BNL-222878-2022-COPA

Machine Learning-based Prediction of Departure from Nucleate Boiling Power for the PSBT Benchmark

U. Rohatgi

Submitted to the Advances in Thermal Hydraulics (ATH 2022) Conference
to be held at Anaheim, CA
June 12 - 16, 2022

Nonproliferation and National Security Department

Brookhaven National Laboratory

U.S. Department of Energy

USDOE Office of Nuclear Energy (NE), Fuel Cycle Technologies (NE-5)

DISCLAIMER

Machine Learning-based Prediction of Departure from Nucleate Boiling Power for the PSBT Benchmark

Chaitee Godbole, Gregory Delipei, Xu Wu, and Maria Avramova
North Carolina State University
Burlington Engineering Laboratory, Raleigh, NC, 27695 USA
cgodbol@ncsu.edu; gkdelipe@ncsu.edu; xwu27@ncsu.edu; mnavramo@ncsu.edu

Upendra Rohatgi
Brookhaven National Laboratory
Upton, New York, USA
rohatgi@bnl.gov

ABSTRACT

Machine Learning (ML) has seen an exponential growth in its applications due to its advanced data-driven prediction capabilities. The study presents a data-driven approach as a preliminary attempt to predict the power at which departure from nucleate boiling (DNB) occurs in pressurized water reactors (PWRs) by constructing an advanced ML algorithm that takes outlet pressure, inlet temperature and inlet mass flux as the input features. DNB is a critical heat flux (CHF) phenomenon seen in PWRs. The experimental data from the PWR subchannel and bundle tests (PSBT) benchmark is first used to train an artificial neural network (ANN) to predict the DNB power, which produces a root mean square error (RMSE) of 6.89 kW/m when tested on a blind subset of the PSBT data. Since the PSBT dataset is relatively small to train an accurate ANN, a data augmentation methodology based on generative adversarial networks (GANs) is used to expand the training dataset. By assuming that the real data follows a certain distribution, GANs try to learn that underlying distribution to generate similar synthetic data to augment the database and to improve the predictive capabilities of the ANN. The data generated from GANs are validated using 1-nearest neighbor and kernel maximum mean discrepancy. To further ensure data from GAN is similar to PSBT, the data is tested and filtered out using the sub-channel thermal-hydraulic code CTF. The results indicate that with the addition of 120 data points from GAN the RMSE reduces to 4.84 kW/m showing promising results for future developments.

KEYWORDS
DNB, ML, GAN, CTF, PSBT

1. INTRODUCTION

Departure from nucleate boiling (DNB), a phenomenon corresponding to critical heat flux (CHF) conditions seen at subcooled and low-quality flow boiling is a major limiting factor in the design and operation of PWRs [1]. This phenomenon is encountered in a variety of two-phase flow boiling systems like high-power microprocessor cooling, refrigeration industry, medical technology fields and nuclear power plants [2]. DNB is said to occur when there is a sharp deterioration of the heat transfer at the

heater/coolant interface due to the formation of a stable vapor film that blankets the heated surface and prevents flow of liquid to the surface leading to a potentially catastrophic escalation of the heater temperature [3]. The paper first aims to predict the power at which DNB occurs in a computationally inexpensive way.

With recent advances in computational capabilities and optimization techniques, the methods based on ML provide an alternative approach to existing data-driven domain knowledge-based tools [4]. These tools are extremely useful in engineering fields where the physical phenomenon in consideration (DNB) is complex. Within this category, an artificial neural network (ANN), also called feed forward neural network, multi-layer perceptron, or deep neural network when there are multiple hidden layers, is a very promising choice as it has been shown to serve as a universal approximator of any nonlinear relations [5] [6]. This paper focuses on the utilization of ANN to predict the linear power at which DNB occurs. The input features are outlet pressure (bar), inlet temperature (ºC) and mass flux (kg/m^2hr), and the output feature is the linear power (kW/m) at which DNB occurs. Different architectures of the ANN are studied to find the optimal number of hidden neurons, hidden layers and the most efficient training algorithm.

The dataset used to train and test the ANN is taken from test series 2, assembly A2 from phase II (DNB benchmark) of the PSBT benchmark which consists of 76 data points [7]. A dataset consisting of only 76 data points is considered very small. Therefore, it is challenging to train an accurate ML model with such a small dataset, especially for an ANN, whose number of parameters (weights and bias) increase quickly with the number of layers and neurons per layer. To alleviate the data scarcity issue, the main focus of this paper lies in augmenting the dataset with deep generative modeling, more specifically, the generative adversarial networks (GANs). GANs have received wide attention for their potential in the field of ML to learn high-dimensional, complex real data distributions [8]. GANs are first trained using data from the PSBT benchmark, then they are used to generate synthetic data which "behaves" very close to the real measurement data. Thus, each data point from GAN consists of outlet pressure, inlet temperature, inlet mass flux and the linear power at which DNB occurs. To validate the data generated by GAN metrics like the leave-one-out accuracy corresponding to 1-nearest neighbor, and the kernel maximum mean discrepancy are used. Both these metrics test how similar the generated data from GAN is to the PSBT data. To further ensure the similarity between the GAN data and the PSBT data; an additional method is applied that utilizes the advanced thermal hydraulic code CTF [9]. To do this, PSBT data is first simulated on CTF and the power values at which DNB occurs as predicted by CTF are compared to the power values in the PSBT data. The range of errors between these power values are then used as an acceptable range to validate artificially generated data from GAN. Data generated by the GAN are simulated on CTF and the power values predicted by CTF are compared to the power values of data from GAN. Only those data points from GAN are then chosen whose error in power lies within the aforementioned acceptable range. This paper mainly focuses on the utilization of GANs to augment the training database for the ANN to improve its predictive capabilities. Thus, this paper presents the ability of GANs to augment database not only for image generation but also to augment data in the field of nuclear engineering.

2. PSBT DATA

This paper utilizes data the test series 2 of the PSBT benchmark to train an ANN and a GAN. The DNB measurements of the PSBT database for test series 2 were performed for full-length partial 5×5 array rod bundles which simulate 17×17 PWR fuel assemblies. Thermocouples attached to the inner surface of the heater rods were used to detect the power at which DNB occurs. The bundle power, which is axially uniform for test series 2, is gradually increased by small steps of 0.328 kW/m at the vicinity of DNB power, which is based on preliminary analysis. The occurrence of DNB is confirmed by a rod temperature rise of more than 11ºC (20ºF) as measured by the thermocouples. The DNB power is then defined as the power corresponding to the step just before the step where the temperature increased [7]. To focus on data

belonging to a particular geometry and a particular radial power distribution, data is only taken from the steady state test series A2 which consists of 76 data points.

3. GENERATIVE ADVERSARIAL NETWORK (GAN)

GAN is a machine learning algorithm designed to solve the generative modeling problem [10]. The role of a generative model is to learn the probability distribution of data by studying a collection of training examples [10]. Generative models have become highly important and popular recently because of their capability to represent complex and high dimensional data as well as their applicability in various fields like image and music generation, medical images, security and various other academic domains [8]. GANs model high-dimensional distributions of data by training a pair of neural networks that compete with each other, i.e., the generator and the discriminator.

GANs were proposed [11] in which a generative model is pitted against an adversary: a discriminative model that has to learn to determine whether a sample is from a model distribution or from a data distribution. The generative model/generator can be thought of as an art forger whose job is to produce fake art that looks extremely real and to sell it without being caught/detected. The discriminator can then be thought of as an art judge whose job is to figure out which art is real and which art is fake. Thus, the generator will do its best to fool the discriminator by making "fake" data look a lot like "real" data; while the discriminator will do its best to distinguish accurately between the "fake" and "real" data. The generator as such has no access to real images and can thereby only learn by interacting with the discriminator [12]. The discriminator, however, has access to both the synthetic samples (fake art) as well as samples drawn from real data (real art). The error signal to the discriminator is provided by simply knowing the truth of whether the art was real or was forged by the generator, which ultimately leads the generator to produce better quality of forgeries [12]. This competition drives the generator and discriminator to improve their methods until the fake artwork is indistinguishable from the real artwork. This adversarial learning occurs between the generator and discriminator which are both ANNs [10]. The generator aims to generate artificial data as close as possible to the real data (with a distribution denoted by p_r), while the discriminator aims to distinguish between the artificial and real data. To learn the generator's distribution p_g over data x, a prior distribution (normal distribution) p_z is defined on input noise variables z. The prior is the initial distribution for the generator before the beginning of training that is taken from an input noise distribution denoted by $p_z(z)$. The mapping to data space for the generator is then defined as $G(z;\theta_g)$, where G is a differentiable function represented by an ANN with parameters θ_g [13] (the parameters are the weights and biases that build the ANN). The discriminator denoted by the ANN $D(x;\theta_d)$, outputs a single scalar. $D(x)$ represents the probability that x came from the data rather than p_g [14]. The discriminator is trained to maximize the probability of assigning the correct label to both training examples and samples from the generator while simultaneously training the generator to minimize $\log\left(1 - D\big(G(z)\big)\right)$. In other words, the discriminator and generator play the two-player minimax game shown in Equation 1 below with value function denoted by $V(G,D)$. This implies that the discriminator wants to successfully be able to distinguish between real data and data that is created by the generator while the generator wants to successfully fool the discriminator by generating fake data that is as similar as possible to the real data.

$$min_G max_D V(D,G) = \mathbb{E}_{x\sim p_r(x)}[\log D(x)] + \mathbb{E}_{x\sim p_z(z)}[\log(1 - D(G(z)))] \quad (1)$$

The architectures of the discriminator and generator are varied until an optimal structure is obtained. This is done by using two validation metrics: leave-one-out (LOO) accuracy corresponding to 1-nearest neighbor (1NN) and the kernel maximum mean discrepancy (KMMD) which are both explained in the sections below. Both 1NN and KMMD are good metrics to evaluate GANs in terms of discriminability, robustness and efficiency [14].

3.1. 1-Nearest Neighbor (1NN) Classifier

1NN classifiers are one of the most commonly used metrics and are an ideal metric for evaluating GANs [15]. They contain advantages seen in other metrics like inception score, mode score etc. along with having its own advantage of giving an output score in the interval of [0,1], similar to the accuracy/error in classification problems.

1NN classifiers belong to the two-test family for which a binary classifier is applied. Given two sets of samples denoted by R and G, where R represents real data and G represents data generated by the GAN, with |R|=|G|, the LOO accuracy is computed of a 1NN classifier trained on R and G with positive labels for actual data and negative labels for generated data [14]. A commonly used cross validation method is the k-fold cross validation where the data is split into k subsets, and k-1 sets are used for training while the remaining is used as a validation test case [14]. LOO cross validation is a special case of k-fold cross validation where k is equal to the number of observations.

While evaluating the GAN, the optimal LOO accuracy should be very close to 0.5, which signifies that the distributions of the generator and actual data match. An accuracy of 0.5 implies that the discriminator is no longer able to differentiate between the actual data and the data from the generator. When the LOO accuracy is less than 0.5, it implies that the GAN is overfitting to the actual data. If the generator has managed to memorize every single data point from the real data, the LOO accuracy would be 0.0. Higher values of LOO accuracy (close to 1.0) imply that the generated data distribution is completely different to the real data distribution.

3.2. Kernel Maximum Mean Discrepancy (KMMD)

KMMD is a commonly used criterion for model selection by computing the distance between two distributions. KMMD is a method to calculate the distance between embeddings of empirical distributions that are in a reproducing kernel Hilbert space [13]. The KMMD is a metric on the space of probability distributions that uses characteristic kernel to ensure that the distribution embeddings are unique for each probability measure [14]. In a GAN, the KMMD computes the distance of the generated dataset to the reference real dataset. KMMD is defined [14] as shown in Equation 2 below.

$$MMD^2(p_r, p_g) = \mathbb{E}_{x_r, x_r' \sim p_r;\, x_g, x_g' \sim p_g}[k(x_r, x_r') - 2k(x_r, x_g) + (x_g, x_g')] \quad (2)$$

This metric measures the similarity between p_r (real distribution) and p_g(parametrized distribution) for some fixed kernel function k that maps for real data denoted by x_r and generated data denoted by x_g. Given two sets of samples from $\mathbb{P}_r$ and $\mathbb{P}_g$, the empirical KMMD between the two distributions is computed with finite sample approximation of the expectation [15]. Lower the value of KMMD implies that $\mathbb{P}_r$ and $\mathbb{P}_g$ are closer to one another.

KMMD has proven to be a reliable metric to evaluate and validate GANs. KMMDs have low sample complexity and low computational complexity and are able to successfully identify generative data from real data [15]. It is also seen that KMMD can be used in other applications that involve evaluating the hypothesis about network architectures, expected likelihood and prediction accuracy [13]. Thus, this paper uses KMMD (with gaussian kernels) as a tool to evaluate and compare the similarity between PSBT data distribution and GAN generated data distribution.

4. CTF

CTF is an advanced thermal hydraulic code and is an improved version of COBRA-TF (Coolant-Boiling in Rod Arrays-Two Fluids) [9]. CTF is a thermal hydraulic simulation code designed for light water reactor (LWR) vessel analysis. It is currently managed by the Reactor Dynamics and Fuel Management Group (RDFMG) at North Carolina State University. CTF can be used for thermal hydraulic rod bundle analysis and for safety analysis of LWRs, as well as to model transients like loss of coolant accident in PWRs. This code uses a two fluid, three field modeling approach in which the three fields are fluid film, fluid drops and vapor. Each field has a set of conservation equations with the assumption that the liquid and droplet fields are in thermal equilibrium and thereby share an energy equation.

The artificially generated data from the GAN is run on CTF with the same geometry and radial power distribution as that of A2 test series of the PSBT database. CTF is modified so as to stay in accordance with the method of power prediction done in PSBT. To understand the acceptable error is CTF, PSBT data is run on CTF wherein only the inlet mass flux, inlet temperature and outlet pressure are varied based on each data point. The output power from CTF when an 11ºC temperature rise is observed is compared to the power of the PSBT data point. The error percentage ranges are computed. This range behaves as a cap/filter for the GAN data. The power predicted by CTF is then compared with the value of power in the GAN data by computing the error percentage. Only those data points are then accepted that have error percentages that are comparable to the error percentages when PSBT data was run on CTF. To do this, two steps are implemented using CTF which are as follows:

- The first step consists of simulating the PSBT test cases with CTF using the exact geometry and test conditions of A2 test series. In CTF, the inlet mass flux, inlet temperature and outlet pressure from the PSBT data are used as boundary conditions. To stay consistent with the way DNB power was measured, the CTF simulations are performed as follows. The bundle linear power is increased in small steps of 0.328 kW/m (identical to the process in the PSBT benchmark), with steady state being achieved after every increment. The temperatures at every axial node are saved at every power. Once a steady state is achieved after an increment in power, the temperatures at every axial node are compared with the temperature at every axial node in the previous time step. If a temperature rise of greater than or equal to 11ºC is observed, the simulation stops and the power at the previous step is noted down. This power is then compared with the power from PSBT data and the error in percentages is documented. The range of errors for the 76 PSBT data points from then acts as a filtering criterion for the GAN generated data.
- The second step consists of running the data generated from GAN, after evaluation using the 1NN and KMMD metrics. All of the GAN data is simulated on CTF using the values of the inlet mass flux, inlet temperature and outlet pressure from the GAN data. The power values are computed via CTF simulation by using the same methodology as the first step. This power from CTF is then compared with the power values of the GAN data. The errors of each data point are then computed and only those data are chosen whose error lies between the error ranges of the PSBT data (as computed in the first step).

5. ARTIFICIAL NEURAL NETWORKS (ANN)

Different types of ANNs have been applied to a variety of disciplines, including those related to applied thermal engineering: from solar radiation [16] and wind speed [17] to forecasting pressure drops in heat exchangers [18], along with a substantial increase in nuclear engineering applications [4], due to their ability to serve as a universal approximator of non-linear relations in the category of supervised learning. The main advantages of ANNs compared to other expert systems are their speed, simplicity and ability of modeling a multivariate problem to solve complex relationships between the variables and can extract the non-linear relationships by means of training data [19] [20]. ANNs have the ability to overcome the limitations of conventional approaches without the usage of specific analytical equations by extracting the required information using training data [21].

ANNs operate as black box model that try to restore the learning mechanism by trying to mirror the brain functions in a computerized way [21] [22]. ANNs have various capabilities like generalization, convergence, learning and error-toleration with structure of the high-speed parallel processing [21] [22]. ANNs learn relationship between input and output through training data. ANNs consist of elementary processing devices called neurons that are used to build layers of non-linear informational processing devices [21] . Each neuron is multiplied by a connection weight and the products along with the biases are transformed through a transfer function (non-linear function) to generate the final output. The ANN is trained using the back-propagation algorithm along with Adam gradient descent [23].

6. RESULTS

This section of the paper presents the results achieved. The first set of results corresponds to GAN evaluation and validation using the 1NN and KMMD metrics. The next set presents the error percentages of the PSBT data simulated with CTF to get an understanding of the acceptable range of error in the code predictions. The final set corresponds to the ANN algorithm that predicts the power at which DNB occurs with just PSBT data, followed by results with data from PSBT and GAN.

6.1 GAN Validation Results with 1NN and KMMD

To get an idea of what the optimal architecture of the GAN should be, color maps which are also called as heat maps are generated to get an apprehension of the number of hidden neurons needed. The heat maps for 1NN and KMMD, shown in Fig. 1 below are when the generator and discriminator have one hidden layer each and the number of neurons in both are varied in the x axis (generator) and y axis (discriminator). Fig.2 shows the color maps for 1NN and KMMD when the generator has one hidden layer with 10 neurons (fixed) and the number of neurons in the first (y axis) and second hidden layer (x axis) of the discriminator are varied.

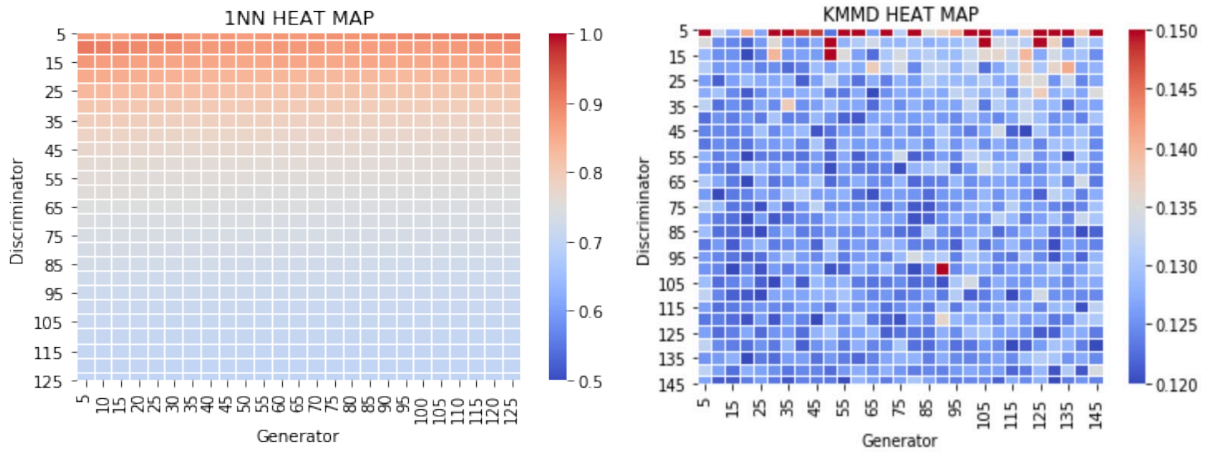


Figure 1. KMMD and 1NN Heat Maps for different number of neurons in one hidden layer of the Discriminator and Generator

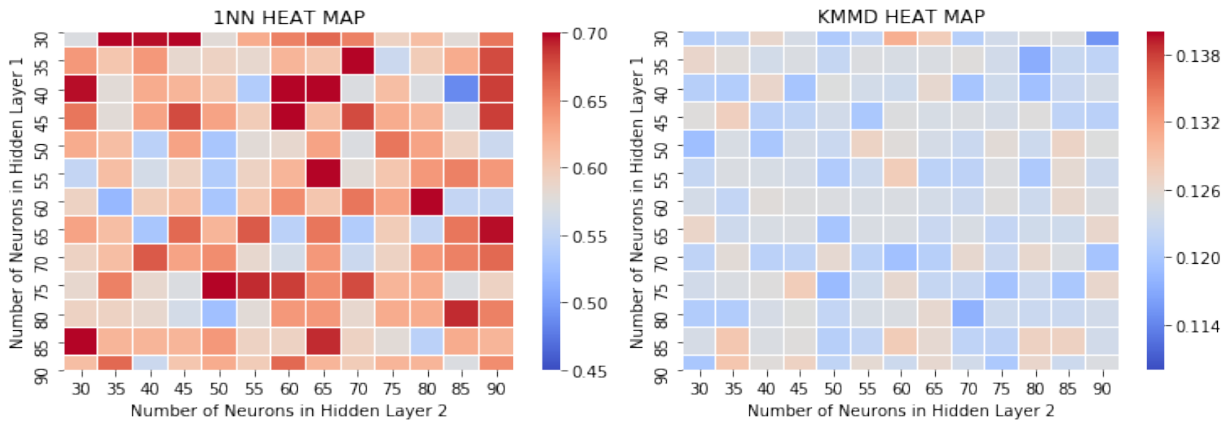
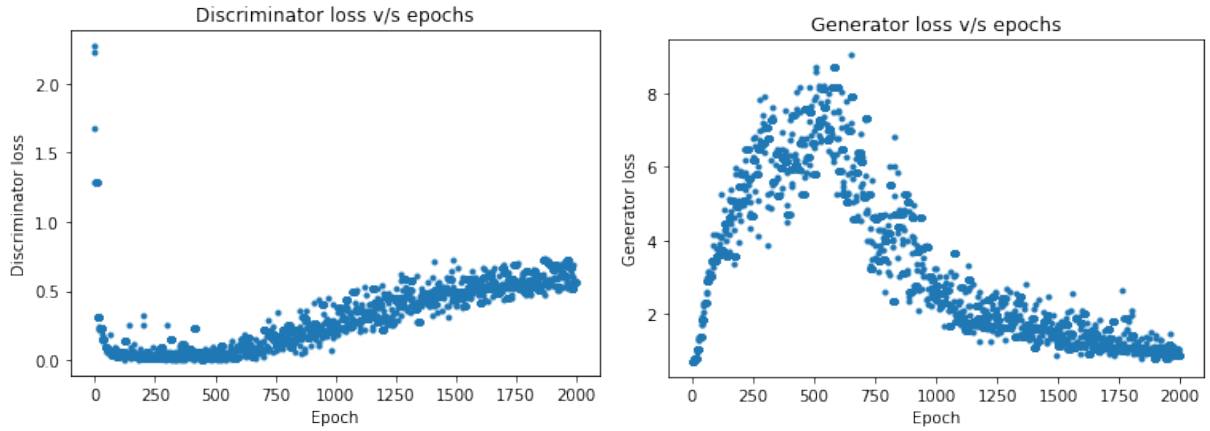


Figure 2. KMMD and 1NN Heat Maps for different number of neurons in the first and second hidden layer of the Discriminator

After further evaluation using the heat maps shown in Fig. 1 and Fig. 2, it is seen that a GAN with 10 neurons in the single hidden layer of the generator and 50 and 40 neurons in two hidden layers of the discriminator respectively resulted in a 1NN value of 0.546 (close to the optimal value of 0.5) and a KMMD of 0.1206 (among the lowest compared to KMMD for other architectures). Each hidden layer of both the discriminator and generator have a drop out of 0.3 and the training is run for 2000 epochs and a batch size of 19. Epoch denotes the number of times an algorithm goes through the entire training set. The graphs shown in Fig. 3: (a) and (b) represent the generator and discriminator loss over 2000 epochs respectively.



(a) (b)

Figure 3. (a) Discriminator loss over every epoch, (b) Generator loss over every epoch

As seen in Fig 3. (a), the discriminator loss reaches a value of around 0.5 which is the optimal [24]. This is because a value of 0.5 for discriminator loss implies that the discriminator is no longer able to distinguish between real and generated data. It can also be seen in Fig 3. (b), over 2000 epochs the generator loss increases and then stabilizes over a value around 1.0 which is a commonly seen in GANs as this implies that the data from the generator was similar to the real data and thereby was successfully able to fool the discriminator [25].

6.2 CTF Results

CTF is used to further validate the data generated by the GAN. The data from GAN is simulated using only the outlet pressure, inlet temperature and inlet mass flux values from the data are implemented in the CTF simulation. The power output from CTF is then compared with the power value from the GAN data and the error in percentage is computed. Only those data points from GAN are taken that fall within a certain range of errors. Thus, to compute this range of acceptable error percentage in DNB power, PSBT test cases are simulated with CTF. The graph shown in Fig. 4 depicts the error in percentages of each of the 76 data points of PSBT that were run on CTF.

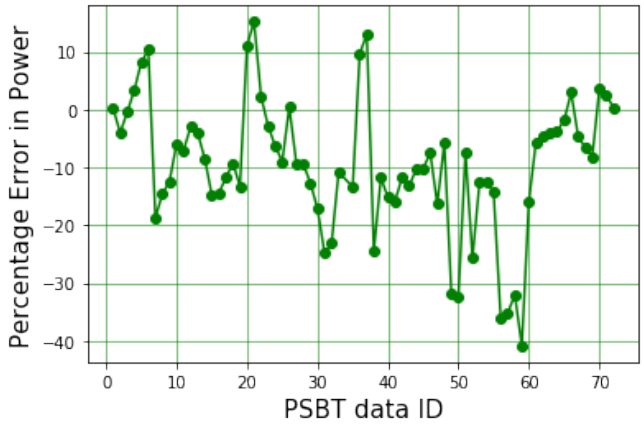


Figure 4. Error percentage in DNB power of PSBT data run on CTF

From the graph, it can be seen that the acceptable values in error percentage in power should lie between -40.86% and 15.2%. It is seen that high values of errors were found in cases with high temperature. It should also be noted that CTF was unable to simulate data points with low pressure values (<50 bar). The error values also arise due to the fact that CTF checks temperature at every axial node location in the rod as opposed to the PSBT data where the temperature is only measured at a set number of axial locations where the thermocouples were placed.

Thus, to further validate and evaluate data from GAN; the generated data is simulated with CTF and the error in power are compared between the power predicted by CTF and the power data from GAN. The percentage error in power for randomly chosen 63 data points generated from GAN that are simulated on CTF is shown in Fig. 5.

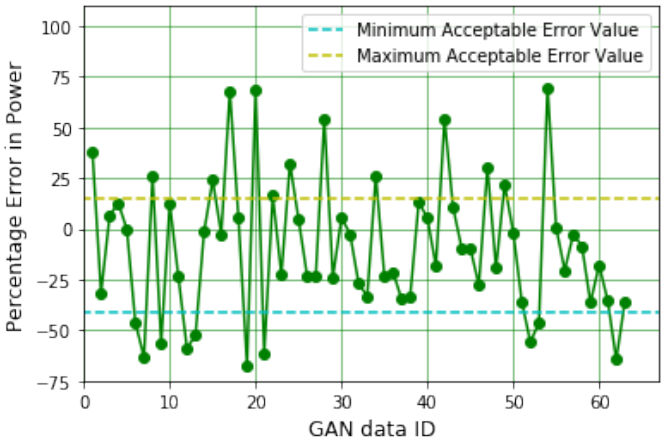


Figure 5. Error percentage in DNB power of GAN data run on CTF

Only those data, for which the error falls within the PSBT measurement error (as shown in the graph above using the two dotted horizontal lines), are accepted to augment the database. The graph above gives an example of how GAN data points are filtered out and only those data points are chosen that lie between the dotted horizontal lines shown in the graph. Out of 500 data points generated by the GAN, only 300 data points were accepted who had acceptable errors as discussed above.

6.3 ML Regression for DNB Power Prediction

An ANN is built using the MLP Regressor in scikit-learn [26]. The ANN has two hidden layers of 50 neurons each and uses the Adam optimizer with tanh activation function. The total number of epochs is set to 500 with a learning rate of 0.005. The hyperparameters for the ANN were chosen through trial and error.

To test the predictive accuracies of the ANN, a constant set of data from PSBT is reserved irrespective of what the training data comprises of. This is done to have a fixed set of data that can be used to check the optimal requirement of additional data from GAN. Fig 5. compares the mean absolute error (MAE) and root mean square error (RMSE) computed on the fixed 17 data points from PSBT. The plot shows the variance in the errors when the training set is varied by increasing the number of data points from GAN. With every addition of 40 data points, the hyperparameters of the ANN and the optimal set of hyperparameters are chosen through trial and error. The beginning of the graph indicates no data points from GAN have been included in the training.

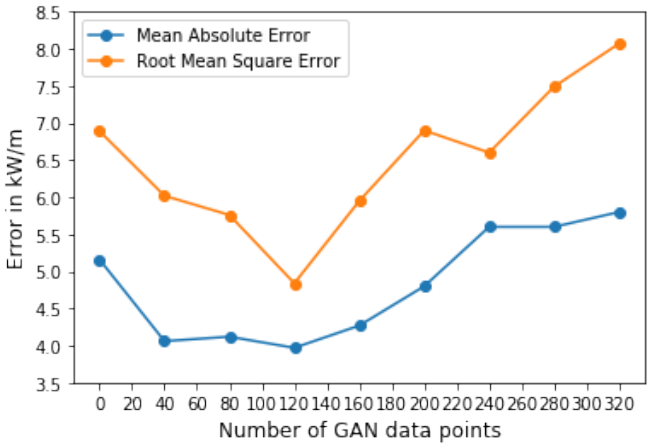
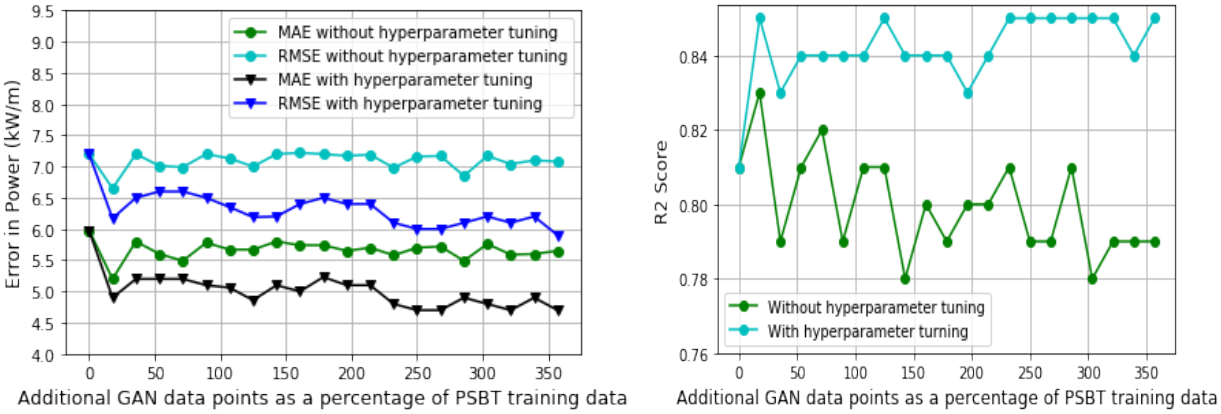


Figure 6. MAE and RMSE errors on PSBT test data with addition of GAN data points in the training dataset

It can be seen from the results shown above that the error values reduced with the addition of more data points from the GAN up to 120 data points from GAN. After 120 data points the errors kept increasing. This could be due to a large amount of GAN data present in training which could result in the ANN overfitting on the training set.

To test the applicability of GAN data on different test cases, different sets of arbitrary 17 data points were chosen to test the error reduction after adding GAN data to the training. It was seen that even after choosing different test sets of PSBT data, with the addition of data from GAN the error always reduced by 20-30%.

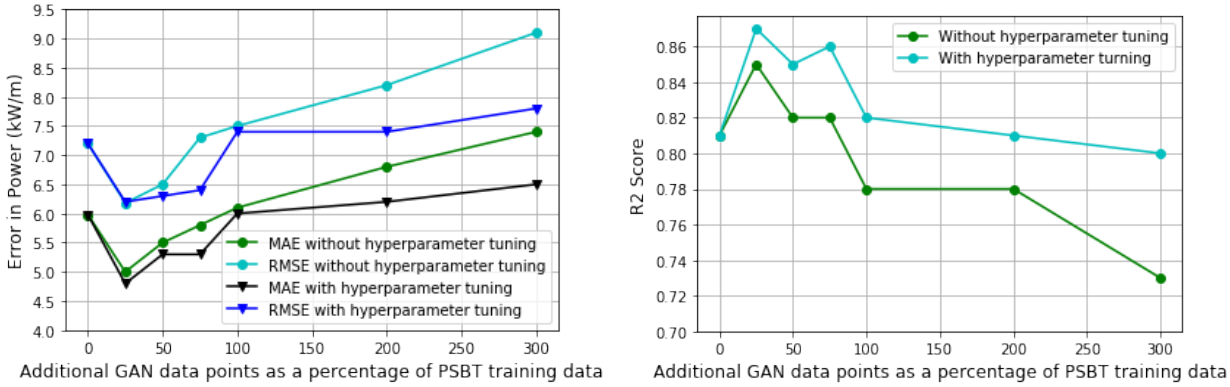
To further evaluate the performance of the ANN after the addition of GAN data, training and testing is done over 50 shuffles of randomly chosen data from PSBT and GAN. The training data consists of randomly chosen 75% of PSBT data plus randomly chosen 75% of "X" GAN data, where "X" represents the number of GAN data points chosen and is varied from 0 to 200. Similarly, the test dataset consists of the remaining 25% of PSBT data and 25% of "X" GAN data. For each value of "X", 50 randomly chosen datasets from both PSBT and GAN are used and the values of MAE, RMSE and R2 are averaged over the 50 shuffles. The graphs shown in Fig. 7. below show the (a) MAE and RMSE and (b) R2 score with and without hyperparameter tuning of the ANN.



(a) **(b)**

Figure 7. (a) MAE and RMSE with and without hyperparameter tuning (b) R2 with and without hyperparameter tuning

It can be seen from the graphs in Fig. 7. that MAE and RMSE values decrease only slightly without hyperparameter tuning and remain almost constant while R2 score decreases slightly. However, with hyperparameter tuning the MAE and RMSE decrease significantly and the R2 score increases significantly. A second similar test as above is conducted where the test data comprises only of PSBT data. The training data is 75% of randomly chosen PSBT data plus "X" GAN data where "X" here is number of additional GAN data points as a percentage of total PSBT training data points. This "X" ranges from 0 to 300%. Each value of MAE, RMSE and R2 score computed is averaged over 50 shuffles of randomly chosen PSBT and GAN data. Graphs in Fig. 8. show the (a) MAE and RMSE and (b) R2 score averaged over 50 shuffles and are computed with and without hyperparameter tuning.



(a) **(b)**

Figure 8. (a) MAE and RMSE with and without hyperparameter tuning (b) R2 with and without hyperparameter tuning

It can be seen from the graphs in Fig. 8. that the MAE, RMSE and R2 score improve significantly more with the addition of GAN data up to around 75% on the x-axis. This shows that adding GAN data whose size is up to 75% of the original training data size helps in the improvement in the predictive accuracies of the ANN. Adding more than 75% decreases the predictive accuracy of the ANN which can be a consequence of overfitting. Hence, it can be seen that using GAN to augment the database for the ANN to predict the power at which DNB occurs improves the predictive accuracy of the ANN.

7. CONCLUSIONS

This paper presents a preliminary attempt to develop an ANN to predict the power at which DNB occurs without relying on modeling limitations that are present in subchannel codes. Recently GANs have found a major augmentation in its uses from image generation, medical images, music etcetera but has seen only limited applications in data augmentation for regression purposes. Thus, this paper successfully demonstrates the abilities of GANs to augment a database by mimicking the PSBT data distribution. This paper looks into two metrics to validate and evaluate the GAN: 1NN and KMMD. It is seen that generator architecture with 10 neurons in the hidden layer and a discriminator architecture with 50 and 40 neurons in its two hidden layers resulted in the lowest value of KMMD around 0.12 and a 1NN LOO accuracy very close to 0.5.

This paper also confirms that with the help of augmented data from GAN, the ANN was able to improve its predictive accuracy by overcoming the dearth of data present in test series A2 of PSBT. This is seen by computing the MAE and RMSE. With the addition of augmented data from GAN, the MSE and RMSE

errors computed on a fixed dataset from GAN reduced from 5.16 kW/m and 7.54 kW/m to 3.97 kW/m and 4.84 kW/m respectively.
Future work entails enlarging the database by adding more experimental data and in turn generating more data from the GAN. Future work will also focus on building a more generalized ML algorithm that can predict the power at which DNB occurs for various geometries and test cases.

ACKNOWLEDGEMENTS

This work has been supported by the Office of Nuclear Energy of US Department of Energy.

REFERENCES

[1] E. Demarly, A New Approachh to Predicting Departure from Nucleate Boiling (DNB) from Direct Representation of Boiling Heat Transfer Physics, Massachusetts Institute of Technology, Department of Nuclear Science and Engineering, 2020.

[2] M. Bruder, G. Bloch and T. Sattelmayer, "Critical Heat Flux in Flow Boiling-review of the current understanding and experimental approaches," *Heat Transfer Engineering,* vol. 38, pp. 347-360, 2017.

[3] Y. A. Çengel, Heat and Mass Transfer, India: Mc Graw Hill, 2019.

[4] X. Zhao, K. Shirvan, R. K. Salko and F. Guo, "On the Prediction of Critical Heat Flux using a Physics-Informed Machine Learning-Aided Framework," *Applied Thermal Engineering,* vol. 164, no. 114540, 2020.

[5] H. Kurt, M. Stinchcombe and H. White, "Multilayer Feedforward Networks are Universal Approximators," *Neural Networks,* vol. 2, no. no. 5, pp. 359-366, 1989.

[6] T. Poggio and F. Girosi, "Networks for Approximation and Learning," vol. 78, 1990.

[7] A. Rubim, A. Schoedel, M. Avramova, H. Utsuno, S. Bajorek and A. Velazquez-Lozada, "OECD/NRC Benchmark Based on NUPEC PWR Sub-Channel and Bundle Test (PSBT) Volume 1: Experimental Database and Final Problem Specifications," *Nuclear Energy Agency of the OECD (NEA),* 2012.

[8] Y. Hong, U. Hwang, J. Yoo and S. Yoon, "How Generative Adversarial Networks and their Variants Work: An Overview," *ACM Computing Surveys,* vol. 52, pp. 1-43, 2019.

[9] R. Salko , M. Avramova, A. Wysocki, A. Toptan, J. Hu, N. Porter, T. Blyth, C. Dances, A. Gomez, C. Jernigan, J. Kelly and A. Abarca, CTF Theory Manual, U.S. Department of Energy, 2020.

[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, "Generative Adversarial Networks," *Communications of the ACM,* vol. 63, pp. 139-144, 2020.

[11] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, "Generative Adversarial Nets," *Advances in Neural Information Processing Systems,* vol. 27, 2014.

[12] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta and A. A. Bharath, "Generative Adversarial Networks: An Overview," *IEEE Signal Processing Magazine,* 2018.

[13] W. Bounliphone, E. Belilovsky, M. B. Blaschko, I. Antonoglou and A. Gretton, "A Test of Relative Similarity for Model Selection in Generative Models," 2016.

[14] J. Cheng, J. C. Dekkers and R. L. Fernando, "Cross-Validation of Best Linear Unbiased Predictions of Breeding Values using an Efficient Leave-One-Out Strategy," *J Anim Breed Genet,* pp. 519-527, 2021.

[15] Q. Xu, G. Huang, C. Guo, Y. Sun, F. Wu and K. Weinberger, "An Empirical Study on the Evaluation

Metrics of Generative Adversarial Networks," 2018.

[16] B. Amrouche and X. LePivert, "Artificial Neural Network Based Daily Forecasting for Global Solar Radiation," *Applied Energy,* vol. 130, pp. 333-341, 2014.

[17] A. P. Marugán, F. P. Márquez, J. M. P. Perez and D. Ruiz-Hernández, "A survey of Artificial NEural Networks in Wind Energy Systems," *Applied Energy,* vol. 228, pp. 1822-1836, 2018.

[18] A. Khosravi, J. G. Pabon, R. N. Koury and L. Machado, "Using Machine Learning Algorithms to Predict the Pressure Drop during Evaporation of R470C," *Applied Energy,* vol. 133, pp. 361-370, 2018.

[19] S. A. Kalogriou, "Artificial Intelligence for the Modeling and Control of Combustion Processes: A Review," *Progress in Energy and Combustion Science,* vol. 29, no. no. 6, pp. 515-566, 2003.

[20] A. Mellit and S. A. Kalogirou, "Artificial Intellgence Techniques for Photovoltaic Applications: A Review," *Progress in Energy and Combustion Science,* vol. 34, no. no. 5, pp. 574-632, 2008.

[21] M. Mohanraj, S. Javaraj and C. Muraleedharan, "Applications of Artificial Neural Networks for Refrigeration, Air-Conditioning and Heat Pump systems-A REview," *Renewable and Sustainable Energy Reviews,* vol. 16, no. no. 2, pp. 1340-1358, 2012.

[22] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, A. Mohamed and H. Arshad, "State-of-the-Art in Artificial Neural Network Applications: A Survey," *Heliyon,* vol. 4, no. no. 11, 2018.

[23] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *ICLR,* 2015.

[24] J. Kim, K. Jeong, H. Choi and K. Seo, "GAN-based Anomaly Detection in Imbalance Problems," *Springer International Publishing ,* 2020.

[25] J. Brownlee, "How to Identify and Diagnose GAN Failure Modes," 2019.

[26] F. Pedregosa, "Sciki-learn:Machine Learning in Python," *JMLR,* vol. 12, pp. 2825-2830, 2011.