Machine Learning based Modeling of Power Electronic Converters

Harish S. Krishnamoorthy
Department of Electrical and Computer Engineering
University of Houston
Houston, TX – 77204, USA
Email: hskrishn@uh.edu

Tulasi Narayanan Aayer
Department of Mathematics
University of Houston
Houston, TX – 77204, USA
Email: tulasi@math.uh.edu

***Abstract*— This paper proposes Bayesian Regularization (BR) along with artificial neural network (ANN) and random forest (RF) based machine learning to model power converters and analyze their performance. Unlike techniques such as PWM switch, state-space averaging, etc., this approach uses data from simulations or hardware experiments to develop the system model. The different steps involved in this modeling process are: (i) using an existing simulation or hardware prototype of the converter, collect data of different system parameters such as time, inductance, capacitance, resistance, voltages, currents, etc., (ii) categorize these parameters into inputs and outputs of the model, (iii) use BR-ANN to model transient performance and RF with bootstrap aggregation to model steady state response, (iv) validate the models with training and testing data sets, and (v) use the model to test with new values of input parameters. The proposed methodologies are tested using conventional buck and boost converters for proof of concept. It is envisioned that this technique can be used to create digital twins of power converters in practical circuits, optimize performance and predict fault conditions.**

Keywords*— *Machine Learning, Power Converter Modeling, Digital Twin, Random Forest, Bayesian Regularization

I. Introduction

Several modeling techniques have been developed for the analysis of power converters in the last 3 to 4 decades. Approaches such as PWM switch model, transformer model and state-space averaged model have been pivotal especially for DC-DC converters [1, 2]. They made it possible to estimate the converter losses, converter transfer function, etc. even before the circuit is physically built. Moreover, these methods form the core of several power electronics simulation tools [3]. As power converters become faster, smaller and cheaper, there is a clear trend towards research in improving the reliability, resiliency and performance across applications [4, 5, 6]. There is also greater emphasis on model based and predictive control strategies [7]. All these, combined with enhanced data collection and processing techniques, have opened immense opportunities in the area of power electronics. This also calls for additional developments in modeling and analysis of power converters, which take a wholistic approach to system design.

One such methodology is the use of analytics to interpret system characteristics and performance. A custom accelerated aging platform for power MOSFETs was introduced in [5], which also used a data-driven approach to estimate the remaining useful life (RUL) of a power device. Whereas, [6] presented a new methodology for reliability evaluation of power converters by focusing on the on-line monitoring of a parameter variation over the time and deploying Bayesian algorithm for data exploitation. It may be envisioned that many of these techniques will eventually lead to enhanced implementation of 'digital twins' [8] for power converters, which may help develop systems with better performance and reliability. The digital twin technology relies on co-simulations of power converter hardware and simulation models running on a computer. While such a scheme can be attractive for a mission critical system, the simulation software may be bulky and expensive for a smaller one. This can be addressed if the power converters and associated circuits themselves can be implemented using inexpensive or open source software.

Data driven modeling using machine learning has been gaining popularity on multiple domains, such as image classification, speech recognition, power systems, waveform magnitude estimation, control, etc. [9 – 15] to model and predict system behavior under a wide range of operating conditions. Artificial neural networks (ANNs) can not only be used to precisely model the transient and steady state characteristics of power converters, but also be implemented using inexpensive software. Machine learning using ANNs have been widely used in several applications including In [16], a deep feed forward neural network to model gallium nitride device has been proposed. Whereas [17] proposes ANN based automated design for reliability. With such efforts, machine learning is seeing widespread adoption in modern power electronics industry.

In this work, a Bayesian Regularization (BR) back-propagation [18, 19] based ANN is introduced to model and analyze power electronic converters, along with random forest method to model the transient and steady state performance. For simplicity of explanation, this paper uses basic DC-DC converters (buck and boost) for proof of concept. But this methodology can be extended to several other power converters as well. The main advantages of the proposed technique are:

- Precision of modeling can be controlled based on the extent of data and resolution; this enables the estimation of ripples in voltage and current if sufficient data is obtained.
- Power converter modeling can be implemented using inexpensive or open source software.
- Modeling can be performed without needing any complicated mathematics or equations.
- Model can adapt with real-time data, which is important for the development of a digital twin.
- Results from the actual hardware and the model can be compared to predict health of power converters or optimize the performance of the same; and
- The control system can be modeled together with the converters using ANN.

This paper focuses on the BR-ANN and random forest (RF) based model development aspect of boost and buck converters.

II. Proposed Machine Leanring based Converter Model Development

The main objective of the proposed machine learning based modeling technique is to synthesize the response of a power converter. The model in this case should be able to take in different converter parameters (both *circuit elements* such as inductance, capacitance, resistance, etc. and *sources* such as input voltage, frequency, duty cycle, etc.) and evaluate the time domain response of load voltage, source current, etc. To achieve this, 'time' should also be an input parameter to the model. The model development process consists of five main sections:

(i) Data Collection:

The developed model can only be as good as the data, so this is an important first step. Data can be collected from a hardware prototype or an accurate simulation representation of the same. In case of a DC-DC converter, the response of output voltage and input current depend on several factors – from input voltage, load resistance, inductances, capacitors, etc. to on-state resistances and parasitic capacitances of the switches. While larger amount of data can lead to a more precise model, it will be computationally more demanding. In this work, since it was desired to focus on the methodology, data was generated and collected from MATLAB simulations. The data consisted of converter system parameter values of *inductance, capacitance, load resistance, input voltage, switching frequency, duty cycle, initial capacitance voltage*; and their corresponding time domain responses of output DC voltage and inductor current. For this study, the parasitics of the converter were kept constant under most circumstances, except for steady state prediction. Values of the parameters was swept within certain ranges and their corresponding responses were recorded as a function of time with 1 µs resolution for 1 s.

(ii) Data Processing:

As can be interpreted, the eventual table has 1 million data points of time for every set of other input parameters, which have less than 10 values each. This disparity made the ANN overfit the data. So, the data had to be down-sampled and processed before training the network. Steady state inductor current ripple estimation requires more data points of the system parameters (*inductance, capacitance, resistance, etc.*), instead of down-sampling the time data to reduce the disparity. This increases the data size but provides much more accuracy to the model. The inductor current ripple was evaluated as a part of the steady state response.

(iii) Machine Learning Algorithm:

a) BR-ANN based Model Development: Bayesian Regularized ANN [13] was used in this work, primarily to reduce lengthy cross-validation and the tendency of neural networks to over-fit the data. MATLAB's neural network tool-box was used for this purpose, though open source frameworks such as 'R' or 'Python' may also be used to achieve equally good results. The BR-ANN has 2 layers – a Hidden Layer (in this case, twice the number of nodes as the number of inputs) and an Output Layer (with the number of nodes being same as the number of outputs), as shown in Fig. 1.

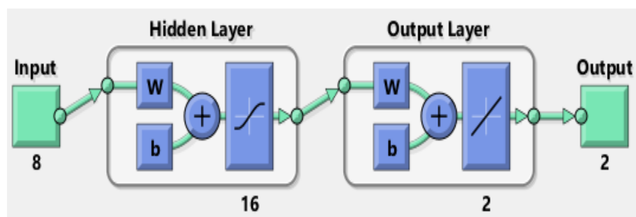

Figure 1: BR-ANN set up for model development with a hidden layer and an output layer

The formulation of the model can me expressed as a minimization cost function *C(w)* with respect to the weights 'w' [20] as shown in eqn. (1) in order to reduce the mean squared error (MSE).

$$C(w) = \beta \cdot \sum_{i=1}^{N_D}[Sys\_Out_i - f(Sys\_In_i)]^2 + \alpha \cdot \sum_{j=1}^{N_W}\left[w_j\right]^2 \quad (1)$$

Here, α and β are hyper-parameters, N_D and N_W are number of data rows and weights, respectively. *Sys_Out* and *Sys_In* are the converter system's output and input data tables, respectively; and *f(Sys_In)* gives the predicted output table from the model, which can be used to find the MSE. It should be noted that there are other back propagation algorithms (such as *Levenberg-Marquardt*) which can be used for modeling as well.

b) Random Forest method - based Model Development: While BR-ANN was able to track the peak overshoot, settling time and steady state average values of the power converters under different scenarios, the steady state ripple was not captured properly, probably because of the singularities that exist in the waveforms. This project then utilized random forest approach to track steady state ripple. Random Forest is an ensemble machine learning technique capable of predicting tasks using multiple decision trees and statistical technique called *bagging* (or bootstrap aggregation). This technique improves the stability of the model and also helps to reduce the variance. The bootstrap aggregation creates multiple models from the same training set [20, 21]. Since the random forest method usually works well with conditional evaluation, the steady state response was captured precisely. A case study of tracking inductor current ripple in a buck converter is discussed in this paper. A brief overview of the random forest method is provided below.

Let B be the training set where Y (for example, inductor current, output voltage, etc.) is the response variable and $X = (x1, x2, x3 \dots .xN)$ are the independent variables. In the Random Forest approach, the overall training set B is further represented by eqn. (2).

$$B = \{(X_1, Y_1), (X_2, Y_2), (X_3, Y_3), \dots., (X_n, Y_n)\} \tag{2}$$

where, $X_1, X_2, \dots X_n$ are subsets of X and $Y_1, Y_2, \dots Y_n$ are subsets of Y. Then take L sets of n elements from B (with replacement) $B_1, B_2, \dots B_L$. Train on each B_i, with $i = 1, \dots, L$ and obtain a sequence of L outputs $f(X_1), f(X_2) \dots \dots f(X_L)$. Now, $f(\bar{X})$ in eqn. (3) gives the *Ensemble Random Forest* regression model.

$$f(\bar{X}) = \sum_{i=1}^{L} f(X_i) \tag{3}$$

(iv) <u>Model Verification:</u>

Typically, 70 % of the data rows are used for training and 30 % are used for testing of the model (though this can change). This level of verification is performed by sampling the data and hence form the first layer of verification. Unless the model is able to achieve good performance in this stage, it would require re-training. This can be analyzed using error histogram and regression plots.

(v) <u>Model Implementation:</u>

Once the model provides satisfactory results with the test data, it needs to be evaluated for new values of test inputs within the data range. While using MATLAB's ANN toolbox, it is possible to extract the model as a code or a Simulink block; similar code can be achieved using 'R' as well. For a given set of input values, the model will give an output, which can be assessed for accuracy comparing with the hardware or its simulation model. This stage completes the modeling process.

III. EXAMPLES FOR TRANSIENT PERFORMANCE USING BR-ANN

A. *Boost Converter*

The MATLAB Simulink model used to extract data is shown in Fig. 2. The main parameters and their range of values (less than 10 values within the range for each parameter other than 'Time') trained are given in TABLE I. The output DC voltage [Vout] and the inductor current [IL] were recorded for each set of inputs in a table, until they achieved steady state. Since the parameter 'Time' had significantly higher number of values compared to other parameters, number of data rows had to be sampled to achieve better performance. It may be noted that if it is desired to capture the high frequency ripple in the waveforms, residual analysis can be performed without sampling the acquired data.

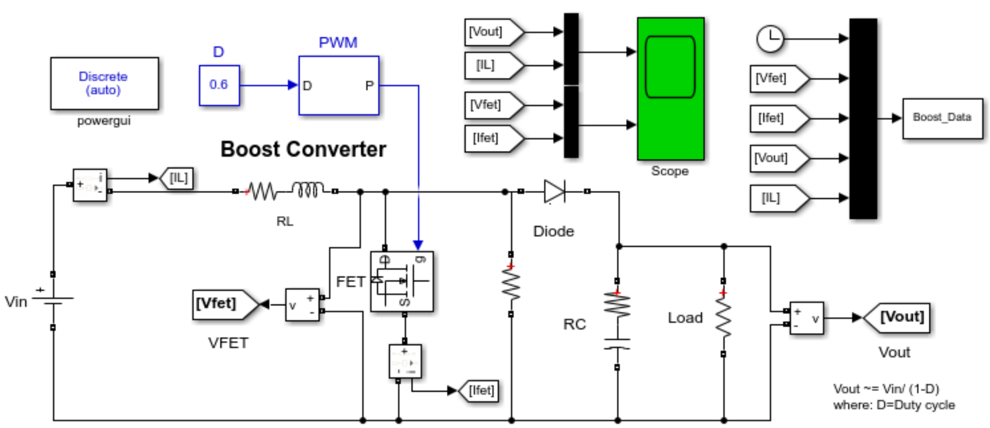

Figure 2: MATLAB Simulink model for data extraction

TABLE I: Range of parameters used in training the BR-ANN with boost converter

Parameter	Vin (V)	Duty (%)	Frequency (kHz)	Load (Ω)	Inductance (µH)	Capacitance (µF)	V_0_Cap (V)	Time (ms)
Range	15 – 24	40 – 60	20 – 50	0.5 – 2.5	100 – 300	400 – 800	0 – 20	0 to 20

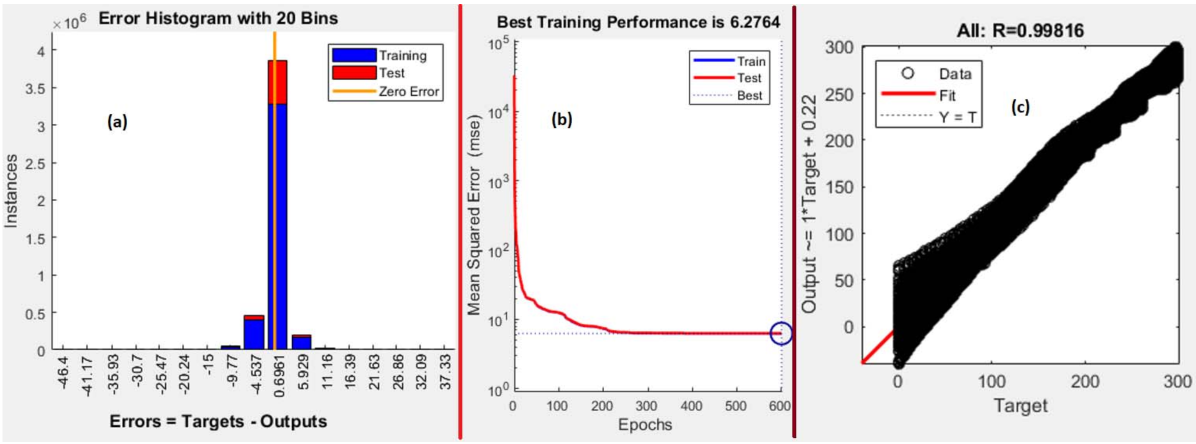

Figure 3: Results from the BR-ANN training and testing on the extracted data using MATLAB's neural network toolbox: (a) Error Histogram, (b) Convergence of MSE, and (c) Regression Plot showing an R-value of ~0.9982

Figure 3 (a), (b) and (c) show the performance of the model development process and results from training/test data. An error histogram is shown in Fig. 3 (a). It can be seen that the histogram has a single long bar with a few shorter bars. Initially, two long bars appeared at the center of the histogram, but the training was not good in that case and required re-training. Here, the mean of the error resides around 0.7. Figure 3 (b) shows the convergence of the minimization cost function shown in eqn. (1). It can be seen that the MSE came down from about 40,000 to 6.28 and the minimization has converged. Figure 3 (c) shows the regression plot from both training and testing set, with a good R-value of about 0.9982. The data curve distribution seen about the mean can be further improved by taking more data points in the training set. But it takes longer time to arrive at the model, which may need a trade-off analysis.

Once the model was satisfactory from the performance graphs in Fig. 3, it was extracted in the form of a Simulink block. Next step is the implementation and validation with new parameter values. Figure 4 (a) and (b) show the time domain transient responses of a boost converter's output voltage [Vout] and inductor current [IL] using the proposed BR-ANN based model, compared with the respective responses from the MATLAB Simulink model (see Fig. 2). Figure 4 (a) was evaluated using a discrete simulation of fixed time step 2.5 µs with the parameter values – [*Vin = 16 V; Duty = 0.55; Frequency = 50 kHz, Load = 1.65 Ω; Inductance = 250 µH; Capacitance = 500 µF; and Initial Cap. Voltage = 5 V*]. Figure 4 (b) was evaluated using a variable time-step simulation with the parameter values – [*Vin = 21 V; Duty = 0.50; Frequency = 25 kHz, Load = 0.5 Ω; Inductance = 250 µH; Capacitance = 450 µF; and Initial Cap. Voltage = 10 V*]. Several values of the parameters, including most values of 'Time' under these simulation conditions were different from the training set used to build the model. Yet, it can be seen that the responses of the BR-ANN based model very closely matches the Simulink based model, whether it is discrete time-step or variable time-step. This validates the proposed technique and shows that BR-ANN can be used to model boost converters' transient behavior. Since ripple was not considered in the training of the model, it is not seen in the response. It is possible to include other parameters such as inductor's or capacitor's series resistances in the training set as well, to get an even more precise model and to adapt to varying parasitics, if desired.

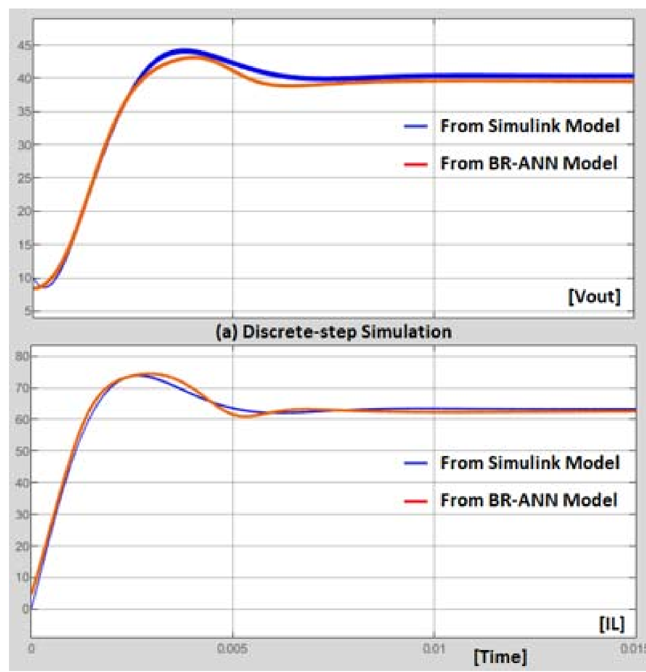

Figure 4 (a): Model implementation and training – comparison of a MATLAB Simulink model based response with the proposed BR-ANN model based response under Discrete Fixed Time-step simulation of 2.5 µs, and (b) Variable Time-step simulation

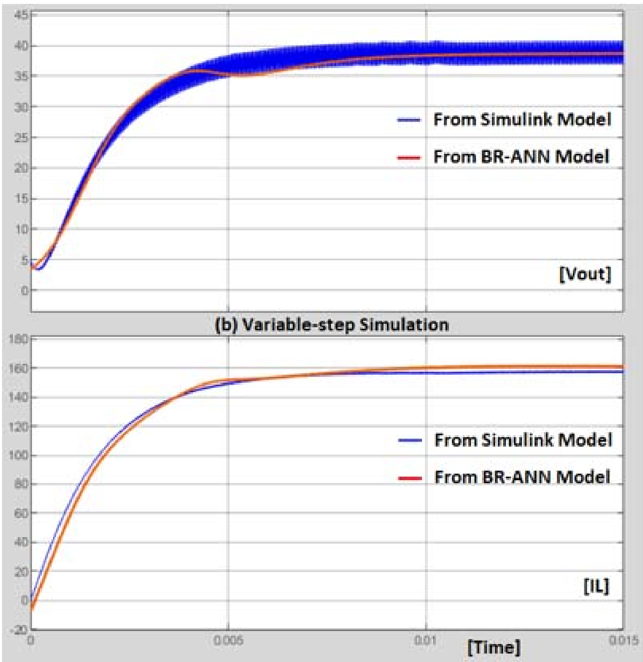

Figure 4 (b): Comparison of a MATLAB Simulink model based response with the proposed BR-ANN model based response under Variable Time-step simulation

B. Buck Converter

As above, a similar modeling (as in Fig. 2) was performed for an open loop buck converter with parameters as in TABLE II. It was possible to achieve close match in its response as well. Two test scenarios are shown in Fig. 5 and Fig. 6.

TABLE II: Range of parameters used in training the BR-ANN with buck converter

Parameter	Vin (V)	Duty (%)	Frequency (kHz)	Load (Ω)
Range	45 – 54	40 – 60	20 – 50	0.5 – 2.5
Parameter	Inductance (µH)	Capacitance (µF)	V_0_Cap (V)	Time (ms)
Range	100 – 300	400 – 800	0 – 20	0 to 20

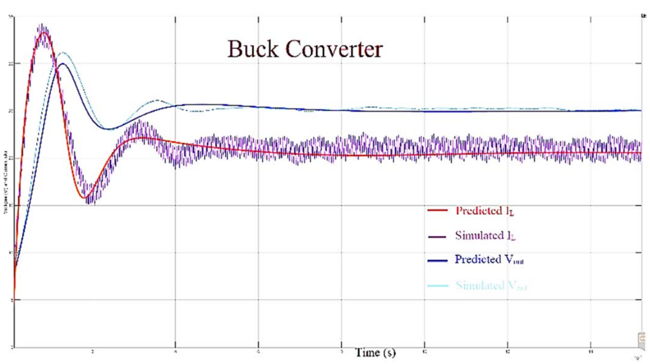

Figure 5: Buck converter BR-ANN based modeling with mixed set of in-sample and out of sample values [*L = 250 µH; C = 500 µF; Duty = 50 %; Vin = 50 V; Fsw = 20 kHz; Load = 1.2 Ω; Vc0 = 10 V*]

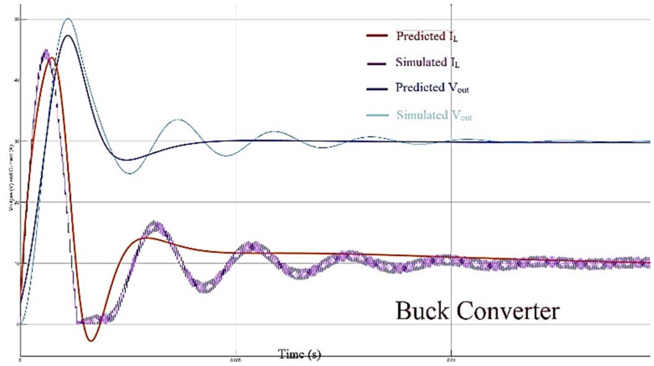

Figure 6: Buck converter BR-ANN based modeling with significantly out of sample values [*L = 250 µH; C = 450 µF; Duty = 53 %; Vin = 54 V; Fsw = 24 kHz; Load = 2.9 Ω; Vc0 = 0 V*]

It can be seen than the BR-ANN based model is able to capture the peak overshoot, rise time, settling time and steady state values reasonably well. The model is even able to predict the tendency of the waveform to go to the negative side, though it gets clipped by the devices in a real scenario. But, the damped oscillations are not being captured that prominently. An approach integrating the benefits of BR-ANN and random forest method is being investigated to track the damped oscillations as well as the clipped operation.

IV. Case Study for Steady State Performance using Random Forest Method

The random forest method detailed in section II was implemented using *'R' software* to demonstrate that the proposed approach can work in multiple platforms. The *randomForestExplainer* package was used extensively. The range of the important parameters were chosen as in TABLE II, but the impact of the capacitor effective series resistance (ESR) was considered to assess the steady state inductor current ripple.

Even with the random forest approach, the peaks of the waveforms were not captured. Subsequently, bootstrap aggregation was performed for the error residuals (considering the error had a pattern and was following uniform distribution), which was added to the original model. This resulted in a very good waveform prediction with in-sample data and most cases of out-of-sample data. Two in-sample waveforms and an out of sample waveform of the inductor current ripple are shown in Fig. 7 and Fig. 8, respectively. An out of sample waveform prediction, which requires further improvement is shown in Fig. 9. It can be seen that while the algorithm tracks the rate of change of the waveform, the peaks are not captured well. Further work is being done to address all the scenarios. Having a much larger supervised training set with a wider range of parameters will be able to solve most of the issues.

It may be noted that the minimum steady state values of the waveforms were captured, trained using BR-ANN and predicted along with random forest, as it was required to estimate the waveforms according to the gating signals.

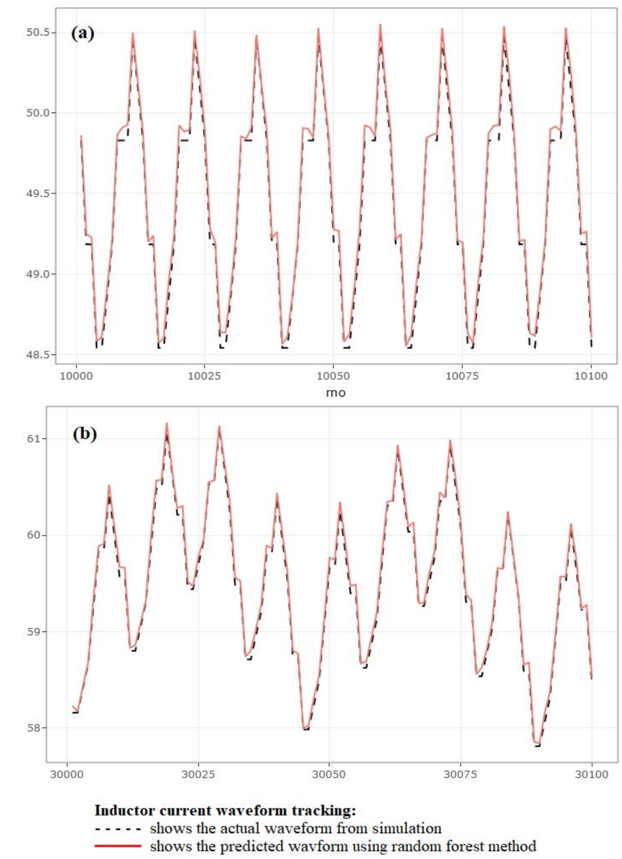

Figure 7: In-sample Buck converter steady state ripple prediction using random forest approach [*(a) Vin = 50 V, D = 0.5, and R = 0.5* Ω*; (b) Vin = 54 V, D = 0.55, and R = 0.5* Ω]

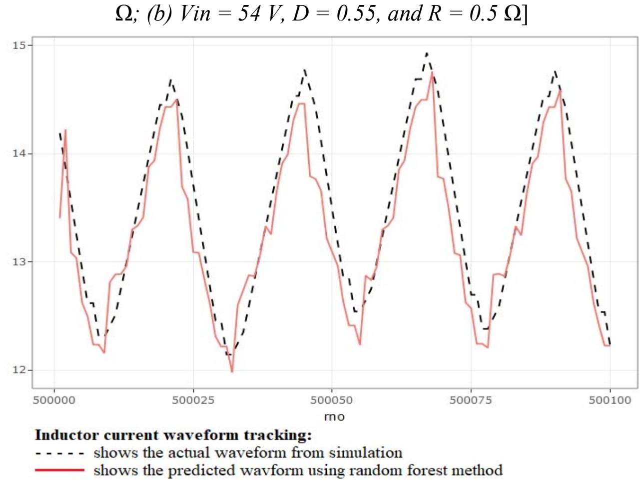

Figure 8: Out of sample Buck converter steady state ripple prediction using random forest approach [*Vin = 52 V, D = 0.51, and R = 2.1* Ω]

The Fig. 10 shows the maximum and minimum value tracking of the steady state waveforms under two scenarios discussed before (waveforms are shown in between the predicted limits). It can be seen that the algorithm is tracking the inductor current steady state peaks and valleys properly.

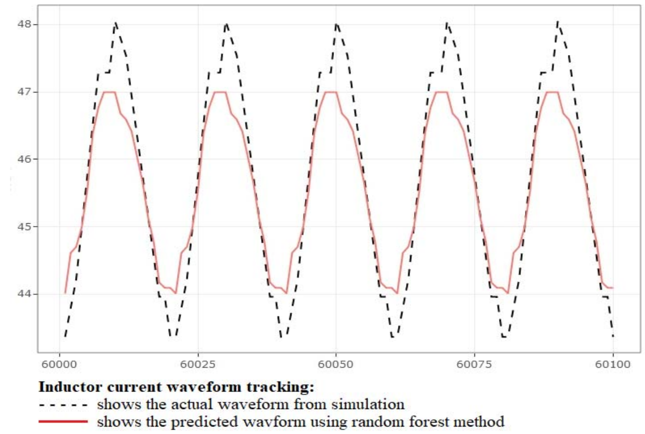

Figure 9: Out of sample Buck converter steady state ripple prediction using random forest approach [*Vin = 49 V, D = 0.47, and R = 0.5* Ω]

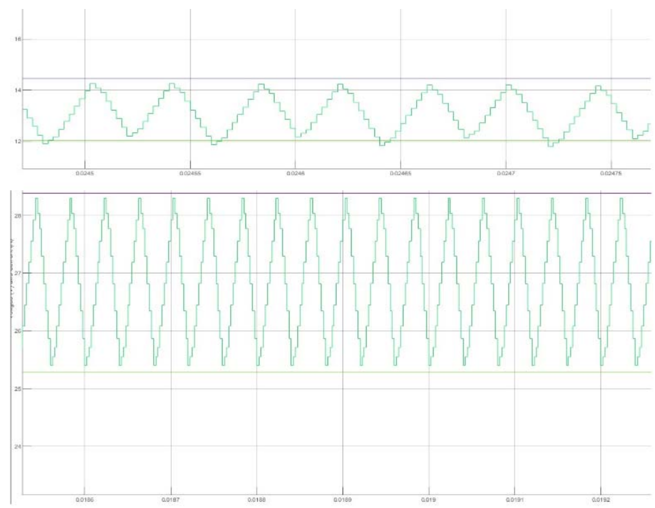

Figure 10: Buck converter steady state inductor current ripple - prediction of maximum and minimum values [*L = 200 μH; C = 500 μF; Duty = 50 %; Vin = 50 V; Fsw = 25 kHz; Load = 2 and 1* Ω]

As the bagged decision trees are made during the random forest process, variable importance plot can be generated by assessing how much error can be calculated by dropping each variable. This plot can be created using H2O machine learning framework and it provides a relative importance of all the parameters considered for training the models. Such a plot is shown in Fig. 11 with the rankings of different variables.

From the variable importance plot in Fig. 11, it can be seen that the ordering of the variables by their mean minimal depth seems quite accurate when looking at the distribution of minimal depth, though it could be argued that the variable *L*' (inductor) be ranked higher than gating signal 'V_g', as the latter is not that often used for splitting at the root. Nevertheless, this plot can be very useful when the importance of the input variables is hard to determine. The last predicted value of the inductor current '$I_{L(n-1)}$', (derived using the minimum value evaluated earlier and fed back as input to the model), has the highest importance in the prediction, as expected.

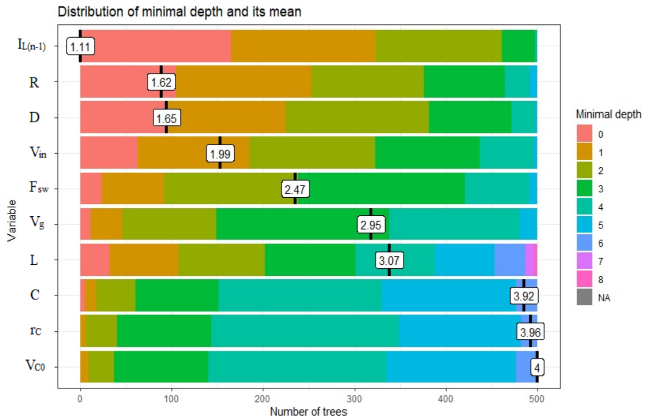

Figure 11: Variable importance plot during random forest evaluation

V. CONCLUSIONS

This paper introduced a new modeling technique for DC-DC converters using Bayesian Regularization based ANN as well as random forest method. Developmental steps of the model, including data collection, processing, training, testing and validation were discussed for the boost converter. It was seen that the machine learning based models was able to provide very close responses to the simulation results. The proposed methods are envisioned to be attractive to the implementation of modern concepts including digital twin, circuit automation, and software defined electrical networks as they offer fast and cost-effective modeling platforms. Moreover, machine learning based models themselves are light in computation and can be implemented on graphics processing units (GPUs) or FPGAs – a critical feature to enable next generation *Edge Computing* in power converters.

The proposed techniques can also model the voltage across (and current through) the switches if large enough data is trained. This paper presents an introduction into this topic of modeling power converters using machine learning techniques. Future research in this area involves further improving the models to capture different scenarios of operation, exploring alternative machine learning techniques for better results and extending them to cover other converters in general.

REFERENCES

[1] R. W. Erickson, and D. Maksimović (2001), “Fundamentals of power electronics”, Kluwer Academic

[2] R. D. Middlebrook and S. Cuk, "A general unified approach to modelling switching-converter power stages," 1976 IEEE Power Electronics Specialists Conference, Cleveland, OH, 1976

[3] E. Van Dijk, J. N. Spruijt, D. M. O'Sullivan and J. B. Klaassens, "PWM-switch modeling of DC-DC converters," in IEEE Transactions on Power Electronics, vol. 10, Nov. 1995

[4] S. Yang, D. Xiang, A. Bryant, P. Mawby, L. Ran and P. Tavner, "Condition Monitoring for Device Reliability in Power Electronic Converters: A Review," in IEEE Transactions on Power Electronics, vol. 25, no. 11, Nov. 2010

[5] S. Dusmez, H. Duran and B. Akin, "Remaining Useful Lifetime Estimation for Thermally Stressed Power MOSFETs Based on on-State Resistance Variation," in IEEE Transactions on Industry Applications, vol. 52, no. 3, pp. 2554-2563, May-June 2016

[6] M. Biglarbegian et al., "On condition monitoring of high frequency power GaN converters with adaptive prognostics," 2018 IEEE Applied Power Electronics Conference and Exposition (APEC), San Antonio, TX, 2018

[7] S. Vazquez et al., "Model Predictive Control: A Review of Its Applications in Power Electronics," in IEEE Industrial Electronics Magazine, vol. 8, no. 1, March 2014

[8] M. Schluse et al., "Experimentable Digital Twins—Streamlining Simulation-Based Systems Engineering for Industry 4.0," in IEEE Transactions on Industrial Informatics, vol. 14, no. 4, April 2018

[9] M. Charkhgard and M. Farrokhi, "State-of-Charge Estimation for Lithium-Ion Batteries Using Neural Networks and EKF," in IEEE Transactions on Industrial Electronics, vol. 57, no. 12, pp. 4178-4187, Dec. 2010

[10] E. Manitsas, R. Singh, B. C. Pal and G. Strbac, "Distribution System State Estimation Using an Artificial Neural Network Approach for Pseudo Measurement Modeling," in IEEE Transactions on Power Systems, vol. 27, no. 4, pp. 1888-1896, Nov. 2012

[11] Y. Chen, Y. Tan and D. Deka, "Is Machine Learning in Power Systems Vulnerable?," 2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm), Aalborg, 2018, pp. 1-6

[12] J. Moreno, M. E. Ortuzar and J. W. Dixon, "Energy-management system for a hybrid electric vehicle, using ultracapacitors and neural networks," in IEEE Transactions on Industrial Electronics, vol. 53, no. 2, April 2006

[13] S. Mohagheghi, R. G. Harley, T. G. Habetler and D. Divan, "Condition Monitoring of Power Electronic Circuits Using Artificial Neural Networks," in *IEEE Transactions on Power Electronics*, vol. 24, no. 10, Oct. 2009

[14] Min-Huei Kim, M. G. Simoes and B. K. Bose, "Neural network-based estimation of power electronic waveforms," in IEEE Transactions on Power Electronics, vol. 11, no. 2, pp. 383-389, March 1996.

[15] K. Medjaher, D. A. Tobon-Mejia and N. Zerhouni, "Remaining Useful Life Estimation of Critical Components With Application to Bearings," in IEEE Transactions on Reliability, vol. 61, no. 2, pp. 292-302, June 2012.

[16] N. Hari, S. Chatterjee and A. Iyer, "Gallium Nitride Power Device Modeling using Deep Feed Forward Neural Networks," 2018 1st Workshop on Wide Bandgap Power Devices and Applications in Asia (WiPDA Asia), Xi'an, China, 2018, pp. 164-168

[17] T. Dragičević, P. Wheeler and F. Blaabjerg, "Artificial Intelligence Aided Automated Design for Reliability of Power Electronic Systems," in IEEE Transactions on Power Electronics, vol. 34, no. 8, pp. 7161-7171, Aug. 2019

[18] L. M. Saini, “Peak load forecasting using Bayesian regularization, Resilient and adaptive backpropagation learning based artificial neural networks”, Electric Power Systems Research, Volume 78, Issue 7, 2008

[19] F. Burden, D. Winkler (2008), “Bayesian Regularization of Neural Networks”. In: Livingstone D.J. (eds) Artificial Neural Networks. Methods in Molecular Biology™, vol 458. Humana Press

[20] T. Hastie, R. Tibshirani, and J. Friedman, “The Elements of Statistical Learning”, Springer Series in Statistics, Springer New York Inc., New York, NY, USA, (2001)

[21] L. Breiman, “Bagging Predictors,” Department of Statistics, University of California Berkeley, CA, Report number: 421, Sep. 1994