2022

A Neural-network Approach to Develop Algebraic Correlations for Heat Transfer and Fluid Flow

Lingnan Lin

Lei Gao

Yunho Hwang

Mark A. Kedzierski

A Neural-network Approach to Develop Algebraic Correlations for Heat Transfer and Fluid Flow

Lingnan LIN[1,*], Lei GAO[2], Yunho HWANG[2], Mark KEDZIERSKI[1]

[1]National Institute of Standards and Technology,
Gaithersburg, Maryland, USA
lingnan.lin@nist.gov

[2]Center for Environmental Energy Engineering, University of Maryland
College Park, Maryland, USA

* Corresponding Author

ABSTRACT

Many heat transfer and fluid flow problems are too complex to model using traditional regression methods. Machine learning (ML) offers a new way to develop predictive models with high accuracy. However, current ML models are often uninterpretable and used as "black boxes". This paper presents an approach to develop explicit, algebraic correlations from neural networks. An interpretable neural network, namely DimNet, is designed. One can train DimNet with experimental or simulation data and then convert the trained network to an explicit, power-law-like piecewise function. Besides being interpretable, DimNet inherits advantages of neural networks in modeling complex nonlinear problems. The mechanism and effectiveness of DimNet and the correlation development approach is further demonstrated by two case studies: 1) correlating simulation data for the friction factor of flow in smooth pipes; 2) correlating experimental data for flow boiling heat transfer coefficient within microfin tubes. Both case studies show DimNet can produce simple, explicit, algebraic correlations that are both statistically and phenomenologically accurate. The presented approach can be potentially used to develop correlations for various thermal-hydraulic problems, such as the pressure drop and heat transfer of single- and multi-phase flow, heat exchangers, and other thermal-hydraulic equipment.

1. INTRODUCTION

Empirical algebraic correlations are an essential element in thermal hydraulic engineering to describe transfer processes, as transfer processes generally involve turbulence for which theoretical solutions are typically not possible. Traditionally, correlations are developed by fitting experimental data to an assumed basis function. This approach often leads to large error or regression divergence for complex nonlinear problems such as multi-phase flow and heat transfer.

Machine learning (ML) is a promising alternative class of methods to develop empirical models, and it is increasingly popular in recent years (Hughes, Kini, and Garimella 2021). Among numerous ML methods, the neural network is the most successful one and has a vast spectrum of applications (Bishop 2006; Schmidhuber 2015). The ability to divide the input space based on the data pattern afford neural networks a powerful capability to model complex problems. The modeling capability is embodied in the universal approximation theorems, which state that neural networks can approximate arbitrary continuous functions to any desired degree of accuracy (Cybenko 1989; Sonoda and Murata 2017).

Despite the superior modeling capability and a wide range of applications, neural networks (and other ML models) remain somewhat controversial, particularly in the field of heat transfer and fluid flow. The major criticism is concerned about the lack of transparency or interpretability, as most of neural networks and other ML models are used as "black boxes". By contrast, traditional regression methods always give simple algebraic equations where the input–

output relationship is clear. The prevailing reluctance toward neural networks and ML also stems from the seemingly lack of connection of these methods to the existing body of knowledge.

This paper presents a new approach to develop algebraic correlations using an interpretable neural network, namely DimNet. DimNet was designed such that it can be converted to a piecewise function of power-law-like equations, which are especially suitable for modeling complex problems of heat transfer and fluid flow. DimNet also inherits neural networks' advantages in data pattern recognition, making it excellent in capturing the different input–output trends in different physical regimes. In what follows, Section 2 describes the methodology of the proposed approach, Section 3 provides two exemplary cases with different number of input dimensions, and Section 4 draws the conclusions.

2. METHODOLOGY

2.1 Formulation

DimNet (Dimensionless Neural Network) is designed primarily for correlating a set of dimensionless quantities with power-law-like relations. The design is based on the consideration that most heat transfer or fluid flow correlations are based on power-law functions. A DimNet model can be expressed by a function f that maps a set of input dimensionless quantities $\{\Pi_1, \Pi_2, \dots, \Pi_L\}$ to the target dimensionless quantity Π_0:

$$\Pi_0 = f(\Pi_1, \Pi_2, \dots, \Pi_L) \tag{1}$$

DimNet is fully connected, feedforward neural network that consists of an input layer, two hidden layers, and an output layer. The notation *L*–*M*–*N*–1 is used to describe the DimNet's configuration, where *L*, *M*, *N*, and 1 denote the width (i.e., number of neurons) of each layer, respectively.

The input layer corresponds to a *L*-dimensional vector, **x**, that consists of the logarithms of the input dimensionless quantities:

$$\mathbf{x} = [x_1\ x_2\ \cdots x_L] = [\ln \Pi_1\ \ln \Pi_2\ \cdots\ \ln \Pi_L] \tag{2}$$

The first hidden layer is activated by the rectified linear unit (ReLU): $\mathrm{ReLU}(z) \triangleq \max(0, z)$, and the second hidden layer is activated by the exponential function: $\mathrm{Exp}(z) \triangleq e^z$. For convenience, the two hidden layers are referred to as the ReLU-layer and Exp-layer, respectively. The output layer computes the linear combination of the outputs of the Exp-layer, giving the final output *y*, which corresponds to the target dimensionless quantity Π_0.

The whole DimNet can be expressed in the following form:

$$y = \sum_{k=1}^{N} w_k^{(3)} \mathrm{Exp}\left[\sum_{j=1}^{M} w_{kj}^{(2)} \mathrm{ReLU}\left(\sum_{i=1}^{L} w_{ji}^{(1)} x_i + b_j^{(1)}\right) + b_k^{(2)}\right] + b^{(3)} \tag{3}$$

where *w* and *b* denote the weight and bias, respectively, both of which are adjusted parameters; the superscripts denote the number of the layer[1].

To ensure numerical stability and facilitate the training efficiency, the input vector is standardized:

$$\mathbf{x}_n \triangleq (\mathbf{x}_n - \mathbf{u})/\mathbf{s} \tag{4}$$

[1] The layers herein are counted using the convention that only includes the hidden layers and the output layer, and the input layer is excluded because they do no computation (Reed and Marks 1999). In case of the DimNet, the first layer is the ReLU layer.

where **u** and **s** are vectors of the mean and the standard deviation of the samples $\{\mathbf{x}_1, \ldots, \mathbf{x}_{\mathcal{N}}\}$, respectively; the output are scaled by the maximum:

$$\hat{y}_n \triangleq \hat{y}_n / \hat{y}_{\max} \tag{5}$$

where $\hat{y}_{\max}$ is the maximum of the labels $\{\hat{y}_1, \ldots, \hat{y}_{\mathcal{N}}\}$.

As explained in Chen (2016) and Hansson and Olsson (2017), the ReLU-layer essentially acts as generating hyperplanes that partition the input space into several regions. Each hyperplane corresponds to a neuron of the ReLU layer. The *j*-th (*j* = 1, 2, …, *M*) hyperplane can be expressed by

$$z_j = 0 \tag{6}$$

where

$$z_j = \sum_{i=1}^{L} w_{ji}^{(1)} x_i + b_j^{(1)} \tag{7}$$

Substituting Eqs. (2), (4), and (5) to Eq. (7) leads to

$$z_j = \ln\left(\prod_{i=1}^{L} \Pi_i^{w_{ji}^{(1)}/s_i}\right) + b_j^{(1)} - \sum_{i=1}^{L} \frac{w_{ji}^{(1)} u_i}{s_i} \tag{8}$$

Each ReLU hyperplane divides the space into two half-spaces, i.e., $z_j^{(1)} > 0$ and $z_j^{(1)} \leq 0$. Thus, a DimNet with *M* neurons in the ReLU layer can create *M* hyperplanes and partition the input space into 2^M regions.

Define a binary variable δ_j:

$$\delta_j \triangleq \begin{cases} 1 \text{ for } z_j > 0 \\ 0 \text{ for } z_j \leq 0 \end{cases} \tag{9}$$

Then an arbitrary region with the index $r = 1,2, \ldots, 2^M$ can be represented by a vector:

$$\mathbf{\Delta}_r = \left(\delta_0, \delta_1, \ldots, \delta_j, \ldots, \delta_M\right) \tag{10}$$

As proved in Lin et al. (2022), the DimNet's equivalent function for the *r*-th region is

$$y = p_0 + \sum_{k=1}^{N} \left(p_k \cdot \prod_{i=1}^{L} \Pi_{\mathrm{i}}^{q_{k,i}}\right) = p_0 + p_1 \prod_{i=1}^{L} \Pi_i^{q_{1,i}} + p_2 \prod_{i=1}^{L} \Pi_i^{q_{2,i}} + \cdots + p_N \prod_{i=1}^{L} \Pi_i^{q_{N,i}} \tag{11}$$

where $p_0 = \hat{y}_{\max} b^{(3)}$ and

$$p_k = \hat{y}_{\max}\, w_k^{(3)} \mathrm{Exp}\left[\sum_{j=1}^{M} \delta_j w_{kj}^{(2)} \left(b_j^{(1)} - \sum_{i=1}^{L} w_{ji}^{(1)} \frac{u_i}{s_i}\right) + b_k^{(2)}\right] \tag{12}$$

$$q_{k,i} = \frac{1}{s_i} \sum_{j=1}^{M} \delta_j w_{kj}^{(2)} w_{ji}^{(1)} \tag{13}$$

2.2 Procedure

DimNet can be trained in the same way that common neural networks are trained, and common neural network training algorithms and techniques generally apply to DimNet. The procedure used for training DimNets in the present study is as follows. The Adam algorithm (Kingma and Ba 2014) with a fixed learning rate of 0.001 and the default Adam parameters ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 10^{-8}$, no weight decay) were used for training. In each iteration (or "epoch"), the entire training set was used to compute the gradients and update the parameters, and the training samples were shuffled to avoid biased optimization. The training was considered converged when the loss (i.e., mean squared error, or MSE) did not decrease by 0.0001% in 100 consecutive iterations. The convergence typically occurred after 10^4 to 10^5 epochs for the studied cases, which took one to several minutes on an average personal computer. It is well known that the solution for neural network training is sensitive to the initial values of the network parameters, and the training algorithm can be usually trapped in local minima or saddle points, meaning the global minimum is not guaranteed (in fact, it is very rare). In this study, each model was randomly initialized using the Kaiming algorithm (He et al. 2015) and repeatedly trained for 100 times to select a solution with the lowest training loss. All neural networks in this study were constructed and trained based on the PyTorch package.

The trained DimNet can be converted to a piecewise function following the procedure below:

i. Obtain the DimNet's parameters, i.e., weights and biases.
ii. Compute hyperplane expressions z_j using Eqs. (6) and (7) for $j = 1, 2, \ldots, L$ where L is the dimension of the input vector.
iii. Generate region vectors $\mathbf{\Delta}_r$, as defined in Eq. (10), for $r = 1, 2, \ldots, 2^M$ where M is the number of neurons in the ReLU-layer, then substitute Eq. (9) to write $\mathbf{\Delta}_r$ in terms of the inequalities of z_j. For example, if $M = 2$, all possible $\mathbf{\Delta}_r$ are [1,1], [1,0], [0,1], and [0,0], which are equivalent to [$z_1 > 0$, $z_2 > 0$], [$z_1 > 0$, $z_2 \leq 0$], [$z_1 < 0$, $z_2 > 0$], and [$z_1 \leq 0$, $z_2 \leq 0$], respectively.
iv. For each region $\mathbf{\Delta}_r$, the corresponding algebraic function can be written in Eq. (11), with the coefficients computed by Eqs (12) and (13). These functions collectively constitute the piecewise function.

3. CASE STUDIES

3.1 One-dimensional case: friction factor for flow within smooth pipes

The flow in smooth pipes is a classical fluid mechanics problem where the friction factor (λ) is a sole function of the Reynolds number (Re). Three distinct physical regimes exist over the entire range of the Reynolds number: laminar flow (Re $\lesssim$ 2300), transitional flow (2300 $\lesssim$ Re $\lesssim$ 4000), and turbulent flow (Re $\gtrsim$ 4000). Avci and Karagoz (2019) proposed a universal model that combines laminar, transitional, and turbulent regime into a single equation:

$$\lambda = \alpha\lambda_{\mathrm{L}} + (1-\alpha)\lambda_{\mathrm{T}} \tag{14}$$

where λ_{L} and λ_{T} are the friction factor for fully-developed laminar and turbulent flows, respectively; α represents the ratio of laminar component to the flow. For smooth pipe, α is computed by:

$$\alpha = \exp[-(\mathrm{Re}/2560)^8] \tag{15}$$

The laminar friction factor has an analytical solution:

$$\lambda_{\mathrm{L}} = 64/\mathrm{Re} \tag{16}$$

The turbulent friction factor can only be empirically or semi-empirically computed. A classic semi-empirical model for smooth pipe was derived by Prandtl from the logarithmic velocity profile (Durand 1935):

$$\frac{1}{\sqrt{\lambda_{\mathrm{T}}}} = -2\log_{10}\left(\frac{2.51}{\mathrm{Re}\sqrt{\lambda_{\mathrm{T}}}}\right) \tag{17}$$

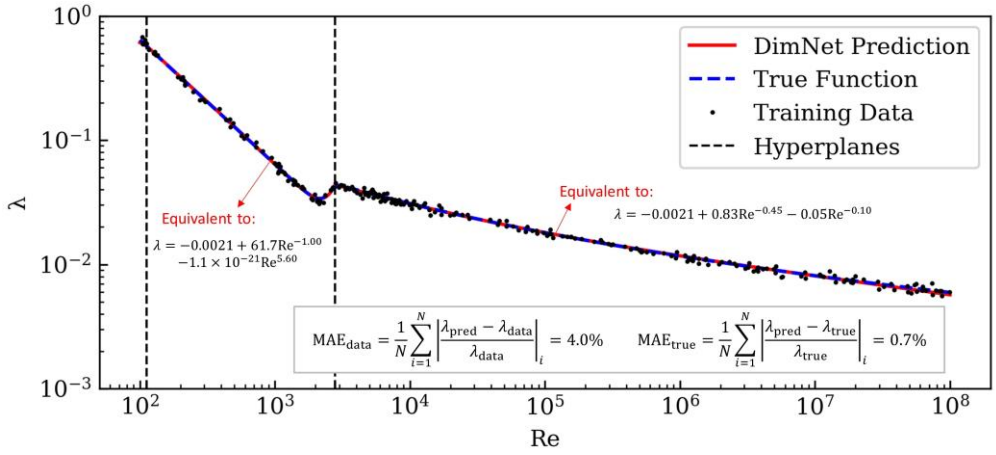

Figure 1 Training data, DimNet-λ model prediction, and true function's calculation for friction factor (λ) of smooth pipe flow

In this case study, Eqs. (14) – (17) were collectively used as the "true function" for DimNet to "learn", with Re as the only input variable and λ as the output variable. It is noted that this paper is focused on how well DimNet "learns" the "true function", while the intrinsic accuracy of the true function is beyond the scope.

A training dataset with 250 synthetic data samples were generated using the true function. 200 of the samples were randomly drawn from the log-uniform distribution for the interval of Re from 10^2 to 10^8. The rest 50 samples were sampled in the same way for the interval [10^3, 10^4] in order to increase the data density in this region where the laminar–turbulent transition occurs. The output value of each sample was calculated by the true function added by a random noise to simulate experimental errors. The noise was drawn from a Gaussian distribution with zero mean and the standard deviation being 5 % of the true value. The DimNet with 1-2-2-1 configuration was trained with the 250 noisy samples following the procedure in Section 2.2. The trained network is hereafter referred to as the DimNet-λ model, and was converted to the following explicit piecewise function:

$$\lambda = -0.0021 + 61.7\mathrm{Re}^{-1.00} - 1.1\times10^{-21}\mathrm{Re}^{5.60} \quad (\text{for } z_1 > 0,\ z_2 > 0) \tag{18-1}$$
$$\lambda = -0.0021 + 0.83\mathrm{Re}^{-0.45} - 0.05\mathrm{Re}^{-0.10} \quad (\text{for } z_1 > 0,\ z_2 \le 0) \tag{18-2}$$
$$\lambda = -0.0021 + 7.49\mathrm{Re}^{-0.55} - 6.7\times10^{-22}\mathrm{Re}^{5.70} \quad (\text{for } z_1 \le 0,\ z_2 > 0) \tag{18-3}$$
$$\lambda = 0.1287 \quad (\text{for } z_1 \le 0,\ z_2 \le 0) \tag{18-4}$$

where

$$z_1 = \ln(\mathrm{Re}^{0.484}) - 2.284 \tag{19-1}$$
$$z_2 = \ln(\mathrm{Re}^{-0.485}) + 3.850 \tag{19-2}$$

Note that the coefficients for the above piecewise function were rounded to give finite decimal numbers. Caution should be exercised when rounding the coefficients, especially when the values are very small. For example, the coefficient 1.1×10^{-21} in Eq. (18-1) should not be rounded to zero, otherwise the rightmost term would vanish, and the resulting function would not have the reversed, increasing trend for Re between approximately 2000 and 3000 (see Figure 1).

Figure 1 shows the true function's curve, the training samples, and the DimNet-λ model's predicted curve. The mean-absolute-error (MAE) between the model prediction and the training data is 4.0 %, which is within the standard deviation of the artificial noise imposed to the data (i.e., 5 %). The MAE between the DimNet prediction and the true function is even lower — 0.7 %, showing that DimNet adequately learns the true function from the noisy data.

As indicated by Eqs. (18) and (19), the 1-2-2-1 DimNet creates two hyperplanes: $z_1 = 0$ and $z_2 = 0$, dividing the entire input space into 4 regions. Solving $z_1 = 0$ and $z_2 = 0$ with Eq. (19) leads to Re = 112 and Re = 2788, respectively. Then

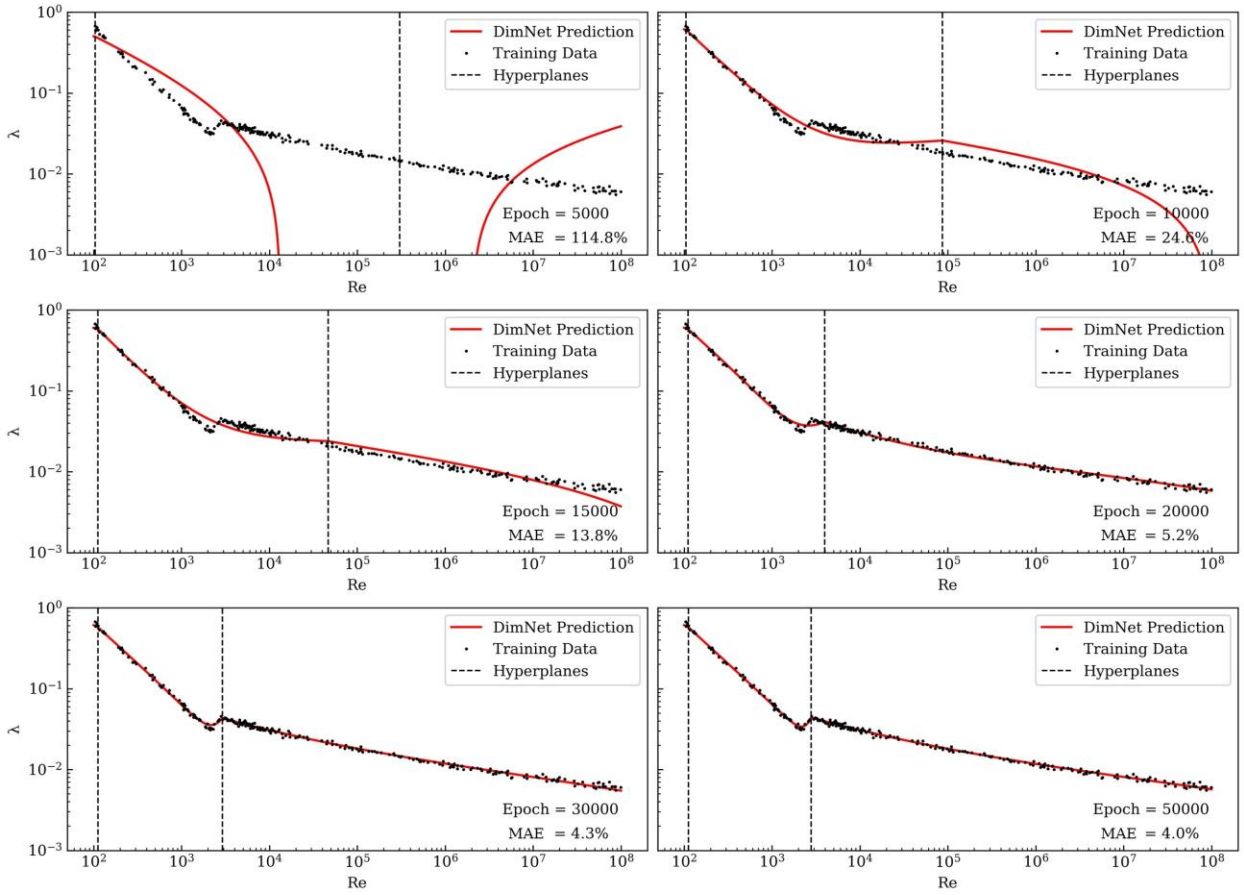

Figure 2 Evolution of the DimNet-λ model and the associated hyperplanes during the training process.

the four regions can be represented in terms of Re: 1) $112 < Re < 2788$; 2) $Re \geq 2788$; 3) $Re \leq 112$; 4) $Re \leq 112$, $Re \geq 2788$ (which is not a valid interval). The sub-equation of Eq. (18) corresponding to each region can be considered as the best-fit model or curve to the training data in that region. Then, it can be implied that DimNet essentially uses the hyperplanes to classify the training data and simultaneously fit the data to a piecewise curve, which may be described as "simultaneous classification and regression".

The hyperplanes and the functional forms of the fitting curves are controlled by the ReLU-layer and the Exp-layer of DimNet, respectively. In this case, the ReLU-layer has 2 neurons, thereby 2 hyperplanes and $2^2 = 4$ regions are created. The Exp-layer has 2 neurons, then the functional form is the sum of two power-law terms. Obviously, more flexibility can be achieved by increasing neurons in ReLU-layers and the Exp-layers, which, however, also increases the risk of overfitting. Knowing the relationship between the DimNet's configuration and characteristics of its equivalent piecewise function is important to choose a proper configuration for problems to be solved.

Note that the hyperplane Re = 2788 is approximately the onset of the turbulent regime for the true function. This is a great embodiment that the DimNet recognizes the pattern (or trend) variation underlying the training data and makes classification accordingly. One may ask why one of the hyperplanes was not placed in the end of the laminar regime, which would ideally separate laminar and transitional regimes. The reason is that current optimization algorithms for training neural networks are generally unable to reach the global minimum of the error surface, and hence the trained neural network is typically near optimal but not globally optimal. Consequently, when the algorithm "found" the neural network parameters that satisfactorily fit the training data, it reached a local minimum that gave a relatively small error, and were unable to escape this local minimum to search for lower error surfaces.

Figure 2 shows the evolution of the DimNet-λ model and the associated hyperplanes over the course of the training process. It can be seen that the hyperplanes and fitting curves are simultaneously and continuously adjusted until a satisfactory fit is obtained. An intuitive analogy for this one-dimensional DimNet is the spline, while the hyperplanes

are essentially knots for the spline. The difference between the DimNet's equivalent spline and a conventional interpolating spline is that the latter's knots are placed "manually" and before curve-fitting, while the former's knots are placed automatically by an optimization algorithm and in parallel with curve-fitting.

Because the hyperplane Re = 112 (or $z_1 = 0$) is near the boundary of the computation domain and almost not possible in practice, the regions #3 and #4 (i.e., $z_1 \leq 0$ or $\mathrm{Re} \leq 112$) are practically redundant and can be omitted. Then, the DimNet-λ model, or Eqs. (18) and (19), can be simplified as:

$$\lambda = -0.0021 + 61.7\mathrm{Re}^{-1.00} - 1.1 \times 10^{-21}\mathrm{Re}^{5.60} \quad (\text{for } \mathrm{Re} < 2788) \tag{20-1}$$

$$\lambda = -0.0021 + 0.83\mathrm{Re}^{-0.45} - 0.05\mathrm{Re}^{-0.10} \quad (\text{for } \mathrm{Re} \geq 2788) \tag{20-2}$$

3.2 Multi-dimensional case: flow boiling heat transfer within microfin tubes

In our recent paper (Lin et al. 2022), DimNet was applied to developing a universal correlation for flow boiling heat transfer within microfin tubes using real experimental data. The following is a brief description for this application.

The modeling objective here was to correlate the pre-dryout flow boiling heat transfer coefficient (h_{tp}) with pertinent parameters (including those representing operational conditions, geometries of microfin tubes, and fluid properties). These parameters are grouped into one target dimensionless quantity: $\mathrm{Nu} = h_{\mathrm{tp}}D_{\mathrm{h}}/k_{\mathrm{f}}$ and seven input dimensionless quantities: [Bo, $\mathrm{P_R}$, $\mathrm{M_w}$, $\mathrm{Re_f}$, $\mathrm{Pr_f}$, Co, Bd], which were identified based on their physical significances and model performances.

The database was compiled by carefully examining and collecting flow boiling measurements in the existing literature. It has 7349 data points from 31 sources for 16 refrigerants. The refrigerants include 12 pure refrigerants (CO_2, R1234yf, R1234ze(E), R1234ze(Z), R1233zd(E), R1224yd(Z), R134a, R32, R125, R161, R22, R245fa) and 4 refrigerant mixtures (R513A, R410A, R410B, R450A). The refrigerant mixtures are with temperature glides less than 1 K, thus they were treated in the same manner as pure refrigerants.

A DimNet with 7-2-2-1 configuration was used to develop the correlation. Considering DimNet was designed to deal with dimensionless inputs and output, an extra step was added to the network to multiply the output, which corresponds to Nu, by a factor of $k_{\mathrm{f}}/D_{\mathrm{h}}$ so that the resulting network's output corresponds to h_{tp}. Note that this step was incorporated into the forward- and back-propagation of the network. The resulting network is referred to as the DimNet-HTC model.

The DimNet-HTC model was first trained with 80 % of the entire database, while the rest of data were withheld for validation. The trained model predicted the withheld dataset with a MAE of 13.9 %, and 89.1 % of the withheld data were within ± 30 % of the prediction. The DimNet-HTC was compared with two extant non-ML-based models (Mehendale 2018; Tang and Li 2018), which were chosen because they were the most recent, and both of them were reported to outperform earlier models. The MAEs (evaluated on the withheld data) for the Tang–Li model and the Mehendale model are 20.3 % and 33.9 %, respectively, which are both significantly higher than that of the DimNet-HTC model.

The DimNet-HTC model was then trained with the entire database. Following the procedure in Section 2.2, the final model can be converted to the following function:

$$h_{\mathrm{tp}} = \begin{cases} \left(\frac{\lambda_{\mathrm{f}}}{D_{\mathrm{h}}}\right)\left[\begin{matrix}7730\left(\mathrm{Re_f}^{0.1929}\mathrm{Pr_f}^{0.3359}\mathrm{Co}^{-0.0318}\mathrm{Bo}^{0.1912}\mathrm{P_R}^{0.0555}\mathrm{M_w}^{-0.3428}\mathrm{Bd}^{0.2880}\right) \\ +3.094\left(\mathrm{Re_f}^{-0.1039}\mathrm{Pr_f}^{-0.8458}\mathrm{Co}^{-0.2494}\mathrm{Bo}^{-0.2031}\mathrm{P_R}^{-0.1398}\mathrm{M_w}^{0.7766}\mathrm{Bd}^{-0.0485}\right) - 581.3\end{matrix}\right], & \text{for } z_1 > 0, z_2 > 0 \\ \left(\frac{\lambda_{\mathrm{f}}}{D_{\mathrm{h}}}\right)\left[\begin{matrix}7803\left(\mathrm{Re_f}^{0.2044}\mathrm{Pr_f}^{0.3081}\mathrm{Co}^{-0.0529}\mathrm{Bo}^{0.1954}\mathrm{P_R}^{0.0509}\mathrm{M_w}^{-0.3207}\mathrm{Bd}^{0.3128}\right) \\ +2.675\left(\mathrm{Re_f}^{-0.2797}\mathrm{Pr_f}^{-0.4217}\mathrm{Co}^{0.0724}\mathrm{Bo}^{-0.2674}\mathrm{P_R}^{-0.0697}\mathrm{M_w}^{0.4389}\mathrm{Bd}^{-0.4281}\right) - 581.3\end{matrix}\right], & \text{for } z_1 > 0, z_2 \leq 0 \\ \left(\frac{\lambda_{\mathrm{f}}}{D_{\mathrm{h}}}\right)\left[\begin{matrix}528.9\left(\mathrm{Re_f}^{-0.0115}\mathrm{Pr_f}^{0.0278}\mathrm{Co}^{0.0211}\mathrm{Bo}^{-0.0042}\mathrm{P_R}^{0.0046}\mathrm{M_w}^{-0.0221}\mathrm{Bd}^{-0.0248}\right) \\ +121.5\left(\mathrm{Re_f}^{0.1758}\mathrm{Pr_f}^{-0.4241}\mathrm{Co}^{-0.3218}\mathrm{Bo}^{0.0643}\mathrm{P_R}^{-0.0701}\mathrm{M_w}^{0.3377}\mathrm{Bd}^{0.3796}\right) - 581.3\end{matrix}\right], & \text{for } z_1 \leq 0, z_2 > 0 \\ \left(\frac{\lambda_{\mathrm{f}}}{D_{\mathrm{h}}}\right) \times 57.7, & \text{for } z_1 \leq 0, z_2 \leq 0 \end{cases} \tag{21}$$

where h_{tp}, λ_{f}, and D_{h} are in SI units, and z_1, z_2 are given by:

$$z_1 = \ln\left(\mathrm{Re_f^{0.4197}Pr_f^{0.6327}Co^{-0.1086}Bo^{0.4012}P_R^{0.1045}M_w^{-0.6585}Bd^{0.6423}}\right) + 5.507 \quad (22\text{-}1)$$

$$z_2 = \ln\left(\mathrm{Re_f^{0.1527}Pr_f^{-0.3684}Co^{-0.2796}Bo^{0.0559}P_R^{-0.0609}M_w^{0.2933}Bd^{0.3298}}\right) + 0.1263 \quad (22\text{-}2)$$

Figure 3 shows the overall performance of the DimNet-HTC model in comparison with the Tang–Li model and the Mehendale model. The MAE, P_{30} (i.e., percentage of data predicted to within ± 30%), and RMSE (root-mean-square-error) of the DimNet-HTC model for the entire dataset are 13.8 %, 89.6 %, and 1.1 kW/m^{2}K, respectively. As shown in Figure 3, the DimNet-HTC model outperforms the Tang–Li model and the Mehendale model in terms of each of these metrics.

Figure 4 compares the predicted h_{tp} vs. x_q curves with experimental data of R134a and CO_2 for selected conditions. The data of R134a and CO_2 are from Longo et al. (2017) and Gao et al. (2007), respectively. Overall, the model prediction agrees with experimental data in terms of both magnitude and trend. The MAE for each h_{tp} vs. x_q curve in Figure 4 is between 5 % and 15 %. All data points in Figure 4 are within ± 30 % predictive error bands, with the only exception of CO_2 under the conditions of $x_q > 0.7$, q = 5.8 kW/m^2 and G = 380 kg/m^{2}s, as shown in Figure 4 (c).

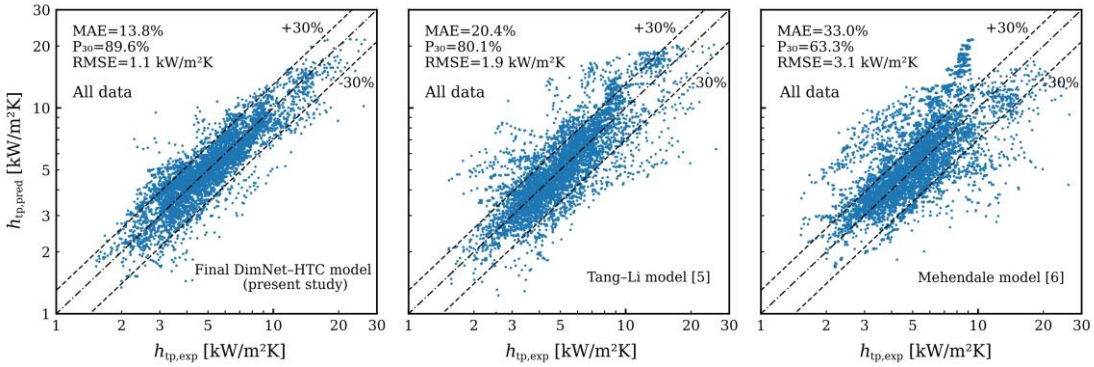

Figure 3 Comparison of experimental data versus the prediction of the DimNet–HTC model and extant correlations (Mehendale 2018; Tang and Li 2018). Logarithmic scale is used on both horizontal and vertical axes.

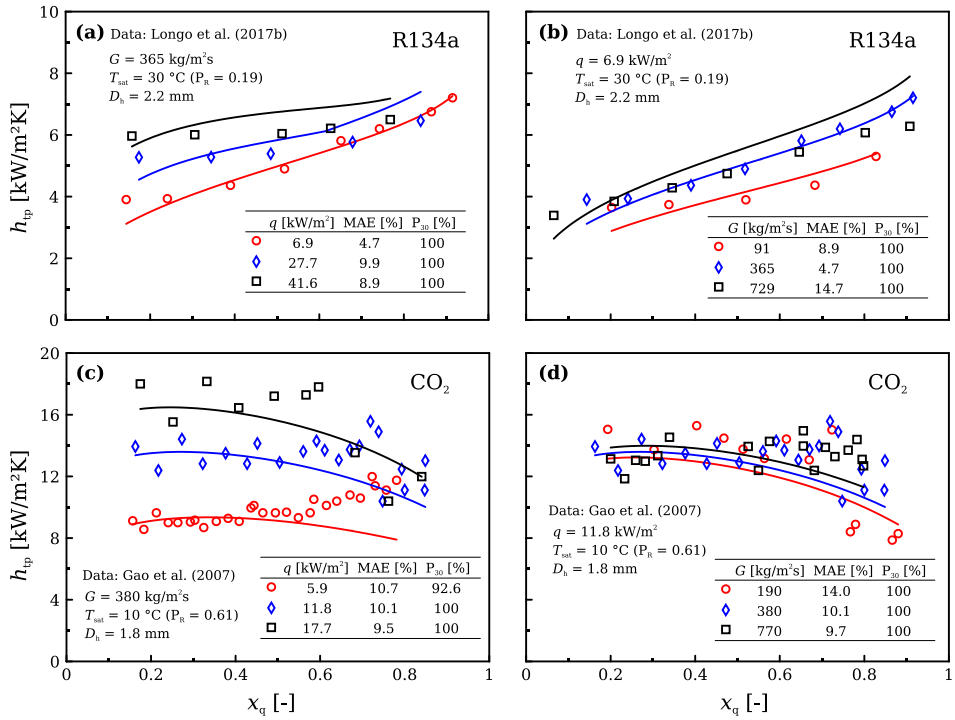

Figure 4 Comparison of the predicted h_{tp} vs. x_q curve by the DimNet-HTC model with experimental data.

4. CONCLUSIONS

The approach presented in this paper enables one to develop explicit algebraic correlations by training neural networks. This approach overcomes the interpretability bottleneck of neural networks, allowing the trained network to be converted to an explicit, algebraic, piecewise function, while keeping the superior predictive power of neural network.

A special neural network, DimNet, is designed such that it can be converted to a power-law-like piecewise function. This design is based on the consideration that most heat transfer or fluid flow correlations are based on power-law functions. The properties of the equivalent piecewise function of DimNet are related to its network configuration parameters, which is demonstrated mathematically in Section 2 and by examples in Section 3. The knowledge of this relationship can be used to guide the construction of the network (i.e., setting the number of neurons in each of the layers) and effectively adjust the model's flexibility based on the prior domain knowledge.

Two case studies have been presented to illustrate the feasibility of this approach and the modeling capability of DimNet. Both case studies show DimNet can produce simple, explicit, algebraic correlations that are both statistically and phenomenologically accurate. The excellent prediction capability of DimNet was attributed to the neural network's ability to automatically classify the data into optimal regions and simultaneously correlate the data within each region. It is expected that DimNet and the presented approach can be used to develop correlations for various thermal-hydraulic problems, such as the heat transfer and pressure drop of single- and multi-phase flow, heat exchangers, and other thermal-hydraulic equipment.

NOMENCLATURE

$\mathbf{b}, b$	bias of neural network
Bd	modified Bond number, $\pi g \rho_{\mathrm{f}} e D_{\mathrm{t}}/(8 N_{\mathrm{fin}} \sigma)$
Bo	boiling number, $q/(G h_{\mathrm{lv}})$
Co	convection number, $\left[(1-x_{\mathrm{q}})/x_{\mathrm{q}}\right]^{0.8} (\rho_{\mathrm{v}}/\rho_{\mathrm{L}})^{0.5}$
c_p	specific heat
D_{h}	hydraulic diameter
D_{t}	inner diameter at the fin tip
e	fin height
G	mass velocity
h_{tp}	two-phase heat transfer coefficient
h_{lv}	enthalpy of vaporization
L, M, N	parameters of DimNet's architecture
k	thermal conductivity
$\mathrm{M_w}$	molecular mass
N_{fin}	number of fins in the microfin tube
Nu	two-phase Nusselt number, $h_{\mathrm{tp}} D_{\mathrm{h}}/k$
$\mathrm{Re_f}$	two-phase Reynolds number for liquid flow, $(1-x_{\mathrm{q}}) G D_{\mathrm{h}}/\mu_{\mathrm{f}}$
p, q	parameters in the piecewise form of DimNet
p_{sat}	saturation pressure
$\mathrm{P_{30}}$	percentage of data predicted to within ± 30 %
$\mathrm{P_R}$	reduced pressure
Pr	Prandtl number, $c_p \mu/k$
q	heat flux
$\mathbf{s}, s$	standard deviation of inputs
T_{sat}	saturation temperature
$\mathbf{u}$	mean of neural network input
$\mathbf{W}, w$	weights of neural network
We	Weber number, $G^2 D_{\mathrm{h}}/(\rho_{\mathrm{f}} \sigma)$
$\mathbf{x}, x$	input of neural network (or "feature")
x_{q}	vapor quality
y	output of neural network (or "target")
z	inactivated value of a hidden unit

Greek symbols

α	helix angle
β	apex angle
λ	friction factor
μ	viscosity
Π	dimensionless group
ρ	density
σ	surface tension
δ	binary variable

Subscripts

f	liquid
v	vapor

REFERENCES

Avci, A., and I. Karagoz. 2019. "A New Explicit Friction Factor Formula for Laminar, Transition and Turbulent Flows in Smooth and Rough Pipes." *European Journal of Mechanics, B/Fluids* 78:182–87.

Bishop, Christopher M. 2006. *Pattern Recognition and Machine Learning*. Springer.

Chen, Kevin K. 2016. "The Upper Bound on Knots in Neural Networks." (November):1–19.

Cybenko, G. 1989. "Approximation by Superpositions of a Sigmoidal Function." *Mathematics of Control, Signals, and Systems* 2(4):303–14.

Durand. 1935. *Aerodynamic Theory*. Vol. 3. Springer.

Gao, Lei, Tomohiro Honda, and Shigeru Koyama. 2007. "Experiments on Flow Boiling Heat Transfer of Almost Pure CO2 and CO2-Oil Mixtures in Horizontal Smooth and Microfin Tubes." *HVAC and R Research* 13(3):415–25.

Hansson, Magnus, and Christoffer Olsson. 2017. "Feedforward Neural Networks with ReLU Activation Functions Are Linear Splines."

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. "Delving Deep into Rectifiers: Surpassing Human-Level Performance on Imagenet Classification." *Proceedings of the IEEE International Conference on Computer Vision* 2015 Inter:1026–34.

Hughes, Matthew T., Girish Kini, and Srinivas Garimella. 2021. "Status, Challenges, and Potential for Machine Learning in Understanding and Applying Heat Transfer Phenomena." *Journal of Heat Transfer* 143(12).

Kingma, Diederik P., and Jimmy Ba. 2014. "Adam: A Method for Stochastic Optimization." *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings* 1–15.

Lin, Lingnan, Lei Gao, Mark A. Kedzierski, and Yunho Hwang. 2022. "A General Model for Flow Boiling Heat Transfer in Microfin Tubes Based on a New Neural Network Architecture." *Energy and AI* 8:100151.

Longo, Giovanni A., Simone Mancin, Giulia Righetti, Claudio Zilio, and Luca Doretti. 2017. "Saturated R134a Flow Boiling inside a 4.3 Mm Inner Diameter Microfin Tube." *Science and Technology for the Built Environment* 23(6):933–45.

Mehendale, Sunil. 2018. "A New Heat Transfer Coefficient Correlation for Pure Refrigerants and Near-Azeotropic Refrigerant Mixtures Flow Boiling within Horizontal Microfin Tubes." *International Journal of Refrigeration* 86:292–311.

Reed, Russell, and Robert J. II Marks. 1999. *Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks*. Cambridge, MA, USA: The MIT Press.

Schmidhuber, Jürgen. 2015. "Deep Learning in Neural Networks: An Overview." *Neural Networks* 61:85–117.

Sonoda, Sho, and Noboru Murata. 2017. "Neural Network with Unbounded Activation Functions Is Universal Approximator." *Applied and Computational Harmonic Analysis* 43(2):233–68.

Tang, Weiyu, and Wei Li. 2018. "A New Heat Transfer Model for Flow Boiling of Refrigerants in Micro-Fin Tubes." *International Journal of Heat and Mass Transfer* 126:1067–78.

ACKNOWLEDGMENT

This work was funded by the National Institute of Standards and Technology (NIST), United States.