average mean square errors for each surface standoff (h_E) using the round nozzle. Average MSE was computed for all points and for all points where $x_E < 12$.



(a) $x_E = 1.3$, $h_E = 0$ **(b)** $x_E = 12$, $h_E = 0$

(c) $x_E = 1.3$, $h_E = 5$ **(d)** $x_E = 12$, $h_E = 5$

Figure 4: Spectra predicted by the machine learning model compared to the experimental data for the $M_a = 0.5$ round nozzle for the $x_E = 1.3, 12$ and $h_E = 0, 5$ surfaces for the shielded observer. The line pairs are observer angles $\Theta = 60^\circ$ (bottom), $\Theta = 90^\circ$, $\Theta = 120^\circ$, and $\Theta = 150^\circ$ (top). Note each spectra is separated by 10 dB from the previous for clarity.



(a) $x_E = 1.3$, $h_E = 0$ **(b)** $x_E = 12$, $h_E = 0$

(c) $x_E = 1.3$, $h_E = 5$ **(d)** $x_E = 12$, $h_E = 5$

Figure 5: Spectra predicted by the machine learning model compared to the experimental data for the $M_a = 0.5$ round nozzle for the $x_E = 1.3, 12$ and $h_E = 0, 5$ surfaces for the reflected observer. The line pairs are observer angles $\Theta = 60°$ (bottom), $\Theta = 90°$, $\Theta = 120°$, and $\Theta = 150°$ (top). Note each spectra is separated by 10 dB from the previous for clarity.

to perform the initial trials. This model, however, included all 15 features used in the original round nozzle model (Section A)[f] Figures 6 and 7 compare experimental and modeled spectra at the shielded and reflected observers, respectively, for the 2:1 aspect ratio nozzle with the longest and shortest surfaces. In general, the model does a good job of reproducing the spectra without over-fitting the data. For example, the modeled spectra is a fairly smooth line through the peaks and valleys in the data at the reflected observers when $x_E = 12$, $h_E = 0$ (Figure 9b). The model also reproduces the spectra well for the $h_E = 5$ standoff distance. Interestingly, the data shows that the spectral peak is flattened at $x_E = 1.3$, $h_E = 0$ for the shielded observer (Figure 6a) and the model does not capture this behavior.

Spectral comparisons for the 8:1 aspect ratio nozzle are shown in Figures 8 and 9 for the shielded and reflected observers respectively. Again, the experimentally measured spectra has a flatter peak to the shielded observer at $x_E = 1.3$, $h_E = 0$ that is not well captured by the machine learning model. One possible explanation for both the 2:1 and 8:1 aspect ratio nozzles is that this behavior is too localized for the machine learning algorithm to give it much attention (i.e. it is not in enough samples). Physically, the shortest surface will move out of the flow at relatively small standoff distances and the spectra will assume the shape and scaling characteristics of jet-mixing noise with some limited shielding upstream. Thus, the machine learning may find the optimum model for a given number of nodes largely neglects the surface when it is sufficiently short. Tables 5 and 6 show the $\overline{MSE}$ for the 2:1 and 8:1 aspect ratio nozzles as functions of x_E and h_E respectively. Like the round nozzle, the model performs better for short surfaces than the longer surfaces and better for larger h_E than surface closer to the nozzle lip. Also, the model performs better for the 2:1 aspect ratio nozzle than the 8:1 aspect ratio nozzle. Finally, note that one model was trained for all the rectangular nozzles; the $\overline{MSE} \approx 1$ dB when computed for all surface configurations, flow conditions and nozzle aspect ratios.

	2:1		8:1	
x_E	shield	reflect	shield	reflect
1.3	0.456	0.407	0.732	0.562
2.7	0.614	0.488	1.045	0.942
4	0.671	0.570	1.498	1.168
8	0.836	0.759	1.334	1.341
12	0.836	0.826	1.342	1.069

Table 5: Average mean square errors for each surface length using the 2:1 and 8:1 aspect ratio nozzles.

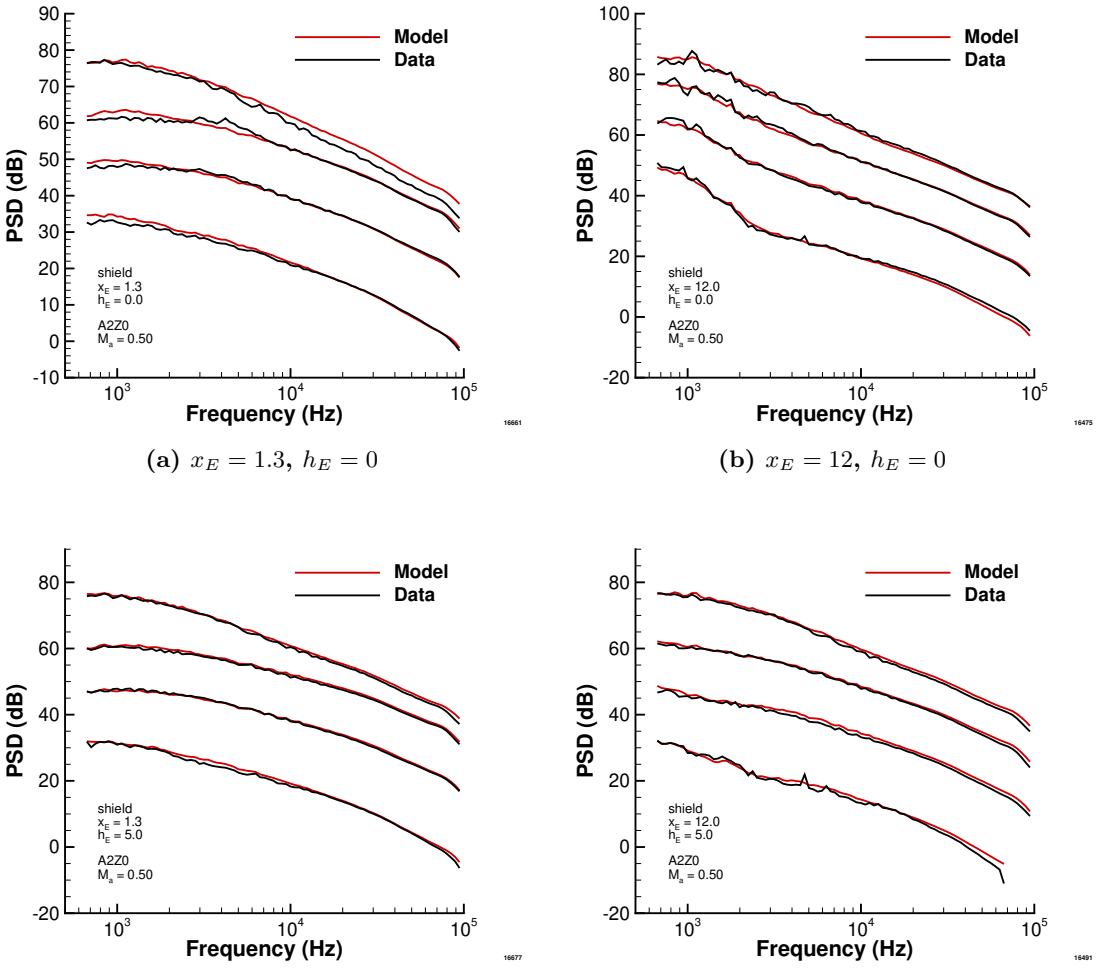
IV. Analysis

Separate models for round and rectangular jets near surfaces have been developed using machine learning techniques. These models, which have relatively simple node structures, were trained using standard machine learning methods available in Python Keras toolbox and, therefore, were up and running within a few weeks of starting the project. This is the first promise of machine learning: the development time required to develop a model from a dataset can be significantly shortened compared to traditional data fitting methods. In fact, these models were used in Sections A and B because they were created quickly and reproduce the data with a reasonable level of accuracy. However, there are a couple of issues that need to be considered. First, do these models effectively interpolate and/or extrapolate across different input features. Second, do more advanced machine learning algorithms significantly improve the results and, if so, what is the trade-off between model accuracy and complexity (of both the machine learning process and the resulting model).

Figure 10 shows examples of the model performance when interpolating across observer angles and surface length. The angular interpolation (Figure 10a) was generated by running the model at $\Theta = 2.5^\circ$ increments so that every other point is interpolated. The results show a generally smooth variation (± 1 dB) between angles as is expected with these data and small increments.

Unlike observer angle, the variations in surface length were logarithmically spaced and there were fewer overall in the data set making x_E a much more difficult test of the model's ability to interpolate. Figure 10b

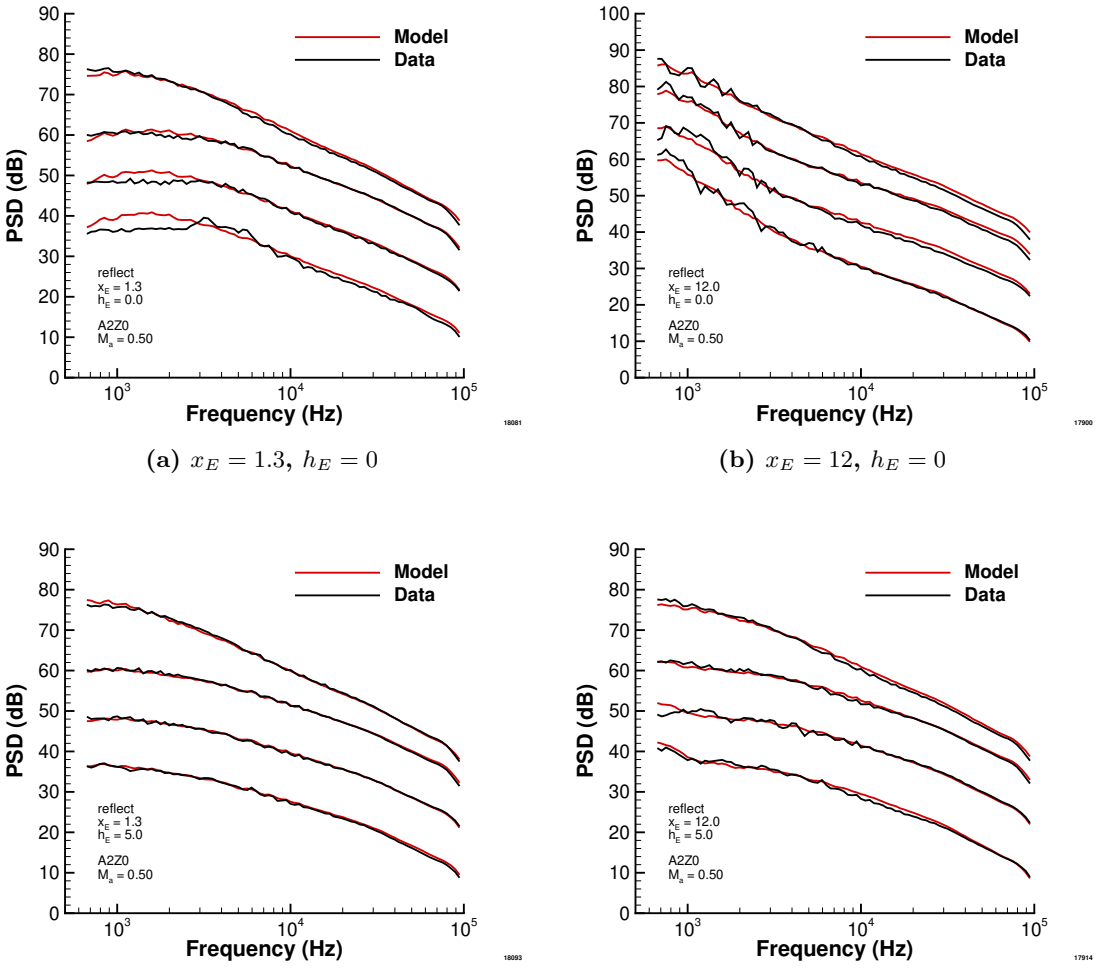
[f]It is likely that this number could be reduced using methods similar to the round nozzle. However, feature reduction was not attempted with this particular model.



(a) $x_E = 1.3$, $h_E = 0$ **(b)** $x_E = 12$, $h_E = 0$

(c) $x_E = 1.3$, $h_E = 5$ **(d)** $x_E = 12$, $h_E = 5$

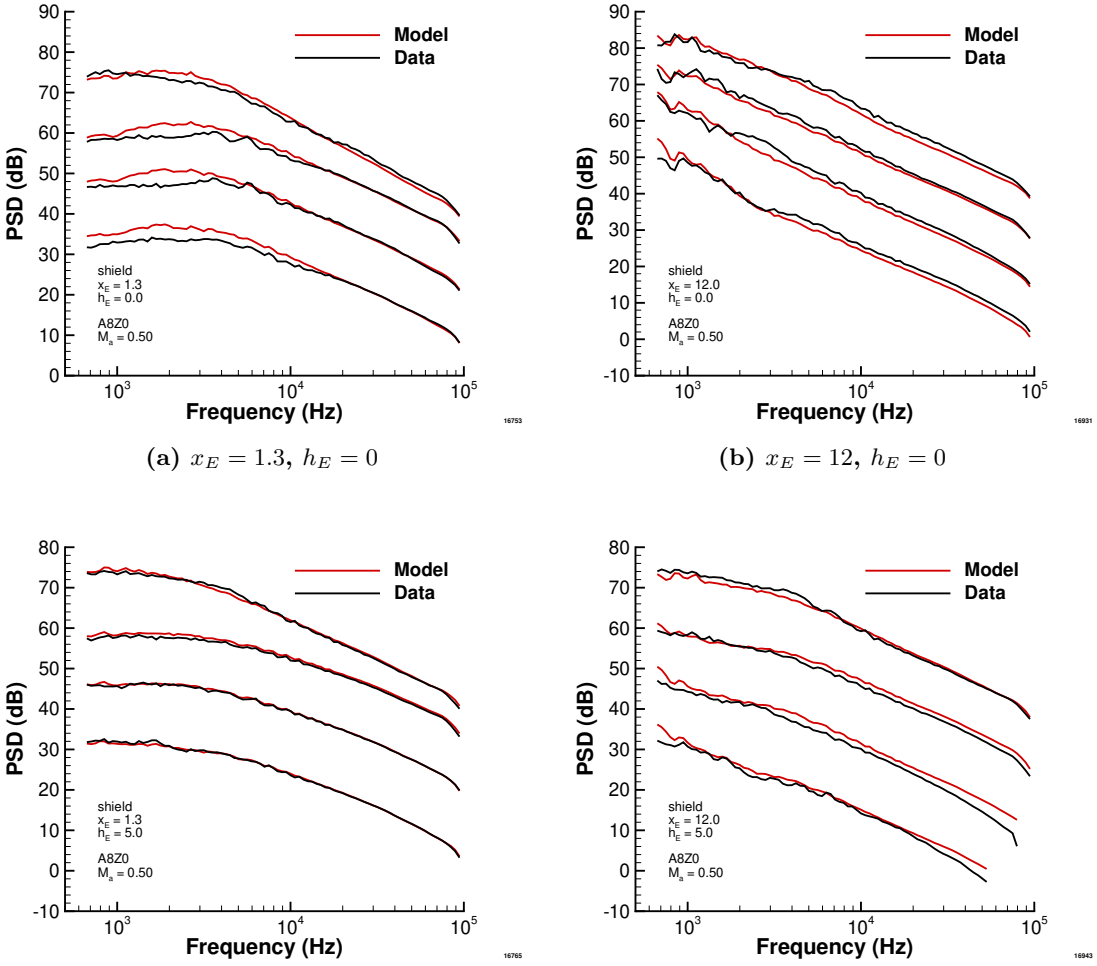
Figure 6: Spectra predicted by the machine learning model compared to the experimental data for the $M_a = 0.5$ 2:1 rectangular nozzle for the $x_E = 1.3, 12$ and $h_E = 0, 5$ surfaces for the shielded observer. The line pairs are observer angles $\Theta = 60°$ (bottom), $\Theta = 90°$, $\Theta = 120°$, and $\Theta = 150°$ (top). Note each spectra is separated by 10 dB from the previous for clarity.



(a) $x_E = 1.3$, $h_E = 0$ **(b)** $x_E = 12$, $h_E = 0$

(c) $x_E = 1.3$, $h_E = 5$ **(d)** $x_E = 12$, $h_E = 5$

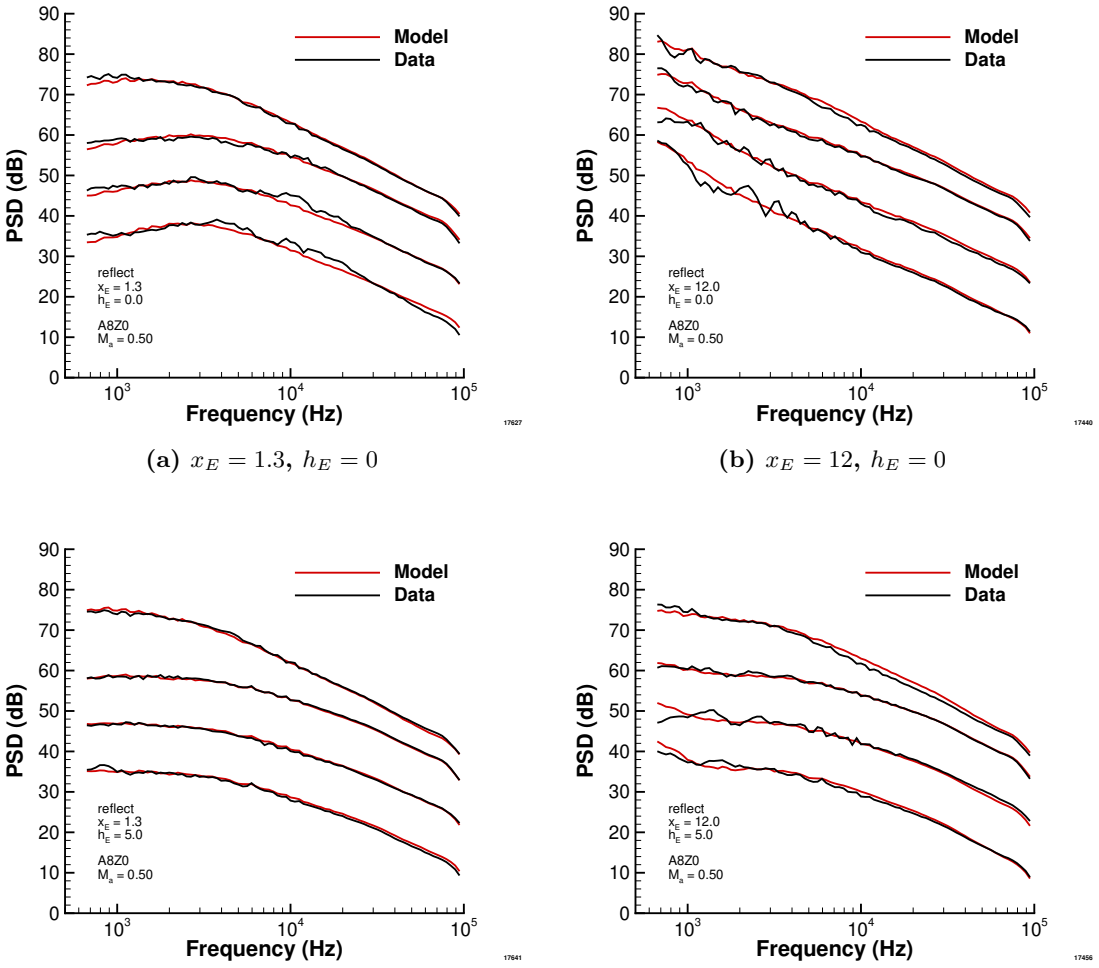
Figure 7: Spectra predicted by the machine learning model compared to the experimental data for the $M_a = 0.5$ 2:1 rectangular nozzle for the $x_E = 1.3, 12$ and $h_E = 0, 5$ surfaces for the reflected observer. The line pairs are observer angles $\Theta = 60°$ (bottom), $\Theta = 90°$, $\Theta = 120°$, and $\Theta = 150°$ (top). Note each spectra is separated by 10 dB from the previous for clarity.



(a) $x_E = 1.3$, $h_E = 0$ **(b)** $x_E = 12$, $h_E = 0$

(c) $x_E = 1.3$, $h_E = 5$ **(d)** $x_E = 12$, $h_E = 5$

Figure 8: Spectra predicted by the machine learning model compared to the experimental data for the $M_a = 0.5$ 8:1 rectangular nozzle for the $x_E = 1.3, 12$ and $h_E = 0, 5$ surfaces for the shielded observer. The line pairs are observer angles $\Theta = 60°$ (bottom), $\Theta = 90°$, $\Theta = 120°$, and $\Theta = 150°$ (top). Note each spectra is separated by 10 dB from the previous for clarity.



(a) $x_E = 1.3$, $h_E = 0$ **(b)** $x_E = 12$, $h_E = 0$

(c) $x_E = 1.3$, $h_E = 5$ **(d)** $x_E = 12$, $h_E = 5$

Figure 9: Spectra predicted by the machine learning model compared to the experimental data for the $M_a = 0.5$ 8:1 rectangular nozzle for the $x_E = 1.3, 12$ and $h_E = 0, 5$ surfaces for the reflected observer. The line pairs are observer angles $\Theta = 60°$ (bottom), $\Theta = 90°$, $\Theta = 120°$, and $\Theta = 150°$ (top). Note each spectra is separated by 10 dB from the previous for clarity.

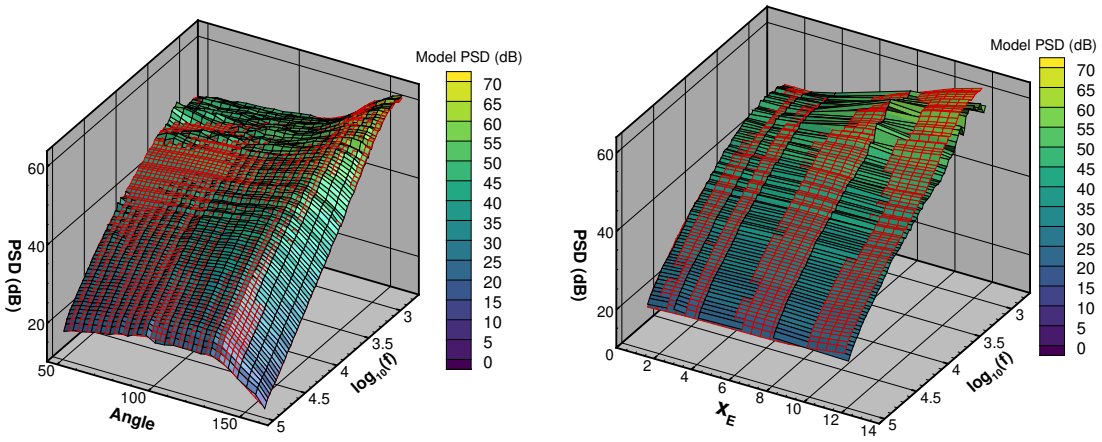
	2:1		8:1	
h_E	Shielded	Reflected	Shielded	Reflected
0	1.027	1.233	1.614	1.344
0.1	0.880	0.850	1.124	1.110
0.2	0.840	0.774	1.205	1.189
0.3	0.807	1.003	1.700	1.900
0.5	0.917	0.841	2.735	1.836
0.7	0.868	0.708	1.487	1.221
1	0.782	0.649	1.486	1.326
1.4	0.623	0.569	1.354	1.182
1.9	0.472	0.462	0.904	1.003
2.5	0.417	0.325	0.697	0.609
3.2	0.407	0.262	0.539	0.430
4	0.376	0.266	0.489	0.350
5	0.360	0.267	0.503	0.310

Table 6: Average mean square errors for each surface standoff (h_E) using the 2:1 and 8:1 aspect ratio nozzles.

shows an example case where the model was used to compute the expected PSD at $x_E = 0.5$ increments. In this case, the model does a good job of smoothly interpolating between points and does not appear that it should be significantly worse than the model evaluated against the training data (again relying on the expectation, based on experience with these data, that the behavior should be fairly smooth between measured x_E). However, as discussed in Section A, the model performance got worse as the surface got longer, particularly for the reflected observer. While the model interpolates between smoothly between $8 \leq x_E \leq 12$, the prediction at at $x_E = 12$ that is not a good representation of the underlying data. Thus, the interpolation aspect of the machine learning model is well behaved but the prediction is not accurate. In general, however, the machine learning models interpolate smoothly between data points which is the desired behavior for this data set. There appear to be no dramatic overshoots or undershoots that might indicate the model is over-fitting the data.

A few attempts were made to extrapolate for different features using the model and the models do not work beyond the range of the training data; this is a known and common problem of machine learning methods. However, it was surprising just how sensitive some models could be: one model returned values that were off by 40 dB given a nozzle area of 3.14 in^2 because it had not seen any training values below 3.14159 in^2.

A second series of models were developed using the PyTorch toolbox. These models explored non-linearities in the data using both machine learning algorithms (Pytorch functions ELU, ReLU, LeakyReLU, and Tanh) and by adding non-linear relations (e.g. x_E^2, M_a^6, etc.). The PyTorch Dropout module was tested at each inner layer (dropout probability at 0.1) to minimize over-fitting issues. The least absolute shrinkage and selection operator (LASSO) was used to find the sparse basis of features used by the neural network and minimize its ability to memorize the dataset. The best implementation used the ReLU and Dropout modules run for approximately 500 epochs (training cycles) and using many feature combinations. However, while these were very insightful from the machine learning point of view, the final model was still limited to the ± 0.5 dB experimental uncertainty so they did not offer a dramatic over the simpler models presented above. This is in part due to the well behaved nature of this dataset that (1) allowed the machine learning techniques to work on a relatively small number of training points while (2) using the more basic machine learning algorithms.



(a) Angle Interpolation ($x_E = 4$) **(b) x_E Interpolation ($\Theta = 90°$)**

Figure 10: Interpolation in angle (left) and x_E (right) from the round nozzle at $M_a = 0.7$. The red mesh represents points that are not in the training or validation data set.

V. Summary

A machine learning approach to jet-surface interaction noise is being studied as a way to quickly generate empirical models for aircraft noise studies. Basic machine learning techniques were used to train models for round and rectangular nozzles to reasonable accuracy relative to the experimental uncertainty in the underlying database. Furthermore, the initial models were generated much faster (a few weeks) than than previous models created using traditional techniques. Additional models were created to investigate the use of more advanced machine learning methods, and although they were not significantly better prediction tools, did provide some insight into over-fitting and non-linearities in the underlying data.

The JSI noise spectra is reasonably well behaved across the range of input parameters and, therefore, was a good candidate for machine learning even though the number of training and validation points was small by machine learning standards. However, this work demonstrates the potential of machine learning to rapidly develop predictive models from experimental data.

Acknowledgements

This work is a collaboration between the Acoustics and Scientific Computing and Visualization groups. The authors would like to thank Dr. James Bridges for supporting and reviewing this work. This work was supported by the Commercial Supersonic Technology Project.

References

[1]Brown, C., "Jet-Surface Interaction Test: Far-Field Noise Results", ASME GT2012-69639, 2012.

[2]Podboy, G., "Jet-Surface Interaction Test: Phased Array Noise Source Localization Results", ASME GT2012-69801, 2012.

[3]Brown, C., "Jet-Surface Interaction Test: Far-Field Noise Results", J. Eng. Gas Turbines Power, **1**35(7), Jun. 2013.

[4]Bridges, J., "Noise from Aft Deck Exhaust Nozzles—Differences in Experimental Embodiments", AIAA 2014-0876, 2014.

[5]Brown, C., "An Empirical Jet-Surface Interaction Noise Model with Temperature and Nozzle Aspect Ratio Effects", AIAA 2015-0229.

[6]Brown, C., "Empirical Models for the Shielding and Reflection of Jet Mixing Noise by a Surface", AIAA 2015-3128, 2015.

[7]Brown, C., "Developing an Empirical Model for Jet-Surface Interaction Noise", AIAA 2014-0878, 2014

[8]Brown, C., Wernet, M.P., "Jet-Surface Interaction Test: Flow Measurement Results", AIAA 2014-3198, 2014.

[9]Brown, C., Bridges, J., "Small Hot Jet Acoustics Rig Validation", NASA/TM 2006-214234.

[10]Dowdall, J., "Applying Machine Learning to Jet Noise Prediction", Acoustics and Urban Air Mobility Technical Working Group Meetings, Oct. 16-18, 2018; Cleveland, OH.