

Potentials of Machine Learning in Vacuum Electronic Devices Demonstrated by the Design of a Magnetron Injection Gun

Liang Zhang and Adrian W. Cross

Abstract— Great progress has been made on machine learning and its applications are expanding rapidly nowadays. Through the case study of optimizing a magnetron injection gun for gyrotron devices, the functions of machine learning were investigated by using two supervised learning algorithms, regression trees and artificial neural networks. They showed excellent performance in predicting the outputs, exploring the importance of the input parameters and the relationship with the output parameters. Machine learning can be a useful tool in the development of microwave vacuum electron devices.

Index Terms— machine learning, microwave vacuum electron device, magnetron injection gun, regression tree, artificial neural networks, supervised learning.

I. INTRODUCTION

Microwave vacuum electron devices (MVED) have a long history dating back to the 1930s when the magnetrons and the first coherent radiation sources the klystron led to revolutionary advances in microwave applications [1, 2]. Since then many different types of vacuum electron devices have been invented with their applications expanded to underpin society's needs from air traffic control to medical radiotherapy treatments. In the early days, the major advances in MVED's development were achieved by improving the machining and adaptive design by trial and error. This development process is long and greatly relied on the developer's experience. Approaching 100 years for the invention of the magnetron and klystron, one of the major research areas nowadays is millimeter and sub-millimeter MVEDs [3]. As the operating frequency becomes higher, the machining requires tighter tolerance and the previous design process becomes increasingly expensive. Computer-aided design (CAD) is now playing an increasingly important role in the development process.

Various methods have been developed for the design of MVEDs, for example the linear and nonlinear models and particle-in-cell (PIC) method [4] to simulate different beam-wave interaction circuits, the beam optics based on particle tracking or PIC for the electron gun and collector design, the finite-difference time-domain (FDTD), mode-matching (MM), method of moments (MoM), multilevel fast multipole method (MLFMM), and others that can be used to design the waveguide

components such as the input/output couplers and microwave windows [5, 6].

Benefiting from the advances in computer science and the boost of the computing resources in recent decades, global optimization algorithms, such as the Simulated Annealing Algorithm, Evolutionary Algorithms, Stochastic Methods, Particle Swarm Optimization, and others, have been used to explore the parameter space and search for the optimal design of different components in MVEDs [7-10]. However, one of the challenges is the long overall optimization time it needs when the free parameter number is large. Another challenge is the components to be designed normally have to satisfy multiple requirements, for example, the gain, bandwidth and frequency for an interaction circuit. The optimal design is normally a trade-off between these parameters. Multiple objective optimizations are possible solutions however they require longer computing time.

In recent years, the huge leaps in machine learning and data mining make them revolutionary techniques and greatly impact many areas such as self-driving, finance, image classification, and so on [11]. Alpha Go, a computer program that combines advanced search trees with deep neural networks, beat a Go world champion in 2016 [12, 13]. It is a landmark achievement that shows artificial intelligence based on deep learning can make better decisions than humans in the most challenging classical game of Go. Machine learning can possibly be a useful tool in the development of MVEDs since it helps explore the relationships between coupled parameters. It is a powerful tool that enables the researcher to use them to investigate deeper into the problem and reduce the complicity and reduce the optimization time.

In this paper, the machine learning algorithm that suits the design of MVEDs was investigated and is presented in section II. The functions and performance were studied via the case study of optimizing a magnetron injection gun (MIG) for gyrotron devices, as presented in sections III and IV. Section V concludes the findings on the potential of machine learning technology in the MVED research field.

II. MACHINE-LEARNING ALGORITHMS THAT SUIT MVED DESIGN

Machine learning algorithms aim to optimize the performance of a certain task by using examples and/or experience [14-16]. They can be divided into three categories, supervised learning, unsupervised learning, and reinforcement learning. Each category includes different algorithms as briefly listed in Table 1. New algorithms are been invented as a result of rapid development in various research fields, and they can be found in recent review papers [17-19]. This paper does not intend to dive deeply into each method as this can be achieved by reviewing the referenced papers but to select the suitable methods for MVED's development. Supervised learning is trained from data with labeled outputs. After training and validation, it can be used to predict the outcomes from unknown inputs. Unsupervised learning does not have labeled outputs for training. The method tries to find the hidden pattern directly from the input data. It can be used to find the connections and clusters among the inputs. Reinforcement learning is used to find the optimal strategy from interactions, for example, self-driving and games.

Table 1 The classification of the deep learning algorithm [20].

Unsupervised learning	Principal Component Analysis; DBSCAN; K-means; Hierarchical clustering.
Supervised learning	Decision Tree; Naive Bayes; Neural Networks; Support vector Machine; linear regression; logistic regression; Similarity learning.
Reinforce learning	Monte Carlo; Deep Q Network; State-action-reward-state.

One of the common features in the design of MVED components is that their performances can be relatively accurately predicted by solving Maxwell equations, or the coupled equations between Maxwell equations and Lorentz equation when a beam-wave interaction is considered. However, the large number of variations in the boundary conditions and the long computing time for each parameter set make it not practical to travel over the whole parameter space. As a result, the sample data that can be used in machine learning contains labeled outputs, and the number of sample data is relatively small, on the scale of tens of thousands or smaller. Supervised learning is therefore a better option as the labeled data can be used for training and predicting performance.

Artificial neural networks [21, 22] and decision trees [23, 24] are two major supervised learning algorithms. Artificial neural networks are inspired by biological neural networks that are based on a collection of neurons. It naturally has the advantage of exploring the nonlinear relations between the inputs and outputs. The network can be trained to fit (or learn) any samples and predict the results from given inputs after training. The artificial neural networks operate as a black box. When the network size is large and a complicated neuron connection model is used, it will be hard to interpolate the relationship between the inputs and outputs. Decision tree models are regarded as white boxes. They were initially used to represent classification rules on the input attributes. The decision-making process can be visualized and provide more information on the

connections among the input data or features. A regression tree is a type of decision tree that can deal with continuous output values. Its disadvantage includes overfitting the training data to reduce the actual prediction accuracy when the relations between the inputs and outputs are complicated. The regression trees are normally for single output while artificial neural networks can deal with multiple-output problems. In this paper, both methods were used.

Benefiting from the rapid development of deep learning algorithms and architectures, many commercial and open-sourced machine learning packages have been developed with excellent flexibility and are convenient to use. For example, the machine learning package in Python Scikit-learn [25], the C++ library SHARK [26], and the Statistics and Machine Learning Toolbox in MATLAB [27]. This allows the researchers to process their data without knowing every detail of the machine learning algorithm.

III. THE CASE STUDY USED IN THE RESEARCH

Among the components in an MVED, the interaction circuit requires the most resource when the PIC method is used. The passive components require less computing resource since no electrons are needed in the simulations. The complexity of electron guns design lies between these two cases. The magnetic-field immersed guns, including the MIG [28] and cusp electron gun [29] for the gyrotron devices are more complicated. In this paper, the MIG gun was used as a case study. Although a theoretical model based on the conservation of angular momentum can be used to estimate some parameters, and a synthesis method was developed to provide a good starting point for the design, further optimization was still required as the electric and magnetic fields have to match well to achieve an optimal design.

A schematic of the standard triode-type MIG is shown in Fig. 1 [30]. Its geometry is to be determined by the parameters labeled in Fig. 1. Together with the parameters used to define the magnetic field, there were 15 parameters to be optimized. From the theoretical model and the experience in the electron gun design, the geometry parameters were strongly coupled with each other and the relations among them were complicated. The variation of one parameter could possibly be compensated by a combination of other parameters, which makes it more difficult to find a systematic way to achieve an optimal design by manual tuning.

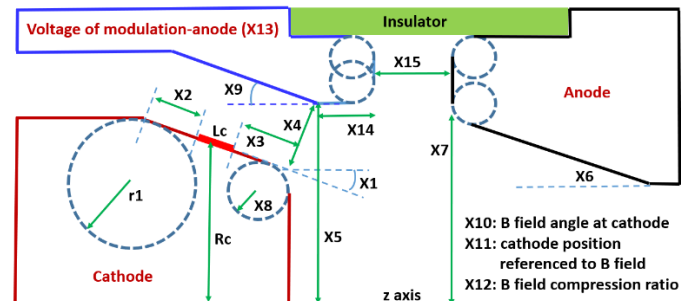


Fig. 1 The parametrized geometry of the triode-type MIG.

The challenge in the optimization is the long computing time needed. Most computer resources are consumed by processes that calculate the particles' trajectories. An iteration process

was used to consider the space charge effect. Normally, 12-20 iterations are normally needed until the solution converges, depending on the strength of the space charge force. When also considering the effect of thermal velocity spread and angular spread of the electrons emitted from the cathode, larger electron samples are needed to represent the distribution. To achieve a higher simulation accuracy, fine meshes are used because the trajectories can be sensitive to the magnetic field.

A MIG gun was simulated and optimized using the genetic algorithm for a high-power gyrokystron [30]. The genetic algorithm is inspired by natural evolution [31, 32]. It stimulates the natural selection process with a certain population generated by crossover, reproduction, and mutation from the initial individuals. Those individuals with higher fitness scores have a higher chance to be kept and used to generate the springs. To find a global optimal MIG gun geometry and to avoid trapping in a local minimum, the parameter ranges were set relatively large which leads to large parameter space. a large population size of 100 was used, and the mutation rate was set relatively high. The optimization was stopped after 400 generations. The computing time for the whole optimization process is three weeks on a PC with a 10-core Intel Core i9 CPU and 32 GB RAM. More details on the simulation setup and final optimized results can be found in the paper [30].

During the optimization, it was found that many parameter sets were labeled as a “bad beam transportation rate” as the entire electron beam did not reach the interaction region, whose location was defined at the exit of the electron gun region. Multiple geometries and magnetic field parameters contribute to this after carefully examining the trajectories of the electrons. When the acceleration electric field strength at the radial direction is too high or the magnetic field at the emitter surface is too low, the electrons may not have enough axial momentum and will directly strike on the modulation anode. When the transverse-to-axial velocity ratio (beam alpha value) at the acceleration region is high, the axial velocities of the electrons will keep reducing at the magnetic field compression region and the electrons will eventually be reflected back to the cathode due to the magnetic mirror effect.

The genetic algorithm was not able to discover the physics inside the electron gun design and the connections among the parameters. It was unavoidable that a large computing resource was “wasted” to simulate the “bad” parameter sets. If the performance of these parameters can be predicted before the simulations, then it would significantly speed up the optimization process.

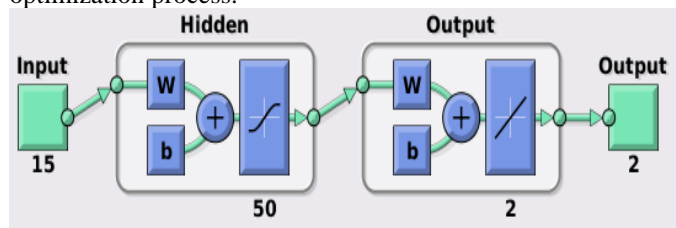


Fig. 2 The structure of artificial neural networks.

In this paper, both the artificial neural networks and decision trees were used to explore the in-depth connections among the geometry parameter, magnetic field parameters as well as beam quality of the given parameter set. For both supervised learning

processes, the inputs were those parameters used for the electron gun simulations. 70% of the input data was used for training and 30% was used for validation. The regression trees were trained with Least-squares boosting and a training iteration of 2000. The feedforward neural networks were chosen as the artificial neural networks with a hidden layer size of 50 and a delay layer size of 20, as shown in Fig. 2. The Levenberg-Marquardt algorithm was used as the training function.

IV. SIMULATION RESULTS

Different functions of the machine learning algorithm were examined, including:

(a) identity of the “bad” parameter sets

Data mining can assist to identify the “bad” parameter sets whose beam transportation rate is poor without the need of running the particle tracking simulations. In supervised learning, the beam transportation rates calculated from the simulation results were used as the labeled output for training and validation. After the validation, the artificial neural networks and the regression trees were used to predict the transportation rate of future populations of the genetic optimization algorithm. The parameter sets predicted with a beam transportation rate less than 100% were identified as “bad” parameters otherwise it would fall within the “good” parameters set, which became a classification problem. The prediction accuracy on the transportation rates for both machine learning algorithms as the function of generation number is shown in Fig. 3.

The sample data number is the product of generation number and population size (100 in the optimization). Both algorithms had similar excellent performance. The accuracy rate was higher than 94% after 4 generations, which was surprisingly high since only a small number of the sample data (400 in this case) was required. Further increasing the sample number would increase the prediction accuracy, however, the accuracy only became stable after 10000 sample data (100 generations). The improvement was not significant considering the sample data used. As only those parameters been predicted to have unity transportation rates were used for the particle tracking simulations, a 94% prediction accuracy resulted in 28% of the particle tracking simulations being saved which could significantly reduce the optimization time.

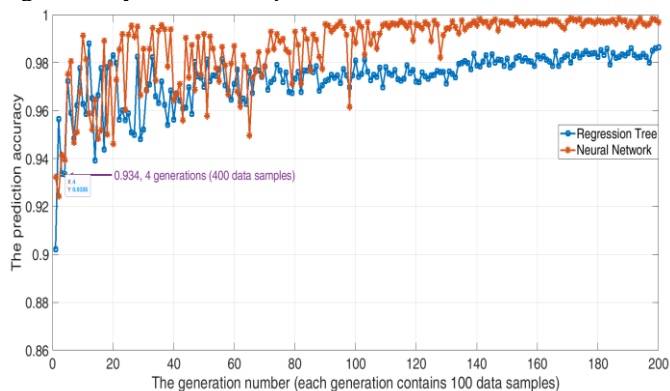


Fig. 3 The prediction accuracy of the artificial neural networks regression trees as the function of the generation number.

(b) Explore the relations between the inputs and outputs

The regression tree can visualize the tree been built and show the importance of the input parameters. Fig. 4(a) shows the importance of the input parameter on the transportation rate, and Fig. 4(b) shows the main branch of the regression tree. It showed the transportation rate was mainly determined by the gap distance of the cathode to modulation anode (X4), the magnetic field angle at the cathode (X10) which was not predicted from the theoretical analysis, magnetic field compression ratio (X12) and the voltage of the modulation anode (X13). The most significant parameter was the gap distance of the cathode to the modulation anode. A small gap distance would result in a strong radial electric field. The electrons might propagate straight across and strike the modulation anode surface if the magnetic field was too weak to bend the electrons.

The relation among the parameters was nonlinear. The regression tree can simplify the complicated relations and provides more suitable parameter ranges, which is useful to narrow down the parameter space during the optimization. A low beam loss can only be obtained by proper combinations of X4, X12, X13 and X10, which are marked as the branches in thicker wires in Fig. 4(b). The compression ratio should not be too big, and the suggested parameter range could also be obtained from the regression tree model. And by giving a modulation anode voltage, the electron beam could reach the interaction region by changing the gap distance X4 or the magnetic field angle. Other parameters also affected the beam transportation rate. However, their contribution was relatively small, which could be examined from other branches.

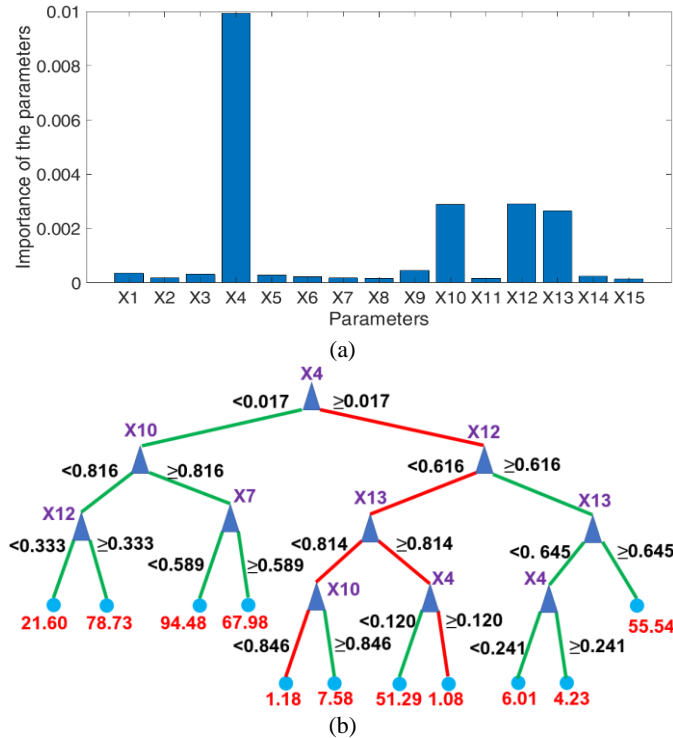


Fig. 4 The dependence of the parameters to the beam lost rate (a), and the structure of the main branch of the regression tree (b).

The output of the optimal MIG gun not only required a high beam transportation rate but also the desired beam properties, including the beam guiding center, average alpha value and the

spread. One of the optimization goals was to achieve the desired alpha value with a minimum spread value, which was expressed as $f_1 = (\sum_n (\alpha_n - \alpha_{desired})^2)^{0.5}$.

A similar analysis was used to study the relationship between f_1 and X_i , as shown in Fig. 5(a), and the tree branch to achieve the small f_1 value is shown in Fig. 5(b). The most important parameters were X12, X13 and X4, which are the same as in Fig. 4(b). The magnetic field compression ratio (X12) was the most important parameter which agreed with the theoretical prediction. X4 and X13 were nearly equally important. In the theoretical analysis, they were a pair of parameters that could be adjusted accordingly to maintain the center alpha value. The partial dependence plot of the regression tree could provide more information, as shown in Fig. 5(c). In the area close to the minimum f_1 , either increasing X4 or reducing X13 would achieve similar f_1 , as marked by the arrows in Fig. 5(c). It also showed how sensitive f_1 depended on X4 and X13. The alpha spread grew rapidly if X4 was too small (<0.1).

As shown in Fig. 5(a), X10 was not as important as X12, X13 and X4, which indicated that if the whole beam could reach the interaction region, then the magnetic field angle had a relatively large tuning range to maintain a small alpha spread.

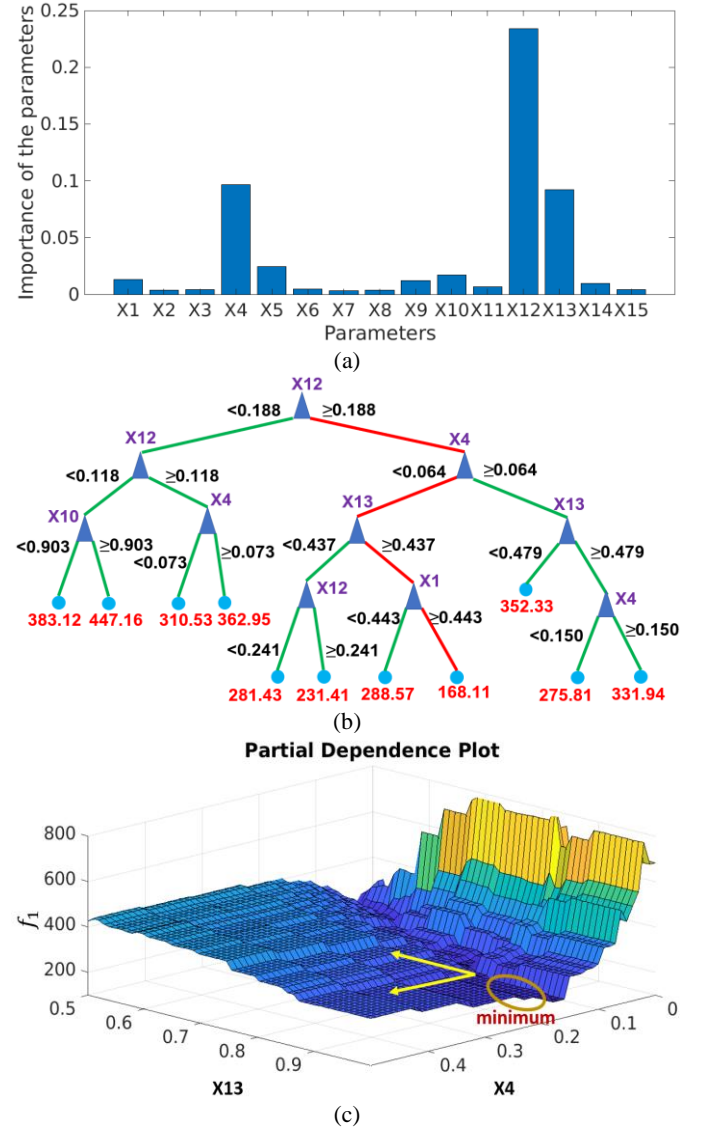


Fig. 5 The dependence of the parameters to the alpha spread (a), the structure of the main branch of the alpha spread (b), and the alpha spread as the function of X4 and X13 (c).

(c) predict the performance of parameter sets

The artificial neural networks and regression trees can also predict the goal function values of the new parameter sets without running the particle tracking simulations. In the genetic optimization algorithm, the goal objective was a combination of beam properties and the optimal design was a trade-off of these requirements. The predicted values obtained from the machining learning algorithms would have errors compared with the ones from the particle tracking simulation due to the complexity of the goal function, which could be expressed as $\Delta = \text{abs}(f_{\text{predicted}}/f_{\text{simulated}} - 1)$. When evaluating the prediction accuracies of the regression trees and the artificial neural networks, the correct prediction was counted to have the error Δ within a defined value. Fig. 6 shows the results at different tolerate error levels.

The regression trees had poor prediction accuracy at tight tolerate error. A 75% prediction rate was achieved if $\Delta \leq 15\%$. It showed that the regression trees did not fit well for complicated nonlinear problems. The artificial neural networks showed much better performance and the prediction accuracy was close to 90% at $\Delta \leq 10\%$. Both algorithms had better prediction accuracy when more training data was used. Artificial neural networks converged faster (~ 120 generations) and required less training data.

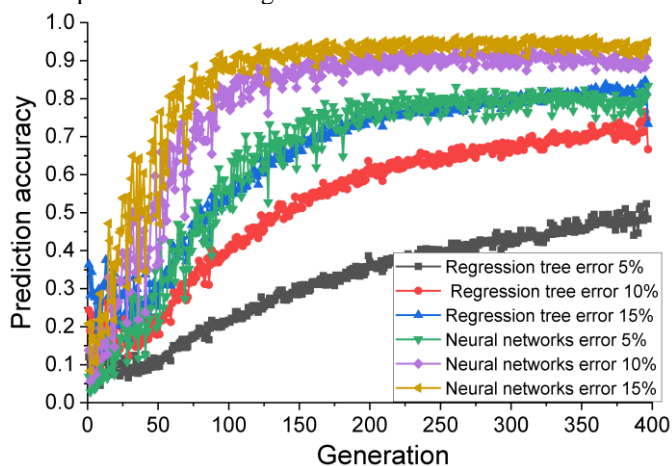


Fig. 6 The prediction accuracies of the goal function by regression trees and neural networks with different tolerate errors Δ .

V. DISCUSSIONS

Two supervised learning algorithms, regression trees and artificial neural networks, were down-selected as the machine learning algorithms in the design of microwave vacuum electron devices. Through the case study of optimizing a MIG for gyrotron devices, the functions of the machine learning algorithm were explored. They include:

(1) Identify the “bad” parameter sets. Both regression trees and artificial neural networks can achieve high prediction accuracy for such classification problems at a small number of

sample data. When integrating this feature into an optimization routine, a significant optimization time can be saved, $\sim 28\%$ in the case study.

(2) The regression trees can be visualized and show the importance of the input parameters. They are excellent tools to explore deep inside the importance of the initial input parameters and to gain knowledge on the components to be designed, to constrain the parameter ranges and narrow down the parameter space, and to discover novel beam-wave interaction circuits and advance the research in millimeter and submillimeter-wave MVEDs.

(3) Predict the performance of the components under investigation. A relatively large number of training data is required to achieve a converged prediction accuracy. The training data size was ~ 8600 in the case study and fewer data may be needed depending on the problem to be solved. With the same training data set, artificial neural networks have much better prediction accuracy than regression trees.

It should be noted that in the case study, the trained network is an assistant to the multiple-objective genetic optimization. The number of particle tracking simulations can be significantly reduced by further applying the knowledge learned. For example, if the importance of the parameters is known, the particle tracking simulations can be carried out more effectively on more important parameter sets, and the parameter range of the unimportant parameters can be narrowed.

The drawback is the data used for training is with specific design requirements, for example, the fixed beam voltage and peak magnetic field. It is difficult to offer a parameter set close to optimal design for other design requirements. However this problem can be possibly solved by increasing the sample data number (ten times the size of the current sample data) at different design conditions, including different beam voltages, beam currents, magnetic fields and others. The future target is to allow machine learning to achieve an optimal MIG design with arbitrary requirements at a minimum particle tracking simulation number of around 1000.

With the rapid progress of machine learning and data mining technologies, it is believed that more powerful features can be integrated into the current CAD tools for MVEDs, to significantly advance the research.

REFERENCES

- [1] A. N. Korolev *et al.*, "Traditional and novel vacuum electron devices," *IEEE Trans. Electron Devices*, vol. 48, no. 12, pp. 2929-2937, 2001, doi: 10.1109/16.974731.
- [2] R. K. Parker, R. H. Abrams, B. G. Danly, and B. Levush, "Vacuum electronics," *IEEE Trans. Microwave Theory Techn.*, vol. 50, no. 3, pp. 835-845, 2002, doi: 10.1109/22.989967.
- [3] R. J. Barker, N. C. Luhmann, J. H. Booske, and G. S. Nusinovich, *Modern Microwave and Millimeter-Wave Power Electronics*. Wiley-IEEE Press, 2005.
- [4] D. Tskhakaya, K. Matyash, R. Schneider, and F. Taccogna, "The Particle-In-Cell Method," *Contributions to Plasma Physics*, vol. 47, no. 8 - 9, pp. 563-594, 2007/12/01 2007, doi: 10.1002/ctpp.200710072.
- [5] S. D. Gedney, "Introduction to the Finite-Difference Time-Domain (FDTD) Method for Electromagnetics," *Synthesis Lectures on Computational Electromagnetics*, vol. 6, no. 1, pp. 1-250, 2011/01/26 2011, doi: 10.2200/S00316ED1V01Y201012CEM027.

- [6] D. B. Davidson, *Computational Electromagnetics for RF and Microwave Engineering*. Cambridge University Press, 2005.
- [7] H. M. Torun and M. Swaminathan, "High-Dimensional Global Optimization Method for High-Frequency Electronic Design," *IEEE Trans. Microwave Theory Techn.*, vol. 67, no. 6, pp. 2128-2142, 2019, doi: 10.1109/TMTT.2019.2915298.
- [8] F. Girolamo and M. Luciano, Eds. *Swarm Intelligence for Electric and Electronic Engineering*. Hershey, PA, USA: IGI Global, 2013, pp. 1-368.
- [9] K. Deb and D. Kalyanmoy, *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, Inc., 2001.
- [10] L. Zhang, W. He, C. R. Donaldson, and A. W. Cross, "Bandwidth Study of the Microwave Reflectors with Rectangular Corrugations," *Journal of Infrared, Millimeter, and Terahertz Waves*, vol. 37, no. 9, pp. 846-856, September 01 2016, doi: 10.1007/s10762-016-0280-y.
- [11] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, p. 255, 2015, doi: 10.1126/science.aaa8415.
- [12] D. Silver *et al.*, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484-489, 2016/01/01 2016, doi: 10.1038/nature16961.
- [13] D. Silver *et al.*, "Mastering the game of Go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354-359, 2017/10/01 2017, doi: 10.1038/nature24270.
- [14] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*. Cambridge: Cambridge University Press, 2014.
- [15] C. Ming-Syan, H. Jiawei, and P. S. Yu, "Data mining: an overview from a database perspective," *IEEE Transactions on Knowledge and Data Engineering*, vol. 8, no. 6, pp. 866-883, 1996, doi: 10.1109/69.553155.
- [16] D. J. Hand and N. M. Adams, "Data Mining," *Wiley StatsRef: Statistics Reference Online*, pp. 1-7, 2015/06/22 2015, doi: 10.1002/9781118445112.stat06466.pub2.
- [17] A. Shrestha and A. Mahmood, "Review of Deep Learning Algorithms and Architectures," *IEEE Access*, vol. 7, pp. 53040-53065, 2019, doi: 10.1109/ACCESS.2019.2912200.
- [18] F. Emmert-Streib, Z. Yang, H. Feng, S. Tripathi, and M. Dehmer, "An Introductory Review of Deep Learning for Prediction Models With Big Data," *Frontiers in Artificial Intelligence*, vol. 3, p. 4, 2020, doi: 10.3389/frai.2020.00004.
- [19] W. Cao, X. Wang, Z. Ming, and J. Gao, "A review on neural networks with random weights," *Neurocomputing*, vol. 275, pp. 278-287, 2018/01/31/ 2018, doi: <https://doi.org/10.1016/j.neucom.2017.08.040>.
- [20] Soniya, S. Paul, and L. Singh, "A review on advances in deep learning," in *2015 IEEE Workshop on Computational Intelligence: Theories, Applications and Future Directions (WC1)*, 14-17 Dec. 2015 2015, pp. 1-6, doi: 10.1109/WCI.2015.7495514.
- [21] A. K. Jain, M. Jianchang, and K. M. Mohiuddin, "Artificial neural networks: a tutorial," *Computer*, vol. 29, no. 3, pp. 31-44, 1996, doi: 10.1109/2.485891.
- [22] T. K. Paradarami, N. D. Bastian, and J. L. Wightman, "A hybrid recommender system using artificial neural networks," *Expert Systems with Applications*, vol. 83, pp. 300-313, 2017/10/15/ 2017, doi: <https://doi.org/10.1016/j.eswa.2017.04.046>.
- [23] W.-Y. Loh, "Classification and Regression Tree Methods," *Wiley StatsRef: Statistics Reference Online*, 2014/09/29 2014, doi: 10.1002/9781118445112.stat03886.
- [24] B. Kamiński, M. Jakubczyk, and P. Szufel, "A framework for sensitivity analysis of decision trees," *Central European Journal of Operations Research*, vol. 26, no. 1, pp. 135-159, 2018/03/01 2018, doi: 10.1007/s10100-017-0479-6.
- [25] "Scikit-learn." <https://scikit-learn.org/stable/> (accessed 2020).
- [26] "SHARK C++ Library." <https://github.com/Shark-ML/Shark> (accessed 2020).
- [27] "Statistics and Machine Learning Toolbox." MATLAB. www.matlab.com (accessed 2020).
- [28] W. Lawson, "Magnetron injection gun scaling," *IEEE Trans. Plasma Sci.*, vol. 16, no. 2, pp. 290-295, 1988, doi: 10.1109/27.3827.
- [29] L. Zhang, W. He, C. R. Donaldson, and A. W. Cross, "Investigation on the optimal magnetic field of a cusp electron gun for a W-band gyro-TWA," *Physics of Plasmas*, vol. 25, no. 5, p. 053104, 2018/05/01 2018, doi: 10.1063/1.5027070.
- [30] L. Zhang, L. J. R. Nix, and A. W. Cross, "Magnetron Injection Gun for High-Power Gyroklystron," *IEEE Trans. Electron Devices*, vol. 67, no. 11, pp. 5151-5157, 2020, doi: 10.1109/TED.2020.3025747.
- [31] D. S. Weile and E. Michielssen, "Genetic algorithm optimization applied to electromagnetics: a review," *IEEE Trans. Antennas Propag.*, vol. 45, no. 3, pp. 343-353, 1997, doi: 10.1109/8.558650.
- [32] S. Katoch, S. S. Chauhan, and V. Kumar, "A review on genetic algorithm: past, present, and future," *Multimedia Tools and Applications*, vol. 80, no. 5, pp. 8091-8126, 2021/02/01 2021, doi: 10.1007/s11042-020-10139-6.