

**GT2017-63403**

# **A COMPARATIVE STUDY OF CONTRASTING MACHINE LEARNING FRAMEWORKS APPLIED TO RANS MODELING OF JETS IN CROSSFLOW**

**Jack Weatheritt\***

**Richard D. Sandberg**

Department of Mechanical Engineering  
 University of Melbourne  
 Parkville, Victoria 3010  
 Australia

Email: jack.weatheritt@unimelb.edu.au

**Julia Ling**

Thermal/Fluid Science and Engineering  
 Sandia National Laboratories  
 Livermore, California, 94551  
 jling@sandia.gov

**Gonzalo Saez**

**Julien Bodart**

ISAE-Supaero  
 Université de Toulouse,  
 Toulouse, 31400  
 France.

julien.bodart@isae-supaero.fr

## **ABSTRACT**

*Classical RANS turbulence models have known deficiencies when applied to jets in crossflow. Identifying the linear Boussinesq stress-strain hypothesis as a major contribution to erroneous prediction, we consider and contrast two machine learning frameworks for turbulence model development. Gene Expression Programming, an evolutionary algorithm that employs a survival of the fittest analogy, and a Deep Neural Network, based on neurological processing, add non-linear terms to the stress-strain relationship. The results are Explicit Algebraic Stress Model-like closures. High fidelity data from an inline jet in crossflow study is used to regress new closures. These models are then tested on a skewed jet to ascertain their predictive efficacy. For both methodologies, a vast improvement over the linear relationship is observed.*

## **NOMENCLATURE**

$a_{ij}$  Anisotropy tensor.  
 $\alpha_{ij}^x$  Extra anisotropy tensor.  
 $d$  Pipe diameter.  
 $h$  Channel height.  
 $k$  Turbulent kinetic energy.  
 $I_k$  Scalar invariants of anisotropy tensor.  
 $r$  Velocity ratio:  $U_j/U_\infty$

$J$  Momentum flux ratio:  $\rho_j U_j^2 / \rho_\infty U_\infty^2$   
 $S_{ij}$  Strain rate tensor.  
 $s_{ij}$  Non-dimensional strain rate tensor.  
 $V_{ij}^k$  Basis for anisotropy tensor.  
 $\delta$  Incoming boundary layer thickness.  
 $U$  Reynolds-averaged velocity  
 $\tau_{ij}$  Reynolds stress.  
 $t_l$  RANS turbulent timescale.  
 $\theta$  Angle of pipe and bulk velocity direction, in the wall parallel plane.  
 $\phi$  Angle of pipe and bulk velocity direction, in the wall normal plane.  
 $\rho$  Fluid density.  
 $p$  Pressure.  
 $\Omega_{ij}$  Rotation rate tensor.  
 $\omega_{ij}$  Non-dimensional rotation rate tensor.  
 $\omega$  Specific dissipation rate.

## **SUBSCRIPTS**

$j$  Jet quantity (when Einstein summation is not implied).  
 $\infty$  Bulk quantity.

---

\*Address all correspondence to this author.

## INTRODUCTION

Accurate prediction of the jet in crossflow (JiC), the ejection of secondary flow through an orifice into traveling fluid [1], is vital to the efficiency and lifespan of gas turbine components. In the pursuit of said efficiency, the inlet temperatures of turbine rotors exceed the melting point of the blade material. To circumvent this, JiCs are used to film cool the surface. At operating conditions, marginal decreases in surface temperature can result in large extensions to the life expectancy of turbine components. Such parts are extremely expensive to replace (tens of thousands of dollars), where a reasonable fraction of the cost is spent on the film cooling holes themselves [2].

For practical calculations, Direct and Large Eddy simulations (DNS/LES) are unfeasible due to intractable computational requirements. Therefore, primarily the industry is limited to the use of Reynolds-Averaged Navier-Stokes (RANS) tools. Whilst being computationally affordable and numerically robust, RANS has known theoretical deficiencies [3, 4, for example]. From a practical perspective, the JiC is particularly troublesome. For the velocity ratios  $r$  indicative of film cooling applications, the dominant highly three-dimensional structures manifest as horse-shoe/necklace vortices on the windward side of the orifice that wrap around a pair of counter-rotating vortices [1]. A RANS model must accommodate these structures (in a mean sense) for accurate predictions.

In reality, the jet penetration distance is overpredicted by models based on linear and non-linear stress-strain relationships [5, 6] and second moment closures [7]. Further, the turbulent mixing is poorly represented [8], which results in slow spreading of the jet [6]. Unsteady RANS calculations have also been performed [9], which better captured the aforementioned structures, yet incorrectly predicted the Reynolds stress anisotropy.

The above paragraph highlights model form uncertainty, particularly attributed to the linear Boussinesq approximation. There have been several efforts to reduce uncertainty for JiC applications, many with the assistance of high fidelity numerical or experimental data. A turbulence model that neglected Reynolds stress transport terms was derived to achieve a higher lateral turbulence diffusivity to boost the spreading rate [10]. For this model, the spanwise eddy viscosity was curve fit to experimental data. Simulations of suction side cooling [11, 12] show that enforcing bounds on the time scale appearing in the eddy viscosity, thus limiting turbulent kinetic energy production in regions of strong strain, improve the heat transfer prediction made by RANS. Further, these simulations showed that using the  $v^2$ - $f$  turbulence model [13], which induces anisotropy by damping the wall normal stress, improves predictions for moderate blowing ratios. Specific damping functions have been formulated by considering DNS data in an *a priori* manner [14], which reduced the error in the eddy viscosity.

The model uncertainty attributed to the linear stress-strain

relationship also impacts on heat transfer prediction. Scalar flux models have therefore been considered *a priori*, via the use of LES data, which showed that the non-linear diffusion models drastically improved predictions [15, 16].

These studies showcase a variety of possibilities for model improvement via *a priori* reference data considerations. Consequently, in recent times there has been a growing trend towards complex machine learning tools that utilize data to produce models with minimal constraints and assumptions. These data-driven approaches are still relatively dormant in the turbulence modeling community, yet some applications have newly emerged [17–20, for example].

In this paper, we compare and contrast two such methodologies: Gene Expression Programming [21] (GEP) and Deep Neural Networks [22] (DNN). The former applies an evolutionary analogy to ‘evolve’ non-linear mathematical expressions whilst the latter is loosely based on neurological processing, producing non-linearity by optimizing edge weightings along a graph. Both approaches are applied to reducing the model form uncertainty of the Reynolds stress and have the ultimate goal of being deployed in existing codes predictively. The GEP is trained on the LES velocity field to learn the mapping between the LES mean velocity gradients and the stress anisotropy. The aim is to learn the physics of the problem and improve on the standard Boussinesq approximation used in linear turbulence models. The DNN is trained on the RANS velocity field to predict the LES anisotropy. When deployed it therefore requires first converging the default RANS model, then iteratively re-converging using the DNN Reynolds stress predictions. These two approaches reflect two different philosophies on how to develop a robust, stable turbulence model using data-driven closures. These different approaches reflect the current aims of the users, not limitations in the methodologies, and as such we also discuss the impact on the methodology and resulting output.

In both machine learning frameworks the objective is a non-linear stress-strain relationship, of the Explicit Algebraic (Reynolds) Stress Model (EASM) class [23, 24], that is better suited to JiC applications. To this end, an inline JiC (IJC) is used to train new Reynolds stress closures using the two frameworks and a skewed JiC (SJiC) is used to test the predictive capability of these new models. This represents a typical machine learning application — first build a model on a related (yet different) flow problem, then test on a new flow case. In this instance, the SJiC represents an extrapolation in physics learnt in the IJC because of the broken symmetry. This provides a challenging test for machine learnt Reynolds stress models.

The paper is structured as follows. In the next section we briefly discuss the Reynolds stress-strain relationship, in particular EASM-like non-linear algebraic extensions possible and the target for optimization in both frameworks. Then the IJC and SJiC flow cases are described. Next, the two optimization philosophies are discussed, in reference to the data prepara-

tion, followed by introductions to the GEP and DNN techniques. Once the frameworks are discussed, we show results and follow with discussions and conclusions.

## THE RANS STRESS-STRAIN RELATIONSHIP AND MACHINE LEARNING OBJECTIVES

Typically a RANS model closes the stress-strain relationship by invoking Boussinesq's linear hypothesis,

$$a_{ij} \equiv \frac{\tau_{ij}}{2k} - \frac{1}{3}\delta_{ij} \quad (1a)$$

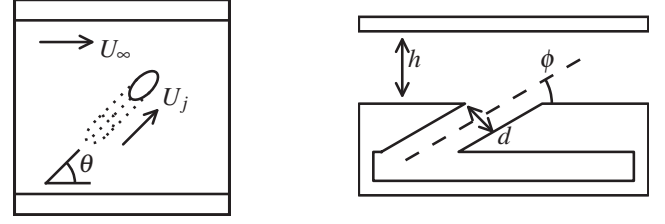
$$= -t_I S_{ij} \quad (\text{linear}) \quad (1b)$$

$$= a_{ij}(V_{ij}^1, V_{ij}^2, \dots, I_1, I_2, \dots) \quad (\text{EASM, GEP}) \quad (1c)$$

$$= a_{ij}(V_{ij}^{1*}, V_{ij}^{2*}, \dots, I_1^*, I_2^*, \dots), \quad (\text{EASM, DNN}) \quad (1d)$$

where the non-dimensional anisotropy  $a_{ij}$  is related to the mean strain rate tensor  $S_{ij}$  via the turbulent timescale coefficient  $t_I$ .  $k = \frac{1}{2}\tau_{kk}$  is the turbulent kinetic energy. Note, the \* in Eq. 1d denotes variables from a RANS solution — this is discussed below. It is precisely Eq. 1b that the machine learning frameworks are trying to improve upon. A class of turbulence models, known as EASMs, begin from the weak equilibrium hypothesis [24] to write down non-linear terms in  $\partial_{x_j} U_i$  for the anisotropy tensor. The  $V_{ij}^k$  and  $I_k$  of Eq. 1c and Eq. 1d are a basis and set of scalar invariants, found using the Cayley-Hamilton theorem [23]. They are non-dimensional functions of the strain and rotation rate  $\Omega_{ij}$  tensors. By defining  $s_{ij} = t_I S_{ij}$  and  $\omega_{ij} = t_I \Omega_{ij}$ , we can write  $V_{ij}^k$  and  $I_k$  as

$$\begin{aligned} V_{ij}^1 &= s_{ij}, & V_{ij}^2 &= s_{ik}\omega_{kj} - \omega_{ik}s_{kj}, \\ V_{ij}^3 &= s_{ik}s_{kj} - \frac{1}{3}\delta_{ij}s_{mn}s_{nm}, \\ V_{ij}^4 &= \omega_{ik}\omega_{kj} - \frac{1}{3}\delta_{ij}\omega_{mn}\omega_{nm}, \\ V_{ij}^5 &= \omega_{ik}s_{kl}s_{lj} - s_{ik}s_{kl}\omega_{lj}, \\ V_{ij}^6 &= \omega_{ik}\omega_{kl}s_{lj} + s_{ik}\omega_{kl}\omega_{lj} - \frac{2}{3}\delta_{ij}s_{kl}\omega_{lm}\omega_{mk}, \\ V_{ij}^7 &= \omega_{ik}s_{km}\omega_{mn}\omega_{nj} - \omega_{ik}\omega_{km}s_{mn}\omega_{nj}, \\ V_{ij}^8 &= s_{ik}\omega_{km}s_{mn}s_{nj} - s_{ik}s_{km}\omega_{mn}s_{nj}, \\ V_{ij}^9 &= \omega_{ik}\omega_{km}s_{mn}s_{nj} + \\ &\quad s_{ik}s_{km}\omega_{mn}\omega_{nj} - \frac{2}{3}\delta_{ij}s_{km}s_{mn}\omega_{np}\omega_{pk}, \\ V_{ij}^{10} &= \omega_{ik}s_{km}s_{mn}\omega_{np}\omega_{pj} - \omega_{ik}\omega_{km}s_{mn}s_{np}\omega_{pj}, \\ I_1 &= s_{mn}s_{nm}, & I_2 &= \omega_{mn}\omega_{nm}, \\ I_3 &= s_{km}s_{mn}s_{nk}, \\ I_4 &= \omega_{km}\omega_{mn}s_{nk}, \\ I_5 &= \omega_{km}\omega_{mn}s_{np}s_{pk}. \end{aligned} \quad (2)$$



**FIGURE 1.** TOP AND SIDE VIEW OF JET IN CROSSFLOW SCHEMATIC, HIGHLIGHTING RELEVANT PARAMETERS.

It is thus the objective of the machine learning frameworks to construct new expressions from Eq. 2 for the Reynolds stress anisotropy; the GEP process targets Eq. 1c and the DNN targets Eq. 1d. The main difference in the current work is that the GEP uses the LES data as inputs and the DNN takes RANS data as inputs. The DNN uses the RANS data as it is designed to be deployable on flows for which high fidelity data is not available. This is detailed in the Data Preparation Philosophy section below. Note, the GEP methodology takes the LES data, because the overall aim is a turbulence model based on the observed physics. Whilst in this work the models are tested using the LES data, other studies [19, 25] have successfully used the GEP models exactly as any other EASM turbulence model. This is because the GEP framework returns an algebraic expression that is easily inserted into CFD codes. The previous studies added simple non-linear terms to the  $k$ - $\omega$ -SST model to improve internal [25] and separated [19] flow prediction using OpenFOAM. In this study, we do not test the models *a posteriori* converging from initial conditions, however prepare the LES data to mimic RANS quantities. This is detailed in the Machine Learning Methodology section below. Note in the previous studies, models that better reproduced anisotropy on high fidelity data were better as predictive RANS models [19].

## FLOW CONFIGURATIONS

Two flow cases are required to test the efficacy of the machine learning frameworks. The first, a training case, is used to actually build the models and second, a test case, is used to validate the learnt closures. The following briefly outlines these two cases.

### Training: Inline Jet In Crossflow

The training case, an incompressible JiC, is schematically shown in Fig. 1. The relevant flow parameters are detailed in Table 2. The jet was at a  $30^\circ$  angle to mimic a film cooling application and injected into a channel of height  $h$ . Note, the plenum

**TABLE 1.** SUMMARY OF JiC PARAMETERS.

	$\phi$	$\theta$	$r$	$J$	$Re_j$	$\delta/d$	$Re_h$
IJiC	30°	0°	1	1	5,400	1	45,800
SJiC	30°	30°	1	1	5,800	1.9	50,000

was included in order to capture the interaction of the boundary layer with the vorticity inside the pipe [26]. The physical setup replicates a hydrodynamic experiment [27], with the exception of the molecular diffusivity (water versus air) — but this is deemed negligible with respect to the turbulent diffusion.

**LES details.** The LES was performed using the massively parallel code *CharLES*<sup>x</sup> [28], with the Vremen subgrid-scale model [29] and synthetic turbulence [30] added at  $-40d$  with the outflow at  $35d$  (the orifice was placed at the origin). The spanwise extent was  $10d$ . The finite-volume nature of the solver allows for unstructured cells in regions not vital for accuracy and consequently the total cell count was 52 million. For more information see Ref [31].

**RANS details.** The RANS was performed with the standard  $k-\epsilon$  turbulence model, implemented in FLUENT (v. 13.0). The grid was structured, consisting of 2.5 million cells. Note, the streamwise extent was  $(-15d, 35d)$  For more information see Ref. [27].

### Testing: Skewed Jet In Crossflow

The testing case, an incompressible JiC is schematically shown in Fig. 1 and summarized with the parameters in Table 2. This flow is similar to the IJiC described above, except that the angle  $\theta$  made with the  $x$  axis and pipe center line in the  $y-z$  plane is non-zero. Because of the non-zero spanwise jet velocity, the channel is wider.

**LES details.** The same numerical method and streamwise extent was used as described above. The spanwise extent was  $18d$ . The grid consisted of 101 million cells. For more information see Ref. [15].

**RANS details.** The RANS was performed using the realizable  $k-\epsilon$  model implemented in FLUENT (v. 13.0). The grid was the same as that used for the LES.

## DATA PREPARATION PHILOSOPHY

For the inputs to the optimization algorithms, described in the next section, two philosophies are tested. These differ in the nature of produced model. It should be reiterated that the data preparation steps are not machine learning algorithm dependent, but reflect current directions being explored in the literature. We provide a description of both, such that the reader may understand the purpose of such a step. Note, both make use of the training (IJiC) and testing (SJiC) cases described above to optimize models and then test their efficacy respectively.

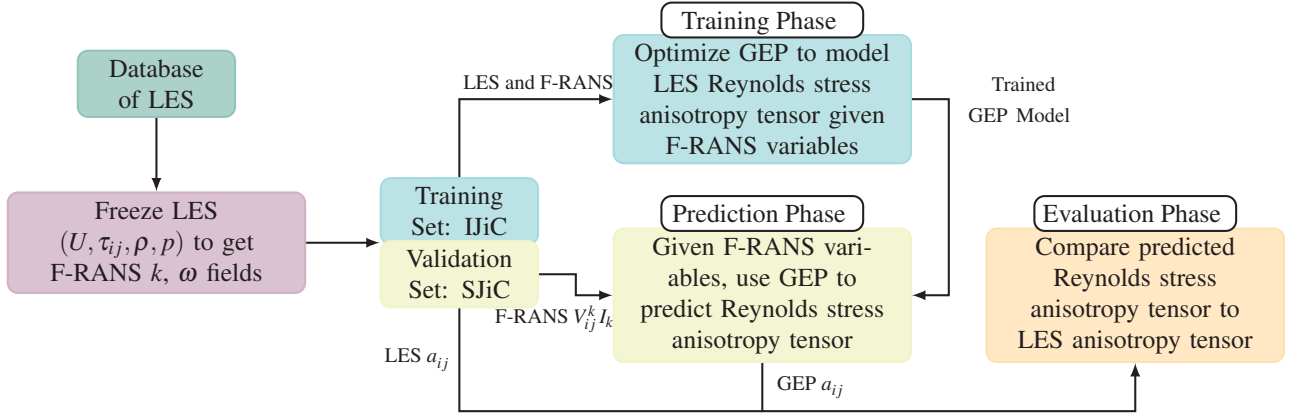
### GEP Data Preparation

When regressing a model for the anisotropy, of the form Eq. 1c — which is designed as an extension to the  $k-\omega$ -SST turbulence model [32] — one requires  $V_{ij}^k$  and  $I_k$  from the high fidelity database to make expressions for  $a_{ij}$ . The velocity gradients appearing in Eq. 2 are calculated readily enough from the LES mean flow field, however the non-dimensionalization through the coefficient  $t_l$  is more problematic. It is not necessarily trustworthy to calculate the time scale from LES, as the RANS transport equations contain many modeled terms and consequently expect something very different from LES quantities. Further,  $k$  and  $\omega$  from a companion RANS would introduce an error, because the velocity fields are different, thereby changing the Reynolds stress. If the global flow features are incorrectly captured by the RANS and the flow fields for  $a_{ij}$  and  $t_l$  differ significantly, then produced models will be optimized including this inconsistency and future predictions, when using the trained model in a CFD code, are unlikely to give satisfactory results (certainly on vastly differing flow problems).

To mitigate these issues, we chose an approach in which we ‘freeze’ the mean LES flow field  $(U, \rho, p, \tau_{ij})$  and solve the  $\omega$  transport equation from the  $k-\omega$ -SST turbulence model. The Frozen RANS (F-RANS) fields are then the ‘correct’ RANS solution for a given LES field. Note, other frameworks [33] have been implemented to circumvent this problem at optimization time, however the F-RANS pre-processing approach [34] has previously shown good results in *a posteriori* CFD [25]. This data preparation step is shown in Fig. 2, as part of the overall GEP framework. Note, in order to test the models, we also apply the same data preparation to the SJiC case and apply the learnt anisotropy model. This is quicker than full *a posteriori* calculations using the new non-linear  $k-\omega$ -SST turbulence model.

### DNN Data Preparation

Instead of using the frozen RANS approach as above, the DNN used the RANS results directly, so the F-RANS calculation is not used. The idea behind the DNN approach is that if you have a fully converged RANS solution to the flow, that can be used as the starting point for the DNN. The RANS mean velocity field is fed into the DNN, which then predicts a new Reynolds



**FIGURE 2.** GENE EXPRESSION PROGRAMMING WORK FLOW WITH DATA PREPARATION.

stress tensor field. This updated Reynolds stress tensor can be fed back into the RANS solver to allow iterative re-convergence, now using the DNN closure model for the Reynolds stresses. At this stage, this iterative re-convergence has not yet been demonstrated using the DNN closure. However, previous work [20] has shown that for two different flow configurations, if the DNN Reynolds stresses are applied as a fixed field (instead of iteratively) and the RANS mean velocity field is re-converged using the DNN Reynolds stresses, then improved accuracy in the mean velocity field is recovered. There are continuing efforts to develop the iterative convergence framework for forward propagation.

This data preparation step is shown schematically below, in Fig. 4, as part of the DNN methodology.

## MACHINE LEARNING METHODOLOGY

### Evolutionary Framework

With a given set of  $N$  data points  $x^\alpha = (V_{ij}^k, I_k, a_{ij})^\alpha$ ,  $\alpha = (1, \dots, N)$ , GEP *symbolically* regresses an algebraic expression of the form Eq. 1c. That is, the optimized model is a tangible mathematical equation in  $V_{ij}^k$  and  $I_k$ , of a functional form not specified by the user. Evolutionary algorithms, which employ a ‘survival of the fittest’ analogy, brought symbolic regression into the computer domain [35]. For example, an algebraic expression of the form described by Eq. 1c,

$$a_{ij} = (V_{ij}^8 + I_3 \exp(-I_2) V_{ij}^1) - V_{ij}^2 \quad (3)$$

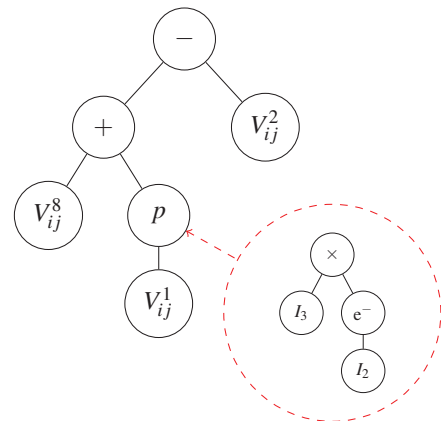
is displayed in Expression Tree (ET) form in Fig. 3. In the GEP algorithm, ETs are constructed from linear strings, known as chromosomes. In the current example, the following chromosomes produce the required ET,

```

--+V8pV1|V2V3V2V4V1V9
*I3E|I2I1I2I4

```

The chromosomes are read left to right, with each representing a tree in Fig. 3. The  $|$  denotes a split between head and tail of the chromosome and the  $E$  denotes the reciprocal exponential. The head may consist of operators and variables and the tail only of variables. This distinction guarantees mathematical syntax — see Ref. [21] for more details. The ETs are read recursively in order to reconstruct Eq. 3. The  $p$  symbol, known as a plasmid [19], is a symbiotic join between the two chromosomes. The first is a tensor expression and the second is a scalar field. The plasmid has made it possible to regress variables that are not all of the same dimension. For a full introduction to the algorithm see Ref. [19]. Figure 2 describes the overall framework where, once the training data has been prepared, the GEP algorithm begins during the training phase. Initially, a population of chromosomes is randomly created. These are then non-deterministically evolved according to survival of the fittest. During each succes-



**FIGURE 3.** EXAMPLE ET.



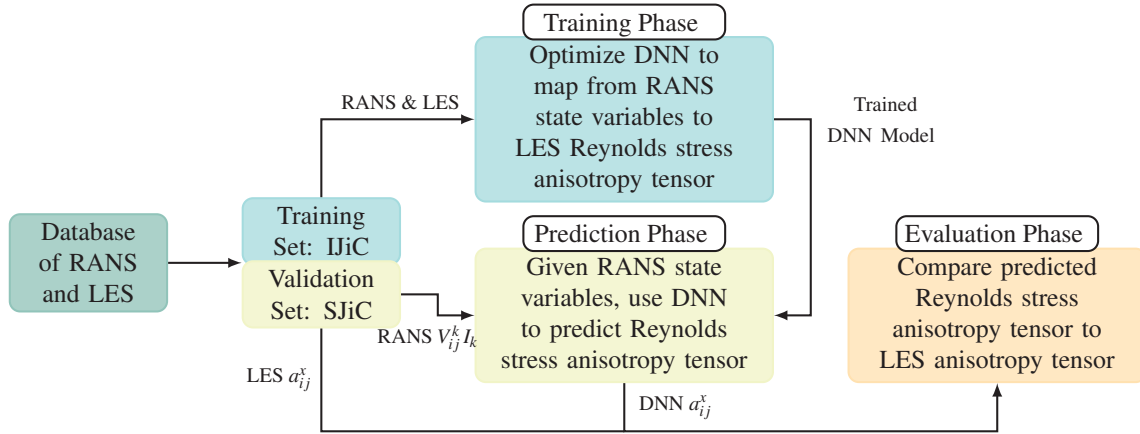


FIGURE 4. DEEP NEURAL NETWORK WORK FLOW WITH DATA PREPARATION.

sive generation, expressions that better approximate  $a_{ij}$  for the IJiC data are more likely to mate and provide genetic material to the next generation. Mutations are mimicked by randomly introducing error when copying the chromosome into the next generation. For a full description of such genetic operators see Refs. [21, 35]. Once a stopping criteria has been met, usually after a finite number of generations, the best expression from the population is taken as the output.

The fitness of an expression is calculated as a relative distance from the training values,

$$\text{Fit}(a_{ij}^{\text{gep}}) = 1 - \sum_{\alpha=1}^N \frac{|a_{ij}^{\text{gep}}(x^\alpha) - a_{ij}(x^\alpha)|}{|a_{ij}(x^\alpha)|}. \quad (4)$$

$|\cdot|$  is the  $\ell^{2,2}$  norm and the summation is over the  $N$  training data points  $x^\alpha$ . This defines a maximal fitness of 1 and the model closest is passed onto the prediction phase in Fig. 2. Note, in reality we perform the training phase multiple times and we take an ensemble of predictions. This is to overcome overfitting the training case and is made possible by the randomness inherent in the algorithm. For this particular case, the training of a single algebraic expression is of the order of core minutes.

### Deep Neural Network Framework

Neural networks are a type of machine learning algorithm that allow for complex, hierarchical feature interactions. Each neuron in a neural network represents a non-linear mapping. Layers of neurons feed into each other to generate the hierarchical interactions. Many customized network architectures, with different size layers and different connection pathways between layers, have been proposed for different learning tasks.

The output of each neuron in the neural network is  $f(w^T x)$ , where  $f(x)$  is the activation function,  $w$  is the weight vector, and

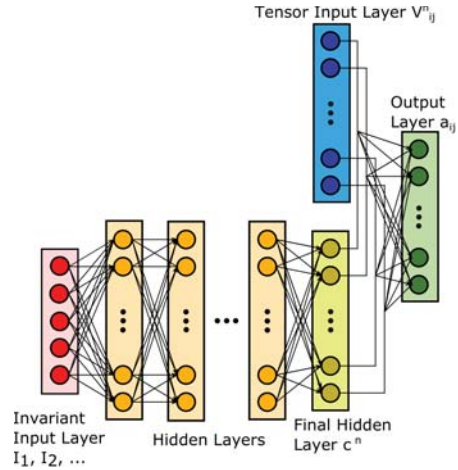


FIGURE 5. NEURAL NETWORK ARCHITECTURE

$x$  is the input vector to the neuron. Typical activation functions include sigmoids, tanh, and rectified linear units (ReLU). The leaky ReLU function ( $f(x) = \max(\epsilon x, x)$  for small pre-specified slope  $\epsilon < 1$ ) was used in this study because of its previously demonstrated ability to allow the training of deep neural networks [36]. The network architecture used in this project is shown in Fig. 5, and has been previously described in detail in Ref. [20]. This architecture was developed to explicitly embed Eq. 1d to guarantee Galilean invariance. It has an input layer that takes in the scalar invariants  $I_1, I_2, \dots$  and non-linearly transforms them through a series of hidden layers to obtain the coefficients  $c^n$  to the tensor basis. These coefficients are then multiplied by the tensor basis  $V_{ij}$  to obtain the final prediction of the Reynolds stress anisotropy tensor  $a_{ij}$ . This specialized architecture has been shown [20] to provide more accurate predictions than a generic architecture that does not embed Galilean invari-

ance.

The weight vectors  $w$  of each neuron in the hidden layers were learned during the training phase through a process called *back-propagation*, in which the sensitivity of the mean squared error objective function to each weight was propagated from the output layer back through the nodes of the network. These sensitivities were used to iteratively update each weight vector. The network was trained on the IJiC case to learn a mapping from the RANS velocity gradient tensor to the LES Reynolds stress anisotropy tensor. Predictions were then made on the SJiC case based on the RANS results for that flow field. This work flow is shown schematically in Fig. 4.

## RESULTS

The trained GEP and DNN anisotropy models, in the evaluation phase (see Figs. 2 and 4), are applied to the SJiC. For the GEP this is using the F-RANS solution and for the DNN the companion RANS solution. The resulting anisotropy tensor components, Eq. 1a, are plotted in Fig. 6 alongside the LES. Also included is the companion RANS solution as a visual guide to the improvement possible. The components of particular interest include the anisotropy of the leeward side of the jet near the orifice, the development downstream and the near-wall behavior.

Both machine learning methods vastly improve on the linear  $k$ - $\epsilon$  model solution. By a comparison of the anisotropy plots in the first (LES) and last (RANS) columns of Fig. 6, we can see that no component is satisfactorily computed when using Eq. 1b. This is because, apart from at the orifice itself, the streamwise velocity gradient component  $\partial_x U_1$  is small (for the LES, but especially for the RANS). Then, by continuity the statement

$$a_{22} \approx -a_{33} \quad (5)$$

holds for the linear model. This unphysical relationship between wall normal and spanwise Reynolds stress seriously limits the prediction.

In contrast, the machine learning algorithms have used terms from Eq. 2, that provide more spatially varying components. These redistribute the Reynolds stress components using higher order velocity gradient terms that improve the prediction of  $a_{11}$  (see Figs. 6(b)-(c)). Consequently, the remaining normal stress components are not dictated by Eq. 5. This is true, even for the DNN — which uses the RANS velocity gradient tensor to make predictions.

Another major issue with Eq. 1b, used in the RANS calculation, is the vanishing near-wall anisotropy of the diagonal components. This is because the off-diagonal velocity gradients are not redistributed using  $V_{ij}^1$  alone. We can see that both machine learning frameworks have successfully utilized  $V_{ij}^2$  and  $V_{ij}^3$ , which

TABLE 2. SUMMARY OF RMSE.

GEP	F-RANS	DNN	RANS
0.078	0.141	0.093	0.152

provide variance near the wall without side effects in other parts of the flow.

Note, other components — such as  $V_{ij}^4$ , the anisotropy of the rotation matrix squared — can provide near-wall improvements, but at the price of strong anisotropy on the leeward side of the orifice. A small region, approximately  $1 \leq x/d \leq 3$  depending on the methodology, has high vorticity which means that usage of  $V_{ij}^k$  components that are strong functions of  $\omega_{ij}$  have caused errors. Hence the region of inaccuracy at  $x/d = 2$  for the GEP and DNN — seen most predominantly in the  $a_{33}$  component. This is the region that is the biggest extrapolation from the training case; the higher levels of streamwise vorticity, a result of the non-zero  $\theta$  in the SJiC case, are not observed in the IJiC, which is symmetric about  $z$ . That said, this region is relatively localized and therefore encouraging.

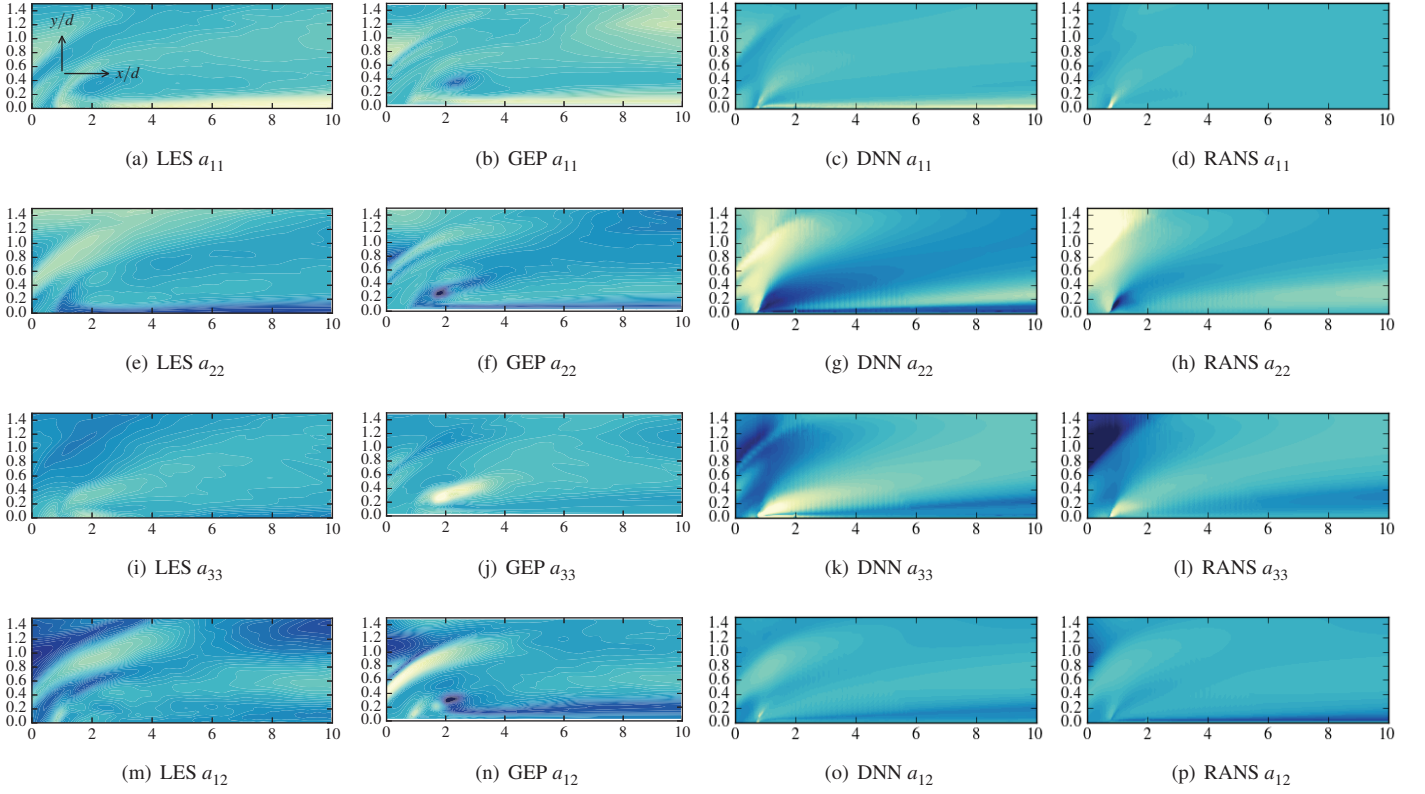
This sensitivity to streamwise vorticity, something previously unseen during training, whilst local, is a potential cause for concern should the models be used for *a posteriori* prediction for large  $\theta$ . In order to alleviate this, as part of a CFD code, the machine learnt models could be applied with limiters.

With respect to machine learning differences, these are seen predominantly as a result of data preparation and the philosophy of usage. The DNN uses the RANS velocity gradients, which contains an incorrect penetration height and spreading rate, so larger differences may be expected visually.

This difference in framework philosophy means that the machine learning algorithms cannot be absolutely compared for performance. One way of providing a tentative indication is via an error metric of a modeled tensor  $a_{ij}^{\text{mod}}$  and compare to their respective baseline anisotropy. For the GEP and DNN models, these are the F-RANS and RANS solutions respectively. We define the root mean square error (RMSE) of the anisotropy tensor,

$$\text{RMSE}(a_{ij}^{\text{mod}}) = \sqrt{\frac{1}{6N} \sum_{\alpha=1}^N \sum_{i=1}^3 \sum_{j=1}^i (a_{ij} - a_{ij}^{\text{mod}})^2} \quad (6)$$

taking care to include only the off-diagonal components once. The RMSE for all models is reported in table 2. This is a 44.7% decrease over the F-RANS and a 38.8% decrease over the RANS errors for the GEP and DNN respectively. These values quantify the excellent reduction in error possible via both methodologies.



**FIGURE 6.** CONTOURS ALONG THE CENTRELINE. TOP TO BOTTOM:  $a_{11}$ ,  $a_{22}$ ,  $a_{33}$ ,  $a_{12}$ . LEFT TO RIGHT: LES, GEP, DNN, RANS.

## CONCLUSIONS

In this paper, we have presented two machine learning methodologies applied for improved Reynolds stress anisotropy predictions for JiCs. GEP, an evolutionary algorithm, was applied in an *a priori* sense and a DNN was used for correcting a linear RANS solution post simulation. These applications present two possible ways of applying machine learning for turbulence modeling. Resulting models could now be tentatively plugged into CFD codes. We began by building a model for the anisotropy using an IJiC, then extrapolated the learnt physics to an SJiC. Whilst predictions were greatly improved over a linear baseline calculation, in the region of strongest streamwise vorticity — the biggest extrapolation of physics — both models were found to return incorrect levels of anisotropy. This level of extrapolation should be further explored. With this information, limiters may be incorporated in order to provide a more trustworthy and robust model.

With regards to extensions of this study, it would be interesting to test new models in CFD codes. This would be achieved by inserting the GEP model into Eq. 1c and the DNN model into Eq. 1d. However, the code is likely to be unstable because of the high order velocity gradients. With the knowledge that  $V_{ij}^2$  and

$V_{ij}^3$  do not contribute to the model extrapolation error, it would be sensible to regress new closures including only these basis functions alongside  $V_{ij}^1$ . This would result in a quadratic model, that would not contain the local inaccuracy on the leeward side of the orifice. Further, as briefly mentioned, limiters may be required on basis coefficients. These could be based on simple min/max functions that limit the inputs to values seen in the training data.

In short, the incorporation of such anisotropy models is not yet a trivial matter, yet the potential predictive capability of machine learnt turbulence closures has been demonstrated for these canonical, but difficult, film cooling applications. Via these machine learning algorithms, more generally, this study has demonstrated the significant potential of data-driven modeling frameworks to provide improved turbulence closures.

## ACKNOWLEDGMENT

The University of Melbourne acknowledges the funding of a veski fellowship. Funding for J. Ling's work was provided by the Sandia LDRD program. Sandia National Laboratories is 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



## REFERENCES

- [1] Mahesh, K., 2013. "The interaction of jets with crossflow". *Annual Review of Fluid Mechanics*, **45**, pp. 379–407.
- [2] Bogard, D., and Thole, K., 2006. "Gas turbine film cooling". *Journal of Propulsion and Power*, **22**(2), pp. 249–270.
- [3] Hunt, J., and Savill, A., 2005. "Guidelines and criteria for the use of turbulence models in complex flows". *Prediction of Turbulent Flows*, pp. 291–343.
- [4] Leschziner, M., 2015. *Statistical turbulence modelling for fluid dynamics — demystified: an introductory text for graduate engineering students*. World Scientific.
- [5] Ajersch, P., Zhou, J.-M., Ketler, S., Salcudean, M., and Gartshore, I. S., 1995. "Multiple jets in a crossflow: detailed measurements and numerical simulations". In ASME 1995 International Gas Turbine and Aeroengine Congress and Exposition, American Society of Mechanical Engineers, p. V004T09A009.
- [6] Hoda, A., and Acharya, S., 2000. "Predictions of a film coolant jet in crossflow with different turbulence models". *Journal of Turbomachinery*, **122**(3), pp. 558–569.
- [7] Demuren, A., 1993. "Characteristics of three-dimensional turbulent jets in crossflow". *International journal of engineering science*, **31**(6), pp. 899–913.
- [8] Demuren, A., Rodi, W., and Schonung, B., 1986. "Systematic study of film cooling with a three-dimensional calculation procedure". *Journal of turbomachinery*, **108**(1), pp. 124–130.
- [9] Hoda, A., and Acharya, S., 2012. "Unsteady rans simulation of film cooling jets in a cross-flow with an improved k-tau model". In ASME 2012 Gas Turbine India Conference, American Society of Mechanical Engineers, pp. 433–441.
- [10] Bergeles, G., Gosman, A., and Launder, B., 1978. "The turbulent jet in a cross stream at low injection rates: a three-dimensional numerical treatment". *Numerical Heat Transfer, Part A: Applications*, **1**(2), pp. 217–242.
- [11] Medic, G., and Durbin, P. A., 2002. "Toward improved prediction of heat transfer on turbine blades". *Journal of turbomachinery*, **124**(2), pp. 187–192.
- [12] Medic, G., and Durbin, P. A., 2002. "Toward improved film cooling prediction". *Journal of turbomachinery*, **124**(2), pp. 193–199.
- [13] Durbin, P., 1996. "On the  $k$ - $\epsilon$  stagnation point anomaly". *International Journal of Heat and Fluid Flow*, **17**(1), pp. 89–90.
- [14] Muldoon, F., and Acharya, S., 2006. "Analysis of  $k$  and  $\epsilon$  budgets for film cooling using direct numerical simulation". *AIAA Journal*, **44**(12), pp. 3010–3021.
- [15] Ryan, K. J., Bodart, J., Folkersma, M., Elkins, C. J., and Eaton, J. K., 2015. "Turbulent scalar mixing in a skewed jet in crossflow: experiments and modeling". In International Symposium on Turbulence and Shear Flow Phenomena.
- [16] Ling, J., Ryan, K. J., Bodart, J., and Eaton, J. K., 2016. "Analysis of turbulent scalar flux models for a discrete hole film cooling flow". *Journal of Turbomachinery*, **138**(1), p. 011006.
- [17] Weatheritt, J., and Sandberg, R., 2015. "A new reynolds stress damping function for hybrid RANS/LES with an evolved functional form". *Advances in Computation, Modeling and Control of Transitional and Turbulent Flows*, p. 330.
- [18] Xiao, H., Wu, J.-L., Wang, J.-X., Sun, R., and Roy, C., 2016. "Quantifying and reducing model-form uncertainties in Reynolds-averaged Navier–Stokes simulations: A data-driven, physics-informed bayesian approach". *Journal of Computational Physics*, **324**, pp. 115–136.
- [19] Weatheritt, J., and Sandberg, R. D., 2016. "A novel evolutionary algorithm applied to algebraic modifications of the rans stress–strain relationship". *Journal of Computational Physics*, **325**, pp. 22–37.
- [20] Ling, J., Kurzawski, A., and Templeton, J., 2016. "Reynolds averaged turbulence modelling using deep neural networks with embedded invariance". *Journal of Fluid Mechanics*, **807**, pp. 155–166.
- [21] Ferreira, C., 2001. "Gene expression programming: A new adaptive algorithm for solving problems". *Complex Systems*, **13**(2), pp. 87–129.
- [22] LeCun, Y., Bengio, Y., and Hinton, G., 2015. "Deep learning". *Nature*, **521**(7553), pp. 436–444.
- [23] Pope, S., 1975. "A more general effective-viscosity hypothesis". *Journal of Fluid Mechanics*, **72**(02), pp. 331–340.
- [24] Rodi, W., 1976. "A new algebraic relation for calculating the reynolds stresses". In Gesellschaft Angewandte Mathematik und Mechanik Workshop Paris France, Vol. 56, p. 219.
- [25] Weatheritt, J., and Sandberg, R. D., 2016. "The development of algebraic stress models using a novel evolutionary algorithm". In 11<sup>th</sup> International ERCOFTAC Symposium on Engineering Turbulence Modelling and Measurements.
- [26] Acharya, S., Tyagi, M., and Hoda, A., 2001. "Flow and heat transfer predictions for film cooling". *Annals of the New York Academy of Sciences*, **934**(1), pp. 110–125.
- [27] Coletti, F., Benson, M., Ling, J., Elkins, C., and Eaton, J., 2013. "Turbulent transport in an inclined jet in cross-flow". *International Journal of Heat and Fluid Flow*, **43**, pp. 149–160.
- [28] Khalighi, Y., Ham, F., Moin, P., Lele, S. K., Colonius, T., Schlinker, R. H., Reba, R. A., and Simonich, J., 2010. "Unstructured large eddy simulation technology for prediction and control of jet noise". In ASME Turbo Expo 2010: Power for Land, Sea, and Air, American Society of Me-

- chanical Engineers, pp. 57–70.
- [29] Vreman, A., 2004. “An eddy-viscosity subgrid-scale model for turbulent shear flow: Algebraic theory and applications”. *Physics of Fluids (1994-present)*, **16**(10), pp. 3670–3681.
  - [30] Touber, E., and Sandham, N. D., 2009. “Large-eddy simulation of low-frequency unsteadiness in a turbulent shock-induced separation bubble”. *Theoretical and Computational Fluid Dynamics*, **23**(2), pp. 79–107.
  - [31] Bodart, J., Coletti, F., Bermejo-Moreno, I., and Eaton, J., 2013. “High-fidelity simulation of a turbulent inclined jet in a crossflow”. *Center for Turbulence Research Annual Research Briefs*.
  - [32] Menter, F. R., 1994. “Two-equation eddy-viscosity turbulence models for engineering applications”. *AIAA journal*, **32**(8), pp. 1598–1605.
  - [33] Parish, E. J., and Duraisamy, K., 2016. “A paradigm for data-driven predictive modeling using field inversion and machine learning”. *Journal of Computational Physics*, **305**, pp. 758–774.
  - [34] Parneix, S., Laurence, D., and Durbin, P., 1998. “A procedure for using DNS databases”. *Journal of fluids engineering*, **120**(1), pp. 40–47.
  - [35] Koza, J. R., 1992. *Genetic programming: on the programming of computers by means of natural selection*, 1 ed. MIT press.
  - [36] Maas, A., Hannun, A., and Ng, A., 2013. “Rectifier nonlinearities improve neural network acoustic models”. *Proceedings of ICML*, **30**, pp. 1–6.



Minerva Access is the Institutional Repository of The University of Melbourne

**Author/s:**

Weatheritt, J; Sandberg, RD; Ling, J; Saez, G; Bodart, J

**Title:**

A Comparative Study of Contrasting Machine Learning Frameworks Applied to RANS Modeling of Jets in Crossflow

**Date:**

2017-01-01

**Citation:**

Weatheritt, J., Sandberg, R. D., Ling, J., Saez, G. & Bodart, J. (2017). A Comparative Study of Contrasting Machine Learning Frameworks Applied to RANS Modeling of Jets in Crossflow. Proceedings of the ASME Turbo Expo: Turbine Technical Conference and Exposition, 2017, VOL 2B, 2B-2017, American Society of Mechanical Engineers. <https://doi.org/10.1115/GT2017-63403>.

**Persistent Link:**

<http://hdl.handle.net/11343/241854>

**File Description:**

Accepted version