Received: 14 February 2024 | Revised: 16 August 2024 | Accepted: 22 September 2024

The Journal of Engineering

DOI: 10.1049/tje2.70014

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH

TCAD-enabled machine learning framework for DC and RF performance evaluation of InGaAs sub-channel DG-HEMTs

Leeban Moses M[1] | **Saravana Kumar R**[2] | **Muhammad Faheem**[3] | **Ramkumar K**[4] | **Shoukath Ali K**[5] | **Arfat Ahmad Khan**[6]

[1]Bannari Amman Institute of Technology, Sathyamangalam, Tamil Nadu, India

[2]School of Electronics Engineering, Vellore Institute of Technology, Chennai, India

[3]School of Technology and Innovations, University of Vaasa, Vaasa, Finland

[4]Department of Electronics and Communication Engineering, Christ University, Bengaluru, India

[5]Department of Electronics and Communication Engineering, Presidency University, Bengaluru, Karnataka, India

[6]Department of Computer Science, College of Computing, Khon Kaen University, Khon Kaen, Thailand

Correspondence
Muhammad Faheem, School of Technology and Innovations, University of Vaasa, Vaasa 65209, Finland.
Email: muhammfa@uwasa.fi

Funding information
Academy of Finland and the University of Vaasa, Finland

Abstract

This research presents a machine learning (ML)-based model that determines the DC and RF characteristics of InGaAs sub-channel double gate high electron mobility transistors (DG-HEMTs) to optimize the device structure. We employ technology computer-aided design (TCAD) simulations to analyze the DC and RF performance of InGaAs sub-channel DG-HEMTs, generating a range of datasets by varying the material composition, layer width, and thickness of different layers in the device structure. We then train and optimize support vector regression (SVR) models using 5-fold cross-validation, varying the kernel function and degree parameters, and achieve better performance with the radial basis function (RBF) kernel. The simulated results indicate that the ML model predicts physical parameters more effectively than experimental analysis, offering a compact modeling solution that requires fewer computing resources than traditional methods.

1 | INTRODUCTION

To overcome the performance limitations of traditional silicon devices, III–V materials are preferred for their superior mobility, reduced impurity scattering, and enhanced electron confinement at heterointerfaces, making them suitable for HEMT design. With the increasing demand for high-performance, low-cost RF devices, InP HEMTs are emerging as strong competitors in high-frequency applications due to their excellent frequency response, power density, and breakdown voltage [1]. These InP-based HEMTs show significant potential in monolithic microwave integrated circuits (MMICs), especially for atmospheric monitoring, inter-satellite communication, radio astronomy, and space applications [2]. The performance of InP-based HEMT device structure is influenced by the materials used for the channel and gate, the separation distance between the gate and channel, doping techniques, indium concentration in the channel, spacing for gate recess, thickness of both the channel and barrier, as well as the presence of electron traps at the surface and within bulk interfaces [3]. Surface passivation with dielectric materials like silicon nitride (Si_3N_4), silicon dioxide (SiO_2), and aluminium nitride (Al_2O_3)[4] is crucial for enhancing the performance and reliability of InP-based HEMT device structures by reducing surface trap effects and protecting against external damage [5]. Achieving high breakdown voltage, transconductance, and drain current simultaneously in conventional HEMTs is challenging, but asymmetric gate recess technology effectively enhances these parameters

without compromising speed performance by reducing parasitic capacitances [6]. Increasing indium content in InP HEMT channels and reducing gate length enhances electron mobility, velocity, and performance metrics like transconductance, I_{ds}, f_T, and f_{max}, though it can also lead to impact ionization in InGaAs subchannels [7].

Electronic design automation has played a critical role in boosting developments in the electronics industry in the past several decades by significantly reducing the design complexity and time consumption of related tasks. Particularly in semiconductor devices, TCAD has liberated researchers and engineers from solving extraordinarily complicated math problems and providing easy access to simulated results that can hardly be solved manually. Moreover, TCAD has also paved the way for a successful and efficient way to dramatically shrink the cost of experiments during device design so that, in most cases, it is no longer necessary to fabricate numerous batches of devices in a cleanroom to optimize just one geometrical parameter. However, with the rapid development of the semiconductor device industry, TCAD has started to encounter many formidable challenges. Simulating complex three-dimensional device structures consumes considerable computational resources and strains the computational capability of conventional computers to its limit. Although many excellent models and methods have been developed to reduce computational resource consumption and shorten time, exploring novel methodologies of TCAD is still a burning issue in relieving the contradiction between the accuracy and time consumption of sophisticated physics simulations. While large-scale, programmable quantum computers might act as a remedy in the future, they still must overcome many barriers first.

As semiconductor devices shrink, traditional compact modelling becomes increasingly complex, with more parameters and a more difficult calibration process. Conventional HEMT mechanisms face two key challenges: improving new devices requires expertise and time, and achieving full automation with top design accuracy is difficult. The lookup table (LUT) based modelling methodology [8] has been proposed as an alternative to equation models, which is attractive because of the speedy turnaround time and the lack of loss with fitting data. A circuit simulator is used to hold the analysis by designing variables and tables. Optimization is done among the simulator values and design values with the proposed scheme. The process is repeated till an approximate value is attained. Due to the nonlinear behaviour of many active components and the need to cope with the latest conditions in design, many researchers have focused on semiconductor design [9, 10] Optimization is tried with many algorithms, such as an artificial bee colony, particle swarm optimization (PSO), improved PSO [11] and ant lion optimizer algorithms [12].

Recent advancements in TCAD-augmented machine learning techniques enable precise and efficient identification of correlations between device structure and electrical performance, facilitating optimized designs in a shorter timeframe [13]. Machine Learning solutions have attracted strong attention and have been successfully employed in many cases of device modelling due to their better processing speed in real-time implementation [14–18]. Deep neural networks (DNNs) help in building computational models with multiple hidden layers [9, 11, 13, 19–21]. These will help us learn the underlying relationship of data with many levels of abstraction. All these promising applications benefit from the advantage of ML-based modelling, that is, the huge computational resource is only required in model training where a large amount of regression weights is set and calibrated by multiple algorithms. After the training, the model can work efficiently in a low resource consumption mode, and the required computational resource for semiconductor device modelling is within the capability of computer clusters nowadays.

A method to predict transistor current–voltage and capacitance–voltage characteristics using machine learning with TCAD-generated data was generated to map the latent variables to device parameters [18]. The framework described in [16] utilizes ML-TCAD for device parameter and operating condition validation through *I–V* predictions, while also identifying the root cause of discrepancies. A multi-layer neural network model was explored [17] as a replacement for numerical TCAD simulations, with training conducted on predictions across thousands of transistors. A GaN-based prediction model [15] was developed, incorporating numerous memory elements and gate components, where various switching elements were used as training and testing data. However, the models mentioned above are limited to identifying operating bands within a predefined range and cannot predict behaviour outside this range. An algorithmic approach for GaN devices utilizing back propagation regression models is discussed [22], where neural networks are used to calculate the switching characteristics of GaN under various conditions between the source and drain terminals. This strategy enhances ML-based device modelling through self-augmentation and auto-encoder techniques, reducing the need for extensive experimental data by augmenting data points to minimize the mean absolute error.

The work in [23] systematically explores and compares InGaAs sub-channel and InAs composite channel DG-HEMTs by fine-tuning structural and geometrical parameters. To predict structural characteristics and defects, a second-order linear regression algorithm with 1326 expanded features, ensuring balanced training and validation was utilized. A machine-learning-assisted modelling framework for design technology that predicts device and circuit electrical characteristics by solving non-linear equations for multivariate inputs using neural networks. A compact modelling approach is detailed, where sparse datasets and ANN sizing techniques are employed to validate numerous devices and circuits with high accuracy [13]. On comparing ML and artificial neural networks for big data, regression algorithms [20] are more appropriate in sectoring them into small-scale data sets. Simple relationships with models are built between inputs and outputs and provide a series of weights to describe the impact of individual components from a set of input data. On considering the advantages of the linear regression algorithms, an ML-based model using regression methods is selected to investigate the influence of DC and RF characteristics of InGaAs sub-channel double gate high electron mobility transistors. In this work, machine learning

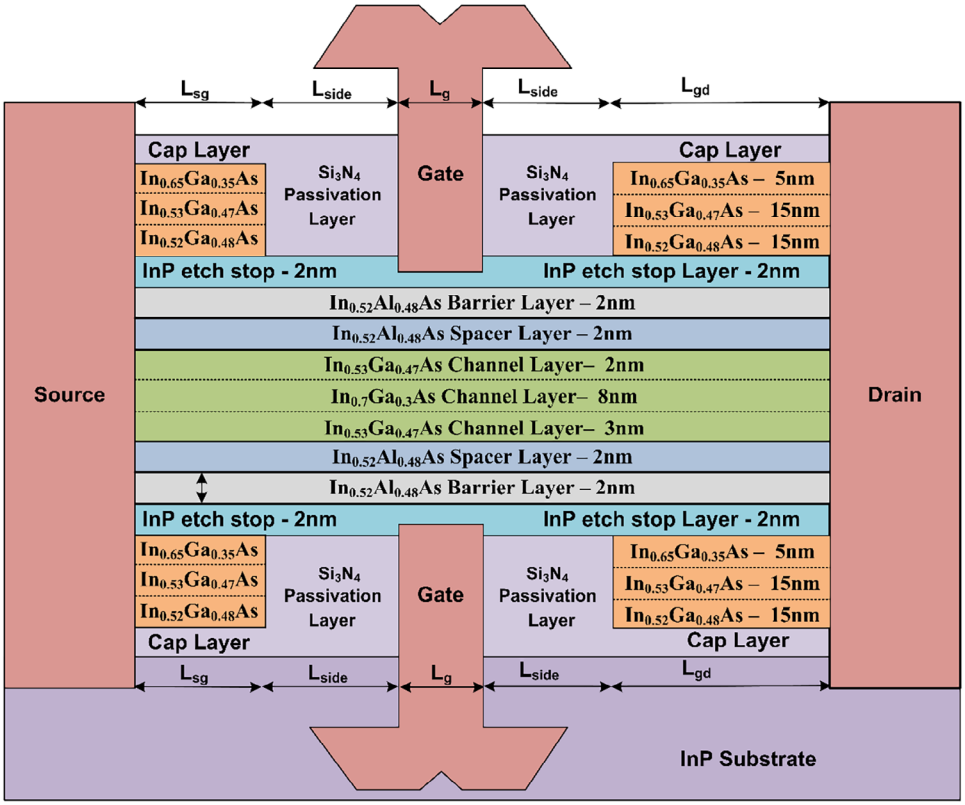

FIGURE 1 Optimized model of the proposed InGaAs sub-channel double gate high electron mobility transistors.

methods and frameworks have been created to make use of ML's capability and apply it to automate and optimize double gate (DG)-HEMTs design procedures and data analysis.

2 | DEVICE STRUCTURE SIMULATION USING TCAD AND DATA GENERATION FOR ML MODEL

Research and applications in the sub-millimeter-wave have drastically improved terahertz microelectronics, which extended its services in imaging systems for security/medical, radars for collision avoidance, and next-generation transport communications. Effective utilization of a sub-millimetre waveband is possible with current-gain cut-off frequency (f_T) and maximum oscillation frequency (f_{max}) closer to 1 THz for semiconductor transistor technologies. Design and optimization of the channel structure for InAlAs/ InGaAs HEMTs is the primary challenge in attaining high sheet electron concentration and electron mobility [24]. A significant rise in electron mobility and saturation velocity is attained by channel InAs mole fraction. The alloy scattering rate is related by x ($1-x$) that increases with x when $x < 0.5$ and decreases for $x > 0.5$ [25]. The trade-off between speed and reliability arises in the epilayer composition of high indium concentration InGaAs. Due to the smaller bandgap in InGaAs, the ionization threshold voltage is also small in the channel of InAlAs/InGaAs HEMTs as shown in Figure 1.

In this work, we propose a DG-HEMT structure on an InP substrate that can be simulated using the TCAD to generate the data set required for the ML model. In the device structure, a 5 and 15 nm InGaAs layer, along with a 15 nm InAlAs multicap layer with variation in mole content, are introduced to decrease the contact and source/drain parasitic resistance, followed by an 84 nm Si_3N_4 passivation layer and a 2 nm InP etch stop layer below the cap layer to decrease the surface defects by providing good quality interface. Next, a thin 2 nm InAlAs barrier layer with thickness varied from 1 to 5 nm is introduced, followed by a 2 nm silicon delta doping layer for supplying the required number of electrons for depletion. To mitigate alloy scattering and enhance channel mobility, a 2 nm undoped spacer layer is placed below the barrier. The double gate structure of the device is provided with InGaAs upper and lower channel layers of 3 and 2 nm with an 8 nm InGaAs sub-channel layer located between the upper and lower channel layers to enhance the RF performance of the device by improving the carrier properties. Shortening the gate length is essential for increasing device speed, but sub-micrometre lengths are limited by lithography, leading to the development of T-gate architecture, which reduces gate resistance and parasitic capacitance and enhances RF and DC performance through the recessed T-gate technique [26]. The gate is recessed up to the etch stop layer to enhance gate control over the channel, with the gate length varied from 1 to 50 nm and gate-to-source and gate-to-drain spacing of 0.5 and 1 μm, respectively. The source and drain contacts, extended to the spacer layers, facilitate immediate electron injection into the channel, with ohmic contacts formed by highly doped InGaAs regions with active arsenic doping. All active layers are identical to the channel from top to bottom,

forming a double gate HEMT structure that operates under a common mode condition, with equal voltages applied to both gate terminals.

To calibrate and validate the proposed device's performance, the simulated I_{ds}–V_{gs} characteristics are compared with experimental results of a single-gate HEMT device with similar material combinations, showing excellent agreement through the appropriate use of physical models. The hydrodynamic (HD) transport model, which accurately simulates non-equilibrium conditions like quasi-ballistic transport, velocity-overshoot in depleted regions, and carrier energy-dependent impact ionization rates, is used for device simulation, also accounting for variable effective mass and doping-dependent mobility at high electric fields. The Canali, Lombardi, and Arora models are used to simulate velocity saturation and mobility degradation due to high electric fields, while the Shockley–Read–Hall and van Overstraeten–de Man models accurately represent carrier lifetime, density, and hot-carrier effects. From the experimental simulations, we can observe that higher peaks are obtained for I_{ds} and g_m by controlling the thickness of the barrier layer (T_b), the drain to source voltage (V_{ds}) and the gate length (L_g). When L_g is increased, the channel resistance gets increased, and this effect makes I_{ds} and g_m lower the peak values. To optimize the structure for higher peaks of I_{ds} and g_m curve, the data set has been created for a few values of L_g, T_b and V_{ds}. To analyze the V_{ds} and I_{ds} curve, L_g, T_b and V_{ds} vary from their lower limit to the higher limit value, and the analysis concludes that I_{ds} increase when L_g decreases. To optimize the proposed structure, a data set has been created for a few values of L_g, T_b and V_{ds} for obtaining higher values of I_{ds}. It is observed that V_T increases on decreasing the T_b, and V_T remains more positive for high values of L_g with controlled values of V_{gs}, I_{ds} and V_{ds}. From the TCAD simulations, we can notice that the I_{ON}/I_{OFF} ratio gets diminished by increasing the value of T_b, and this happened because of an increase in I_{ON} and I_{OFF}; the I_{ON}/I_{OFF} ratio gets improved by increasing the values of L_g since it reduces the off-state leakage current.

The drain induced barrier lowering (DIBL) parameter is lowered when both T_b and L_g get increased, and this effect is due to the reduction of electrostatic gate control and lateral electric field. DBIL gets improved by increasing the values of L_g and T_b and by maintaining V_{ds} to a constant value. The RF performance measure for DG-HEMT structure is determined by f_T and f_{max}. Higher peaks of the frequency curves are obtained by increasing and decreasing the values of V_{gs}, L_g, T_b, I_{ds} and V_{ds}. From the TCAD simulations, it is noted that f_T and f_{max} increase and reach maximum when we increase V_{gs} and V_{ds} and decrease L_g and T_b to an acceptable level. From the experimental analysis, it is observed that the structure of the device and the biasing voltages have to be varied in order to obtain higher performance. To optimize the structure using ML model, a data set has been created in this work for V_{gs}–I_{ds} curve with g_m, V_{ds}–I_{ds} curve, V_T, I_{ON}/I_{OFF} ratio, DBIL, f_T and f_{max} by increasing and decreasing the values of T_b and L_g to an acceptable level. The device with material properties and parameters proposed in [21] provides an appropriate match between experimental and simulation results used in the device structure.

The backpropagation model is used to train machine learning during the development process, but it fails to guess the initial dependence, and by doing so, the best-fitting function may not be obtained at its output. The wrong choice of parameters will produce fitting curves that degrade the performance of the ML model. To neglect this inadequacy, optimization in the data set is needed for the initiation of the training process by exploring the search space to find an optimal solution. Combining optimization and machine learning techniques in a hybrid algorithm enhances decision-making by leveraging both methodologies for improved accuracy and efficiency. Metaheuristic algorithms, used for complex optimization problems, are classified by their exploitation and exploration trade-offs and should be benchmarked on tasks like the target design optimization problem. Hybrid optimizations dynamically select the most suitable algorithm from a set at compile time using heuristics to optimize each piece of code. The introduction of an improved particle swarm whale optimizer and Brownian movement and improved Levy's flight-based dragonfly optimizer prior to the training process enhances the efficiency and accuracy of the ML model and are considered as an extra benefit of the model. The optimized data set is fed into the regression models such as linear regression, random forest regression, KNN regression and support vector regression with linear, polynomial, sigmoid and RBF kernels to predict the accurate values of I_{ds}, g_m, V_T, I_{ON}/I_{OFF} ratio, DBIL, f_T and f_{max} values for an optimistic device variation of T_b and L_g.

2.1 | Optimization in data set obtained from TCAD using improved particle swarm whale optimizer (IPSWO) algorithm

Particle swarm optimization is a computationally efficient, nature-inspired algorithm that mimics the social behaviour of fish and birds to find optimal solutions [27]. Particles in a swarm update their individual best positions and the swarm's best position as they navigate the search space. Inspired by the hunting behaviours of humpback whales, the whale optimization algorithm uses strategies of encircling prey, searching for prey, and spiral bubble-net attacking to find optimal solutions. The algorithm assumes the best current candidate is near the optimal solution, and other search agents update their positions towards this best candidate. Balancing exploration and exploitation is crucial in optimization algorithms. WOA excels in exploration but may take longer to converge if the distance to the optimal solution is large, while PSO converges quickly but can get trapped in local optima [28] due to its limited global search capability. The improved particle swarm whale optimizer (IPSWO) algorithm combines WOA's exploration with PSO's exploitation to avoid premature convergence and improve solution accuracy [29, 30]. Figure 2 depicts the procedure for creating a dataset for the SVR model from the TCAD simulations.

The first step is to generate test data and training data from the experimental analysis, and the training data is normalized by removing the redundant and repeated data by considering this as the initial population for the PSO algorithm [31]. The

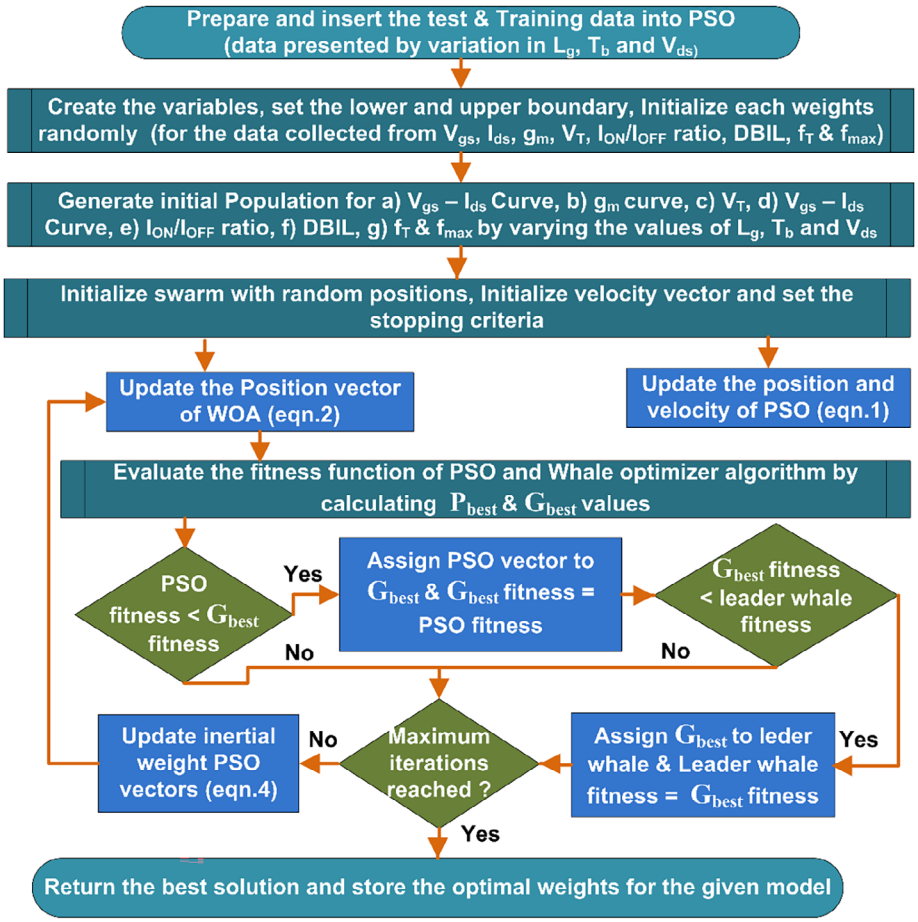

FIGURE 2 Flow chart representing the steps of dataset creation for the proposed model using improved particle swarm whale optimizer (IPSWO) algorithm.

initial population is created based on the values obtained from (a) V_{gs}–I_{ds} curve, (b) g_m curve, (c) V_T, (d) V_{gs}–I_{ds} curve, (e) I_{ON}/I_{OFF} ratio, (f) DBIL, (g) f_T and f_{max} by varying the values of L_g, T_b and V_{ds}. Calculating the fitness value, the PSO algorithm sets particle-best (P_{best}), which is the finest fitness value attained by a particle and global-best (G_{best}), which is the finest fitness value grasped by the population.

$$v_{t+1}^{PSO} = \omega * v_t + c_1 * r_1 \left(p_{best}^t - x_t\right) + c_2 * r_2 \left(g_{best}^t - x_t\right)$$
$$x_{t+1} = x_t + v_{t+1} \tag{1}$$

where r_1 and r_2 are randomly selected amongst the maximum and minimum values. The current velocity function is defined as V_T and updated velocity can be denoted as V_{T+1}, the current position is defined as x_t and updated position can be denoted as x_{t+1}. The positions of the WOA are selected randomly, when the prey is selected randomly among the search space and by selecting the best prey, the position vectors are updated.

$$\overline{x_{t+1}^{WOA}} = \overline{x_{random}} - \left\{\left(2\bar{i} * \bar{r} - \bar{i}\right) * \left|\left(\bar{c} * \overline{x_{random}}\right) - \bar{x}\right|\right\}$$
$$\overline{x_{t+1}^{E}} = \overline{x_{up-prey}} - \left\{\left(2\bar{i} * \bar{r} - \bar{i}\right) * \left|\left(\bar{c} * \overline{x_{up-prey}}\right) - \bar{x}\right|\right\} \tag{2}$$

Here, r denotes the randomly chosen value between 0 and 1, i depends upon the number of iterations by reducing the value between 2 and 0, x_{radom} denotes the vector with the movement of prey randomly and $x_{up\text{-}prey}$ denotes the updated positions upon the prey. IPSWO combines WOA and PSO by using WOA's logarithmic spiral function for broader exploration and faster convergence, replacing PSO's personal best vector with WOA's updated vector to achieve the optimal solution efficiently. The PSO's adaptive weight function is used alongside WOA to enhance exploitation and improve convergence speed, updating the search agent's velocity based on the best position in the search space. The position and velocity can be updated by using the equation.

$$v_{t+1}^{IPSWO} = \omega * v_t + c_1 * r_1 \left(\overline{x_{t+1}^{WOA}} - x_t\right) + c_2 * r_2 \left(g_{best}^t - x_t\right)$$
$$x_{t+1} = x_t + v_{t+1}^{IPSWO} \tag{3}$$

Using PSO, low inertial weight can cause local trapping, while high inertial weight slows convergence. To address this, a self-adaptive parameter control technique based on each individual's fitness data was employed for better searching.

$$\omega_{t+1}^{\mathrm{IPSWO}} = \begin{cases} \omega_x + \left(\omega^t - \omega_x\right) * \dfrac{f\left(x_{t+1}\right) - f_x}{f_a - f_x} & ; r_{\mathrm{next}} < \delta \mathrm{and} f\left(x_{t+1}\right) < f_x \\ r_{\mathrm{next}+1} & ; r_{\mathrm{next}} < \delta \mathrm{and} f\left(x_{t+1}\right) < f_x \\ \omega^t & ; \mathrm{otherwise} \end{cases} \tag{4}$$

Here δ denotes a probability factor introduced to adjust the weight parameters, ω_x denotes the minimum value of the inertial weight, r_{next} denotes a random number depending upon the inertial weight varying from 0 to 1, $r_{\mathrm{next}+1}$ denotes a weight factor value that varies between minimum and maximum weight.

2.2 | Optimization in data set obtained from TCAD using Brownian movement and improved Levy's flight-based dragonfly (BMILFD) algorithm

Dragonflies exhibit two key behaviours: forming pairs to hunt other insects and separating to evade predators. These behaviours align with the optimization processes of exploration and exploitation. To model this mathematically for optimization problems, five weights are used: separation function (S_{F}), alignment function (A_{F}), cohesion function (C_{F}), food factor representation function (F_{F}), and enemy factor representation function (E_{F}), along with an inertia weight representation function (IW_{F}), guiding the dragonfly towards food and away from enemies [32]. To achieve results in the search space, various dragonfly behaviours are defined. The initial step, representing S_{F}, deals with collision avoidance with the nearest dragonfly to avoid overcrowding of the data presented by variation in L_{g}, T_{b} and V_{ds}. Let μ denote the number of data which represent all the output parameters by varying L_{g}, T_{b} and V_{ds} with the mole content and thickness of each layer in the space, with μ_x denoting the position of the x^{th} dragonfly and μ_y denoting the position of the y^{th} dragonfly. The second step is the alignment function, which is the tendency of dragonflies to match their velocities with the average velocity of their neighbours, promoting synchronized movement. This behaviour ensures cohesive group movement and reduces erratic trajectories for stability.

$$S_F = -\sum_{n=1}^{i+j} \mu_x - \mu_y^{i+j} \tag{5}$$

$$A_F = \frac{1}{M} \sum_{n=1}^{i+j} \nu_{i,j} \tag{6}$$

Here, $i + j$ represents the maximum number of data available in space with respect to V_{gs}, I_{ds}, g_{m}, V_{T}, $I_{\mathrm{ON}}/I_{\mathrm{OFF}}$ ratio, DBIL, f_{T} a $f_{\max}$. M represents the total number of data available, which denotes the dragonflies. $\nu_{i,j}$ velocity of j^{th} dragonfly to i^{th} dragonfly. Cohesion is the tendency of dragonflies to move towards the centre of mass of their neighbours, keeping the group together. This behaviour maintains group integrity and optimal swarm density, balancing exploration and exploitation. The fourth step, which represents the food factor representation function, guides dragonflies toward optimal solutions by drawing them towards a food source, helping in the exploitation of promising areas. The fifth step, the enemy factor representation function, steers the dragonflies away from threats or suboptimal solutions, promoting exploration and preventing premature convergence. Together, F_{F} and E_{F} balance the search process by refining good solutions and avoiding poor ones. In the dragonfly optimization algorithm, dragonflies update their positions based on these influences, effectively navigating the search space to find the best solution.

$$C_F = \left(\frac{\sum_{n=1}^{i+j} \mu_y^{i+j}}{M} \right) - \mu_x \tag{7}$$

$$F_F = \mu_{FA} - \mu_x \tag{8}$$

$$E_F = \mu_{En} - \mu_x \tag{9}$$

Here μ_{FA}, denotes the food availability for the dragonflies in their respective groups formed at cohesion, and μ_{En} denotes the availability of enemies that represent the unnecessary data after grouping. The position ranking based on the good and bad positions are described by these two factors μ_{FA} and μ_{En}. The next step in the optimization algorithm is to update the movement of the dragonflies based on their velocity and direction of movement to determine the best and worst positions of the dragonflies.

$$\mu_x(t+1) = \{a.S_F + B.A_F + C.C_F + d.F_F + e.E_F\} + \beta\left(\delta[\mu(t+1)]\right) \tag{10}$$

Here, the weight factors a, b, c, d, e represents an additional parameter added to S_{F}, A_{F}, C_{F}, F_{F}, and E_{F} to evaluate the dragonflies' best and worst positions. The parameter β denotes the change in values depending upon the input parameter, and t represents the number of times to update the movement of dragonflies. When there are no neighbouring dragonflies, they use the Levy-flight mechanism to search the surrounding space,

and their new position can be updated using the following expression.

$$\mu_{t+1}^{x} = \mu_t^{x} + \left\{ 0.3 * \left(\frac{y_1 * \omega}{|y_2|^{1/\varphi}} \right) \right\} \tag{11}$$

Here, y_1 and y_2 values are selected randomly between 0 and 1 depending upon the Levy nature of dragonfly movement, and the value of φ is fixed at 1.5. The Levy flight factor ω can be represented as,

$$\omega = \left\{ \frac{\Psi(1+\varphi).\sin\left(\frac{\pi}{2} * \varphi\right)}{\Psi\left(\frac{1+\varphi}{2}\right) * \left(\sqrt{\frac{\varphi}{2}} * 2^{(\varphi-1)}\right)} \right\} \tag{12}$$

A Levy flight factor is introduced to fine-tune priority factors and optimize the weights among dragonflies and the available resources. Dragonflies exhibit periodic motion with a normal distribution, characterized by sudden jumps and random movements akin to vibrations. Brownian motion steps are determined using a Gaussian distribution, and the equation is adjusted based on the size and number of agents in the algorithm, finalizing the Brownian motion. The updated position can be expressed as:

$$\mu_{t+1}^{x} = \mu_t^{x} + \left\{ \sqrt{\frac{0.01}{N} rand()} \right\} * \left\{ \frac{1}{b\sqrt{\pi}} \right\} e^{\left(\frac{D-G}{2b^2}\right)} \tag{13}$$

Here, the time period for the velocity of motion is fixed as a constant to 0.01, D denotes the dimension of agents, and G denotes the number of agents as shown in Figure 3.

3 | MACHINE LEARNING ALGORITHM AND PREDICTION USING THE OPTIMIZED DATASET

3.1 | Linear regression

Regression analysis is a statistical technique that uses observed data to develop a model characterizing the relationship between a dependent variable and a set of independent variables [33]. This model helps predict, describe, and control the value of the dependent variable based on the values of the independent variables. In this linear regression model, the dependent variable x is associated with the independent variable y, and the model forecasts x values by maximizing the regression fitness value based on the y variable population. The optimized dataset created by IPSWO and the BMILFD algorithm is given as input to the regression model. The first step in linear regression is to calculate the best-fit line using the slope-intercept form, which minimizes the error between predicted and actual values. Mathematically, this best-fit line is obtained by minimizing the residual sum of squares and is given by,

$$Y = \gamma_0 + \gamma_1 x_1 + \gamma_2 x_2 + \gamma_3 x_3 + \cdots\cdots .. + \gamma_m x_m \tag{14}$$

Here γ_0, γ_1, ….. γ_m represents the parameter used in the regression function to calculate the relation between x and y, n represents the maximum data in the independent variable. The goal of linear regression is to find the optimal values for $B0$ and $B1$ to minimize the error between predicted and actual values, resulting in the best-fit line. This is typically achieved using the mean squared error cost function, which calculates the average squared error between the predicted values and actual data points. The cost function is given as,

$$C_{\text{linear}}(\gamma) = \frac{1}{N}\sum_{i=1}^{n}\left[Y - \overline{Y_n}\right]^2 \tag{15}$$

Here $\overline{Y_n}$ represents the calculation of predicting the n^{th} value of the dependent variable Y. The cost function helps determine the optimal values for γ_0 and γ_1 by using the MSE function and gradient descent method to minimize the cost and achieve the best-fit line. Gradient descent optimizes the cost function by iteratively updating the slope γ_1 and constant γ_0 coefficients to minimize the MSE, achieving the optimal solution for the linear regression model and the learning rate determines the speed at which the algorithm converges to the minimum.

3.2 | Random forest regression

The random forest algorithm is a powerful machine learning technique that creates multiple decision trees using random subsets of data and features, reducing overfitting and improving prediction performance [34]. During prediction, it aggregates the results from all trees by voting for classification tasks or averaging for regression tasks, ensuring stable and precise results. Random forest uses ensemble learning by building multiple independent decision trees, each focusing on different data aspects to minimize the influence of any single tree. Random forest promotes diversity among its decision trees by using random feature selection, where each tree is trained on a different subset of features to capture various aspects of the data. Bagging is a key training strategy in random forest that generates multiple bootstrap samples from the original dataset, enabling each decision tree to train on a unique subset of data and enhancing the model's robustness. In random forest, each decision tree contributes to predictions by voting, with the final classification determined by the mode of the trees' predictions and regression predictions averaged, ensuring a balanced decision-making process.

$$RF_{\text{prediction}} = \frac{1}{M}\sum_{m=1}^{M}\mu_M(x) \tag{16}$$

Regression trees exhibit low bias and high variance, and the random forest prediction is obtained by averaging the mean predictions from N regression trees, $\mu_N(x)$. The samples not selected for training the N^{th} regression tree during bagging form the out-of-bag dataset, which typically comprises about one-third of the original dataset, and the tree's performance is

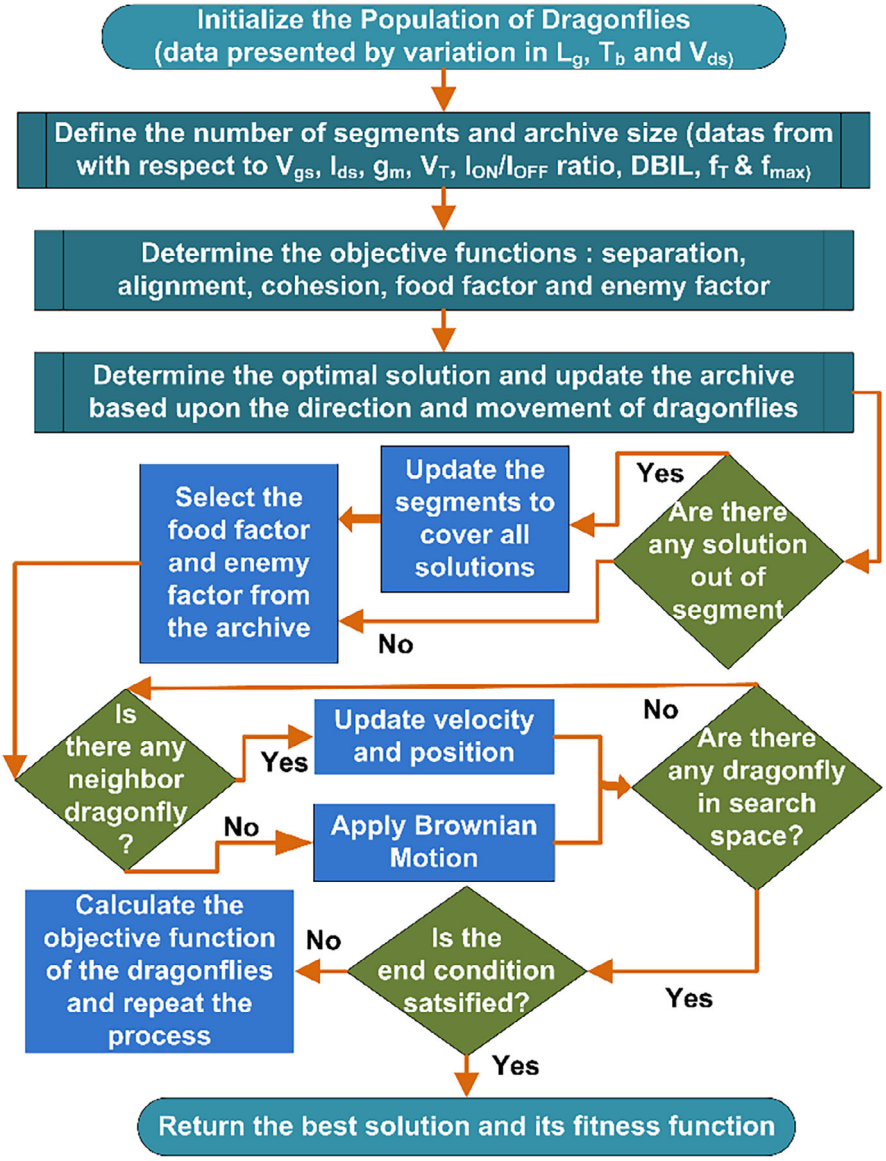

FIGURE 3 Flow chart representing the steps of dataset creation for the proposed model using Brownian movement and improved Levy's flight-based dragonfly (BMILFD) algorithm.

evaluated using the mean squared error cost function.

$$C_{\mathrm{RF}}(\gamma) = \frac{1}{N}\sum_{i=1}^{n}\left[Y - \overline{Y_n}\right]^2 \tag{17}$$

Here Y denotes the n^{th} predicted value, and $\overline{Y_n}$ denotes the mean of n^{th} predicted value from all the trees. The coefficient of determination R^2_{RF} represents the proportion of variance in the dependent variable that can be predicted from the independent variables.

$$R^2_{\mathrm{RF}} = 1 - \frac{\sum_{i=1}^{n}\left[Y - \overline{Y_n}\right]^2}{\sum_{i=1}^{n}\left[Y - \Gamma_n\right]^2} \tag{18}$$

Here, Y denotes the results obtained from the TCAD output model and Γ_n denote the predicted value obtained from the random forest regression model. The denominator term represents the total variance of the output parameter.

3.3 | K-nearest neighbour regression

The K-nearest neighbours (KNN) algorithm, used for regression, stores the entire training dataset and makes predictions by calculating the distance between the input data point and all training examples using a chosen metric like Euclidean distance [35]. The KNN algorithm predicts new data points' values based on their feature similarity to points in the training set. To predict a new point, the KNN algorithm first calculates its distance to each training point using methods like Euclidean, Minkowski, Manhattan and Chebyshev. After measuring the distances, the

k nearest neighbours are selected, where k is a predetermined value. The selected neighbours are then used for further analysis or prediction.

$$\begin{aligned} d_{\text{Euclidean}} &= \sqrt{\sum_{n=1}^{N} (X_n - Y_n)^2} \\ d_{\text{Minkowski}} &= \left(\sum_{n=1}^{N} (X_n - Y_n)^a \right)^{1/a} \\ d_{\text{Manhattan}} &= \sum_{n=1}^{N} |X_n - Y_n| \\ d_{\text{Chebyshev}} &= \max (|X_n - Y_n|) \end{aligned} \tag{19}$$

Here X_n and Y_n denote the independent variables in space vector N and the parameter a denotes a constant Minkowski function. Choosing the hyperparameter k in KNN is a trade-off: low values of k can lead to overfitting due to noisy data, resulting in a flexible regressor with low bias but high variance. Conversely, high values of k may cause underfitting by averaging too many distant data points, leading to a regressor with high bias and low variance. An optimal k balances bias and variance, ensuring a well-constructed model. The best value of hyperparameter k is determined using cross-validation on various randomized training sets, selecting the one that minimizes the prediction error, with lower MSE indicating a more accurate model.

3.4 | Support vector regression model

SVR is a machine learning technique established for nonlinear regression approximation, in which a model learns the importance of a variable in characterizing the relationship between input and output. The flexible nature of SVR is attributed to the kernel function that implicitly converts data into a higher-dimensional feature space and is used to deal with non-linear problems by computing the inner product in a higher-dimensional feature space. A radial basis function (RBF), which is employed under a nonlinear kernel function, provides a non-linear solution in a higher-dimensional feature space and moderates the dimensional input space by reducing the training process and escalating efficiency [36]. This RBF kernel function, decision boundary and box constraint, C, determines the efficiency of SVR and substitutes the lower dimensional dataset into the higher dimensional dataset to acquire the non-linear behaviour. From TCAD simulations: (a) higher peaks of V_{gs}–I_{ds} and g_m curve and V_{ds}–I_{ds} curve are obtained by varying the values of L_g, T_b and V_{ds}; (b) variation in V_T, I_{ON}/I_{OFF} ratio, DBIL, f_T and f_{max} are obtained by varying the values of L_g, T_b and V_{ds}. The predicted value $p(x)$ can be obtained. SVR procedure attempts to discover the regression equation, which is maximally deviated by ε from the true values.

Consider a training data set of input vector $\{I_j\}_{j=1}^{M}$ and output vector $\{O_j\}_{j=1}^{M}$ where M denotes the amount of data points considered. The SVR tries to calculate a function which contains ε variations from the essentially attained targets for the training data. The objective of SVR is to determine the regression function of the output variable and is given by

$$f(I) = w^T * \psi(I) + a \tag{20}$$

The input vector I_j is equated to higher-dimensional space, which can provide a better solution by assigning the input vectors from lower-dimensional space to higher-dimensional space using a simple multiplicative function. The learning equation for the proposed model prescribed in Equation (2) can be remodelled by using the error function ε and is given by,

$$\min \left\{ \frac{1}{2} \|w^2\| + C \sum_{j=1}^{M} L(\varepsilon) \right\} \tag{21}$$

$$L(\varepsilon) = \begin{Bmatrix} 0 & if \ |O - f(I)| < \varepsilon \\ |O - f(I)| - \varepsilon & \text{otherwise} \end{Bmatrix} \tag{22}$$

The equation above represents the learning information based on the error function; this representation will never consider the errors ranging from $-\varepsilon$ to $+\varepsilon$ and determines the distance between the output function O_j and the error function. The parameter C provides the balance between empirical risk and model flatness. Therefore, optimum values should be properly selected for the parameters C and ε by conducting many trials. For a default value of ε as 0.1, the parameter C can be selected as 100 for the RBF kernel. The discrepancies between both the absolute measurements as well as the appropriate boundary variables at the top and bottom of the ε-tube, respectively, are represented by two slack variables. The regression function may then be transformed into the constraint below.

$$\begin{aligned} &\min \left\{ \frac{1}{2} \|w^2\| + C \sum_{j=1}^{M} \left(\xi_j + \xi_j^* \right) \right\} \\ &\text{subject to } |O - f(I)| < \varepsilon + \xi_j; \\ &|f(I) - O| < \varepsilon + \xi_j^*; \ \xi_j, \ \xi_j^* \geq 0 \end{aligned} \tag{23}$$

Since the training variables lie beyond the boundary set by the ε-tube and slack variables, this can be considered as the optimization problem and is solved by using the Lagrange multiplier method.

$$\begin{aligned} L = &\frac{1}{2} \|w^2\| + C \sum_{j=1}^{M} \left(\xi_j + \xi_j^* \right) \\ &- \sum_{i=1}^{N} \left(\chi_i \xi_j + \chi_i^* \xi_j^* \right) + \sum_{i=1}^{N} \left(\beta_i (f(I_i) - O_i - \varepsilon + \xi_j) \right) \\ &+ \sum_{i=1}^{N} \left(\beta_i^* (O_i - f(I_i) - \varepsilon + \xi_j^*) \right) \end{aligned} \tag{24}$$

Here χ_i, χ_i^*, β_i, β_j^* denotes the positive Lagrange multiplier variables and the optimum variables ξ_j, ξ_j^*, w, C are adjusted

by differentiating them and equating to zero, in-order to obtain optimal solution. The optimal solution is obtained by adjusting the variables for prediction and introducing the kernel function by replacing the original function and is given by,

$$\max \left\{ \sum_{j=1}^{M} \left[O_j \left(\beta_j + \beta_j^* \right) - \varepsilon \left(\beta_j + \beta_j^* \right) \right] - \frac{1}{2} \sum_{j=1}^{M} \sum_{k=1}^{N} \left[\left(\beta_j^* - \beta_j \right) \left(\beta_k^* - \beta_k \right) K \left(I_j, I_k \right) \right] \right\} \quad (25)$$

Here $K(I_j, I_k)$ denotes the kernel function. The optimal solution can be obtained by solving the above equation with the kernel function and the obtained solution can be considered as the best regression function and is denoted as

$$F(I) = \sum_{j=1}^{M} \left[\left(\beta_j - \beta_j^* \right) K \left(I_j, I_k \right) \right] + a \quad (26)$$

$$\text{linear Kernel, } K \left(I_j, I_k \right) = I^T * I_j \quad (27)$$

$$\text{polynomial Kernel, } K \left(I_j, I_k \right) = \left(\Gamma (I^T * I_j) - 1 \right)^p \quad (28)$$

$$\text{Sigmoid Kernel, } K \left(I_j, I_k \right) = \tanh \left(\left(\Gamma (I^T * I_j) - 1 \right) \right) \quad (29)$$

$$\text{rbf Kernel, } K \left(I_j, I_k \right) = \exp \left(- \frac{\left\| I - I_j \right\|^2}{2 \sigma^2} \right) \quad (30)$$

Here, I_j and I denote the Euclidean distance between two input vectors considered inside the ε-tube, and σ denotes the hyperparameter, which can also be considered as the variance. For the polynomial kernel function, the value of p should be constant and greater than 0, which defines the order of the polynomial kernel. The linear kernel calculates the dot product for linearly separable data, while the polynomial kernel computes similarity as a polynomial function for non-linear data. The RBF kernel maps data into an infinite-dimensional space for non-linear decision boundaries, whereas the sigmoid kernel, based on the hyperbolic tangent function, is an alternative for non-linear classifiers. Gaussian RBF function is used as the kernel in the SVR algorithm for predicting various parameters in the design of an InAlAs/InGaAs HEMT device structure because they can map the nonlinear input data obtained from simulations to an infinite dimensional space.

4 | RESULTS AND DISCUSSION

In this work, the InGaAs sub-channel DG-HEMTs model was created, and their performances were evaluated using TCAD simulation. Since the analysis of DC and RF characteristics of the device has a greater number of iterations in the TCAD model, the optimal design and their characteristics were obtained from a machine learning framework. To obtain this ML-based framework, a dataset was created to determine the values of V_{gs}, g_m, V_{ds}, I_{ds}, V_T, I_{ON}/I_{OFF} ratio, sub-threshold swing and maximum frequency of oscillation by increasing and decreasing the values of T_b, L_g, V_{gs} and V_{ds} to an acceptable level. The created data set is obtained from the TCAD simulation without any manual feature extraction and is used to train the ML model by optimizing the unknown parameters to a desired value to avoid overfitting issues. To reduce the features by optimizing the data for ML models, we use the improved particle swarm whale optimizer meta-heuristic algorithm and Brownian movement and improved Levy's flight-based dragonfly optimizer meta-heuristic algorithm. The optimized dataset was split into 80% for training and 20% for testing, with training data used to learn hyperparameters in various ML models. Standard scaler and Min-Max-Scaler from Scikit-Learn were applied to standardize the input data and normalize the output data, respectively, during the process. We use four different ML models as linear regression, random forest regression, K-nearest neighbour regression and support vector regression with linear, polynomial, sigmoid and RBF kernel functions to predict the values of the V_{gs}, g_m, V_{ds}, I_{ds}, V_T, I_{ON}/I_{OFF} ratio, sub-threshold swing and f_{max}. The performance of models was evaluated using the coefficient of determination (R^2), mean absolute error (MAE) assessing the model's fit, and mean squared error (MSE) quantifying the average deviation between observed and predicted values. The trained SVR model with RBF kernel in Table 1 shows a reduction in MAE and MSE, which can predict the characteristics of the device simulated for various changes in barrier thickness and gate length with the variations in the gate-to-source voltage and gate-to-drain voltage.

The trained SVR model with RBF kernel, which utilizes IPSWO as an optimization in the data set, is selected for predicting the output parameters due to its reduction in MAE and MSE, as shown in Table 1. The results shown below provide a comparison between the TCAD model and the ML-based predicted model, in which the red colour indicates the ML-based predicted model and the black colour indicates the TCAD model. Figure 4 shows the comparison of modelling results between the ML-based prediction model and TCAD model for the variations that occurred in g_m and I_{ds} when there is a change in V_{gs} and V_{ds} from 0 to 1 V, L_g from 30 to 120 nm and T_b from 1 to 5 nm. From the results shown, it is evident that when T_b is 2 nm, we obtain higher peaks in g_m of 3.068 ms/μm and I_{ds} 0.974 mA/μm of value for a fixed value of V_{ds} as 0.5 V and L_g as 30 nm. The double-gate recessed control in DG-HEMTs enables faster depletion and increased electron density by reducing the gate-to-channel distance. This design leads to higher peak drain current and transconductance due to enhanced carrier concentrations and average drift velocity in the channel. When L_g is increased or decreased beyond 30 nm, due to the increase in channel resistance the I_{ds} and g_m starts to decrease. The SVR-RBF prediction model with IPSWO optimization function has 2086 training data set for different variations in V_{gs}, V_{ds}, L_g and T_b. The results from the graph in

TABLE 1 Comparison of regression models: R^2, mean absolute error (MAE) and mean squared error (MSE) analysis with and without optimization.

	R^2			MAE			MSE		
Regression models	**Without optimizer**	**IPSWO**	**BMILFD**	**Without optimization**	**IPSWO**	**BMILFD**	**Without optimizer**	**IPSWO**	**BMILFD**
Linear regression	0.524	0.728	0.688	4.60×10^{-2}	2.41×10^{-2}	2.89×10^{-2}	2.12×10^{-3}	5.82×10^{-4}	8.36×10^{-4}
Random forest regression	0.618	0.824	0.724	2.60×10^{-2}	4.68×10^{-3}	3.11×10^{-2}	6.76×10^{-4}	5.19×10^{-5}	2.86×10^{-4}
KNN regression	0.622	0.862	0.784	2.88×10^{-2}	9.08×10^{-3}	2.99×10^{-2}	8.28×10^{-4}	8.24×10^{-5}	1.48×10^{-5}
SVR-linear kernel	0.585	0.822	0.736	3.10×10^{-2}	6.81×10^{-3}	3.71×10^{-3}	9.60×10^{-4}	4.64×10^{-5}	1.38×10^{-5}
SVR-poly kernel	0.604	0.894	0.82	1.80×10^{-2}	2.20×10^{-3}	8.98×10^{-3}	3.24×10^{-4}	4.82×10^{-6}	8.06×10^{-5}
SVR-sigmoid kernel	0.652	0.927	0.838	9.92×10^{-3}	1.69×10^{-3}	9.55×10^{-3}	9.84×10^{-5}	2.84×10^{-6}	9.12×10^{-5}
SVR-RBF kernel	0.698	0.968	0.862	3.71×10^{-3}	3.49×10^{-4}	3.04×10^{-3}	1.38×10^{-5}	1.22×10^{-7}	9.27×10^{-6}

Abbreviations: BMILFD, Brownian movement and improved Levy's flight-based dragonfly; IPSWO, improved particle swarm whale optimizer; KNN, K-nearest neighbour; RBF, radial basis function; SVR, support vector regression.

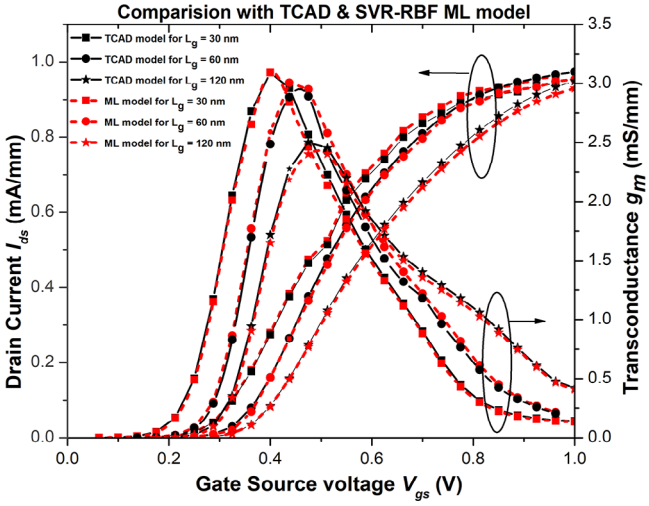

FIGURE 4 Comparison of technology computer-aided design (TCAD) plot and support vector regression-radial basis function (SVR-RBF) based prediction model with improved particle swarm whale optimizer optimization function plot for V_{gs}–I_{ds} and V_{gs}–g_m characteristics of the optimized double gate high electron mobility transistors structure for $V_{ds} = 0.5$ V.

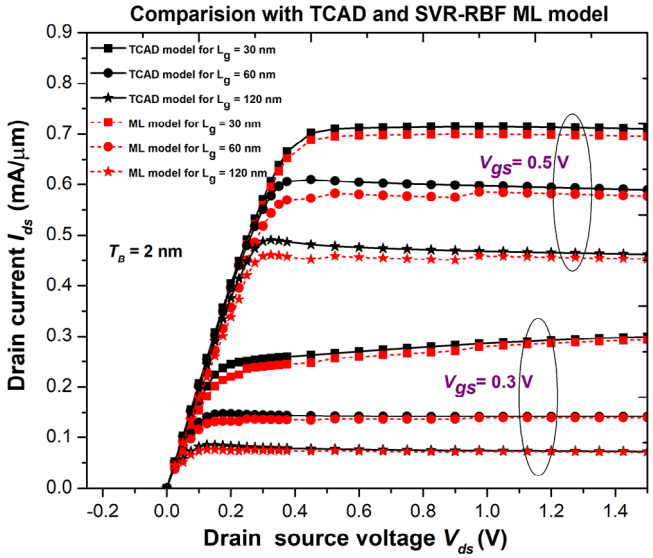

FIGURE 5 Comparison of the I_{ds}–V_{ds} characteristics for the optimized double gate high electron mobility transistors structure at $V_{gs} = 0.3$ and 0.5 V between the technology computer-aided design (TCAD) plot and the support vector regression-radial basis function (SVR-RBF) prediction model with improved particle swarm whale optimizer optimization.

Figure 4 show very minute variations in the ML-based predicted model compared with the TCAD model, and these variations do not affect the performance of the device.

Figure 5 shows the comparison of the TCAD plot and SVR-RBF-based prediction model with IPSWO optimization function plot for I_{ds}–V_{ds} characteristics when V_{gs} and V_{ds} from 0 to 1 V, L_g from 30 to 120 nm and T_b from 1 to 5 nm. From the results shown, it is evident that when T_b is 2 nm, L_g is 30 nm and V_{gs} is 0.5 V, we obtain a higher saturation current of 0.703 mA/μm. The dual gate structure enhances output characteristics due to higher 2DEG density from the top and bottom doping layers, forming a clear quantum well. In the thick InAs composite channel with dual gate structure, these results in superior mobility, 2DEG density, and conductivity, leading to a higher drive current compared to a thinner channel. When L_g is increased from 30 nm, V_{gs} is decreased from 0.5 V, and T_b is increased to 5 nm, the carrier mobility gets suppressed due to the increase in the gate capacitance and channel resistance. This results in the reduction of drain current. To optimize the value of I_{ds} for different variations in V_{gs}, V_{ds}, L_g and T_b, the SVR-RBF prediction model with IPSWO optimization function is

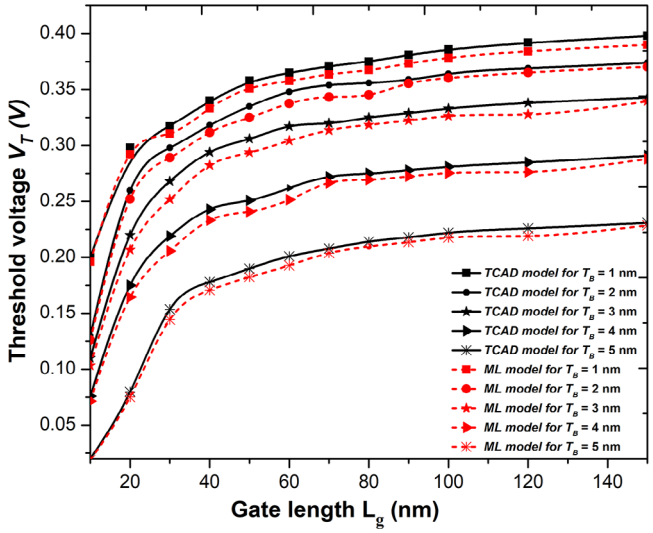

FIGURE 6 Comparison of threshold voltage for the optimized double gate high electron mobility transistors structure for different variations in T_b and L_g between the technology computer-aided design (TCAD) plot and the support vector regression-radial basis function prediction model with improved particle swarm whale optimizer optimization.

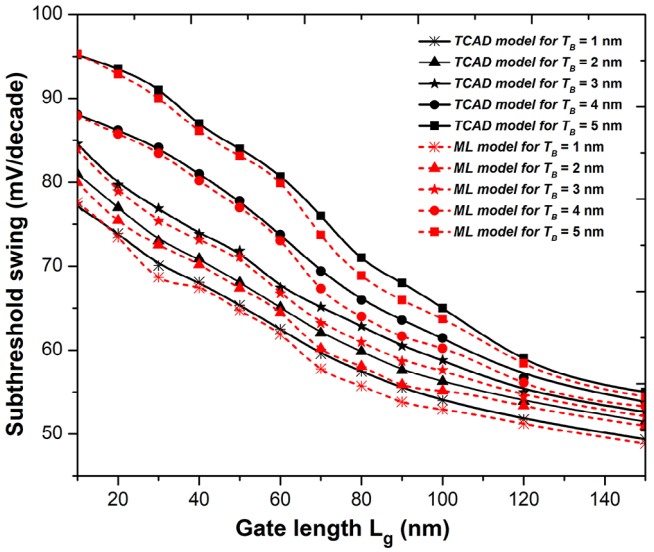

FIGURE 7 Comparison of subthreshold swing for the optimized double gate high electron mobility transistors structure for different variations in T_b and L_g between the technology computer-aided design (TCAD) plot and the support vector regression-radial basis function prediction model with improved particle swarm whale optimizer optimization.

trained with 1084 training data sets and results in minimization of MAE and MSE. The results from the graph in Figure 5 show the small variations in the ML-based predicted model provide a higher saturation current of 0.692 mA/μm, and these variations do not affect the performance of the device.

In enhancement-mode operation, the threshold voltage (V_T), which dictates the gate voltage needed to activate the device and initiate current flow, is influenced by factors such as channel width scaling, recessed gate design, and variations in the mole fraction and T_b. Figure 6 shows the Comparison of threshold voltage for the optimized DG-HEMTs structure when T_b is varied from 1 to 5 nm and L_g from 0 to 140 nm between the TCAD plot and the SVR-RBF prediction model with IPSWO optimization. From the results shown, it is evident that, for a variation in L_g from 0 to 140 nm, there is a maximum V_T variation from 0.18 to 0.389 V for $T_b = 1$ nm and this clearly says the V_T gets increased due to the increase in L_g. As T_b increases from 1 nm to 5 nm, the V_T decreases because the channel moves farther from the gate, leading to reduced electrostatic control due to the increased gate-to-channel distance. The enhancement in V_T is obtained with the variation in L_g from 0 to 140 nm primarily due to the rise in parasitic resistance. To optimize the value of V_T for different variations in L_g from 0 to 140 nm and T_b from 1 nm to 5 nm, the SVR- RBF prediction model with IPSWO optimization function is trained with 1084 training data sets and results in minimization of MAE and MSE. The graph results show that small variations in the ML-based predicted model provide a maximum threshold voltage of 0.38 V.

Device scaling is crucial for improving HEMT performance, but in ultra-short channel devices, it is essential to analyse short channel effects, like subthreshold slope as they significantly impact performance. Figure 7 shows the comparison of subthreshold swing for the optimized DG-HEMTs structure when T_b is varied from 1 to 5 nm and L_g from 10 to 140 nm between the TCAD plot and the SVR-RBF prediction model with IPSWO optimization. Subthreshold swing, determined from the slope of the I_{ds}–V_{gs} curve in the subthreshold region when drain to source voltage is 0.5 V, measures the change in gate voltage needed to achieve a tenfold increase in subthreshold drain current. A thinner T_B with 1 nm enhances electrostatic control, lowering subthreshold swing to 76.9 mV and reducing OFF-state leakage current, while a thicker T_B with 5 nm weakens control and increases subthreshold swing to 94.9 mV when L_g is fixed as a constant to 10 nm. From the results shown, it is evident that, for a variation in L_g from 10 to 140 nm, there is a reduction in subthreshold swing variation from 76.9 to 48.86 mV for $T_b = 1$ nm and this clearly says the subthreshold swing gets decreased due to the increase in L_g. To optimize the value of subthreshold swing for different variations in L_g from 0 to 140 nm and T_b from 1 to 5 nm, the SVR-RBF prediction model with IPSWO optimization function is trained with 1084 training data sets and results in minimization of MAE and MSE. The graph results show that small variations in the ML-based predicted model provide minute variations of subthreshold swing and provide subthreshold swing variation from 77.2 to 47.24 mV for a variation in L_g from 10 to 140 nm with $T_b = 1$ nm.

Drain induced barrier lowering (DIBL), a key parameter for assessing the electrostatic integrity of HEMTs, is defined as the change in threshold voltage in response to changes in voltage obtained from source to drain. Figure 8 shows the comparison of DIBL for the optimized DG-HEMTs structure when T_b is varied from 1 to 5 nm and L_g from 10 to 140 nm between the TCAD plot and the SVR-RBF prediction model with IPSWO optimization. As T_B thickens, L_g varies from 1 to 5 nm and the channel moves farther from the gate, electrostatic gate control

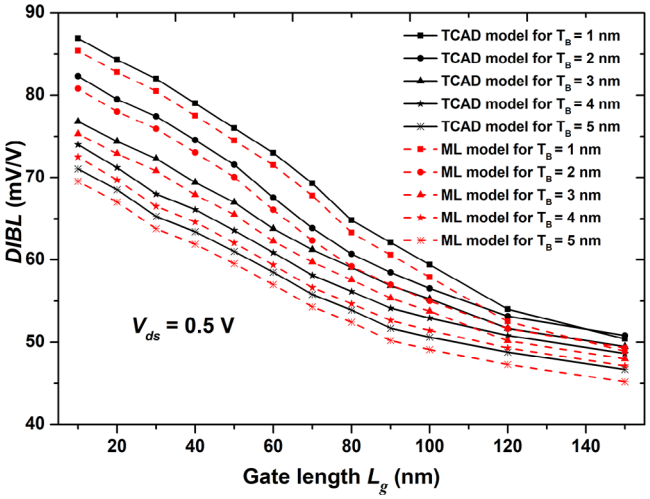

FIGURE 8 Comparison of drain induced barrier lowering (DIBL) for the optimized double gate high electron mobility transistors structure for different variations in T_b and L_g between the technology computer-aided design (TCAD) plot and the support vector regression-radial basis function prediction model with improved particle swarm whale optimizer optimization.

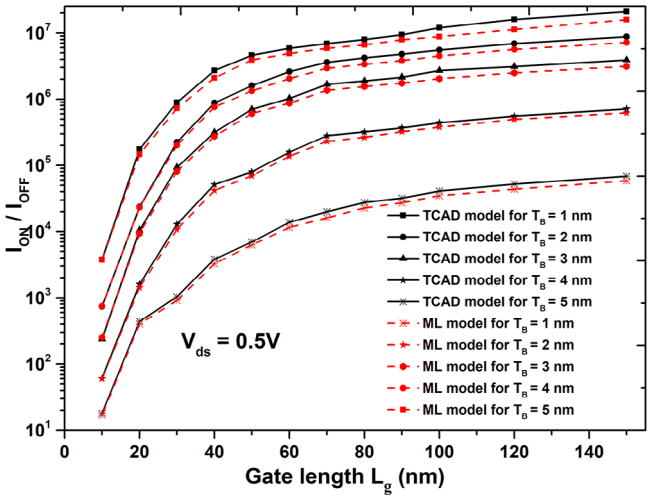

FIGURE 9 Comparison of I_{ON}/I_{OFF} for the optimized double gate high electron mobility transistors structure for different variations in T_b and L_g between the technology computer-aided design (TCAD) plot and the support vector regression-radial basis function prediction model with improved particle swarm whale optimizer optimization.

weakens, leading to DIBL degradation of 87.6 to 69.8 mV/V for $L_g = 10$ nm. The results indicate that as L_g increases from 10 to 140 nm, DIBL decreases from 87.6 mV/V to 52.64 mV for $T_b = 1$ nm, demonstrating that DIBL reduces with longer L_g. To optimize DIBL for L_g ranging from 0 to 140 nm and T_b from 1 to 5 nm, the SVR-RBF prediction model with IPSWO optimization was trained on 1084 datasets, effectively minimizing MAE and MSE. The graph indicates that minor fluctuations in the ML-based predicted model result in slight changes in DIBL, ranging from 86.2 mV/V to 50.24 mV as L_g varies from 10 to 140 nm with $T_b = 1$ nm.

The ON-state current refers to the drive current in saturation, while the OFF-state current encompasses the total leakage current, including subthreshold, gate, and junction leakage. The I_{ON}/I_{OFF} ratio is a key parameter for device performance, significantly impacting power consumption in static and low standby power applications. Figure 9 shows the comparison of I_{ON}/I_{OFF} for the optimized DG-HEMTs structure when T_b is varied from 1 to 5 nm and L_g from 10 to 140 nm between the TCAD plot and the SVR-RBF prediction model with IPSWO optimization. As L_g increases, the device's OFF-state leakage current decreases, raising the I_{ON}/I_{OFF} ratio, whereas an increase in T_B leads to higher I_{OFF} and I_{ON}, lowering the I_{ON}/I_{OFF} ratio. The results indicate that, for 10 nm long L_g with 1 nm thick T_B, we obtain I_{ON}/I_{OFF} as 180 and for 140 nm long L_g with 1 nm thick T_B, we obtain I_{ON}/I_{OFF} as 14.8×10^4. To optimize I_{ON}/I_{OFF} for L_g ranging from 0 to 140 nm and T_b from 1 to 5 nm, the SVR-RBF prediction model with IPSWO optimization was trained on 1084 datasets, effectively minimizing MAE and MSE. The graph indicates that minor fluctuations in the ML-based predicted model result in slight changes in I_{ON}/I_{OFF}, ranging from 182 to 8.4×10^4 as L_g varies from 10 to 140 nm with $T_b = 1$ nm.

Figure 10 shows the comparison of f_T and f_{max} for the optimized DG-HEMTs structure when T_b is varied from 1 to 5 nm, V_{ds} from 0 to 1 V and L_g from 10 to 140 nm between the TCAD plot and the SVR-RBF prediction model with IPSWO optimization. As L_g decreases to 30 nm, f_T and f_{max} improve due to reduced parasitic gate capacitance, better gate control, and shorter electron transit time, but below 30 nm, they saturate and decline due to transconductance reduction from severe short channel effects, as shown in our simulations. The simulation results from the graph indicate that an f_T and f_{max} of 816 and 948 GHz is obtained for L_g with 30 nm and T_b with 1 nm; an f_T and f_{max} of 368 and 678 GHz is obtained for L_g with 140 nm and T_b with 1 nm; an f_T and f_{max} of 661 and 862 GHz is obtained for L_g with 30 nm and T_b with 5 nm and an f_T and f_{max} of 272 and 613 GHz is obtained for L_g with 30 nm and T_b with 1 nm. To optimize f_T and f_{max} for L_g ranging from 0 to 140 nm and T_b from 1 to 5 nm, the SVR-RBF prediction model with IPSWO optimization was trained on 1084 datasets, effectively minimizing MAE and MSE. The graph indicates that minor fluctuations in the ML-based predicted model result in slight changes in f_T and f_{max} of 808 and 939 GHz is obtained for L_g with 30 nm and T_b with 1 nm. Therefore, the experimental verifications of InAlAs/InGaAs HEMTs recommend that the performance parameters predicted by the ML model resemble the same as that of TCAD simulation, and the computational time is very low and can be a compact modelling solution over traditional methods.

5 | CONCLUSIONS

In this work, we have discussed the number of iterations that have been done in TCAD simulations to analyse the characteristics of any new device by varying the device parameters. Initially, we have designed a dual gate HEMT structure using TCAD simulation and obtained the values of I_{ds}, g_m, V_T,

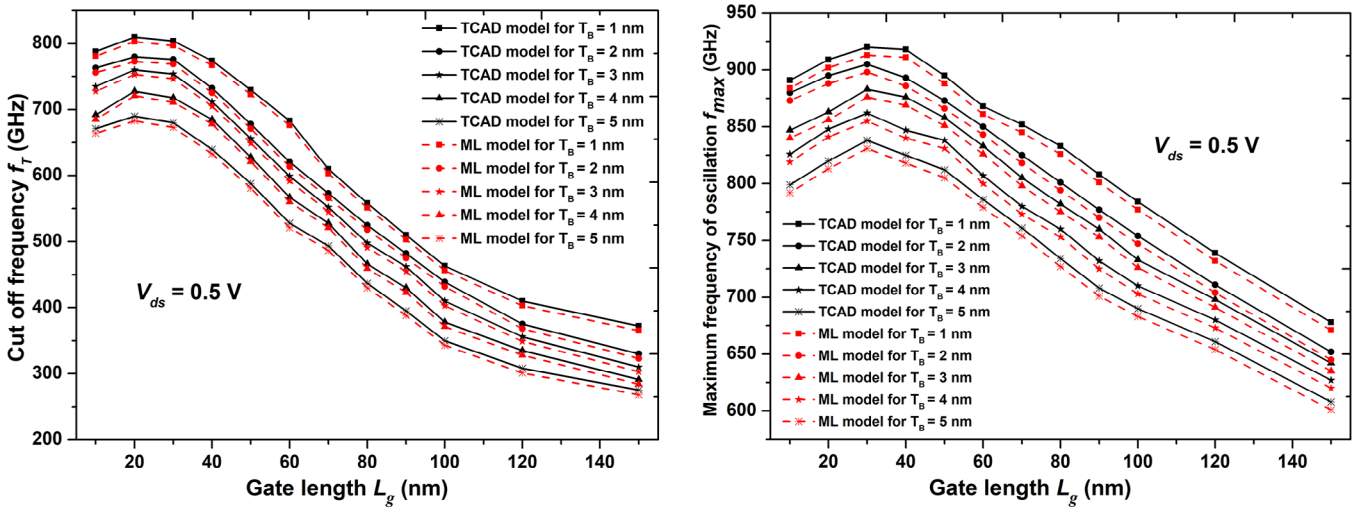

FIGURE 10 Comparison of f_T and f_{max} for the optimized double gate high electron mobility transistors structure for different variations in T_b and L_g between the technology computer-aided design (TCAD) plot and the support vector regression-radial basis function prediction model with improved particle swarm whale optimizer optimization.

I_{ON}/I_{OFF} ratio, DBIL, f_T and f_{max} values for an optimistic device variation of T_b, V_{ds}, V_{gs} and L_g. Due to the extensive iterations required for analysing the DC and RF characteristics of the device in the TCAD model, optimal designs and their features were derived using a machine learning framework. We employed improved particle swarm whale optimizer meta-heuristic algorithm, Brownian movement, and improved Levy's flight-based dragonfly optimizer meta-heuristic algorithm to obtain optimistic values of the dataset created. We employ four different ML models—linear regression, random forest regression, K-nearest neighbours regression, and support vector regression with linear, polynomial, sigmoid, and RBF kernel functions to predict the output characteristics. The results show that the trained SVR model with the RBF kernel, using IPSWO for optimization in the dataset, was chosen to predict the output parameters due to its lower MAE and MSE. The graph results show that small variations in the ML-based predicted model provide minute variations for predicting I_{ds}, g_m, V_T, I_{ON}/I_{OFF} ratio, DBIL, f_T and f_{max} values with low computational time when compared with TCAD models and this can be a compact modelling solution over traditional methods. In the future, the researcher may explore novel technologies [37–42] for the DC and RF systems,

AUTHOR CONTRIBUTIONS

Leeban Moses M: Conceptualization; formal analysis; investigation; methodology; supervision; validation; visualization; writing—original draft; writing—review and editing. **Saravana Kumar R**: Conceptualization; formal analysis; investigation; methodology; funding acquisition; project administration; visualization; writing—original draft; writing—review and editing. **Muhammad Faheem**: Methodology; supervision; writing—original draft; writing—review and editing. **Ramkumar K**: Project administration; software development; validation; writing—review and editing. **Shoukath Ali K**: Investigation; project administration; supervision; validation; visualization; writing—review and editing. **Arfat Ahmad Khan**: Formal analysis; investigation; resources management; supervision; validation; writing—review and editing.

ACKNOWLEDGEMENTS

The research work of Muhammad Faheem is supported by the Academy of Finland and the University of Vaasa, Finland.

CONFLICT OF INTEREST STATEMENT

The authors declare no conflicts of interest.

DATA AVAILABILITY STATEMENT

The authors confirm that the data supporting the findings of this study are available within the article.

ORCID

Leeban Moses M https://orcid.org/0000-0001-7468-1660
Muhammad Faheem https://orcid.org/0000-0003-4628-4486
Shoukath Ali K https://orcid.org/0000-0001-9256-373X
Arfat Ahmad Khan https://orcid.org/0000-0003-0918-8874

REFERENCES

1. Jeong, J., et al.: Influence of channel structure on the subthreshold swing of InGaAs HEMTs at cryogenic temperatures down to 4 K. IEEE Trans. Electron Devices 71(5), 3390–3395 (2024). https://doi.org/10.1109/TED.2024.3369576
2. Thome, F., Leuther, A.: Low-noise power-amplifier MMICs for the WR4.3 and WR3.4 bands in a 35-nm gate-length InGaAs mHEMT technology. IEEE Microwave Wireless Technol. Lett. 34(6), 749–752, (2024). https://doi.org/10.1109/LMWT.2024.3388320
3. Ajayan, J., Nirmal, D.: A Review of InP/InAlAs/InGaAs Based Transistors for High Frequency Applications. Academic Press, Cambridge, MA (2015). https://doi.org/10.1016/j.spmi.2015.06.048
4. Ajayan, J., Nirmal, D., Mohankumar, P., Arivazhagan, L.: Investigation of impact of passivation materials on the DC/RF performances of InP-HEMTs for terahertz sensing and imaging. Silicon 12(5), 1225–1230 (2020). https://doi.org/10.1007/s12633-019-00226-1

5. Jeong, J., et al.: Thermal studies of 3D stacked InGaAs HEMTs and mitigation strategy of self-heating effect using buried metal insertion. IEEE Trans. Electron. Devices 71, 4517–4523 (2024). https://doi.org/10.1109/TED.2024.3404419
6. Ajayan, J., Ravichandran, T., Prajoon, P., Pravin, J.C., Nirmal, D.: Investigation of breakdown performance in L_g = 20 nm novel asymmetric InP HEMTs for future high-speed high-power applications. J. Comput. Electron. 17(1), 265–272 (2018). https://doi.org/10.1007/s10825-017-1086-4
7. Ajayan, J., Nirmal, D.: 20 nm high performance enhancement mode InP HEMT with heavily doped S/D regions for future THz applications. Superlattices Microstruct. 100, 526–534 (2016). https://doi.org/10.1016/j.spmi.2016.10.011
8. Thakker, R.A., Sathe, C., Sachid, A.B., Shojaei Baghini, M., Ramgopal Rao, V., Patil, M.B.: A novel table-based approach for design of FinFET circuits. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. 28(7), 1061–1070 (2009). https://doi.org/10.1109/TCAD.2009.2017431
9. Peng, Y., Deng, W., Wu, W., Luo, Z., Huang, J.: Hybrid modelling routine for metal-oxide TFTs based on particle swarm optimisation and artificial neural network. Electron Lett 56(9), 453–456 (2020). https://doi.org/10.1049/el.2019.4001
10. Şenel, B., Şenel, F.A.: Novel neural network optimization approach for modeling scattering and noise parameters of microwave transistor. Int. J. Numer. Modell. Electron. Networks Devices Fields 35(1), e2930 (2022). https://doi.org/10.1002/jnm.2930
11. Deng, W., Zhang, W., Peng, Y., Wu, W., Huang, J., Luo, Z.: Compact modeling of metal-oxide TFTs based on artificial neural network and improved particle swarm optimization. J. Comput. Electron. 20(2), 1043–1049 (2021). https://doi.org/10.1007/s10825-020-01641-z
12. Kouhalvandi, L., Ceylan, O., Ozoguz, S.: Optimization techniques for analog and RF circuit designs: An overview. Analog Integr. Circuits Signal Process. 106(3), 511–524 (2021). https://doi.org/10.1007/s10470-020-01733-7
13. Zhang, L., Chan, M.: Artificial neural network design for compact modeling of generic transistors. J. Comput. Electron. 16(3), 825–832 (2017). https://doi.org/10.1007/s10825-017-0984-9
14. Cai, J., King, J., Chen, S., Wu, M., Su, J., Wang, J.: Machine learning-based broadband GaN HEMT behavioral model applied to class-J power amplifier design. Int. J. Microwave Wireless Technol. 13(5), 415–423 (2021). https://doi.org/10.1017/S1759078720001385
15. Burhan, M., et al.: A comprehensive survey on the cooperation of fog computing paradigm-based IoT applications: Layered architecture, real-time security issues, and solutions. IEEE Access 11, 73303–73329 (2023). https://doi.org/10.1109/ACCESS.2020.3005457
16. Wong, H.Y., et al.: TCAD-machine learning framework for device variation and operating temperature analysis with experimental demonstration. IEEE J. Electron Devices Soc. 8, 992–1000 (2020). https://doi.org/10.1109/JEDS.2020.3024669
17. Chen, Y., et al.: Evaluation of machine learning models for smart grid parameters: Performance analysis of ARIMA and Bi-LSTM. Sustainability 15(11), 8555 (2023). https://doi.org/10.1109/led.2019.2931839
18. Kawoosa, A.I., et al.: Using machine learning ensemble method for detection of energy theft in smart meters. IET Gener. Transm. Distrib. 17(21), 4794–4809 (2023). https://doi.org/10.1109/LED.2020.3045064
19. Wang, J., Kim, Y.H., Ryu, J., Jeong, C., Choi, W., Kim, D.: Artificial neural network-based compact modeling methodology for advanced transistors. IEEE Trans. Electron. Devices 68(3), 1318–1325 (2021). https://doi.org/10.1109/TED.2020.3048918
20. Wei, J.H., et al.: Advanced MOSFET model based on artificial neural network. In: Proceedings of the China Semiconductor Technology International Conference 2020, CSTIC 2020, pp. 1–3. IEEE, Piscataway, NJ (2020). https://doi.org/10.1109/CSTIC49141.2020.9282457
21. Zhang, L., Chan, M.: Artificial neural network design for compact modeling of generic transistors. J. Comput. Electron. 16(3), 825–832 (2017). https://doi.org/10.1007/s10825-017-0984-9
22. Hari, N., Chatterjee, S., Iyer, A.: Gallium nitride power device modeling using deep feed forward neural networks. In: Proceedings of the 2018 1st Workshop on Wide Bandgap Power Devices and Applications in Asia (WiPDA Asia), pp. 164–168. IEEE, Piscataway, NJ.
23. Radhakrishnan, S.K., Subramaniyan, B., Anandan, M., Nagarajan, M.: Comparative assessment of InGaAs sub-channel and InAs composite channel double gate (DG)-HEMT for sub-millimeter wave applications. AEU Int. J. Electron. Commun. 83, 462–469 (2018). https://doi.org/10.1016/j.aeue.2017.10.029
24. Saravana Kumar, R., Mohanbabu, A., Mohankumar, N., Godwin Raj, D.: Simulation of InGaAs subchannel DG-HEMTs for analogue/RF applications. Int. J. Electron. 105(3), 446–456 (2018). https://doi.org/10.1080/00207217.2017.1376354
25. Poornachandran, R., Mohankumar, N., Saravana Kumar, R., Sujatha, G.: Sheet-carrier density and *I–V* analysis of In0.7Ga0.3As/InAs/In0.7Ga0.3As/InAs/In0.7Ga0.3As dual channel double gate HEMT for THz applications. Int. J. Numer. Modell. Electron. Networks Devices Fields 32(5), e2625 (2019). https://doi.org/10.1002/jnm.2625
26. Poornachandran, R., Mohankumar, N., Saravanakumar, R., Sujatha, G.: Analysis of microwave noise in an enhancement-mode dual-quantum-well InAs HEMT. J. Comput. Electron. 18(4), 1280–1290 (2019). https://doi.org/10.1007/s10825-019-01365-9
27. Moses, M.L., Kaarthick, B.: Multiobjective cooperative swarm intelligence algorithm for uplink resource allocation in LTE-A networks. Trans. Emerging Telecommun. Technol. 30(12), e3748 (2019). https://doi.org/10.1002/ett.3748
28. Choudhury, S., Bhowmick, B., Baishnab, K. L.: A double-gate heteromaterial tunnel FET optimized using an evolutionary algorithm. J. Comput. Electron. 19(1), 277–282 (2020). https://doi.org/10.1007/s10825-019-01426-z
29. Singh, A., Sharma, A., Rajput, S., Bose, A., Hu, X.: An investigation on hybrid particle swarm optimization algorithms for parameter optimization of PV cells. Electronics 11(6), 909 (2022). https://doi.org/10.3390/electronics11060909
30. Sharma, A., Sharma, A., Averbukh, M., Jately, V., Azzopardi, B.: An effective method for parameter estimation of a solar cell. Electronics 10(3), 1–21 (2021). https://doi.org/10.3390/electronics10030312
31. Punyakum, V., Sethanan, K., Nitisiri, K., Pitakaso, R.: Hybrid particle swarm and whale optimization algorithm for multi-visit and multi-period dynamic workforce scheduling and routing problems. Mathematics 10(19), 3663 (2022). https://doi.org/10.3390/math10193663
32. Moses, L., Sambantham, P., Faheem, M., Ali K, S., Khan, A.A.: Joint delay and energy aware dragonfly optimization-based uplink resource allocation scheme for LTE-A networks in a cross-layer environment. J. Eng. 2024(2), (2024). https://doi.org/10.1049/tje2.12353
33. Chellappan, D., Rajaguru, H.: Detection of diabetes through microarray genes with enhancement of classifiers performance. Diagnostics 13(16), 2654 (2023). https://doi.org/10.3390/diagnostics13162654
34. Ganesh, N., Jain, P., Choudhury, A., Dutta, P., Kalita, K., Barsocchi, P.: Random forest regression-based machine learning model for accurate estimation of fluid flow in curved pipes. Processes 9(11), 2095 (2021). https://doi.org/10.3390/pr9112095
35. Keramat-Jahromi, M., Mohtasebi, S.S., Mousazadeh, H., Ghasemi-Varnamkhasti, M., Rahimi-Movassagh, M.: Real-time moisture ratio study of drying date fruit chips based on on-line image attributes using kNN and random forest regression methods. Measurement 172, 108899 (2021). https://doi.org/10.1016/j.measurement.2020.108899
36. Li, J., Sun, S., Xie, L., Zhu, C., He, D.: Multi-kernel support vector regression with improved moth-flame optimization algorithm for software effort estimation. Sci. Rep. 14(1), 16892 (2024). https://doi.org/10.1038/s41598-024-67197-1
37. Faheem, M., Raza, B., Bhutta, M.S., Madni, S.H.H.: A blockchain-based resilient andsecure framework for events monitoring and control indistributed renewable energy systems. IET Blockchain 1–15 (2024). https://doi.org/10.1049/blc2.12081
38. Faheem, M., Kuusniemi, H., Eltahawy, B., Bhutta, M.S., Raza, B.: A lightweight smartcontracts framework for blockchain-based

securecommunication in smart grid applications. IET Gener. Transm. Distrib. 18, 625–638 (2024). https://doi.org/10.1049/gtd2.13103
39. Burhan, M., et al.: A comprehensive survey on the cooperation of fog computing paradigm-based IoT applications: layered architecture, real-time security issues, and solutions. IEEE Access 11(7), 73303–73329 (2023). https://doi.org/10.1109/ACCESS.2023.3294479
40. Faheem, M., Muhammad, Ahmad A-K.: Multilayer cyberattacks identification and classification using machine learning in internet of blockchain (IoBC)-based energy networks. Data in Brief 54(8), 1–24 (2024). https://doi.org/10.1016/j.dib.2024.1100461
41. Faheem, M., Al-Khasawneh, M.A., Khan, A.A., Madni, S.H.H.: Cyberattack patterns in blockchain-based communication networks for distributed renewable energy systems: a study on big datasets. Data in Brief 53(6), 1–14 (2024). https://doi.org/10.1016/j.dib.2024.110212
42. Khan, A.A., et al.: D2PAM : epileptic seizures prediction using adversarial deep dualpatch attention mechanism. CAAI Trans. Intell. Technol. 8(3), 755–769 (2023). https://doi.org/10.1049/cit2.12261

How to cite this article: Moses M, L., Kumar R, S., Faheem, M., K, R., Ali K, S., Khan, A.A.: TCAD-enabled machine learning framework for DC and RF performance evaluation of InGaAs sub-channel DG-HEMTs. J. Eng. 2024, e70014 (2024). https://doi.org/10.1049/tje2.70014