SAND2016-11832C

Proceedings of ASME Turbo Expo 2017
ASME Turbo Expo 2017
June 26-30, 2017, Charlotte, NC, USA

GT2017-63299

DRAFT V0: A MACHINE LEARNING APPROACH FOR DETERMINING THE TURBULENT DIFFUSIVITY IN FILM COOLING FLOWS

Pedro Milani*
Mechanical Engineering Department
Stanford University
Stanford, CA 94305
Email: pmmilani@stanford.edu

Julia Ling
Thermal/Fluid Science and Engineering
Sandia National Laboratories
Livermore, CA 94551

Gonzalo Saez-Mischlich
ISAE
University of Toulouse
Toulouse 31400, France

Julien Bodart
ISAE
University of Toulouse
Toulouse 31400, France

John Eaton
Mechanical Engineering Department
Stanford University
Stanford, CA 94305

ABSTRACT

In film cooling flows, it is important to know the temperature distribution resulting from the interaction between a hot main flow and a cooler jet. However, current Reynolds Averaged Navier Stokes (RANS) models yield poor temperature predictions. A novel approach for RANS modeling of the turbulent heat flux is proposed, in which the simple gradient diffusion hypothesis (GDH) is assumed and a machine learning algorithm is used to infer an improved turbulent diffusivity field. This approach is implemented using three distinct data sets: two are used to train the machine learning algorithm and the third is used for validation. The results show that the proposed method produces significant improvement compared to the common RANS model, especially in the prediction of film cooling effectiveness.

NOMENCLATURE

θ Dimensionless temperature $(T-T_\infty)/(T_{cool}-T_\infty)$
D Hole diameter in the jet in crossflow geometries
H Cube height in the cube in crossflow geometry
BR Blowing ratio in a jet in crossflow configuration
U Velocity scale (either bulk velocity or free-stream velocity)
ν Fluid kinematic viscosity
k Turbulent kinetic energy
ε Turbulent dissipation rate
ν_t Eddy viscosity calculated by RANS, $C_\mu k^2/\varepsilon$
α_t Turbulent diffusivity
$\alpha_{t,RANS}$ Turbulent diffusivity predicted by the fixed $Pr_t = 0.85$ model
$\alpha_{t,LES}$ Turbulent diffusivity extracted from the high-fidelity simulation
$\alpha_{t,ML}$ Turbulent diffusivity predicted by the machine learning algorithm
Pr_t Turbulent Prandtl number ν_t/α_t
u_i i-th component of the velocity
d Distance to the nearest wall

1 INTRODUCTION

Film cooling is a widely used technique to control the thermal stresses and increase the lifespan of gas turbine blades [1]. For design purposes, Reynolds-Averaged Navier-Stokes (RANS) simulations are often used to solve for the temperature field in

*Address all correspondence to this author.

these flows, which can be used to determine the film cooling performance and blade metal temperature distribution. Past studies have shown that RANS models struggle to accurately solve for the velocity and temperature fields in film cooling configurations [2, 3]. There has been considerable effort in closure strategies for the momentum equation: for example, Hoda and Acharya tested seven different models [2]. However, previous work has shown that the largest source of error in film cooling predictions is the model for the turbulent heat flux [4]. Therefore, the present work focuses solely on RANS models for the scalar transport equation. In the case of non-buoyant flows, temperature can be considered a passive scalar.

The simplest model to close the turbulent transport term in the RANS scalar equation is the Gradient Diffusion Hypothesis (GDH), shown in the following equation:

$$\overline{u_i'\theta'} = -\alpha_t \frac{\partial \bar{\theta}}{\partial x_i} \tag{1}$$

u_i represents the velocity in the i-th direction and $\theta = (T - T_\infty)/(T_{cool} - T_\infty)$ is the dimensionless temperature, with T_∞ being the inlet temperature and T_{cool} being the coolant temperature. Primed quantities indicate turbulent fluctuations and overbars indicate time average. The turbulent diffusivity is isotropic and given by α_t. The standard practice in commercial codes is to calculate the eddy viscosity ν_t in the momentum equations and use a fixed turbulent Prandtl number to find α_t, usually $Pr_t = \nu_t/\alpha_t = 0.85$ [5].

Previous work has shown that this approach is not sufficient for film cooling flows [3, 6–9]. He et al. [6] performed RANS simulations in a normal jet in crossflow and recommended a turbulent Schmidt number $Sc_t = 0.2$ to better match experimental results, a value considerably lower than 0.85 (in the current paper, Sc_t and Pr_t are considered indistinguishable because temperature is assumed to behave as a passive scalar). Lakehal [7] obtained improved results for the temperature field in a film cooling geometry after considering an anisotropic diffusivity and a spatially-varying Pr_t. The experimental measurements of Kohli and Bogard [8] suggested that Pr_t varied between 0.5 and 2 in an inclined jet in crossflow. Liu et al. [9] showed that the predicted film cooling effectiveness prediction is sensitive to the choice of Pr_t and that a spanwise-varying Pr_t improved the simulation results. Coletti et al. [3] used MRI data and concluded that ν_t and α_t vary spatially according to different trends, so their ratio is not uniform.

There are more complex algebraic closures of the scalar equation as alternatives to the simple GDH. The Generalized Gradient Diffusion Hypothesis (GGDH) of Daly and Harlow [10] and the Higher Order Generalized Gradient Diffusion Hypothesis (HOGGDH) of Abe and Suga [11] are two commonly cited examples. Ling et al. [12] studied these closures in the context of a discrete hole film cooling geometry and found that they indeed yield slightly more accurate turbulent transport than the GDH. However, they concluded that tuning model parameters in simpler models produces greater improvements than switching model form. In fact, their most accurate results were obtained when the GDH was used in conjunction with a turbulent diffusivity field extracted from a highly resolved Large Eddy Simulation (LES) [12]. Inspired by those results, the present paper seeks to use the simple GDH of Eqn. (1) with a machine learning methodology to select a more appropriate α_t field.

Machine learning consists of a broad class of algorithms that process large amounts of data and extract patterns from them in order to make informed decisions in uncertain situations. Recently, the fast increase in computational capability led to a surge in the availability of large data sets from fluid mechanics simulations. In turn, this sparked interest in using machine learning techniques in the context of turbulence modeling [13–18]. Tracey et al. [13] used machine learning algorithms to improve RANS models and obtain error bounds in a turbulent combustion and non-equilibrium boundary layer simulations. Tracey et al. and Duraisamy et al. [14, 15] applied machine learning algorithms to generate functional forms of turbulence models. Ling and Templeton [16] evaluated different machine learning algorithms applied to the problem of predicting regions where RANS simulations yield uncertain results. Ling et al. [17, 18] predicted the Reynolds stress anisotropy in a jet in crossflow geometry using machine learning and also solved for the RANS velocity field using such anisotropy. Although the research mentioned is still in an early phase, all the authors concluded that machine learning seems a promising technique to inform turbulence modeling.

In this paper, a supervised learning algorithm is applied in an attempt to improve the RANS results of the scalar transport equation, particularly in film cooling geometries. The objective is to predict an improved α_t field that can be used with Eqn. (1). A total of three data sets are used, in which both RANS and high-fidelity simulations are available. The high-fidelity data are used to calibrate the machine learning model and to assess the model performance. Section 2 describes these data sets. Section 3 explains the machine learning approach. Section 4 presents the results of this study, both in terms of diffusivity and in terms of the scalar field. Finally, section 5 contains the conclusions and next steps for this research.

2 COMPUTATIONAL DATA SETS

To apply the machine learning framework, computational data sets of three different geometries are used: a baseline inclined jet in crossflow, a skewed inclined jet in crossflow, and a wall-mounted cube in crossflow. In all cases, a well-validated high-fidelity simulation is available in conjunction with a RANS simulation that uses the standard realizable $k-\varepsilon$ model in the momentum equation and the GDH with $Pr_t = 0.85$ in the scalar

equation. Throughout this paper, buoyancy and molecular diffusion are neglected, so the analogy between temperature and a passive scalar is used. Therefore, the scalar field will be treated as the dimensionless temperature θ and the turbulent Schmidt and Prandtl numbers are indistinguishable. The three cases are summarized in Table 1 and further described in the following subsections.

2.1 Baseline inclined jet in crossflow

A schematic of the baseline jet in crossflow geometry can be seen in Fig. 1. It consists of a single circular cooling hole of diameter D injecting onto a square channel of side $8.62D$. The origin of the coordinate system is located at the center of the hole. The hole is inclined 30° with respect to the y axis and is not angled with respect to the x axis (the latter differentiates it from the skewed case). The Reynolds number based on the channel bulk velocity and hole diameter is $Re_D = 3{,}000$. The flow is incompressible and the velocity ratio between the bulk main flow velocity and the bulk hole velocity is $BR = 1$. The cooling flow is fed from a plenum underneath the channel and has a prescribed concentration of a scalar contaminant. An adiabatic boundary condition is prescribed at all the walls.

The LES of this configuration was performed by Bodart et al. [19] using CharLESx, a second order, unstructured, finite-volume solver developed at Stanford University. It employs the Vreman subgrid scale model in the momentum equation and the GDH with $Sc_{SGS} = 0.9$ in the scalar equation. A posteriori analysis showed that the LES resolved most of the energy containing scales. The authors also extensively validated the LES results against experimental data and found good agreement. More details can be found in Ref. [19].

The RANS simulation of this configuration was performed by Coletti et al. [3] using the software FLUENT. The first cell above the wall was located at $y^+ < 1$ and they confirmed grid-independent convergence. More information can be found in Ref. [3].

2.2 Skewed inclined jet in crossflow

A schematic of this configuration can be found in Fig. 2(a). The skewed jet in crossflow geometry is very similar to the baseline case described in the previous section, except that the circular cooling hole is inclined 30° both with respect to the y direction and the x direction and the channel is wider in the spanwise direction (8.62D x 17.3D). The skew creates a qualitatively distinct mean velocity field: while the baseline case contains a counter-rotating vortex pair (CVP) downstream of injection, the secondary flow in the skewed case is dominated by a single vortex [20]. The Reynolds number based on the main flow bulk velocity and hole diameter is $Re_D = 5{,}800$ and the blowing ratio is $BR = 1$. The skewed hole is fed from a plenum and contains a prescribed passive scalar concentration, and all the walls are adiabatic.

TABLE 1. SUMMARY OF THE DATA SETS

Case	Description and references	Reynolds number
Baseline	Inclined jet in crossflow [3, 19]	3,000
Skewed	Skewed inclined jet in crossflow [20]	5,800
Cube	Wall-mounted cube in crossflow [21]	5,000

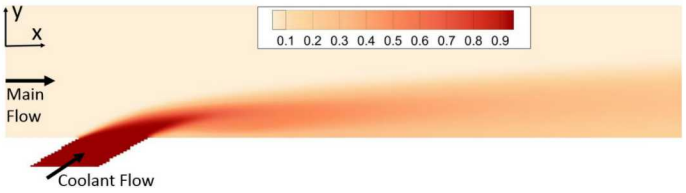


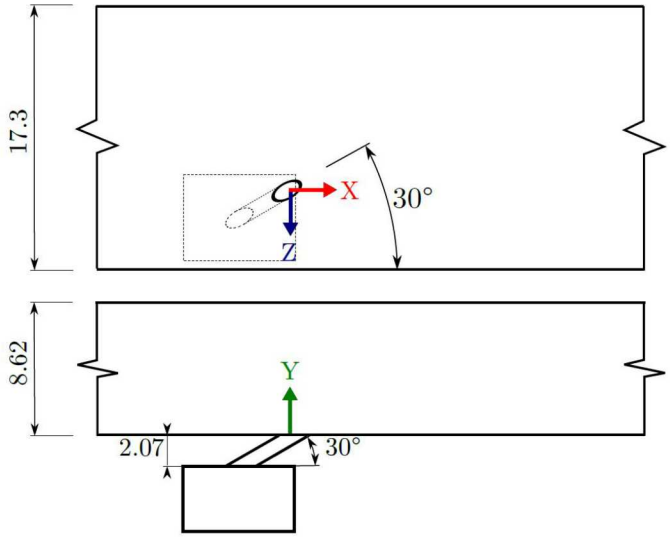
FIGURE 1. Schematic of the baseline geometry. Figure shows contour of $\bar{\theta}$ at the center spanwise plane as calculated by the LES. Plenum that feeds the jet is also simulated, but not shown.

The LES of this geometry was performed by Folkersma [20]. It was a nominally compressible simulation, but since the Mach number was low (approximately 0.2), the density variations were negligible. It also used the CharLESx solver and a posteriori results showed that subgrid scale contribution in the turbulent transport was negligible. The results were validated against experimental data and good agreement was found. More details can be found in Ref. [20].

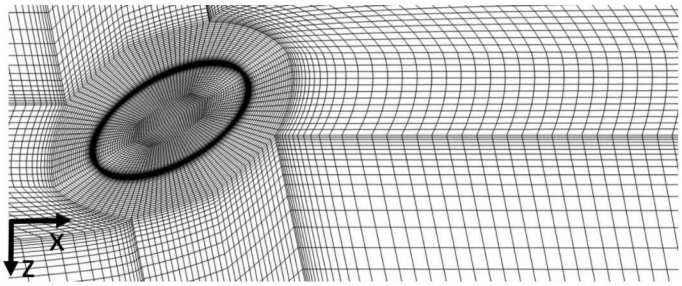
The RANS simulation of this geometry was performed in the scope of the present paper using the software FLUENT. The simulation domain spanned $60D$ in the streamwise direction, divided evenly before and after the injection point, to minimize the influence of inlet and outlet boundary conditions. At the channel inlet, a turbulent velocity profile was picked to best match the LES velocity from Ref. [20]. A slip condition was applied at the top wall, and no-slip was applied at all the other walls. A similar mesh to the one used in the baseline RANS was used in this simulation, and it can be seen in Fig. 2(b). The equations were considered converged when the continuity residuals were below $2.5x10^{-2}$ and the other residuals were below 10^{-6}.

2.3 Wall-mounted cube in crossflow

Figure 3 shows a schematic of the wall-mounted cube in crossflow. A cube of side H is attached to the wall and acts as an obstacle for the incoming flow. The Reynolds number based on H and the free-stream velocity is $Re_H = 5{,}000$. The no-slip boundary condition for the velocity field is imposed at the wall



(a) Schematics



(b) RANS mesh

FIGURE 2. Skewed geometry. (a) is from [20] and has a wall normal plane showing the skewed hole and a spanwise plane showing dimensions and the plenum (the latter is also valid for the baseline case). (b) shows the mesh used in the RANS simulation at the wall around the injection hole.

and cube surface, and a passive scalar is released from a circular source centered on the top face of the cube. A direct numerical simulation (DNS) and a RANS simulation of this geometry were performed by Rossi et al. [21]. Their DNS was validated against experimental data and good agreement was found. More information about the configuration and the simulations can be found in Ref. [21].

3 MACHINE LEARNING APPROACH

The proposed framework to RANS modeling consists of using a supervised learning algorithm to infer the turbulent diffusivity field α_t. Supervised learning is a subset of machine learning algorithms in which the objective is to predict the value of

FIGURE 3. Schematic of the cube geometry. Figure shows contour of $\bar{\theta}$ calculated by the DNS at the center spanwise plane.

a parameter given some information about the problem (the features). The algorithm is calibrated with examples in which the correct values for both the target parameter and the features are known. These are called training examples. In the present work, the parameter that the algorithm tries to predict is the turbulent diffusivity at each cell in the computational domain and the features are RANS variables at that cell.

3.1 Features

The features used to predict a cell's turbulent diffusivity are exclusively quantities that are calculated in a $k-\varepsilon$ RANS simulation. This is because the objective of the present approach is to improve RANS predictions in a situation in which high-fidelity data are not available. The features are chosen as quantities upon which the turbulent diffusivity might depend: the RANS mean velocity gradient ($\nabla\bar{\mathbf{u}}$), the RANS mean scalar gradient as calculated by the fixed $Pr_t = 0.85$ model ($\nabla\bar{\theta}$), the distance to the nearest wall (d), and the eddy viscosity calculated by the $k-\varepsilon$ model (ν_t). Note that if the fixed Pr_t model were sufficient, the only RANS variable upon which α_t would depend is ν_t. By adding other RANS variables, the algorithm is given more information about the local velocity and concentration fields and then "learns" how to use it.

The raw RANS variables mentioned above need to be processed before they are used in the machine learning framework to ensure the algorithm is dimensionally consistent and produces an isotropic result. To guarantee the former, the features are non-dimensionalized by the following local scales: $\nabla\bar{\mathbf{u}}$ is divided by $\frac{k}{\varepsilon}$, as suggested by Pope [22], $\nabla\bar{\theta}$ is divided by $\frac{k^{3/2}}{\varepsilon}$, d is divided by $\frac{\nu}{\sqrt{k}}$, which creates a turbulence Reynolds number, and ν_t is divided by ν. k and ε are the kinetic energy and dissipation values predicted by the RANS equations, and ν is the fluid's kinematic viscosity. The resulting turbulent diffusivity α_t is non-dimensionalized by the global scale UD, where U and D are the relevant velocity and length scales of the problem. Other possible non-dimensional forms were tried, but the ones reported yielded the best overall results.

Since the output of the machine learning algorithm is an isotropic scalar, Ling et al. [23] suggest that this property be

enforced by pre-processing the features and extracting isotropic bases. The idea behind this is to add physical meaning to the predictions: if the same flow were studied under a different coordinate system, the raw features described in the last paragraph would change, but the predicted α_t should not. Therefore, a polynomial basis that is invariant to coordinate transformations is extracted from the dimensionless velocity gradient tensor and the dimensionless scalar gradient vector as described in Smith [24]. This set of invariant scalars is then used together with the dimensionless forms of d and ν_t as the features fed into the supervised learning algorithm, for a total of 19 features. More information on this process can be found in Ref. [23].

3.2 Training Examples

To calibrate the algorithm, training examples in which the turbulent diffusivity is known are needed. High-fidelity, time-resolved simulations are used for this purpose. From the the mean concentration gradient and turbulent transport term, which is directly available in a DNS or LES, the turbulent diffusivity can be inferred by assuming that the GDH of Eqn. (1) holds, according to the following equation:

$$\alpha_{t,LES} = -\frac{\overline{u_i'\theta'}\frac{\partial\tilde{\theta}}{\partial x_i}}{\frac{\partial\tilde{\theta}}{\partial x_i}\frac{\partial\tilde{\theta}}{\partial x_i}} \tag{2}$$

where $\alpha_{t,LES}$ is the turbulent diffusivity inferred from the high-fidelity simulation (either DNS or LES) and the terms on the right-hand side are the statistics calculated from the high-fidelity simulation at that cell. Eqn. (2) assumes that there is an isotropic turbulent diffusivity that relates the mean gradient and the turbulent transport. This assumption is not necessarily valid and would still cause model form error. However, Ling et al. [12] showed that using $\alpha_{t,LES}$ in a RANS simulation dramatically increases the accuracy of the results. So, $\alpha_{t,LES}$ as calculated by Eqn. (2) is considered to be the correct value that is used to train the machine learning algorithm.

3.3 Algorithm

There are several supervised learning algorithms that can be applied to the problem of predicting a continuous variable such as the turbulent diffusivity. These include support vector machines, decision trees, and neural networks. Ling and Templeton [16] tested different algorithms applied to turbulence modeling and concluded that random forests (RF) had the "best combination of good performance and easy implementation". Random forests are capable of non-linear decision boundaries, are robust against overfitting and against the inclusion of non-important features. Also, pre-processing to force the data to fall within a specified range is not required, as it is for some other algorithms.

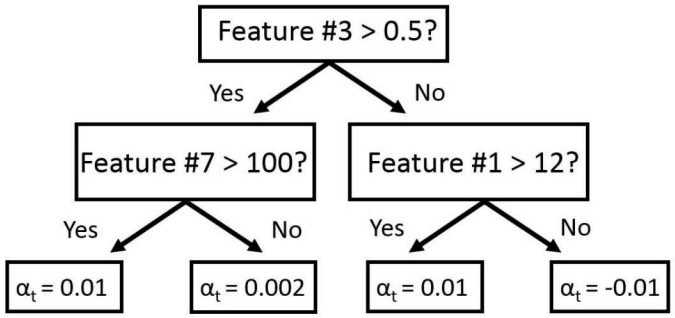


FIGURE 4. Example binary decision tree of height 2. To decide on the value of α_t for any set of features, the rules are followed starting from the top until a leaf is reached.

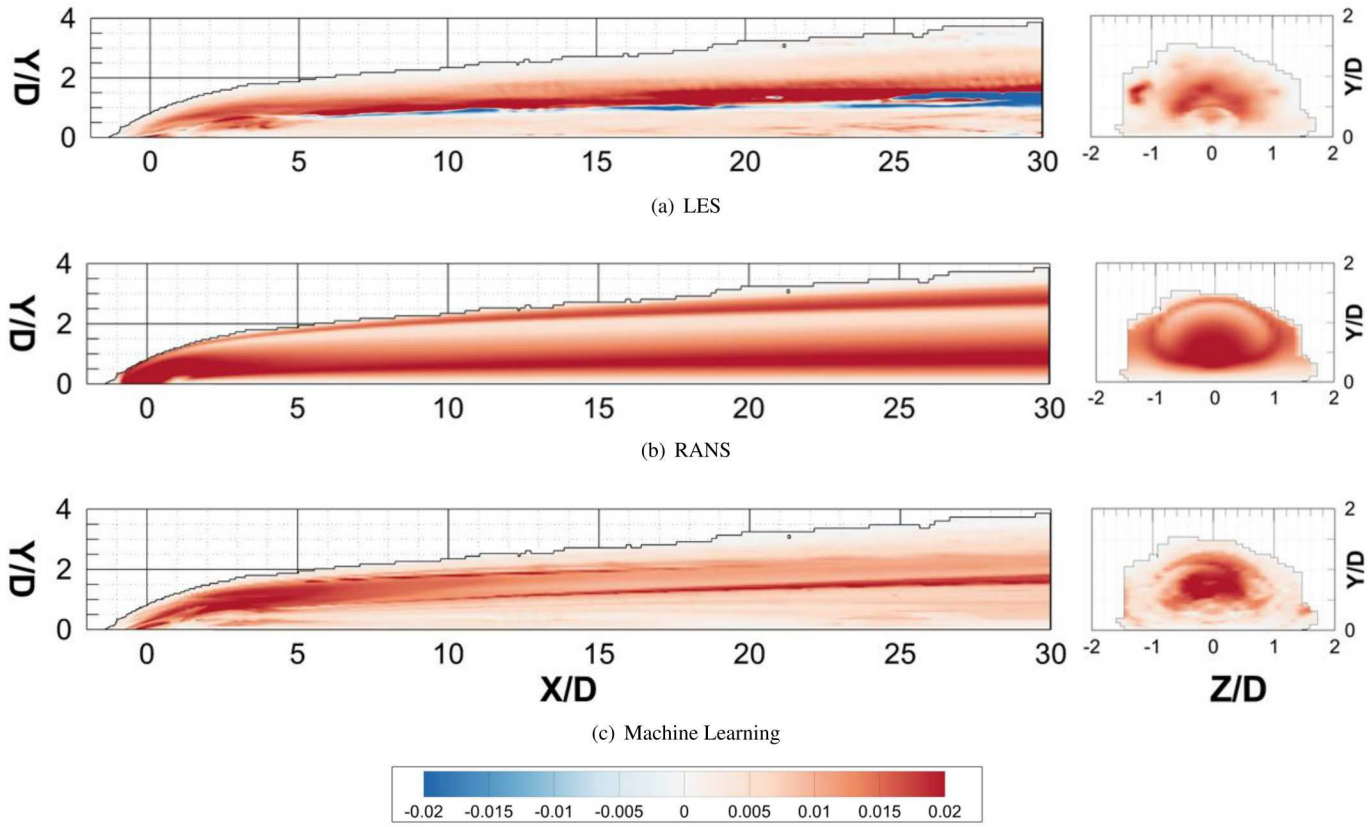
Finally, RFs are relatively cheap to train and test when compared with neural networks [16]. Therefore, to demonstrate the framework proposed in the present work, the random forest algorithm is employed.

Random forests consist of an ensemble of binary decision trees [25]. A diagram with an example binary decision tree is shown in Fig. 4. It consists of a collection of rules (a feature with an associated threshold) that are used to classify an example. When a prediction is needed for a set of features, the decision tree follows its rules starting from the top (the root) until it reaches an endpoint (the leaves), where a value is assigned. To construct a binary decision tree from a set of training examples, typically a greedy algorithm is used to assign rules that maximize the local information gain sequentially to each node, starting from the root.

In a random forest, each decision tree is constructed by taking a random sample, with replacement, of all the training examples (a process called "bagging"). The overall prediction of the RF is an average of the predictions of each individual tree. A single tree can be prone to overfitting, but constructing an ensemble is done to minimize this problem while retaining good accuracy. The main hyperparameter that must be chosen is the number of trees in the ensemble, N_{trees}. Usually, more trees give better answers and lower variance between runs, but with diminishing returns and at a higher computational cost. After testing performance and variance with different values of N_{trees} for this particular problem, $N_{trees} = 1{,}000$ was chosen because higher values of N_{trees} barely change the results.

4 RESULTS

To demonstrate the proposed machine learning approach, the skewed and the cube data sets are used for training the RF, which in turn is tested on the baseline data set. The machine learning routines were coded in Python using the Scikit-learn open source library [26]. The high-fidelity data sets and the ex-



(a) LES

(b) RANS

(c) Machine Learning

FIGURE 5. Contours of non-dimensional turbulent diffusivity field $\alpha_t/(UD)$ in the baseline geometry. The figures on the left are a spanwise plane at the center of the channel, $Z/D = 0$. The figures on the right show streamwise planes at $X/D = 2$. The plots are blanked in regions in which the dimensionless mean scalar gradient calculated by the RANS or by the LES data sets is smaller than 10^{-5}. (a) is the field extracted from the LES using Eq. 2, (b) is the RANS diffusivity calculate via a fixed $Pr_t = 0.85$, and (c) is the field that the machine learning algorithm predicts.

tracted turbulent diffusivity are linearly interpolated onto the respective RANS meshes. The features are calculated and the cells of the skewed and cube cases are used to train the random forest. Not all points are used: only the cells where the 2-norm of the dimensionless mean scalar gradient from both the DNS/LES and RANS is larger than 10^{-5} are used for training and testing. This is because in regions of low gradient, Eq. (2) could yield near singular results and the GDH would predict negligible scalar transport anyway. Also, two small regions in the LES of the skewed case (one close to injection and one near the outlet) had poorly converged values of $\overline{u_i'\theta'}$, which caused the extracted $\alpha_{t,LES}$ to be orders of magnitude higher than what was observed in the rest of the flow. These two regions were removed from the training set, which considerably improved the performance of the resulting algorithm. Overall, around 250k cells in the cube case and 450k cells in the skewed case are used for training, and the algorithm is tested on around 475k cells of the baseline case.

4.1 Turbulent Diffusivity Prediction

Figure 5 shows contours of the non-dimensionalized turbulent diffusivity in the baseline geometry, including the prediction from the machine learning algorithm after it was trained on the skewed and cube cases. In Fig. 5(a), the turbulent diffusivity extracted from the baseline LES according to Eq. 2 is shown. This is the field that the machine learning algorithm is trying to replicate. One interesting feature is that it contains regions where $\alpha_{t,LES}$ is negative, which implies that turbulent diffusion acts to move heat from colder regions to hotter regions. This unphysical behavior hints that underlying assumptions of the GDH, such as turbulent length scales being smaller than the length scales over which the mean temperature changes [27], are not valid in those regions.

Figure 5(b) shows the turbulent diffusivity field that typical RANS codes would use, given by $\alpha_{t,RANS} = (C_\mu k^2/\varepsilon)/Pr_t$, with $C_\mu = 0.09$ and $Pr_t = 0.85$. Comparing this field to Fig. 5(a) helps explain the difficulties that RANS codes have to solve for $\bar{\theta}$. It is

clear that in the region close to injection (especially for $X/D < 2$) and in a layer underneath the jet (Y/D around 0.5), the fixed Pr_t model overestimates the turbulent diffusion, and that is expected to create inaccurate scalar fields.

Figure 5(c) presents the turbulent diffusivity field that was calculated by the random forest, $\alpha_{t,ML}$. Note that there are still differences between this field and the one shown in Fig. 5(a). For example, the RF fails to predict the regions of negative diffusivity. However, it shows significant improvement over the RANS diffusivity field close to injection and around the jet, regions where the fixed Pr_t model was particularly inappropriate.

For a quantitative comparison between the fields in Figs. 5(b) and 5(c), one can define an integral error metric. A possible choice is the sum of the error magnitudes in each cell normalized by the sum of the magnitudes of the LES diffusivity, as shown in the following equation:

$$error_\alpha = \frac{\sum |\alpha_t - \alpha_{t,LES}|}{\sum |\alpha_{t,LES}|} \tag{3}$$

For each turbulent diffusivity field ($\alpha_{t,RANS}$ and $\alpha_{t,ML}$), the error can be calculated by Eq. 3. The sum is evaluated for all cells in the RANS mesh of the baseline case where the 2-norm of the dimensionless mean scalar gradient from the LES and RANS is larger than 10^{-5}. By this metric, the RANS diffusivity has an error of 1.20, while the machine learning diffusivity achieves 0.60, a 50% improvement. The error is still relatively high, but since the improvement is concentrated in regions of the flow with high gradients, such as near injection and close to the wall, using $\alpha_{t,ML}$ to solve the scalar equation could conceivably generate significant improvements.

4.2 Forward Propagation

The machine learning algorithm is designed to predict a turbulent diffusivity field. However, the ultimate goal is to improve the predictions of the mean temperature field $\bar{\theta}$. Therefore, the best way to assess the performance of the RF prediction shown in Fig. 5(c) is to use that field in a RANS code and compare its temperature prediction against the LES results. This is the forward propagation step.

Since the passive scalar assumption is used, the continuity and momentum equations don't need to be solved again; it is enough to take their results as a prescribed mean velocity field to be used in the scalar equation. Assuming the mean fields are steady, using the GDH of Eq. 1 to close the turbulent transport term, and neglecting molecular diffusion produces the Reynolds-Averaged Advection Diffusion (RAAD) equation shown below:

$$\frac{\partial}{\partial x_i}(\bar{u}_i\bar{\theta}) = \frac{\partial}{\partial x_i}(\alpha_t \frac{\partial \bar{\theta}}{\partial x_i}) \tag{4}$$

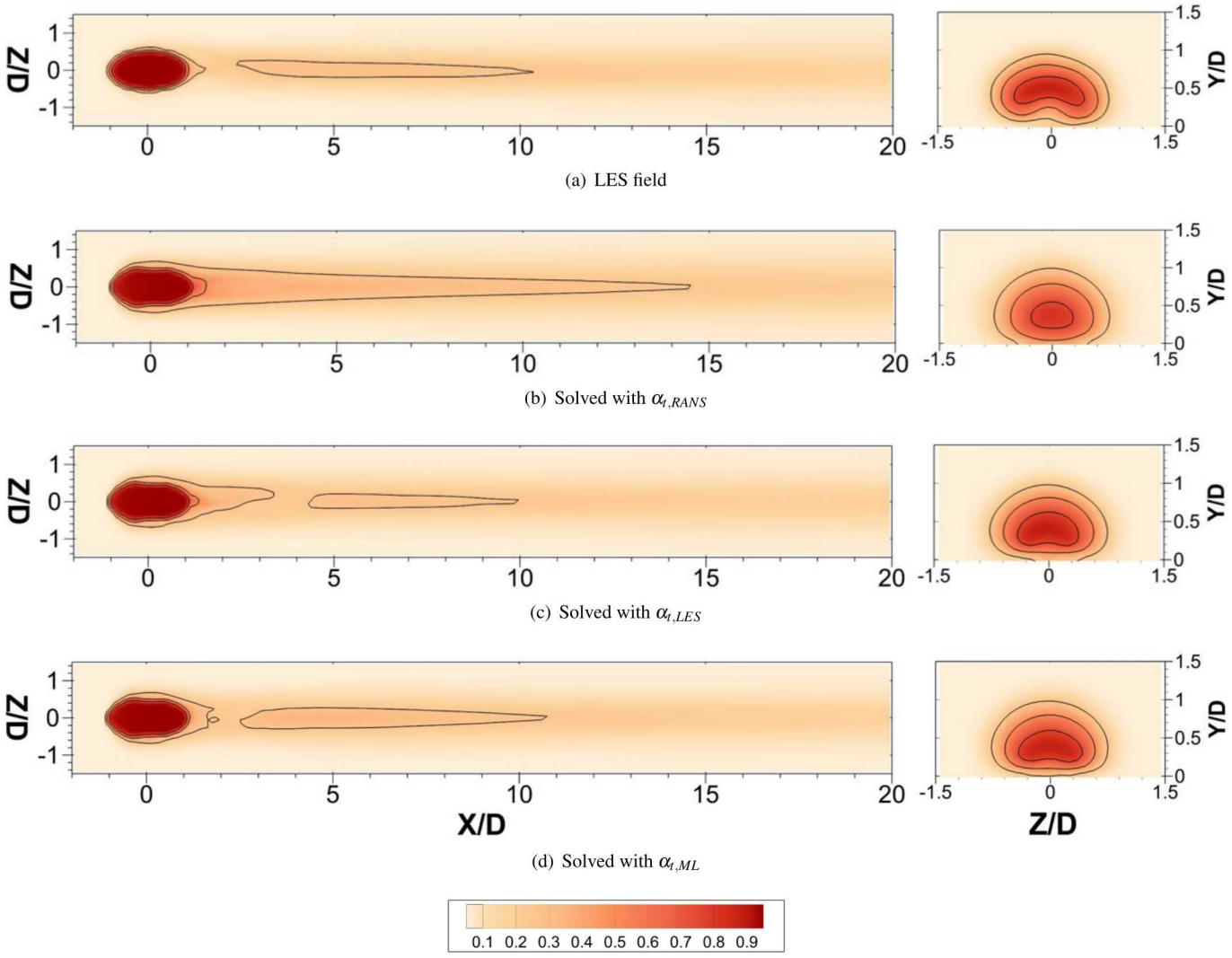
Note that with a known velocity field and turbulent diffusivity, Eq. 4 can be directly solved for the dimensionless temperature field. To test the potential of the machine learning approach, Eq. 4 is solved in the baseline geometry using the RANS velocity field and each of the three turbulent diffusivity fields shown in Fig. 5. The RANS velocity doesn't exactly match the true velocity field from the LES [3]. But in an arbitrary geometry in which RANS predictions are desired, a high fidelity velocity field might not be available and the RANS velocity field might be the only option. Thus, to simulate the predictive capabilities in this scenario, the RANS velocity field is chosen.

Equation 4 is solved using an in-house code with a uniform and structured mesh. It uses a finite volume method with second order central differencing for the diffusion term and first order upwinding for the convective term. The resolutions in the streamwise and spanwise direction are similar to the ones in the RANS simulation of [3] ($dx/D = dz/D = 0.1$), and the wall-normal resolution was picked to match the one Coletti et al. used at $y/D = 1$, $dy/D = 0.05$. This results in the first cell above the wall being at approximately $y^+ = 10$. Despite using simple numerical methods and a coarse mesh, this solver was picked because of the ease of prescribing a velocity and turbulent diffusivity fields. The resolution is still enough to resolve the main features of the mean velocity and temperature, so the results are useful to determine whether the machine learning algorithm can improve the temperature predictions.

The method described above is used the solve the RAAD equation and the results are shown in Fig. 6. The diffusivity fields needed to be marginally pre-processed before being fed into the code. The LES and ML diffusivity were made non-negative to guarantee convergence and a moderate positive value ($\alpha_t/(UD) = 0.01$) was prescribed for the blanked region of Fig. 6 in all three cases. The latter is done because the LES diffusivity field was not extracted and the ML algorithm was not applied in regions of low mean temperature gradient. Note that the value of α_t in those regions is not important because the GDH applied there would predict negligible turbulent transport regardless.

The results of the forward propagation are shown in Fig. 6. Fig. 6(a) shows the LES scalar field from Bodart et al. [19]. This is assumed to be the correct mean temperature field and is used to assess the other three calculations. Figures 6(c)-(d) present results of the RAAD equation solved with the three different turbulent diffusivity fields.

Comparing the LES field of Fig. 6(a) with the field solved for using $\alpha_{t,RANS}$ of Fig. 6(b) demonstrates the difficulties that fixed Pr_t models have in film cooling geometries. In reality, at $BR = 1$, the jet separates after injection and the temperature field stays concentrated within it, in a characteristic kidney shape observed on the right frame of Fig. 6(a). This causes a low adiabatic effectiveness at around $X/D = 2$. However, the diffusivity obtained from $Pr_t = 0.85$ is too high close to injection and between the jet and the wall, as observed in Fig. 5(b). This makes the tem-



(a) LES field

(b) Solved with $\alpha_{t,RANS}$

(c) Solved with $\alpha_{t,LES}$

(d) Solved with $\alpha_{t,ML}$

FIGURE 6. Mean dimensionless temperature field in the baseline geometry. The left panels show wall normal planes at the wall ($Y/D = 0$), which are equivalent to adiabatic effectiveness. The right panels show streamwise planes at $X/D = 2$. The contour lines are shown at $\bar{\theta} = 0.75, 0.5, 0.25$. (a) has the LES field from Bodart et al. [19], (b)-(d) contain the mean scalar field calculated using Eq. 4 with different turbulent diffusivity fields.

perature diffuse towards the wall too quickly, which causes the simulation to overestimate the adiabatic effectiveness and fail to predict the kidney shape at $X/D = 2$.

Figure 6(c) is the result of using the turbulent diffusivity field extracted directly from the LES of the baseline geometry. Here, using the wrong turbulent diffusivity field is eliminated as a source of error; the discrepancies between Fig. 6(a) and Fig. 6(c) are caused from model form error, from using the RANS velocity field, and from numerical discretization. As expected, it yields improved results when compared to the field solved for using the RANS diffusivity field. The adiabatic effectiveness, for example, is qualitatively closer to the true one: particularly, it captures the jet separation and a region of low effectiveness right downstream of injection.

The results of Fig. 6(d) show the temperature field calculated from the turbulent diffusivity field inferred by the machine learning algorithm, $\alpha_{t,ML}$. It shows remarkable improvement over the RANS diffusivity field. The adiabatic effectiveness is considerably more accurate, even though it is slightly overestimated near $X/D = 0.75$, and the separation region at around $X/D = 2$ is well captured. The kidney shape of the temperature field at $X/D = 2$ is not as clearly resolved as in Fig. 6(a), but it is much closer than what was inferred from the $Pr_t = 0.85$ calculation. Surprisingly, some of the qualitative featured mentioned above are better cap-

tured in Fig. 6(d) than in Fig. 6(c). This is probably due to errors in the ML diffusivity field negating part of the other sources of error in the simulation mentioned in the previous paragraph and causing a more appropriate temperature field in certain regions.

For a quantitative comparison, an integral error metric is defined to assess the difference between each field and the LES field. It consists of an average of the magnitude of the difference between the temperature calculated and the LES temperature at each cell, and is shown in the equation below:

$$error_{\bar{\theta}} = \frac{1}{N} \sum \left| \bar{\theta} - \bar{\theta}_{LES} \right| \tag{5}$$

where the sum is performed over all cells in a particular region of interest, N is the number of cells in that region of interest, and $\bar{\theta}_{LES}$ is the LES mean temperature in that cell.

Table 2 shows the error calculated according to Eq. 5 in three distinct regions of interest of the baseline geometry, with their respective definitions. The errors are not high in absolute terms because the cells are uniformly spaced and the regions of interest include portions where $\bar{\theta}$ is low, so any reasonable RANS solution would get results similar to the LES field. However, the numbers on Table 2 can still be assessed relative to each other. As Fig. 6 suggested, both the LES and ML diffusivities produce improvements over the RANS diffusivity calculated with $Pr_t = 0.85$, with the LES diffusivity producing higher gains than the ML field as expected. In the most critical regions of interest, the improvements are more noticeable and the difference between LES and ML diffusivity fields is relatively smaller. In the injection region, the ML diffusivity produced almost 30% improvement, while in the wall region it produced over 60% improvement over the RANS field. Combined with the qualitative improvements seen in Fig. 6, this shows that machine learning approaches have potential to significantly improve RANS turbulence closures.

TABLE 2. ERROR IN CALCULATED $\bar{\theta}$

Diffusivity	Region of interest		
	Total X/D: -1 to 19 Y/D: 0 to 1.75 Z/D: -0.75 to 0.75	Injection X/D: -1 to 4 Y/D: 0 to 1.75 Z/D: -0.75 to 0.75	Wall X/D: 1 to 19 Y/D: 0 to 0.1 Z/D: -0.75 to 0.75
RANS	0.0277	0.0470	0.0566
LES	0.0204	0.0314	0.0214
ML	0.0231	0.0339	0.0223

5 CONCLUSION

A machine learning approach to improve turbulent mixing models was proposed, with special interest to film cooling geometries. The method consists in using a closure with a simple form for the turbulent heat flux, the gradient diffusion hypothesis, and then using a supervised learning algorithm to better determine the parameter of this model, the turbulent diffusivity α_t. This approach was demonstrated using three data sets: a baseline jet in crossflow, a skewed jet in crossflow, and a wall-mounted cube in crossflow. The last two were used to train the machine learning algorithm, while the baseline case was used to test it. Calculations of the temperature field using the machine learning diffusivity showed significant qualitative and quantitative improvements over the usual $Pr_t = 0.85$ closure. In particular, predictions of the temperature at the wall (the adiabatic effectiveness) were remarkably better.

The proposed approach was designed to be used in complex configurations where no high-fidelity data are available. The machine learning algorithm can be trained in a few geometries where high-fidelity data for velocity and concentration exist (like the three data sets presented in this paper) and then applied to predict the turbulent diffusivity in an arbitrary geometry. A RANS $k-\varepsilon$ simulation of a complex geometry is needed to produce the features for the supervised learning algorithm, which would then output a predicted turbulent diffusivity field for that geometry. Such process can presently be applied with the code base developed in this work. Finally, a second RANS would need to be run, with the constraint that it should use the turbulent diffusivity field produced by the machine learning algorithm to close the temperature equation.

Future work includes evaluating how robust this approach is in different film cooling geometries. It would also be interesting to generalize it to other classes of turbulent flows. This poses the question of how closely related flows used as training sets need to be to the target flow to guarantee improved answers. The approach itself can be perfected too. For example, new machine learning algorithms and different features might produce improved answers in different flows. Also, more complex closures can be used in conjunction with machine learning: the move to an anisotropic diffusivity could produce significant improvement. Finally, a careful study of how the machine learning algorithm utilizes the features to make decisions could reveal important physics underlying turbulent mixing, which in turn could inspire the development of new physics-based models.

ACKNOWLEDGMENT

Pedro Milani was supported by a Stanford School of Engineering Fellowship. Julia Ling was 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] Bogard, D., and Thole, K., 2005. "Gas turbine film cooling". *Journal of Propulsion and Power,* ***22***(2), pp. 249–270.

[2] Hoda, A., and Acharya, S., 2000. "Predictions of a film coolant jet in crossflow with different turbulence models". *Journal of Turbomachinery,* ***122***(3).

[3] Coletti, F., Benson, M., Ling, J., Elkins, C., and Eaton, J., 2013. "Turbulent transport in an inclined jet in crossflow". *International Journal of Heat and Mass Transfer,* ***43***, pp. 149–160.

[4] Ryan, K., Bodart, J., Folkersma, M., Elkins, C., and Eaton, J., 2016. "Turbulent scalar mixing in a skewed jet in crossflow: Experiments and modeling". *To appear on Flow, Turbulence and Combustion*.

[5] Kays, W., 1994. "Turbulent Prandtl number - where are we?". *Journal of Heat Transfer,* ***116***(2), pp. 284–295.

[6] He, G., Guo, Y., and Hsu, A., 1999. "The effect of Schmidt number on turbulent scalar mixing in a jet-in-crossflow". *International Journal of Heat and Mass Transfer,* ***42***(20), pp. 3727–3738.

[7] Lakehal, D., 2002. "Near-wall modeling of turbulent convective heat transport in film cooling of turbine blades with the aid of direct numerical simulation data". *Journal of Turbomachinery,* ***124***(3), pp. 485–498.

[8] Kohli, A., and Bogard, D., 2005. "Turbulent transport in film cooling flows". *Journal of Heat Transfer,* ***127***(5), pp. 513–520.

[9] Liu, C., Zhu, H., and Bai, J., 2008. "Effect of turbulent Prandtl number on the computation of film-cooling effectiveness". *International Journal of Heat and Mass Transfer,* ***51***(25-26), pp. 6208–6218.

[10] Daly, B., and Harlow, F., 1970. "Transport equations in turbulence". *Physics of Fluids,* ***13***(11), pp. 2634–2649.

[11] Abe, K., and Suga, K., 2001. "Towards the development of a Reynolds-averaged algebraic turbulent scalar-flux model". *International Journal of Heat and Fluid Flow,* ***22***(1), pp. 19–29.

[12] Ling, J., Ryan, K., Bodart, J., and Eaton, J., 2016. "Analysis of turbulent scalar flux models for a discrete hole film cooling flow". *Journal of Turbomachinery,* ***138***(1).

[13] Tracey, B., Duraisamy, K., and Alonso, J., 2013. "Application of supervised learning to quantify uncertainties in turbulence and combustion modeling". *AIAA Aerospace Sciences Meeting*.

[14] Tracey, B., Duraisamy, K., and Alonso, J., 2015. "A machine learning strategy to assist turbulence model development". *AIAA Aerospace Sciences Meeting*.

[15] Duraisamy, K., Zhang, Z., and Singh, A., 2015. "New approaches in turbulence and transition modeling using data-driven techniques". *AIAA Aerospace Sciences Meeting*.

[16] Ling, J., and Templeton, J., 2015. "Evaluation of machine learning algorithms for prediction of regions of high Reynolds Averaged Navier Stokes uncertainty". *Physics of Fluids,* ***27***.

[17] Ling, J., Ruiz, A., Lacaze, G., and Oefelein, J., 2016. "Uncertainty analysis and data-driven model advances for a jet-in-crossflow". *Journal of Turbomachinery,* ***139***(2).

[18] 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.

[19] Bodart, J., Coletti, F., Bermejo-Moreno, I., and Eaton, J., 2013. High-fidelity simulation of a turbulent inclined jet in a crossflow. CTR Annual Research Briefs, Center for Turbulence Research, Stanford University, Stanford, CA.

[20] Folkersma, M., 2015. "Large eddy simulation of an asymmetric jet in crossflow". Master of Science thesis, Tampere University of Technology, Tampere, Finland, October.

[21] Rossi, R., Philips, D., and Iaccarino, G., 2010. "A numerical study of scalar dispersion downstream of a wall-mounted cube using direct simulations and algebraic flux models". *International Journal of Heat and Fluid Flow,* ***31***, pp. 805–819.

[22] Pope, S., 1975. "A more general effective-viscosity hypothesis". *Journal of Fluid Mechanics,* ***72***, pp. 331–340.

[23] Ling, J., Jones, R., and Templeton, J., 2016. "Machine learning strategies for systems with invariance properties". *Journal of Computational Physics,* ***318***, pp. 22–35.

[24] Smith, G., 1965. "On isotropic integrity bases". *Archive for Rational Mechanics and Analysis,* ***18***, pp. 282–292.

[25] Breiman, L., 2001. "Random forests". *Machine Learning,* ***45***, pp. 5–32.

[26] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E., 2011. "Scikit-learn: machine learning in Python". *Journal of Machine Learning,* ***12***, pp. 2825–2830.

[27] Corrsin, S., 1974. "Limitations of gradient transport models in random walks and in turbulence". *Advances in Geophysics,* ***18***, pp. 25–60.