Machine Learning Integration in Computational Fluid Dynamics for Rapid Flow Field Prediction

M. Nemati, A. Jahangirian

Aerospace Engineering Department, Amirkabir University of Technology, Tehran, Iran

Abstract

This study introduces a deep learning approach integrated with Computational Fluid Dynamics (CFD) for efficient prediction of aerodynamic turbulent flow fields, particularly for wing section geometries. Utilizing minimal datasets, the research leverages the PARSEC method for precise airfoil geometry extraction, significantly enhancing prediction speed and accuracy. This method demonstrates versatility and robustness in predicting complex flow patterns, showcasing potential for diverse aerodynamic applications.

Keywords: Flow field prediction, Machine learning, airfoil parameterization, Neural Network, Computational Fluid Dynamics.

1. Introduction

In the field of aerodynamics, the use of Computational Fluid Dynamics (CFD) is common for capturing complex flow patterns, but its computational demands have led to the exploration of surrogate models [1]. These surrogate models, however, often lack detailed flow field data crucial for advanced design. One significant trend is replacing CFD solvers with neural networks, enabling faster simulations f for aerodynamics coefficient [2], pressure distributions [3], or entire flow field [4].

While structured grid representations simplify data processing, real world scenarios often require unstructured grids [5]. To bridge this gap, methods like pixelation have been used to project scattered CFD data onto Cartesian grids [6]. Convolutional Neural Networks (CNNs) are particularly notable in this shift, effectively substituting computationally intensive parts. However, this approach has limitations in precision and efficiency. Furthermore, they often encounter difficulties in accurately capturing subtle flow variations [7]. Another promising approach is viewing CFD data points as a point cloud rather than pixels, offering advantages in predictive accuracy and adaptability to complex geometries [8]. Despite these innovations, challenges remain in accurately predicting flow fields for diverse geometric conditions.

This study introduces a method that leverages minimal datasets for predicting flow fields using neural networks, integrating the geometry construction method of PARSEC [9] and CFD data extraction. The goal is to enhance accuracy in predicting flow fields for varying geometric conditions while optimizing dataset preparation and the machine learning model's structure for the training process. Additionally, this study emphasizes the adaptability of deep learning in addressing complex aerodynamic scenarios such as turbulence.

2. Methodology

The current approach is divided into two sections: the geometry dataset and the flow field dataset. In the first section, to extract the geometry, the PARSEC method is employed instead of traditional image-based methods. A major benefit of this approach is that it avoids the distortion problems commonly faced in image-based techniques. Additionally, the number of parameters needed to define a wing section using PARSEC is fewer compared to universal methods such as class shape function transformation (CST) [10] and Free Form Deformation (FFD) [11].

In the second section, the flow field values preserved at the control volume of the computational grid are extracted without modifications, eliminating the need for flow field images. This ensures that the training and prediction accuracy remains closely aligned with CFD results. As shown by Nemati and Jahangirian [12], these features enable the model to predict flow fields in compressible turbulence scenarios with fewer data compared to other methods. Their findings, for a fixed geometry under various flow conditions, demonstrate the ability to predict the flow field with high accuracy. However, to predict different geometric conditions, geometric parameterization is required to generate an accurate shape with the fewest parameters, as mentioned in the first part.

Since predicting the flow field around variable geometries is a primary objective of this study, we will use the parameterization method suggested in Ref. [12] for this purpose.

2.1 Geometry Parameterization

Using a geometry parameterization method involves three essential principles: efficiency, reliability, and adaptability. Firstly, to achieve optimal computational performance, the number of parameters should be minimal. Secondly, the method should consistently produce shapes that are both physically and geometrically logical. Thirdly, the model must be sufficiently adaptable to generate a wide range of geometries. While methods with numerous design variables can represent intricate shapes and potentially yield innovative designs, they also increase the computational cost due to the expanded design space. Additionally, adding more parameters can sometimes result in the creation of shapes that lack physical realism.

The PARSEC method uses a polynomial function with only few defining parameters for wing section representation. A unique advantage of PARSEC is its inclusion of physical geometric attributes, such as the location and curvature of the airfoil's crest, into its design variables. The PARSEC geometry parameterization method consists of 10 input parameters is used in this work (figure 1).

These parameters encompass the leading edge radius, r_{LE}, upper and lower crest locations (X_{UP},Y_{UP}, X_{LO},Y_{LO}), upperand lower curvatures (Y_{XXUP},Y_{XXLO}), trailing edgeposition (Y_{TE}),and direction trailing edge wedge angles (α_{TE} ,β_{TE}).

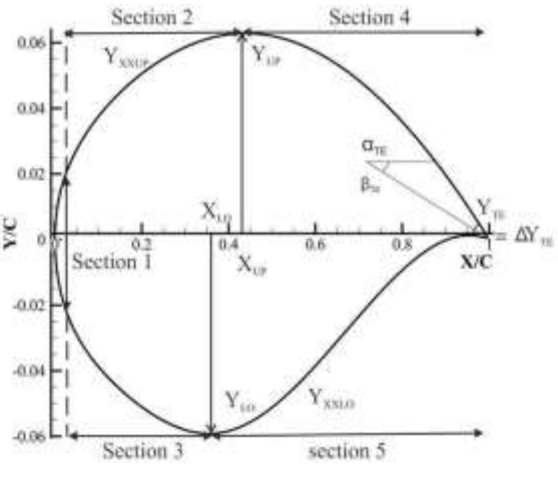

Figure 1 – PARSEC airfoil

The airfoil's geometry is represented by the linear blend of shape functions.

$$Z_K=\sum_{n=1}^{6} a_{n,k}\, X_k^{\frac{n-1}{2}}\ ,\ k=1,2 \tag{1}$$

According to the aforementioned equation, the airfoil is segmented into its upper (k = 1) and lower (k = 2) sections, each developed independently. In this method, with 10 parameters, it's possible to accurately model the geometry for a wide range of desired wing sections which can be utilized for producing input geometries for the neural network.

2.2 Neural Network Architecture

In this section, the architecture of the neural network designed for predicting multiple flow field variables, such as Mach number, pressure, u and v velocity components, and turbulent kinetic energy, within a unified framework is introduced. This neural network employs a loss function calculated independently for each flow field. Equation 2 presents the Mean Squared Error (MSE), which is used as the loss function.

$$\mathrm{MSE}=\frac{1}{n}\sum_{i=1}^{n}(E_i\text{-}P_i)^2 \tag{2}$$

Here, *n* denotes the total number of data points for each output, *E* represents the exact CFD data, and *P* signifies the predicted values. This setup allows the neural network to handle inputs (geometry parameters) and outputs (free stream inputs) and simultaneously output variables, matching the complexity of the problem. The neural network architecture consists of five dedicated input layers, each associated with a specific flow field variable, Mach number, pressure, velocity components and turbulent kinetic energy. The ReLU (Rectified Linear Unit) activation function is employed for nonlinearity and efficiency. During training, 17% validation split is utilized, alongside early stopping and model check pointing. The training process involves a maximum of 2000 epochs and a batch size of 60, ensuring effective learning. Batch normalization, employed in these layers, introduces nonlinearity and improves the learning process. By normalizing the outputs of each layer, it plays a crucial role in enhancing the stability of the training process. This is particularly vital in addressing the challenge of internal covariate shift, a common issue when input variables exhibit significant differences in scale, thereby boosting the model's overall performance.

2.3 Dataset preparation

For effective neural network training, selecting an appropriate dataset is vital. This study uses the Latin Hypercube Sampling (LHS) [13] method, a stratified sampling technique ensuring uniform representation across variable ranges. LHS is particularly advantageous in models with multiple variables, as it captures extreme behaviors and complex interactions better than simple random sampling. The study considers two main input parameters:

1- Geometrical Representation: PARSEC method based on standard form are employed to evaluate neural network training efficiency.

2- Flow Inputs: This includes the Mach number (0.2 to 0.5) and the angle of attack (1 to 5 degrees).

After determining the geometry, a computational grid is generated using a dynamic mesh method, starting with an initial network around a reference airfoil. As the geometry evolves, the grid adapts accordingly. The dynamic mesh method maintains consistent cell numbers by shifting cell centers, allowing for quick grid reconstruction. Linear and torsional spring analogies are used to enhance spatial resolution, numerical accuracy, element angles, and overall computational robustness.

The final computational grid, once established, stores flow parameters within its cells, forming the basis for neural network training outputs.

In this study, the compressible Navier–Stokes equations are solved using a finite volume, cell centered implicit scheme based on the methodology described in Ref. [14] for unstructured grids. Additionally, a two-equation $k-\epsilon$ turbulence model is applied in combination with the Reynolds averaged Navier–Stokes equations.

3. RESULTS

The training process utilizes TensorFlow with Keras, running on top of Python's TensorFlow library [15]. Figure 2 illustrates a typical computational grid used in this study, consisting of 5436 computational cells. Cell values will be used as the neural network's output. The dataset for training the neural network comprises 1300 samples, with 81% allocated to training, 17% to validation, and 2% to testing. After training the network, the mean square error for the trained neural network on the flow field characteristics has reached the order of 10^{-5}. For the kinetic energy, this error has even decreased to 10^{-7}.

Training data is used to train the neural network to recognize patterns and make predictions. Validation data evaluates model performance during training and helps fine tune model parameters. Test data assesses the model's performance on new, unseen data after training and validation, ensuring it generalizes well to new examples. The airfoil parameters are generated based on the range specified in Table 1.

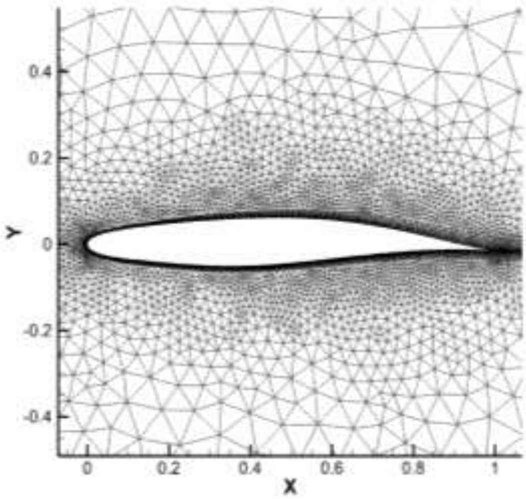

Figure 2 – Computational grid around testing airfoil

TABLE 1: Training range of input parameters.

Parameters	Lower bound	Upper bound
r_{LE}	0.02	0.09
X_{up}	0.4	0.54
Y_{up}	0.05	0.08
Y_{xxup}	-0.61	-0.41
α_{TE}	7.73	17.7
β_{TE}	6.00	16.0
X_{lo}	0.4	0.54
Y_{lo}	-0.08	-0.04
Y_{xxlo}	0.58	1.00
Z_{Te}	0.0	-0.1
Mach number	0.1	0.5
Angle of Attack (degrees)	1	5

Figure 3 displays the MSE results for all flow fields. The neural network training took around 16.30 hours for 1250 samples. With early stopping activated at epoch 1700, training concluded at epoch 1885. All flow variables, except for turbulent kinetic energy, showed similar slopes on the graph. The values of turbulent kinetic energy decreased more rapidly due to their smaller magnitude.
Table 2 presents the prediction results. The Mach number and angle of attack for this test are 0.34 and 3.5 degrees, respectively. The trained network can predict all flow variables for a new case within the range of the trained parameters in approximately 3 seconds, which is 227 times faster than CFD.
Figure 4 displays the predicted contours for the flow field around a testing geometry beside the CFD computed results, emphasizing the network's effectiveness in replicating the flow field distributions and capturing essential flow features. In this figure, the absolute errors between the CFD and predicted results are also presented.

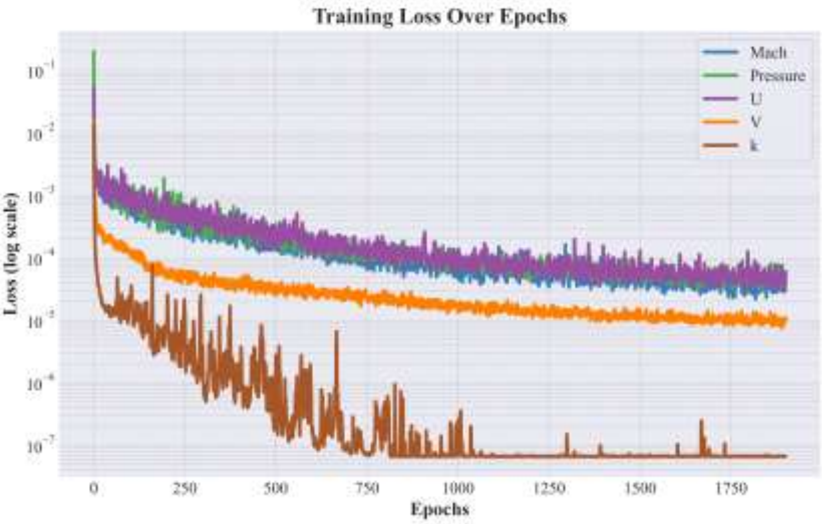

Figure 3 – Training loss over epochs for random test Case, involving Mach number, pressure, u, v, and k flow fields.

TABLE 2: Prediction results of test case at M = 0.34 and AOA = 3.5◦.

Variables	CFD time (s)	Prediction time (s)	MSE (10^{-4})	Speed up
M			5.15	
P			6.18	
U	683	3	5.72	227
V			1.11	
K			0.003	

In certain regions of the pressure field and kinetic energy contours, some discrepancies are observed between the CFD results and the predictions. To visualize the extent of the scatter, the regression results of the predicted fields shown in Figure 5 also indicate some fluctuations compared to the CFD results. Nevertheless, there is a very good overall agreement between the results.
The method used demonstrates that the constructed model has the capability to predict a wide range of arbitrary airfoil sections. This approach can train the process with various flow input parameters and establish their relationship with the flow field, which is the output of the neural network. One of the main features of this method is its ability to train with a very small amount of training data.

Additionally, due to the direct use of CFD data in the training dataset, the predictions are also made in the order of the original results.

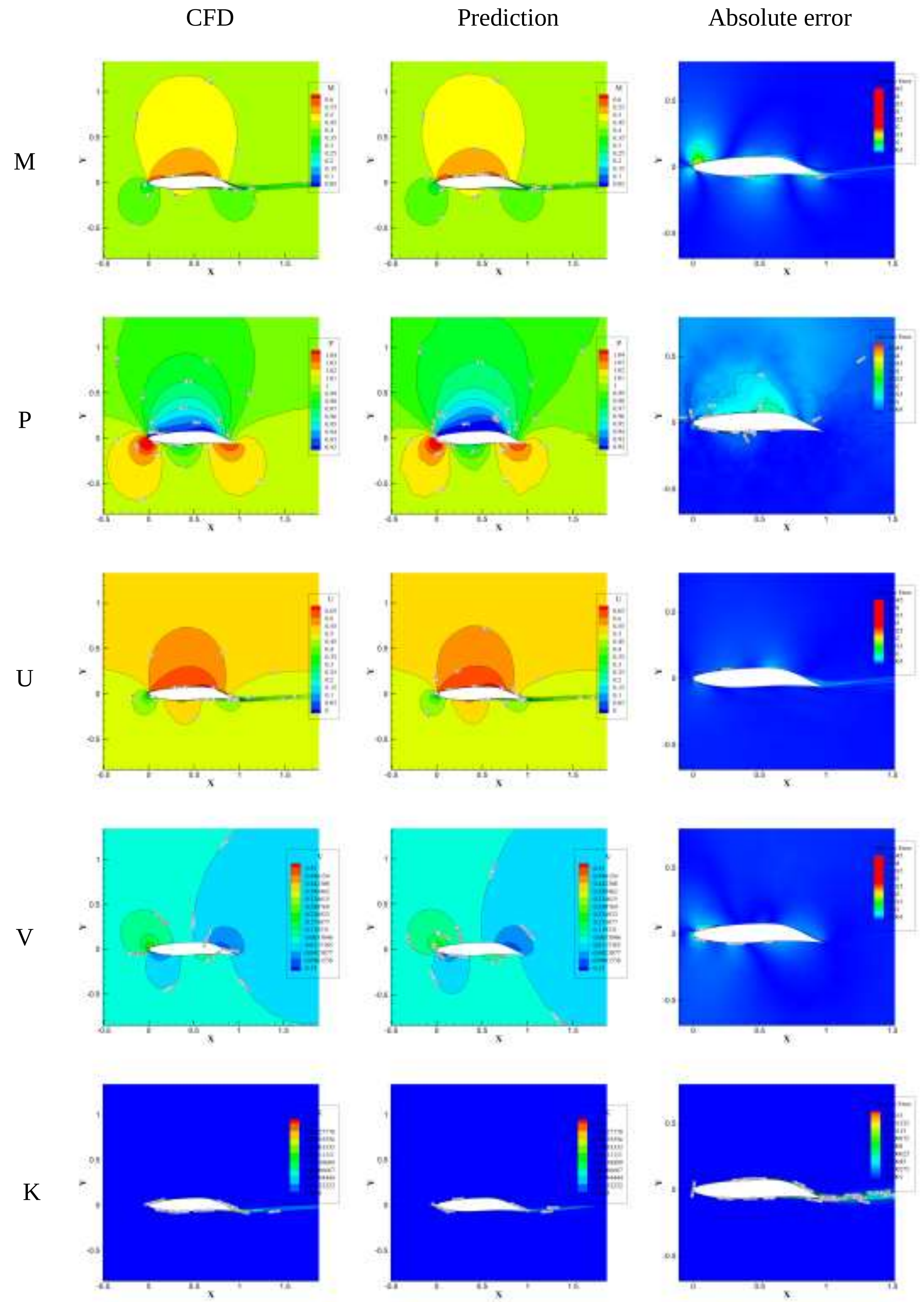

Figure 4 – Computed and predicted results for a testing airfoil at M= 0.34 and AOA= 3.5 degrees.

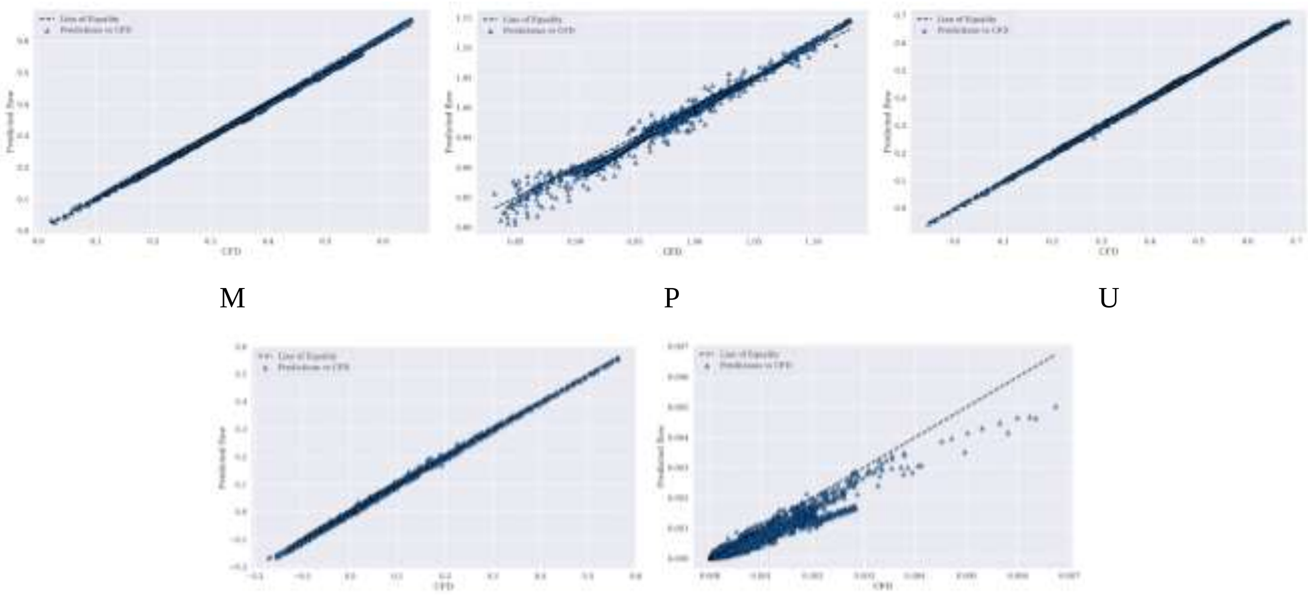

Figure 5 – Dot plot of the predicted flow fields for the test airfoil at M = 0.34, AOA = 3.5 deg.

References

[1] Y. Qiu, J. Bai.Stationary flow fields prediction of variable physical domain based on proper orthogonal decomposition and kriging surrogate model. Chinese J. Aeronaut. 28 44–56, 2015.

[2] A. Shahrokhi and A. Jahangirian, A surrogate assisted evolutionary optimization method with application to the transonic airfoil design, Engineering Optimization 42, 497 (2010).

[3] X. Hui, J. Bai, H. Wang, and Y. Zhang, Fast pressure distribution prediction of airfoils using deep learning, Aerospace Science and Technology 105, 105949 (2020).

[4] X. Guo, W. Li, and F. Iorio, Convolutional neural networks for steady flow approximation, in Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining (2016) pp. 481–490.

[5] T. Miyanawala and R. K. Jaiman, A hybrid data driven deep learning technique for fluid structure interaction, in International conference on off shore mechanics and arctic engineering, Vol. 58776 (American Society of Mechanical Engineers, 2019) p. V002T08A004.

[6] C. Duru, H. Alemdar, O.U. Baran .A deep learning approach for the transonic flow field predictions around airfoils Duru, C., Alemdar, H. and Baran, O.U. A deep learning approach for the transonic flow field predictions around airfoils. Computers & Fluids, 236 105312, 2022.

[7] V. Sekar, Q. Jiang, C. Shu, and B. C. Khoo, Fast flow field prediction over airfoils using deep learning ap- proach, Physics of Fluids 31 (2019).

[8] A. Kashefi, D. Rempe, and L. J. Guibas, A point cloud deep learning framework for prediction of fluid flow fields on irregular geometries, Physics of Fluids 33 (2021).

[9] A. Oyama, S. Obayashi, K. Nakahashi, and N. Hirose, Aerodynamic wing optimization via evolutionary algorithms based on structured coding, Computational Fluid Dynamics Journal 8, 570 (2000).

[10] B. M. Kulfan, Universal parametric geometry representation method, Journal of aircraft 45, 142 (2008).

[11] T. W. Sederberg and S. R. Parry, Free form deformation of solid geometric models, in Proceedings of the 13th annual conference on Computer graphics and interactive techniques (1986) pp. 151–160.
[12] M. Nemati, A. Jahangirian,.A Data-Driven Machine Learning Approach for Turbulent Flow Field Prediction Based on Direct Computational Fluid Dynamics Database J. Appl. Fluid Mech. 17 60–74, 2023.
[13] M. Stein, Large sample properties of simulations using latin hypercube sampling, Technometrics 29, 143 (1987).
[14] A. Jahangirian, M. Hadidoolabi, Unstructured moving grids for implicit calculation of unsteady compressible viscous flows, Int. J. Numer. Methods Fluids 47 (2005) 1107–1113.
[15] F. Chollet et al., Keras: Deep learning library for theano and tensorflow, URL: https://keras. io/k 7, T1 (2015).