Check for updates

IET Intelligent Transport Systems

WILEY

The Institution of Engineering and Technology

ORIGINAL RESEARCH OPEN ACCESS

Development of Deep Neural Network—Decision Tree Hybrid Control Strategy for Regenerative Braking in Electric Vehicles

Omer Ergun[1] | Erkin Dincmen[2] | Ilyas Istif[1]

[1]Mechanical Engineering, Yildiz Technical University, Istanbul, Türkiye | [2]Control and Automation Engineering, Yildiz Technical University, Istanbul, Türkiye

Correspondence: Erkin Dincmen (erkin.dincmen@yildiz.edu.tr)

Received: 13 September 2025 | **Revised:** 14 November 2025 | **Accepted:** 21 November 2025

ABSTRACT

Optimizing regenerative braking in dual-motor electric vehicles (EVs) is critical for extending driving range but presents a complex high-speed control problem. This study proposes a novel, real-time control strategy by training a hybrid deep neural network–decision tree (DNN–DT) model on an optimal dataset generated by offline dynamic programming (DP) considering seven key characteristic variables: road grade, friction coefficient, vehicle load distribution, velocity, braking rate, battery state of charge, and total braking torque. This hybrid methodology combines the high-accuracy, non-linear mapping of DNNs with the interpretability of DTs. The model was validated in a 14-DOF Simulink environment against two reference strategies (fixed-ratio and baseline) across four different scenarios (UDDS, NYCC, WLTP), including interpolation and extrapolation tests. Key experimental results show the hybrid model accurately tracks the DP-optimal torques (average $R^2 \approx 0.97$) and consistently outperforms the reference methods, achieving a 1.26% to 5.06% reduction in net SOC loss. This energy saving translates to a practical gain of 90–383 meters per cycle. Crucially, the model's average inference time of 2.3 ms confirms its computational efficiency and feasibility for real-time implementation on a standard vehicle control unit (VCU).

1 | Introduction

The global shift toward sustainable transportation and concerns about environmental pollution have increased interest in electric vehicles (EVs). This shift is driven by the need to meet increasingly stringent emission standards [1] and improve overall energy efficiency through methods such as advanced eco-driving strategies, as real-world energy consumption at the driver level is heavily influenced by individual driving behavior [2]. While EVs offer a sustainable transportation alternative, challenges such as limited range and reduced battery performance in cold weather hinder their widespread adoption [3, 4]. This necessitates the development of innovative energy management systems (EMS) to enhance energy efficiency. Within EMS, regenerative braking systems (RBS), which can potentially recover 39% to 50% of the energy lost during braking [5], play a crucial role in extending driving range. However, effective utilization of RBS presents a complex control problem due to constraints such as battery state of charge (SOC), charging currents, and motor capacity limitations [6–9]. In this context, the optimal control of RBS presents a solution-oriented approach to enhance EV performance and address range limitations, while also posing a considerable engineering challenge due to the complex and dynamic nature of vehicle systems. The variability of vehicle dynamics (e.g., speed, weight distribution, and braking intensity) necessitates continuous adaptation of an optimal balance between hydraulic and electric motor braking to maximize energy recovery. Traditional control methods often fail to adapt quickly and flexibly enough to such dynamic conditions. This challenge underscores the need for innovative control algorithms capable of responding rapidly and accurately to driver braking demands, adapting to dynamic conditions, and continuously optimizing

This is an open access article under the terms of the Creative Commons Attribution-NonCommercial-NoDerivs License, which permits use and distribution in any medium, provided the original work is properly cited, the use is non-commercial and no modifications or adaptations are made.

© 2025 The Author(s). *IET Intelligent Transport Systems* published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology.

vehicle systems. To maximize the efficiency and effectiveness of regenerative braking management, various approaches and control strategies have been investigated to determine the optimal torque distribution ratio (TDR) between hydraulic braking systems and electric motors. These strategies can be broadly categorized into rule-based (RB), optimization-based (OB), and learning-based (LB) methods [10].

Deterministic RB control strategies represent the simplest approach. However, such simple rules are often too basic for efficient application, as torque distribution in dual-motor modes can be managed more effectively by considering powertrain efficiency. For example, Ruan et al. [5] implemented a similar RB control with eco, sport, and safety modes, along with a "Motor Fault Tolerance" (MFT) fail-safe mechanism, though their study was limited by a single-motor EV architecture. Fuzzy logic control, another RB method, relies on predefined membership functions and rules based on human expertise, lacking systematic methodology and adaptability to dynamic conditions. While RB strategies are simple and easy to implement in a vehicle control unit (VCU) without prior knowledge of driving cycles, their main drawback is their inability to achieve optimal performance.

Optimization-based (OB) methods offer a more systematic approach, requiring less human expertise and focusing on maximizing an objective function (e.g., energy recovery or braking stability) using mathematical models and algorithms. This optimization can be local, calculating optimal control variables at each time step, or global, providing a benchmark for other strategies. Li et al. [11] used a variable-ratio optimization algorithm for a single-motor EV on downhill slopes, but their work was limited to a single driving cycle and longitudinal dynamics. Global optimization strategies, such as the NSGA-II based multi-objective strategy proposed by Zhou et al. [12] or the genetic algorithm approach by Chen et al. [13] for different loads and road grades, have shown improved energy recovery. Tang et al. [14] used an artificial bee colony (ABC) algorithm to determine the optimal braking force distribution considering various constraints and observed significant energy recovery improvements over the fixed-ratio distribution. However, strategies based on ideal torque curves from regulations may not always correspond to the most efficient operating points for all motors in terms of energy efficiency. Li et al. [15] proposed a game theory optimization-based strategy for EVs with in-wheel motors, considering stability, comfort, and energy recovery. Despite their superior performance compared to RB methods, OB approaches suffer from high computational costs and their optimal solutions are often limited to specific driving cycles or vehicle parameters, failing to provide universally optimal results.

To address these limitations, researchers have turned to model predictive control (MPC) and machine learning techniques to identify current traffic situations and driving patterns and determine the most appropriate TDR accordingly. For instance, Tang et al. [16] proposed an MPC-based coordinated control strategy for emergency braking in dual-motor EVs, achieving significant performance improvements in braking distance, time, regenerative energy efficiency (more than 22.76% increase), and slip ratio. Other MPC-based methods have been developed for four-wheel-drive in-wheel motor EVs [17] and adaptive predictive control methods combining estimation with dynamic programming (DP) [18]. However, these predictive approaches typically involve real-time cost function calculations, leading to heavy computational loads, and their performance heavily relies on prediction accuracy, potentially leading to suboptimal EMS performance in unforeseen driving conditions.

Recently, with advancements in artificial intelligence, learning-based methods have demonstrated the ability to achieve near-optimal solutions with high accuracy for complex and dynamic systems. These learning approaches include supervised, unsupervised, reinforcement, or hybrid methods. For example, Kiddee et al. [19] used an artificial neural network (ANN) to calculate braking forces, considering vehicle speed, driver input, and energy storage status. Yavaşoğlu et al. [20] employed ANNs for power management in a dual-motor EV with hybrid energy storage. He et al. [21] used an ANN-based controller to optimize power distribution between the electric motor and hydraulic brakes. Tian et al. [22] proposed an ANN-based controller for energy management in a hybrid electric city bus, using DP-derived optimal results for training. Following a similar hybrid approach, Nguyen et al. [23] proposed an adaptive network-based fuzzy inference system (ANFIS), trained with DP-derived data, for torque distribution optimization in a dual-motor EV. By calculating the total energy consumed from the state of charge (SOC) drop, their proposed strategy achieved a 4.65% improvement relative to the RB method. Prasanth et al. [24] utilized random forest and polynomial regression algorithms to maximize regenerative braking force in EVs. Zeng et al. [25] optimized speed profiles and energy recovery for trains using Q-learning, a reinforcement learning method. More recently, Yang et al. [26] developed a deep reinforcement learning-based regenerative braking strategy for an all-electric mining truck, demonstrating potential energy efficiency improvements of up to 18.15

Beyond these single-method learning-based approaches, recent studies have adopted ensemble and hybrid learning frameworks to combine the strengths of different algorithms. Several studies, for example, have focused on energy consumption estimation to reduce range anxiety, using hybrid convolutional neural network-bagged decision tree (CNN-BDT) models [27] or stacked generalization ensembles (ESG) to predict EV energy consumption, demonstrating superior performance compared to individual learners [28]. Beyond estimation, some comprehensive reviews specifically on AI-based energy management strategies confirm this hybrid trend. Kermansaravi et al. [29] highlight the value of combining complex "black-box" models (like DNNs) with simpler, interpretable models (like DTs) to improve safety. Similarly, a 2025 review by Prakash [30], on AI-based regenerative braking strategies, highlighted that hybrid models integrating neural networks, fuzzy logic, and reinforcement learning achieved higher recovery efficiency and braking stability than single-model approaches. These findings align with the motivation of the present study, where a hybrid DNN–DT framework is developed to exploit both the nonlinear feature extraction capability of deep networks and the interpretability of decision trees, aiming to achieve robust and generalizable control performance across various driving scenarios.

A review of the literature indicates an increasing importance of learning-based methods in regenerative braking and energy

17519578, 2025, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/itr2.70127 by Erkin Dinçmen - Yildiz Teknik University , Wiley Online Library on [21/12/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

management strategies. However, there is a relative scarcity of studies specifically for EVs, and existing approaches often consider a limited number of parameters, failing to comprehensively evaluate full vehicle dynamics and diverse scenario combinations. Although these ensemble methods are proposed as general frameworks, their specific application to the high-speed control of optimal torque distribution in dual-motor regenerative braking–using a DP-trained hybrid structure–remains a largely unaddressed deficiency. This paper proposes and evaluates a novel hybrid DNN–DT based control methodology, trained with optimal torque distribution ratios derived from DP across a comprehensive set of driving scenarios defined by seven characteristic variables. The goal is to maximize energy recovery by meeting the driver's braking demands while adhering to actuator and battery limits, as well as regulations. The performance of this hybrid model is benchmarked against both optimal DP results and two baseline torque distribution strategies.

2 | Vehicle Model and Electro-Hydraulic Braking System

The characterization of the electric vehicle (EV) model and its electro-hydraulic braking system provides the essential foundation for the subsequent control algorithm design. This section includes a detailed presentation of the EV's powertrain components, motor, and battery dynamics. Furthermore, the components and operational principles of the electro-hydraulic braking system are explained, thereby clarifying the physical model and structural constituents underpinning the optimization and control approaches discussed hereafter.

The vehicle is equipped with two electric motors, one on the front axle (EMf) and one on the rear axle (EMr), along with a hydraulic braking system (Hf, Hr), providing four actuators in total (assuming symmetry between left and right hydraulics) for deceleration, (Figure 1). In this configuration, the battery supplies the necessary energy to the EMs for driving and stores the energy recovered during regenerative braking. Thus, during braking, EMf and EMr operate in generator mode to charge the battery, while hydraulic brakes engage to provide additional braking torque when the demand exceeds the limits of the EMs or when regenerative braking is not feasible.

2.1 | Matlab Virtual Vehicle Composer Model

To analyze dynamic processes such as braking and regenerative energy recovery as realistically as possible, a 14-degree-of-freedom (DOF) vehicle model was developed using the MATLAB virtual vehicle composer (VVC) application. The VVC/Simulink environment was chosen for the integration and flexibility it provides. This flexibility was critical to the methodology of this study that enables the implementation and integration of the novel hybrid DNN-DT control architecture within the simulation loop.

The 14-DOF model comprises 6-DOF for the vehicle body and 2-DOF for each wheel (Figure 2). The model features a rigid two-axle vehicle body with six DOFs to calculate longitudinal, lateral, vertical, pitch, roll, and yaw motions. It accounts for body mass, inertia, aerodynamic drag, road grade, suspension between axles, and weight distribution due to external forces and moments. These detailed subsystems make this high-fidelity model a valid proxy for developing and validating the proposed control strategy.

2.1.1 | Vehicle Equations of Motion

To describe vehicle motion, rigid body dynamics, wind drag effects, inertial forces, and coordinate transformations are considered [31]. The translational motion in the body-fixed coordinate system is calculated by the following equation (where applied forces $\begin{bmatrix} F_x & F_y & F_z \end{bmatrix}^T$ are expressed in the body-fixed coordinate system, and the body mass m is assumed constant):

$$\overline{\mathbf{F}}_b = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = m\left(\dot{\overline{\mathbf{V}}}_b + \overline{\omega} \times \overline{\mathbf{V}}_b\right) \tag{1}$$

$$\overline{\mathbf{M}}_b = \begin{bmatrix} L \\ M \\ N \end{bmatrix} = \mathbf{I}\,\dot{\overline{\omega}} + \overline{\omega} \times \left(\mathbf{I}\,\overline{\omega}\right) \tag{2}$$

$$\mathbf{I} = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & I_{yy} & -I_{yz} \\ -I_{zx} & -I_{zy} & I_{zz} \end{bmatrix} \tag{3}$$

where $\overline{\mathbf{F}}_b$ [N] is the net force vector, $\overline{\mathbf{V}}_b$ [m/s] is the velocity vector, $\overline{\omega}$ [rad/s] is the angular velocity vector, $\overline{\mathbf{M}}_b$ [N · m] is the net torque vector, $\mathbf{I}$ [kg · m^2] is the inertia tensor of the vehicle body, and m [kg] is the total vehicle mass.

The relationship between the body-fixed angular velocity vector $\begin{bmatrix} p & q & r \end{bmatrix}^T$ and the rate of change of Euler angles $\begin{bmatrix} \dot{\phi} & \dot{\theta} & \dot{\psi} \end{bmatrix}^T$ is given by:

$$\begin{aligned} \begin{bmatrix} p \\ q \\ r \end{bmatrix} &= \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} \\ &\quad + \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \\ &\equiv \mathbf{J}^{-1} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \end{aligned} \tag{4}$$

By inverting the matrix J, the relationship to determine the Euler angle rates vector is obtained as:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \mathbf{J} \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi\,\tan\theta & \cos\phi\,\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \frac{\sin\phi}{\cos\theta} & \frac{\cos\phi}{\cos\theta} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{5}$$

The applied forces and moments are the sum of drag, gravitational, external, and suspension forces:

17519578, 2025, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/itr2.70127 by Erkin Dinçmen - Yildiz Teknik University , Wiley Online Library on [21/12/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

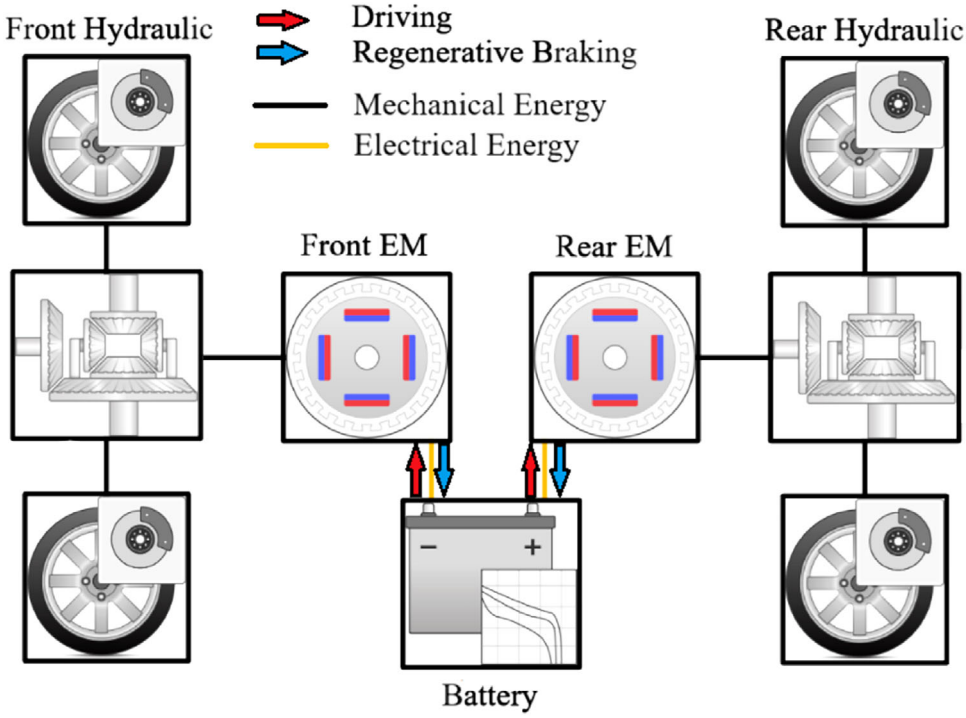


FIGURE 1 | Configuration of the electro-hydraulic braking system.

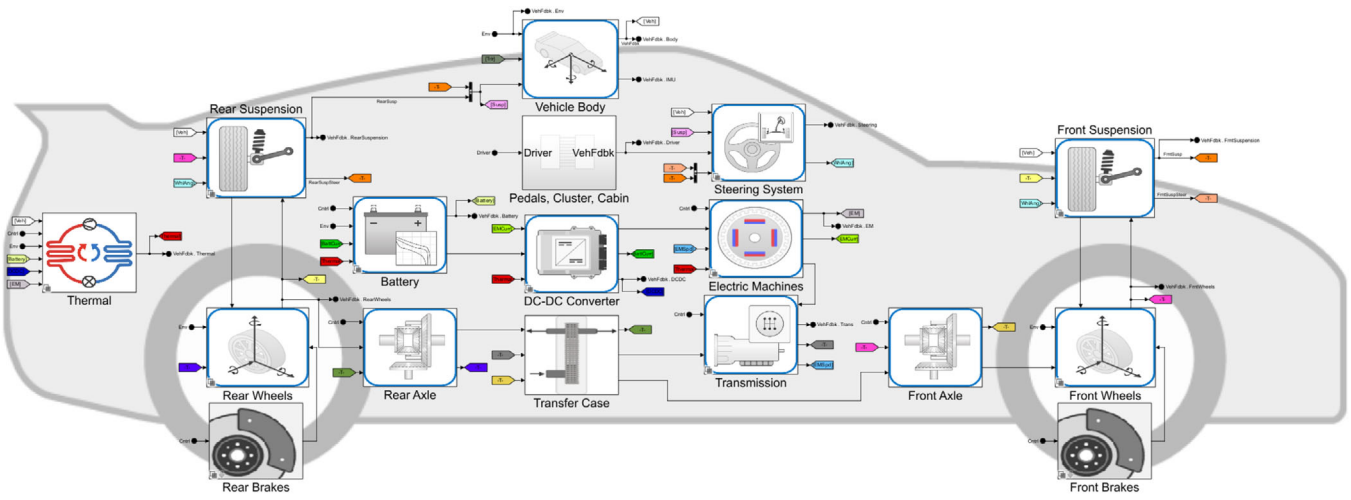


FIGURE 2 | 14-DOF vehicle model.

$$\overline{F}_b = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = \begin{bmatrix} F_{dx} \\ F_{dy} \\ F_{dz} \end{bmatrix} + \begin{bmatrix} F_{gx} \\ F_{gy} \\ F_{gz} \end{bmatrix} + \begin{bmatrix} F_{\text{extx}} \\ F_{\text{exty}} \\ F_{\text{extz}} \end{bmatrix} + \begin{bmatrix} F_{\text{FLx}} \\ F_{\text{FLy}} \\ F_{\text{FLz}} \end{bmatrix} + \begin{bmatrix} F_{\text{FRx}} \\ F_{\text{FRy}} \\ F_{\text{FRz}} \end{bmatrix} + \begin{bmatrix} F_{\text{RLx}} \\ F_{\text{RLy}} \\ F_{\text{RLz}} \end{bmatrix} + \begin{bmatrix} F_{\text{RRx}} \\ F_{\text{RRy}} \\ F_{\text{RRz}} \end{bmatrix} \tag{6}$$

$$\overline{M}_b = \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} M_{dx} \\ M_{dy} \\ M_{dz} \end{bmatrix} + \begin{bmatrix} M_{\text{extx}} \\ M_{\text{exty}} \\ M_{\text{extz}} \end{bmatrix} + \begin{bmatrix} M_{\text{FLx}} \\ M_{\text{FLy}} \\ M_{\text{FLz}} \end{bmatrix} + \begin{bmatrix} M_{\text{FRx}} \\ M_{\text{FRy}} \\ M_{\text{FRz}} \end{bmatrix} + \begin{bmatrix} M_{\text{RLx}} \\ M_{\text{RLy}} \\ M_{\text{RLz}} \end{bmatrix} + \begin{bmatrix} M_{\text{RRx}} \\ M_{\text{RRy}} \\ M_{\text{RRz}} \end{bmatrix} + \overline{\mathbf{M}}_F \tag{7}$$

17519578, 2025, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/itr2.70127 by Erkin Dinçmen - Yildiz Teknik University , Wiley Online Library on [21/12/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

where $\overline{F}_b$ is the net force and $\overline{M}_b$ is the net moment vector acting on the vehicle's center of gravity (CG). Subscripts x, y, z denote the vehicle's body axes, d and g denote the forces/moments due to aerodynamic drag and gravity, respectively, while ext represents external forces/moments. Subscripts FL, FR, RL, RR refer to the suspension forces and moments acting on the four respective wheels (front-left, front-right, rear-left, rear-right) and $\overline{M}_F$ represents the external moment acting on the vehicle body, which arises from the gravitational force acting at CG.

2.2 | Tire Model

The dynamics of the wheels, which provide the vehicle's contact with the road, directly determine the generation of braking forces, and thus the vehicle's total braking torque demand. In this study, magic formula approach is used to reflect the dynamic behavior of the tire by simultaneously considering longitudinal and lateral slip conditions. Accordingly, the longitudinal and lateral forces generated at the wheel are given by:

$$F_x = (D_x \sin[C_x \arctan\{B_x\kappa_x - E_x(B_x\kappa_x - \arctan(B_x\kappa_x))\}] + S_{Vx})G_{x\alpha} \tag{8}$$

$$F_y = G_{y\kappa}F_{yp} + S_{Vy\kappa} \tag{9}$$

where F_x and F_y are the longitudinal and lateral tire forces, respectively. All coefficients (D_x, C_x, B_x, E_x, etc.) are the standard empirical parameters defined by the Magic Formula approach established in [32].

2.3 | Driver Model

The driver model discussed in this section is based on the optimal single-point preview (look-ahead) control model developed by C. C. MacAdam [33–35]. The model represents the driver's control behavior during the tracking of a predefined path or speed profile. The model considers the vehicle dynamics as linear (single-track/bicycle), minimizes the previewed error signal at a single point T^* seconds ahead, and also accounts for the driver lag.

In the model, the longitudinal motion of the vehicle is expressed by the state equations in matrix form as follows:

$$\dot{x} = Fx + \mathbf{g}\bar{u} \tag{10}$$

Here:

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad F = \begin{bmatrix} 0 & 1 \\ \frac{F_r}{m} & 0 \end{bmatrix}, \quad \mathrm{g} = \begin{bmatrix} 0 \\ \frac{K_{pt}}{m} \end{bmatrix} \tag{11}$$

$$\bar{u} = u - \frac{m^2}{K_{pt}} g \sin(\gamma) \tag{12}$$

where x is the vehicle state vector, x_1 is the longitudinal velocity and x_2 is the derivative of x_1. F is the system matrix, and $\mathbf{g}$ is the control coefficient vector. The term m is the vehicle mass, K_{pt} is a constant for the tractive force and brake limit, γ is the grade angle of the road, and g is the gravitational acceleration constant.

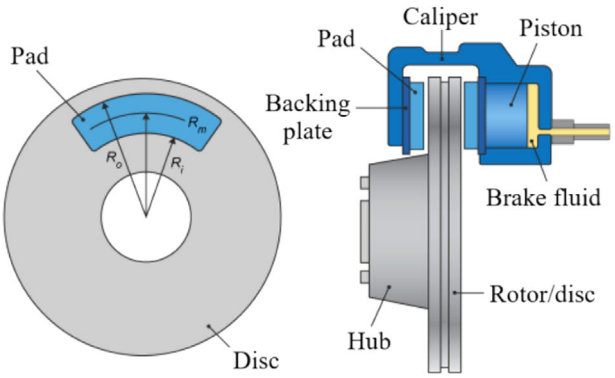


FIGURE 3 | Structure of disc braking.

The rolling resistance (F_r) is calculated with the following equation:

$$F_r = -\left[\tanh(x_1)\left(\frac{a_r}{x_1} + c_r x_1\right) + b_r\right] \tag{13}$$

where a_r, c_r, and b_r are the coefficients of static, linear, and aerodynamic rolling resistance, respectively.

2.4 | Hydraulic Braking Model

The electric vehicle model uses a disc hydraulic brake system on each wheel. The hydraulic braking system is activated during situations requiring high torque, such as emergency braking where the power of the electric motors is insufficient, at very low speeds when regenerative braking is not suitable, or when the battery is nearly full. The general structure of the disc braking is shown in (Figure 3).

The generated braking torque is calculated as follows:

$$\mathcal{T} = \begin{cases} \dfrac{\mu P \pi B_a^2 R_o N_{\text{pads}}}{4}, & \text{if } N \neq 0, \\ \dfrac{\mu_{\text{static}} P \pi B_a^2 R_o N_{\text{pads}}}{4}, & \text{if } N = 0, \end{cases} \tag{14}$$

$$R_o = \frac{R_d + R_i}{2}. \tag{15}$$

where $\mathcal{T}$ is the braking torque generated on one disc (Nm), P is the applied brake pressure (Pa), N is the wheel speed (rad/s), N_{pads} is the number of brake pads used in the disc brake, μ_{static} is the static friction coefficient between pad and disc, μ is the kinetic friction coefficient between pad and disc, B_a is the brake cylinder diameter, R_o is the mean radius of the pad force acting on the brake disc, R_d is the outer radius of the brake pad, and R_i is the inner radius of the brake pad.

The total braking torque required by the vehicle during braking is calculated as the sum of individual torque components from all discs:

$$\mathcal{T}_{\text{brake}}^{\text{total}} = \sum_{j=1}^{4} \left(\mu \cdot \pi \cdot B_a^2 \cdot 2 \cdot R_o \cdot \frac{N_{\text{pads}}}{4}\right) \cdot P_j \tag{16}$$

17519578, 2025, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/itr2.70127 by Erkin Dinçmen - Yildiz Teknik University , Wiley Online Library on [21/12/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

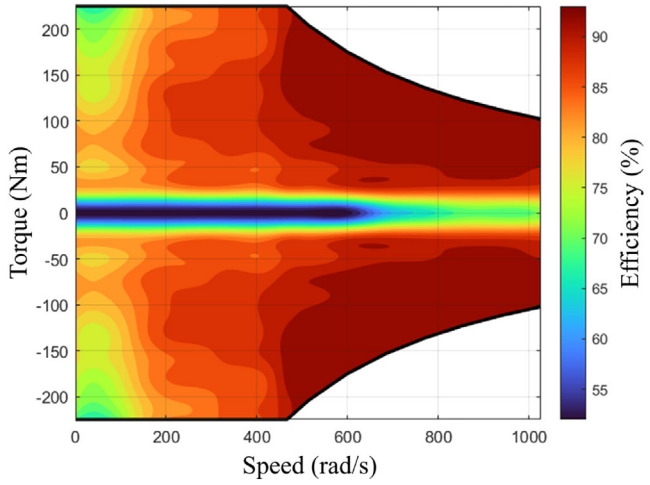


FIGURE 4 | Motor efficiency map.

$$P_j = \begin{cases} y_{dec} \cdot \text{Front factor}\,(1) \cdot K, & \text{front}\,(j = 1, 2) \\ y_{dec} \cdot \text{Rear factor}\,(0.6) \cdot K, & \text{rear}\,(j = 3, 4) \end{cases} \tag{17}$$

where y_{dec} is the normalized deceleration command generated by the driver model, K is the pressure factor (8,000,000 Pa), P_j is the brake pressure transmitted to each disc, and $\mathcal{T}_{\text{brake}}^{\text{total}}$ is the total demanded braking torque.

2.5 | Motor Model

The model includes two electric motors (EMs) with identical characteristics on the front and rear axles. The efficiency map of the motor, dependent on its torque and angular speed, is shown in Figure 4.

The motors have a maximum torque of ±225 Nm and a maximum power of 105 kW. The positive torque region represents conditions where the motor consumes energy, while the negative region indicates conditions where the vehicle's mechanical energy is converted into electrical energy. The power generated by the front and rear axle motors to charge the battery during regenerative braking is defined as:

$$P_{\text{EM}} = \mathcal{T}_{\text{EM}} \cdot \omega \cdot \eta \tag{18}$$

where P_{EM} is the electric motor power (W), $\mathcal{T}_{\text{EM}}$ is the electric motor torque (Nm), ω is the electric motor angular speed (rad/s), and η is the electric motor efficiency.

2.6 | Battery Model

A lithium-ion battery model is used to characterize battery performance at various operating points. The battery consists of 96 cells in series and 94 cells in parallel, with each cell having a mass of 0.0495 kg.

The open-circuit voltage and internal resistance, which vary with SOC and battery temperature and are used to determine the battery output voltage, are shown graphically in Figure 5.

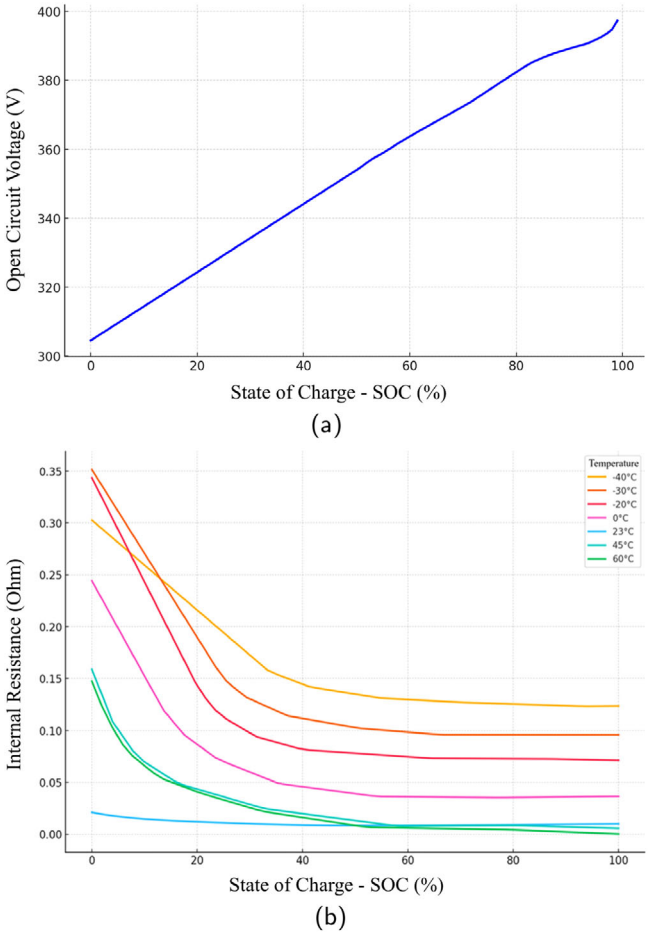


FIGURE 5 | Battery characteristics: (a) Open-circuit voltage vs. SOC, (b) Internal resistance vs. SOC for different temperatures.

$$E_m = f(\text{SOC}) \tag{19}$$

$$R_{\text{int}} = f(T, \text{SOC}) \tag{20}$$

where E_m (V) represents the battery's open-circuit voltage, and R_{int} (Ω) represents its internal resistance.

The SOC dynamics of the battery are given by:

$$\text{SOC}(t) = \text{SOC}_0 - \frac{1}{C_{\text{bat}}} \int_0^t I_{\text{bat}}(\tau)\, d\tau \tag{21}$$

where SOC_0 is the initial state of charge, C_{bat} is the battery capacity, and I_{bat} is the current flowing through the battery.

The discharge power dynamics of the battery system are given by:

$$P_{\text{bat}} = V_{\text{bat}} I_{\text{bat}} \tag{22}$$

$$V_T = E_m - I_{\text{bat}} R_{\text{int}} \tag{23}$$

where P_{bat} is the battery power, and V_{bat} (or V_T) represents the battery's terminal voltage.

17519578, 2025, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/itr2.70127 by Erkin Dinçmen - Yildiz Teknik University , Wiley Online Library on [21/12/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Combining Equations (22) and (23) and solving the resulting quadratic equation for P_{bat} yields the battery current as:

$$I_{\text{bat}} = \frac{E_m - \sqrt{E_m^2 - 4P_{\text{bat}}R_{\text{int}}}}{2R_{\text{int}}} \tag{24}$$

Finally, the power stored in the battery during regenerative braking is expressed as:

$$P_{\text{Stored}} = P_{\text{Bat_in}} - P_{\text{Loss}} \tag{25}$$

where P_{Loss} is the battery power loss, calculated as:

$$P_{\text{Loss}} = N_p N_s I_{\text{bat}}^2 R_{\text{int}} \tag{26}$$

where N_p and N_s represent the number of parallel and series cells, respectively.

3 | Optimization Algorithm

The distribution ratio of the braking force between the front and rear axles, along with the ratio of the regenerative braking forces from both motors to the total braking force, plays a critical role in determining both braking stability and energy recovery efficiency in dual-motor electric vehicles (DM–EVs) equipped with one motor on each axle.

The optimization algorithm developed here focuses on allocating total braking torque between the front and rear axles to maximize energy recovery via regenerative braking during EV braking events. This development is based on the full EV model presented in Section 2. The fundamental framework of this algorithm, including regulations-compliant constraints, vehicle stability criteria, and electric motor limits, is initially expressed mathematically. Following this, the architecture of the dynamic programming (DP) algorithm, along with its state space definition, cost function, and solution steps, is explained comprehensively.

The final part details how the derived optimal results are transformed into a dataset that serves as input to train a hybrid controller based on a deep neural network (DNN) and a decision tree (DT) in the subsequent section.

3.1 | UN/ECE Regulations

To ensure vehicle braking safety, the UN/ECE has established specific regulations. According to the ECE R13–H braking regulations for passenger cars, the braking force distribution of the front and rear axles must meet the following requirements [36]:

1. For all loading conditions of the vehicle, when the braking rate z is in the range $0.15 \sim 0.8$, the adhesion utilization curve of the rear axle must not be greater than that of the front axle [regulation 3.1.(A)];

$$k_f \geq k_r \tag{27}$$

2. When the adhesion coefficient k is in the range 0.2–0.8, the braking rate must satisfy the following condition [regulation 3.1.(B)]:

$$z \geq 0.1 + 0.7(k - 0.2) \tag{28}$$

Substituting the values $k = 0.2$ and $k = 0.8$ into inequality (28), respectively, the range of change for z is found to be $0.1 \leq z \leq 0.52$. The adhesion coefficients of the front and rear wheels within this z–range satisfy the following conditions:

$$k_f \leq \frac{z + 0.04}{0.7} \tag{29}$$

$$k_r \leq \frac{z + 0.04}{0.7} \tag{30}$$

where k_f and k_r are theoretical adhesion coefficients at the front and rear axles, respectively, that is, the coefficients indicating how effectively the wheels utilize road grip during braking. Considering the effect of gravity on a slope, the braking rate of the vehicle on a sloped road, and the front and rear adhesion coefficients are defined as follows, respectively [11]:

$$z_t = z\cos\alpha + \sin\alpha \tag{31}$$

$$k_{\text{f}} = \frac{F_{x,\text{f}}}{F_{z,\text{f}}} = \frac{\gamma z_t L}{l_r\cos(\alpha) + z_t h_{cg}} \tag{32}$$

$$k_{\text{r}} = \frac{F_{x,\text{r}}}{F_{z,\text{r}}} = \frac{(1-\gamma) z_t L}{l_f\cos(\alpha) - z_t h_{cg}} \tag{33}$$

where $F_{x,f}$ and $F_{x,r}$ are the forces applied by the brakes to the front and rear axles, respectively, under normal braking conditions; $F_{z,f}$ and $F_{z,r}$ are the normal forces acting from the road surface to the front and rear axles, respectively, during braking; l_f is the distance between the front axle and the center of gravity (CG); l_r is the distance between the rear axle and the CG; L is the wheelbase; α is the road grade; h_{cg} is the height of the CG; and γ is the braking force distribution ratio.

By substituting Equations (32) and (33) into (27), (29), and (30), the following inequalities related to the braking force distribution ratio are obtained:

$$\gamma \geq \frac{l_r\cos\alpha + z_t h_{cg}}{L\cos\alpha} \quad \text{for } 0.15 \leq z_t \leq 0.8 \tag{34}$$

$$\gamma \leq \frac{(z_t + 0.04)(l_r\cos\alpha + z_t h_{cg})}{0.7 z_t L} \quad \text{for } 0.1 \leq z_t \leq 0.52 \tag{35}$$

$$\gamma \geq 1 - \frac{(z_t + 0.04)(l_f\cos\alpha - z_t h_{cg})}{0.7 z_t L} \quad \text{for } 0.1 \leq z_t \leq 0.52 \tag{36}$$

Thus, under ECE regulation constraints, the adhesion coefficient for the front wheels is always higher than that for the rear wheels, ensuring braking stability.

Furthermore, during the braking process, it is desired that the utilization coefficients of the wheels are lower than the road adhesion coefficient (μ). Considering the inequality in (27), since

17519578, 2025, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/itr2.70127 by Erkin Dinçmen - Yildiz Teknik University , Wiley Online Library on [21/12/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 1 | Initial vehicle parameters.

Description	Value	Unit
Vehicle mass (m)	2000	kg
Distance from CG to front axle (l_f)	1.09	m
Distance from CG to rear axle (l_r)	1.7	m
Height of CG (h_{cg})	0.627	m
Wheel radius (r)	0.327	m
Transmission ratio (i_t)	3.32	—

the utilization coefficient of the front wheels is ensured to be greater than that of the rear wheels, only the condition that the utilization coefficient of the front wheels must be lower than the road adhesion coefficient is considered, which is expressed as:

$$\gamma < \frac{\mu(z_t h_{cg} + l_r \cos\alpha)}{z_t L} \tag{37}$$

3.2 | Vehicle Load Distribution

According to the ISO 10392:2011 standard, changes in vehicle load affect structural parameters of the vehicle such as h_{CG}, l_f, and l_r [37]. These structural parameters influence the distribution of braking forces and overall braking performance.

For the different loading scenarios simulated, the total vehicle mass (m_v) and rear axle load (m_r) are varied. The horizontal distance between the front axle center and the CG, l_f, is then recalculated for each specific scenario using:

$$l_f = \frac{m_r}{m_v} \times L \tag{38}$$

The distance to the rear axle is subsequently updated as $l_r = L - l_f$. As a simplification, the height of the vehicle's CG (h_{cg}) is assumed to be constant based on the assumption that additional loads are located near the CG. The parameters presented in Table 1 represent the initial (baseline) state of the vehicle for the first simulation.

3.3 | Dynamic Programming

A dynamic programming (DP) algorithm is developed for optimizing the distribution of total braking torque between the front and rear axles, with the objective to maximize regenerative braking energy during braking. For this purpose, the front axle electric motor torque space is discretized into a set of torque levels (Figure 6), where each circle corresponds to a specific level. The DP algorithm subsequently identifies the optimal sequence of state transitions among these torque levels to ensure maximum energy recovery.

At each time step of the braking scenario, three constraints are considered to determine possible state transitions:

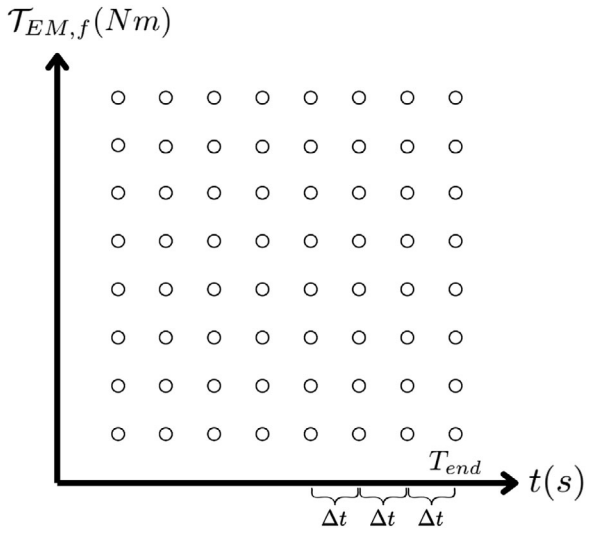


FIGURE 6 | Front axle electric motor torque space.

Constraint 1: The maximum producible electric motor (EM) torque is determined at each time step based on the limits derived from the EM efficiency map (Figure 4).

Constraint 2: Based on the UN/ECE regulations, the following limits are derived by Equations (34)–(37) to determine the permissible torque values for the front axle:

$$\mathcal{T}_{\mathrm{EM,f,min}} = \gamma_{\min} \mathcal{T}^{\mathrm{total}}_{\mathrm{brake}} i_t \tag{39}$$

$$\mathcal{T}_{\mathrm{EM,f,max}} = \gamma_{\max} \mathcal{T}^{\mathrm{total}}_{\mathrm{brake}} i_t \tag{40}$$

where $\mathcal{T}_{\mathrm{EM,f,min}}$ and $\mathcal{T}_{\mathrm{EM,f,max}}$ are the minimum and maximum permissible torque values for the front axle electric motor, respectively; i_t is the transmission ratio; and $\mathcal{T}^{\mathrm{total}}_{\mathrm{brake}}$ is the total demanded braking torque, as defined in the hydraulic braking model.

Constraint 3: Due to the electric motor's finite response time (slew rate), transitions between the torque levels shown in Figure 6, from a torque level at time t to a torque level at time $t + \Delta t$, are constrained by a rise time (Δt) limit of 0.2 s, which was adopted for this study. In this model, this is simplified: it can be limited to a maximum of 3 levels up or down within this time. In practice, the actual value of this constraint is determined according to the characteristics of the electric motor used.

By considering these three constraints, the set of all feasible torque values that can be produced by the EM on the front axle at any time step is determined. The DP algorithm selects the optimal torque from this feasible set. If the torques of the EMs are insufficient during the distribution of total braking torques to the front and rear axles, the remaining braking torques are provided by the hydraulic braking systems.

To formally define the optimization problem, the braking scenario is discretized into time steps t, where $t \in \{0, \Delta t, \ldots, T_{\mathrm{end}} - \Delta t\}$.

17519578, 2025, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/itr2.70127 by Erkin Dinçmen - Yildiz Teknik University , Wiley Online Library on [21/12/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

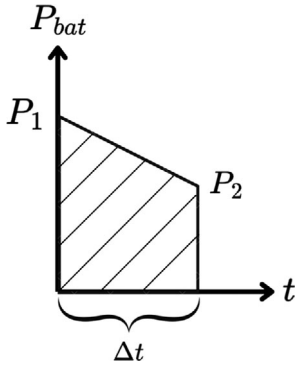


FIGURE 7 | Change in battery power from P_1 to P_2 in time Δt.

- State (i): The state at time t is the discrete torque level i of the front axle EM. $i \in \mathcal{T}_{\text{levels}}$, where $\mathcal{T}_{\text{levels}}$ is the set of all possible torque levels (Figure 6).
- Decision (j): The decision at time t is the selection of the torque level j for the next time step, $t + \Delta t$.
- Constraints: The decision j must satisfy all three constraints. The set of all valid decisions j starting from state i is denoted $\mathcal{U}(i)$.
- Stage Reward ($J_t^{t+\Delta t}(i, j)$): The reward (energy recovered, ΔE) for transitioning from state i at time t to state j at time $t + \Delta t$. This reward is calculated using the 7-step process described below and the trapezoidal rule (Equation 41).
- Value Function ($J_t^*(i)$): The optimal value function (or "reward-to-go"), representing the maximum possible energy to be recovered from time step t until the final time T_{end}, given the system is in state i.

The 7-step process to calculate the stage reward $J_t^{t+\Delta t}(i, j)$ for each feasible transition is as follows:

1. The efficiency of the front electric motor (at state i and j) is calculated by interpolation from the motor map (Figure 4).
2. The power of the front electric motor is calculated (Equation 18).
3. These operations are repeated for the rear axle electric motor.
4. The total power of both electric motors is calculated.
5. The battery's open-circuit voltage (E_m) and internal resistance (R_{int}) are calculated (Figure 5).
6. The battery's power and charging current are calculated (Equations 22–24).
7. The stage reward $J_t^{t+\Delta t}(i, j) = \Delta E$ (the incremental change in battery energy) is calculated using the battery charging powers at t and $t + \Delta t$.

As shown in Figure 7, when the battery's charging power changes from P_1 (at t) to P_2 (at $t + \Delta t$) over the duration Δt, the recovered energy is calculated using the trapezoidal rule:

$$\Delta E = J_t^{t+\Delta t}(i, j) = \frac{(P_1 + P_2)\Delta t}{2} \tag{41}$$

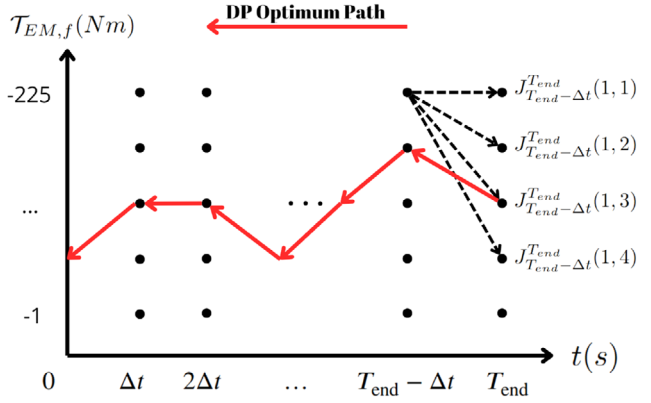


FIGURE 8 | Schematic optimal torque transitions for the front axle electric motor.

The DP algorithm solves this optimization problem by iterating backward in time, from $t = T_{\text{end}} - \Delta t$ down to $t = 0$, using the Bellman equation, which defines the optimal value function recursively:

$$J_t^*(i) = \max_{j \in \mathcal{U}(i)} \left[J_t^{t+\Delta t}(i, j) + J_{t+\Delta t}^*(j) \right] \tag{42}$$

The iteration is initialized at the final time step with the terminal reward: $J_{T_{\text{end}}}^*(i) = 0$ for all $i \in \mathcal{T}_{\text{levels}}$.

At each backward step t, the algorithm computes and stores the optimal decision $j^*(i)$ (the optimal next torque level) that maximizes the right-hand side of Equation (42) for every possible state i.

Once the backward iteration is complete, an optimal torque trajectory (the sequence of optimal states) is found by tracing the stored optimal path forward in time from the initial state ($t = 0$). This trajectory represents the theoretical maximum energy recovery for the given braking profile. An illustrative example of the optimal front axle EM torque transitions during a braking scenario is presented in Figure 8.

The overall operational flow of the algorithm, solving the Bellman equation (Equation 42) and reconstructing the path, is summarized in Algorithm 1.

The optimal solutions obtained with dynamic programming (EMf, EMr) and their corresponding seven input variables will be used as training data for the deep neural network–decision tree hybrid model to be developed in the following section. This dataset is designed to include both the outputs of the optimization and the physical constraints of the system (e.g., regulation limits for γ, motor torque capacities, etc.). The input-output pairs to be used in training the hybrid model consist of various scenarios generated by DP's multi-variable optimization process.

4 | DNN–Decision Tree Hybrid Control System Design

A hybrid control strategy is developed to improve regenerative braking performance in electric vehicles (EVs). This approach

17519578, 2025, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/itr2.70127 by Erkin Dinçmen - Yildiz Teknik University , Wiley Online Library on [21/12/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

ALGORITHM 1 | Dynamic Programming for Optimal Energy Recovery.

Input: State space $\mathcal{T}_{\text{levels}}$, Time horizon T_{end}, Time step Δt
Output: Optimal torque trajectory $Path$

▷ 1. Initialization

for all states $i \in \mathcal{T}_{\text{levels}}$ **do**
  $J^*_{T_{\text{end}}}(i) \leftarrow 0$ *% Set terminal reward to zero*
end for

▷ 2. Backward Iteration

for $t \leftarrow (T_{\text{end}} - \Delta t)$ **down to** 0 **do**
  for all states $i \in \mathcal{T}_{\text{levels}}$ **do**
    $max_reward \leftarrow -\infty$
    for all decisions $j \in \mathcal{U}(i)$ **do** % $\mathcal{U}(i)$ = *Feasible levels (Constraints 1, 2, 3)*
      $stage_reward \leftarrow$ Stage_Reward(t, i, j) *% Calculate reward (7-step process, Equation 41)*
      $reward_to_go \leftarrow J^*_{t+\Delta t}(j)$ *% Retrieve stored value (Bellman recursion)*
      $total_reward \leftarrow stage_reward + reward_to_go$
      if $total_reward > max_reward$ **then**
        $max_reward \leftarrow total_reward$
        $Store_Optimal_Path[t][i] \leftarrow j$
      end if
    end for
    $J^*_t(i) \leftarrow max_reward$ *% Store result of Bellman equation*
  end for
end for

▷ 3. Forward Path Reconstruction

$Path[0] \leftarrow$ Initial_State
for $t \leftarrow 0$ **to** $(T_{\text{end}} - \Delta t)$ **do**
  $Path[t + \Delta t] \leftarrow Store_Optimal_Path[t][Path[t]]$
end for
return $Path$

effectively combines the complex pattern modeling capabilities of DNNs with the high interpretability offered by decision trees (DTs). During its development, hyperparameter optimization techniques were employed, and the training process leveraged optimal control data acquired through dynamic programming (DP). Validation of the developed hybrid model was conducted through comprehensive performance analyses on both internal datasets and independent test scenarios. The model's principal output consists of optimal regenerative torque values (EMf and EMr) for the front and rear axle electric motors during braking.

4.1 | Data Preparation and Preprocessing

The dataset required for training and validating the hybrid control model was generated based on the urban dynamometer driving schedule (UDDS), a standardized urban driving cycle. UDDS, with its varying acceleration, deceleration, and constant speed periods, is a widely used protocol for evaluating regenerative braking potential, representing typical urban driving conditions [38]. Figure 9 illustrates the UDDS speed profile and its key characteristics.

To simulate various operational and environmental conditions, 32 scenarios were derived from the deceleration segments of the UDDS cycle, generating a total of 2253 samples (1720 for training and 533 for independent testing). These scenarios include different combinations of critical parameters such as road grade (%), road surface adhesion coefficient (μ), total vehicle mass (kg), and battery state of charge (SOC, %). The parameter values for the scenarios are summarized in Table 2. To objectively evaluate the model's generalization capability, scenarios 4, 9, 24, 25, 29, and 30 were entirely set aside as independent test data, while the remaining scenarios (1720 samples) were used for the model's training (70%), validation (15%), and internal testing (15%) phases.

The generated dataset includes the following input variables (model inputs):

- Total braking torque demand (Nm),
- Battery state of charge (SOC, %),
- Road surface adhesion coefficient (μ),
- Road grade (%),
- Braking rate (braking rate = vehicle acceleration/gravitational acceleration),
- Vehicle speed (m/s),
- Total vehicle mass (kg).

The output variables that the model aims to predict are the optimal regenerative braking torque values for the front electric motor (EMf, Nm) and the rear electric motor (EMr, Nm). These values serve as reference torques determined by the DP method, maximizing energy recovery.

To ensure repeatability in dataset separation, a random seed of 42 was fixed. To improve model training performance and balance the effect of input variables at different scales, input variables were scaled to the [0,1] range using the `minmax` normalization

17519578, 2025, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/itr2.70127 by Erkin Dinçmen - Yildiz Teknik University , Wiley Online Library on [21/12/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

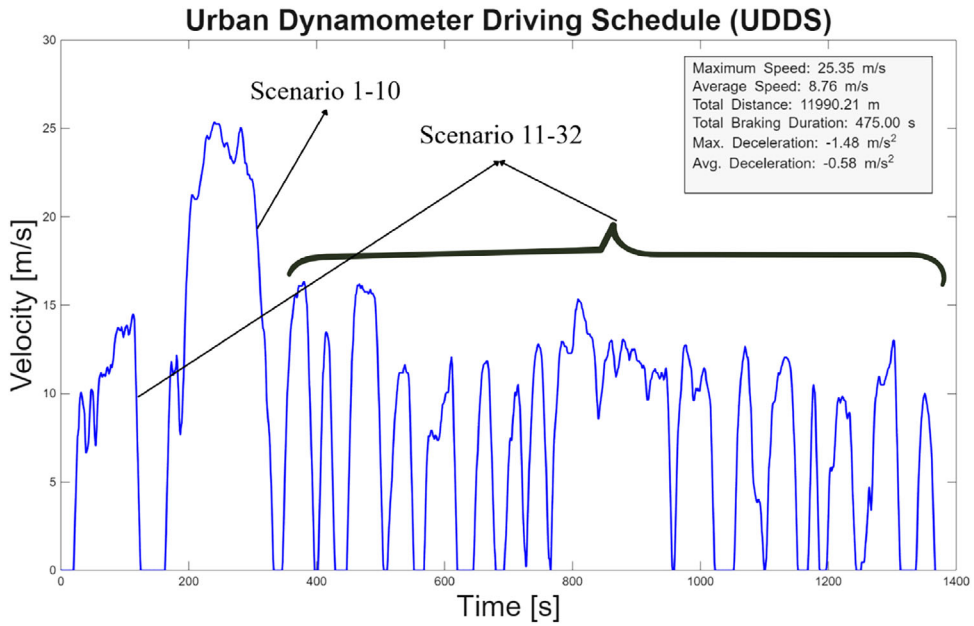


FIGURE 9 | Urban dynamometer driving schedule (UDDS) speed profile.

TABLE 2 | Scenario parameters used for hybrid model development.

Condition	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_10	S_11-32
Road grade (%)	0	0	0	−10	0	0	0	0	10	−10	0
Adhesion coeff. (μ)	0.9	0.6	0.3	0.9	0.9	0.9	0.9	0.9	0.3	0.3	0.9
Total mass (kg)	2000	2000	2000	2000	2200	2400	2000	2400	2200	2400	2400
SOC (%)	60	60	60	60	60	60	20	20	20	60	60

method:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{43}$$

where X represents the original input variable; X_{norm}, the normalized value; and X_{min} and X_{max}, the minimum and maximum values of the respective variable in the training set. The output variables, EMf and EMr torque values, were not subjected to normalization to preserve the direct physical meaning of the predictions.

4.2 | Hybrid System Approach and Architecture

The proposed model for regenerative braking control features a hybrid architecture that combines the strengths of DNN and DT algorithms. DNNs are capable of modeling complex, nonlinear relationships between input variables with high accuracy [39], while DTs are advantageous in capturing localized behaviors and generating interpretable rules, especially under specific sets of conditions [40]. By integrating these two approaches into a hybrid structure, this integration aims to enhance prediction accuracy and enable the model to exhibit more robust and generalizable performance under diverse driving conditions.

The conceptual structure of the developed hybrid model is presented in Figure 10. The system directs input data along two distinct paths: data is normalized using Equation (43) before being fed into the DNN submodule, while it is transmitted to the DT component in its original (unnormalized) form. The DNN, working with normalized data, predicts both EMf and EMr torques within a single network structure. The DT component, which receives unnormalized data, consists of two independent decision trees, separately optimized for each output (EMf and EMr). Predictions from both paths (from DNN: $\hat{y}_{nn,\mathrm{EMf}}$, $\hat{y}_{nn,\mathrm{EMr}}$; and from respective DTs: $\hat{y}_{tree,\mathrm{EMf}}$, $\hat{y}_{tree,\mathrm{EMr}}$) are combined using optimized weight coefficients (w_{nn} and w_{tree}) according to Equations (44) and (45) to form the final EMf and EMr torque predictions of the hybrid model.

$$\hat{y}_{\mathrm{EMf}} = w_{nn} \cdot \hat{y}_{nn,\mathrm{EMf}} + w_{tree} \cdot \hat{y}_{tree,\mathrm{EMf}} \tag{44}$$

$$\hat{y}_{\mathrm{EMr}} = w_{nn} \cdot \hat{y}_{nn,\mathrm{EMr}} + w_{tree} \cdot \hat{y}_{tree,\mathrm{EMr}} \tag{45}$$

where $\hat{y}_{\mathrm{EMf}}$ and $\hat{y}_{\mathrm{EMr}}$ denote the final EMf and EMr torque predictions of the hybrid model; $\hat{y}_{nn,\mathrm{EMf}}$ and $\hat{y}_{nn,\mathrm{EMr}}$ are the torque predictions generated by the DNN; and $\hat{y}_{tree,\mathrm{EMf}}$ and $\hat{y}_{tree,\mathrm{EMr}}$ represent the predictions from the decision trees trained for EMf and EMr, respectively. For the weight coefficients w_{nn} (DNN weight) and w_{tree} (decision tree weight), the condition $w_{nn} + w_{tree} = 1$ applies, and these weights are determined during the

17519578, 2025, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/itr2.70127 by Erkin Dinçmen - Yildiz Teknik University , Wiley Online Library on [21/12/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

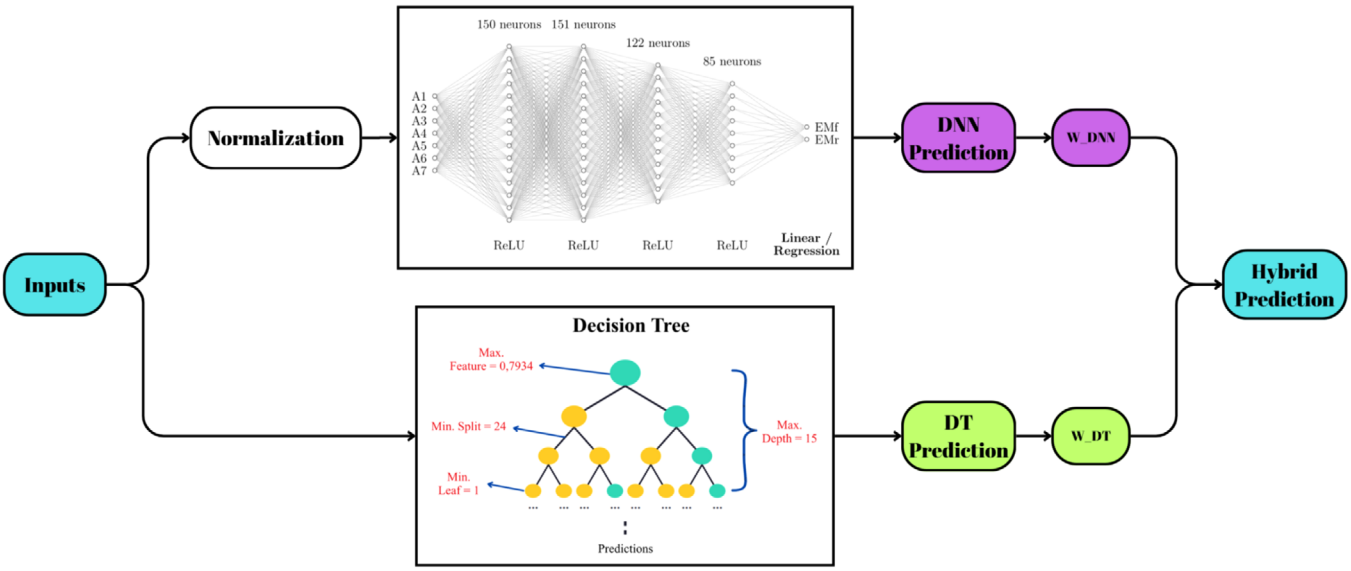


FIGURE 10 | Proposed deep neural network - decision tree hybrid model architecture.

optimization process and used identically for both torque predictions.

4.3 | Hyperparameter Optimization Process

To maximize the hybrid model's potential performance and minimize the risk of overfitting, a comprehensive hyperparameter optimization process was conducted. This process involved optimizing the structural and training parameters of both the DNN and DT sub-models, as well as their weights (w_{nn}, w_{tree}) within the hybrid structure. The primary goal of the optimization was to find the most suitable hyperparameter combination that minimizes the prediction error (e.g., root mean square error - RMSE for both EMf and EMr) on the validation dataset.

4.3.1 | Optimization Approach and Parameter Ranges

Bayesian optimization, which is an effective method in high-dimensional and computationally expensive parameter spaces, was employed [41]. This method builds a probabilistic model from previous trials, which then guides the selection of the next optimal hyperparameter combination. By doing so, it enables achieving near-optimal results with significantly fewer trials compared to random or grid search methods. The optimization process was conducted over 50 trials, each representing a different hyperparameter set.

The key hyperparameters and their ranges examined within the scope of the optimization are listed below:

DNN parameters:

- Number of hidden layers: [1, 4]
- Neurons per layer: [16, 256]
- Learning rate (α): [10^{-5}, 10^{-1}] (logarithmic scale)
- Mini-batch size: [16, 256] (typically powers of 2)
- L2 regularization coefficient (λ): [10^{-10}, 10^{-3}] (logarithmic scale)
- Dropout rate: [0, 0.7]

Decision tree parameters (same for EMf and EMr trees):

- Maximum tree depth: [3, 25]
- Minimum samples split: [2, 30]
- Minimum samples leaf: [1, 15]
- Maximum features: [0.1, 1.0] (ratio of randomly selected features for splitting)

Hybrid model parameters:

- DNN weight (w_{nn}): [0.3, 0.95]
- Decision tree weight (w_{tree}): $1 - w_{nn}$

For each hyperparameter combination, the hybrid model was trained, and its performance (average RMSE) on the validation set was recorded. The progress of the optimization process is shown in Figure 11.

Figure 11 shows a general downward trend in validation error over 50 trials, indicating that the optimization successfully explored hyperparameter regions leading to better performance. The red line represents the best (minimum) error at each step, while blue dots show the result of each individual trial. A specific trial (marked with a star) yielded the lowest validation error and was selected as the optimal configuration.

The individual effects of various hyperparameters on model performance (Avg. RMSE) are visualized in Figure 12.

Several key observations can be drawn from these graphs:

17519578, 2025, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/itr2.70127 by Erkin Dinçmen - Yildiz Teknik University , Wiley Online Library on [21/12/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

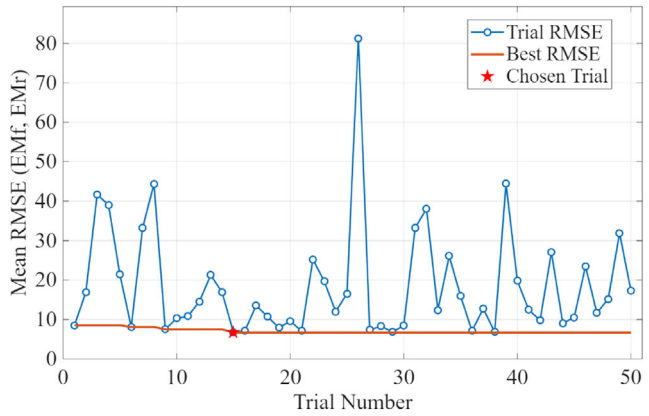


FIGURE 11 | Progress of the Bayesian optimization process: Validation error (Avg. RMSE) for each trial and the best error value found so far.

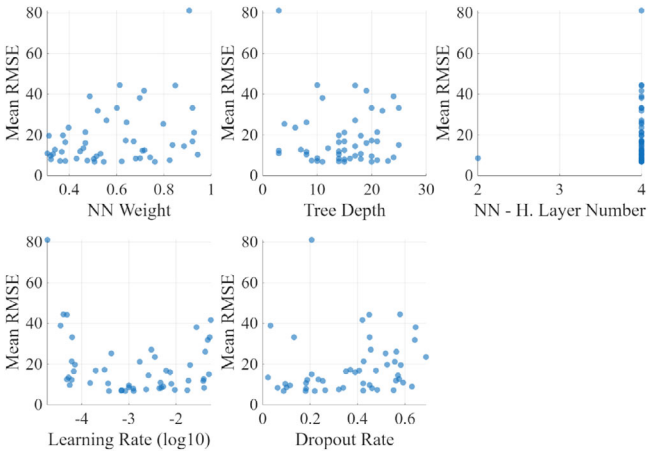


FIGURE 12 | Effects of selected hyperparameters on validation error (Avg. RMSE).

- *DNN Weight* (w_{nn}): While there wasn't a distinct pattern in validation error as the DNN component's weight increased, the lowest errors were generally observed in the 0.3-0.6 range. This suggests that the decision tree component also significantly contributes to model performance. (Note: The final optimized value was 0.7631, which might seem contradictory to the graph, but it's plausible for this value to be optimal due to interactions with other parameters.)
- *Tree Depth*: The optimal depth for decision trees was observed to be between 5 and 25. Shallower trees can lead to underfitting, while very deep trees can cause overfitting.
- *NN Hidden Layer Number*: DNN architectures with 4 hidden layers generally yielded lower errors. This indicates that the complexity of the problem requires deeper network structures.
- *Learning Rate*: On a logarithmic scale, learning rates between 10^{-3} and 10^{-2} (i.e., between -3 and -2) generally resulted in better outcomes. Values outside this range could lead to either slow convergence or instability in the optimization process.
- *Dropout Rate*: The dropout rate, used to prevent overfitting, was observed to be optimal in the 0.2–0.6 range. (Note: The final optimized value is 0.0887, suggesting that a low dropout rate was sufficient or balanced by other regularization methods (L2).)

TABLE 3 | Optimal hyperparameter values determined by Bayesian optimization.

Parameter	Value
DNN parameters	
Hidden layer sizes	[150, 151, 122, 85]
Learning rate	0.0013
L2 regularization (λ)	6.9085e-10
Mini-batch size	49
Dropout rate	0.0887
Decision tree parameters	
Maximum tree depth	15
Min. samples split	24
Min. samples leaf	1
Maximum features ratio	0.5184
Hybrid model parameters	
DNN weight (w_{nn})	0.7631
Decision tree weight (w_{tree})	0.2369

4.3.2 | Optimized Hyperparameters

As a result of the Bayesian optimization process, the hyperparameter configuration yielding the lowest error on the validation set was determined and is presented in Table 3.

According to the optimized structure, the DNN model consists of four hidden layers with 150, 151, 122, and 85 neurons, respectively. For decision trees, maximum depth, splitting, and leaf node conditions were optimized as shown in Table 3. In the hybrid model, the DNN's contribution is determined to be 76.31%, and the decision trees' contribution is 23.69%.

4.4 | Structural Details of the Hybrid Model

4.4.1 | Deep Neural Network Component

The DNN component, designed according to the optimized hyperparameters, has a deep feedforward neural network architecture, and consists of the following layers:

- *Input Layer ('FeatureInputLayer')*: Comprises 7 neurons, corresponding to the 7 normalized input variables.
- *Hidden Layers*: A total of *four* hidden layers are present. Each hidden layer block applies non-linear transformations and regularization before passing the learned feature representations to the next layer:
  - 1^{st} *Hidden Block*: Fully connected layer (150 neurons) + ReLU activation layer + dropout layer (0.0887 rate).

17519578, 2025, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/itr2.70127 by Erkin Dinçmen - Yildiz Teknik University , Wiley Online Library on [21/12/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

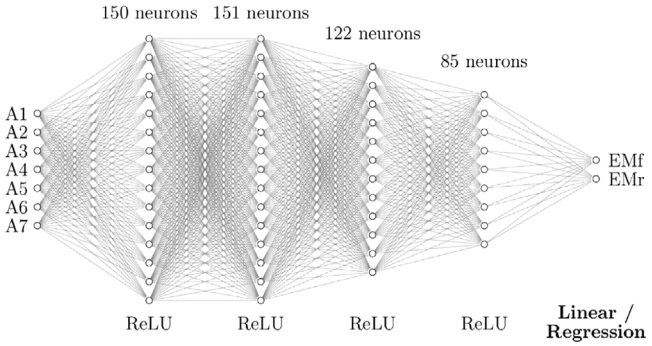


FIGURE 13 | Architecture of the optimized DNN model.

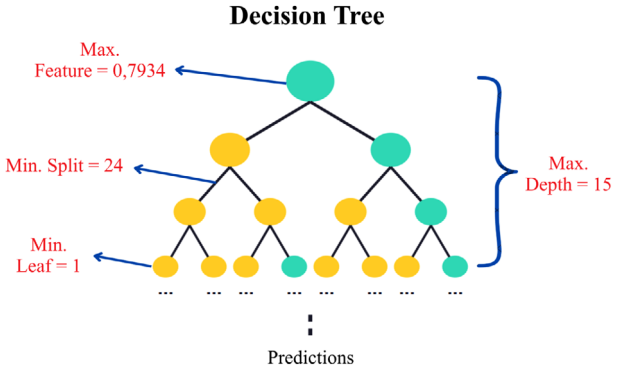


FIGURE 14 | General structure of the decision tree model (separate trees for EMf and EMr).

- – *2nd Hidden Block*: Fully connected layer (151 neurons) + ReLU activation layer + dropout layer (0.0887 rate).
- – *3rd Hidden Block*: Fully connected layer (122 neurons) + ReLU activation layer + dropout layer (0.0887 rate).
- – *4th Hidden Block*: Fully connected layer (85 neurons) + ReLU activation layer + dropout layer (0.0887 rate).

  These layers aim to extract progressively more complex features hierarchically from the input data.

- *Output Layer*: The final part of the network consists of two sequential components: First, a 2-neuron fully connected layer ('FullyConnectedLayer') linearly maps features from the last hidden layer to two target outputs (EMf, EMr). Subsequently, a regression layer ('RegressionOutputLayer') takes these two values and calculates the model error using a loss function like mean squared error (MSE), which is standard for regression tasks.

The layered structure of the DNN model is schematically shown in Figure 13.

The ReLU (rectified linear unit) activation function, commonly used in hidden layers, is defined as $f(x) = \max(0, x)$ and offers benefits such as alleviating the vanishing gradient problem and improving computational efficiency [42]. The 'RegressionOutputLayer' used as the output layer is designed to directly predict continuous values in regression problems and typically works with a loss function such as MSE. The fully connected layer immediately preceding the output (with two neurons) provides a linear mapping that adapts the feature representations to the output dimension.

4.4.2 | Decision Tree Component

The DT component of the hybrid model consists of two regression trees trained independently for each output variable (EMf and EMr). These trees utilize the optimized common hyperparameters specified in Table 3:

- Maximum depth: 15 levels.
- Minimum samples for split: At least 24 data points are required for a node to split.
- Minimum samples per leaf: At least 1 data point must be present for a leaf node to form.
- Maximum features ratio: 51.84% of the input features are randomly selected for searching for the best split at each splitting point.

Decision trees sequentially partition the dataset, creating subregions that satisfy specific conditions and assigning a constant prediction value (the average of samples in that region) to each region. This structure is particularly effective in modeling systems that exhibit different behaviors or are sensitive to threshold values within specific input ranges. Figure 14 represents the general hierarchical structure of the decision tree model.

4.4.3 | Training Parameters and Algorithms

The DNN and DT components of the hybrid model were trained in the MATLAB environment based on the optimized hyperparameters. The main parameters and algorithms determined for the training process are:

- *Optimization Algorithm (for DNN)*: Adam (adaptive moment estimation) [43]. Adam is an efficient gradient-based optimization algorithm that calculates adaptive learning rates for each parameter using estimates of the first and second moments.
- *Maximum Number of Epochs*: 500. The upper limit for the number of passes over the training data.
- *Mini-Batch Size*: 49. The number of samples used for gradient calculation in each iteration.
- *Initial Learning Rate*: 0.0013. The initial learning rate for the Adam algorithm.
- *Weight Decay (L2 Regularization)*: 6.9085e-10. The L2 penalty term coefficient applied to DNN weights to prevent overfitting.
- *Validation Patience (Early Stopping)*: 20 epochs. A criterion that automatically terminates the training process if performance (loss function value) on the validation set does not improve for 20 consecutive epochs, aiming to prevent overfitting [44].

17519578, 2025, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/itr2.70127 by Erkin Dinçmen - Yildiz Teknik University , Wiley Online Library on [21/12/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 4 | Performance metrics of the hybrid model in independent test scenarios.

Scenario	Output	Performance metrics			
		MSE	RMSE	MAE	R^2
Scenario 4	EMf	87.3868	9.3481	6.6158	0.9597
	EMr	131.2672	11.4572	5.9372	0.9119
Scenario 9	EMf	191.4025	13.8348	12.3659	0.9318
	EMr	83.2858	9.1261	5.1275	0.9434
Scenario 24	EMf	17.9908	4.2416	3.7260	0.9977
	EMr	6.8053	2.6087	1.6139	0.9985
Scenario 25	EMf	19.0283	4.3621	3.0739	0.9937
	EMr	14.4299	3.7987	1.7736	0.9930
Scenario 29	EMf	19.6800	4.4362	3.9827	0.9601
	EMr	2.8988	1.7026	1.1539	0.9909
Scenario 30	EMf	11.4395	3.3822	1.9901	0.9887
	EMr	3.6066	1.8991	1.4153	0.9958
Average	EMf	57.8213	6.6008	5.2924	0.9720
	EMr	40.3823	5.0987	2.8369	0.9723

- *Loss Function*: MSE. Calculated by averaging the squares of both EMf and EMr prediction errors, which is then minimized.

Decision trees were trained using standard regression tree algorithms (e.g., CART), under the specified hyperparameter constraints, to minimize MSE.

4.5 | Performance Evaluation and Results

The performance of the developed and optimized hybrid model was comprehensively evaluated using various metrics on a set of completely independent driving scenarios (which were not used during the training or validation phases).

4.5.1 | Performance on Independent Test Scenarios

To measure the model's generalization capability and its robustness under unseen conditions during training and validation, 6 independent test scenarios (Scenarios 4, 9, 24, 25, 29, 30) were used. Quantitative performance evaluation on these scenarios was performed using the metrics presented in Table 4.

The results in Table 4 demonstrate that the hybrid model generally exhibits very high performance across all independent scenarios. Average RMSE values are approximately 6.6 Nm for EMf and 5 Nm for EMr. The coefficient of determination (R^2) values, averaging above 0.97 for both torque predictions, indicate that the model successfully explains a very large portion of the variance in the target torques. Scenario-wise, the model shows near-perfect agreement in most scenarios with R^2 values around or above 0.99. However, in Scenario 9, the error rates (RMSE and MAE) are noticeably higher compared to others. Scenario 4's EMr prediction also has a relatively higher RMSE. This could be attributed to challenging conditions such as low friction, steep road grade, or low SOC.

To provide a visual summary of performance metrics across independent test scenarios and facilitate comparison, error metrics (RMSE, MAE) and R^2 values are shown as bar charts in Figure 15.

This figure visually supports the numerical findings in Table 4. The upper graphs display the calculated RMSE and MAE values for each scenario, highlighting the relatively higher errors in Scenarios 4 and 9 while demonstrating low error levels in other scenarios.

To observe the model's dynamic behavior and prediction accuracy over time in these independent scenarios, DP reference torque values were compared with model predictions in Figure 16.

The plots in Figure 16 demonstrate that the model predictions (red dashed line) closely follow the DP reference torque values (blue solid line). Consistent with Table 4 and Figure 15, the close proximity of predictions to actual values in Scenario 30 is noteworthy. In higher-error scenarios like Scenario 9, small deviations or delays in predictions are observed, especially during sudden torque changes or at peak points. However, the high overall agreement indicates the model's adaptability to dynamic conditions.

In conclusion, these comprehensive evaluations on independent test scenarios strongly demonstrate that the developed hybrid DNN-decision tree model can predict optimal regenerative braking torques for the front and rear electric motors with high accuracy, reliability, and good generalization capability, even under various driving conditions and vehicle parameters not present in the training data.

5 | Simulation-Based Performance Analysis and Comparison of the Hybrid Control Model

This section comprehensively evaluates the performance of the hybrid DNN-DT model, developed and validated offline in Section 4, when integrated into a full electric vehicle model. The primary objective is to measure the energy recovery efficiency of the proposed hybrid control strategy under various driving conditions and parameters, test the model's robustness, and compare it with two reference control methods.

The control strategies compared are:

1. *Hybrid Model Control*: The optimized model developed in Section 4, which dynamically determines EMf and EMr torques based on driving conditions (speed, braking demand, SOC, vehicle weight, road grade, etc.).
2. *Fixed-Ratio Torque Distribution*: A rule-based method where the braking torque demand is distributed between the front and rear axles with a predetermined fixed ratio (70% front–30% rear).

17519578, 2025, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/itr2.70127 by Erkin Dinçmen - Yildiz Teknik University , Wiley Online Library on [21/12/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

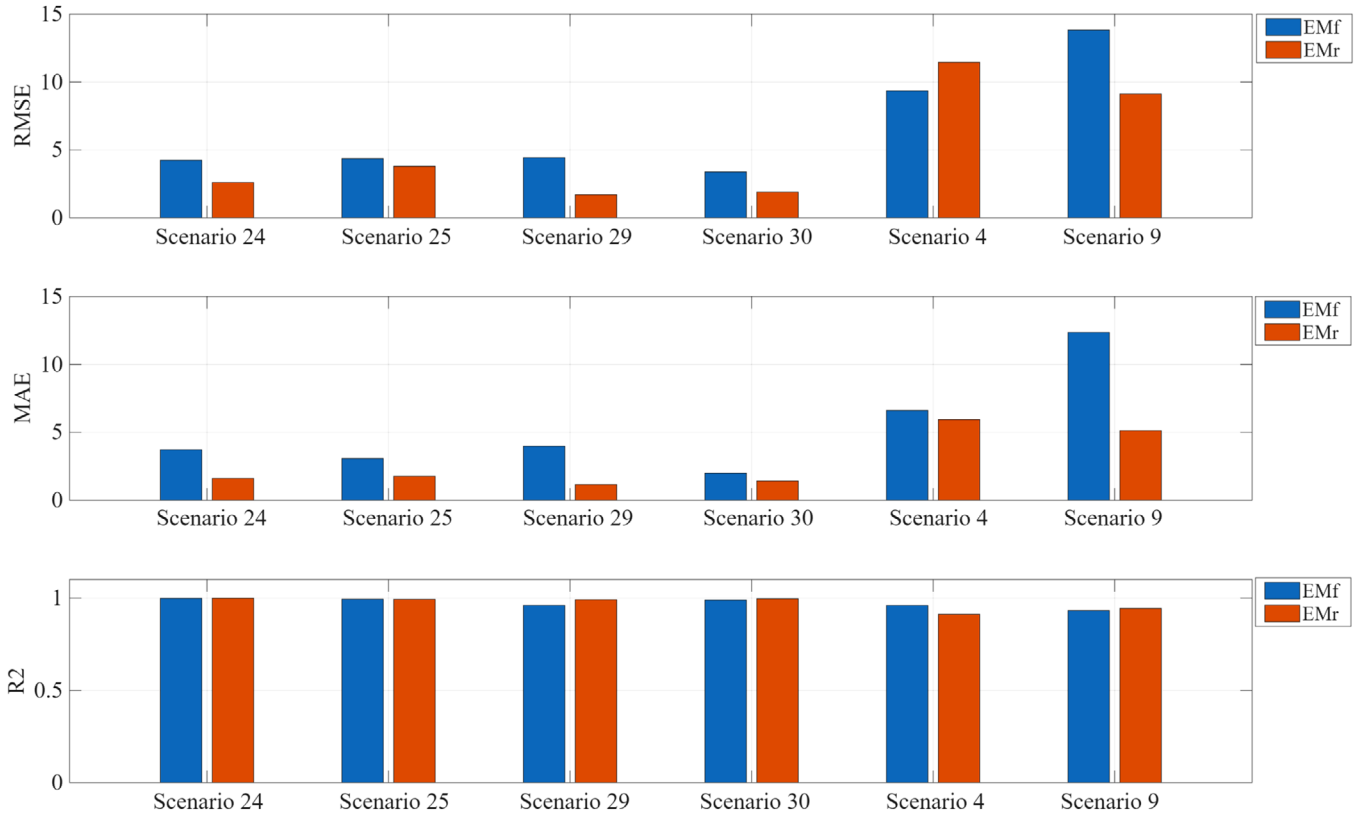


FIGURE 15 | Comparative visualization of error metrics (RMSE, MAE) and coefficient of determination (R^2) values in independent test scenarios.

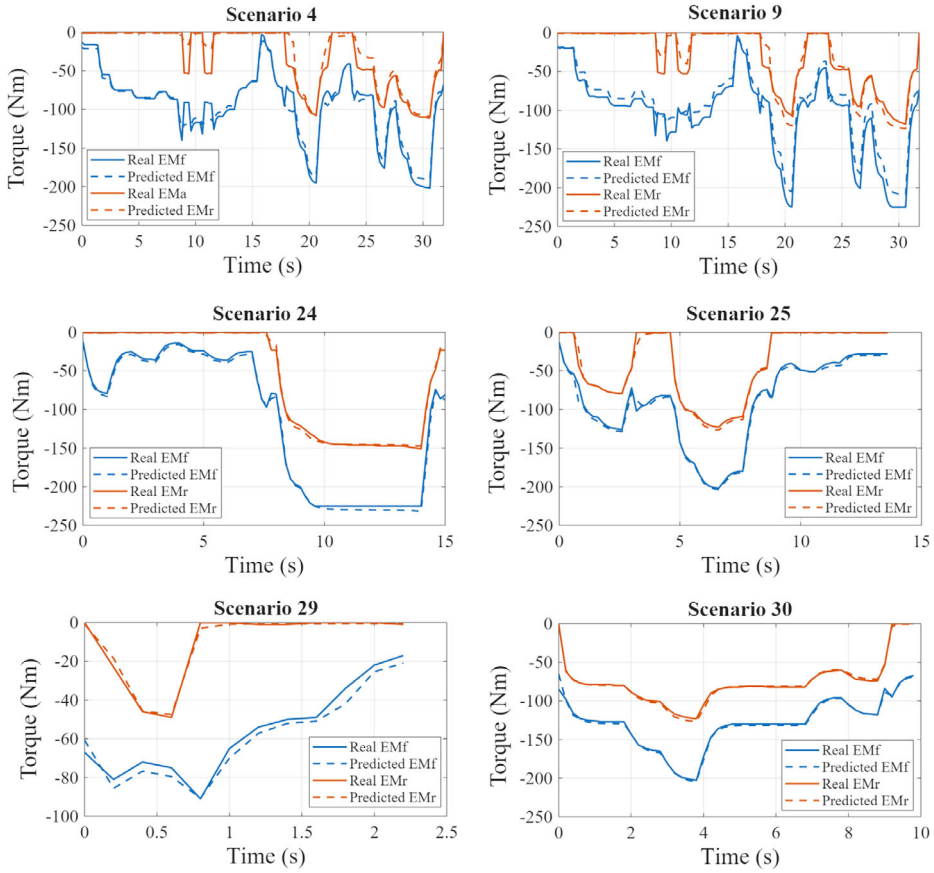


FIGURE 16 | Independent test scenarios: Time-series comparison of DP reference (actual) and hybrid model predictions (predicted) for EMf and EMr torque values.

17519578, 2025, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/itr2.70127 by Erkin Dinçmen - Yildiz Teknik University , Wiley Online Library on [21/12/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

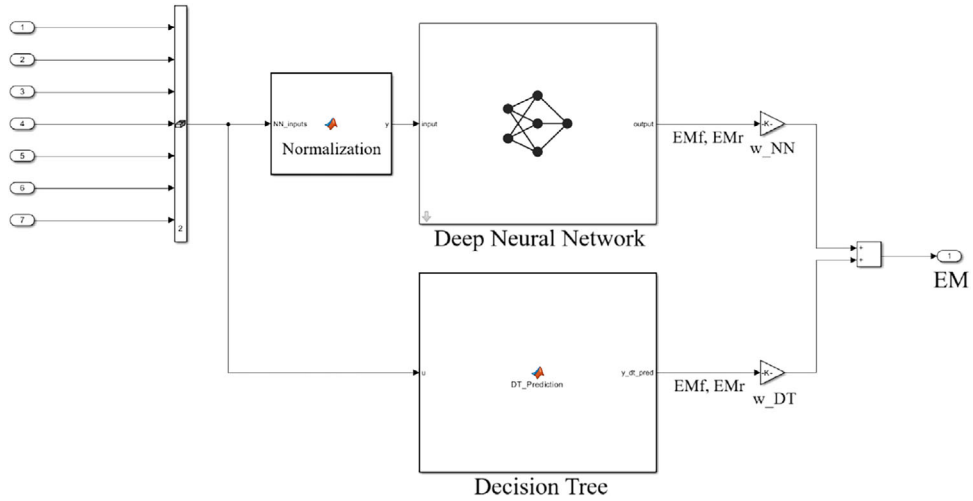


FIGURE 17 | Integration structure of the hybrid control model into the vehicle simulation model.

3. *Baseline Control (Simulink Default)*: This reference method is the default regenerative braking torque distribution strategy within the Simulink vehicle model. According to this strategy, the total braking torque demand is first directed to the front electric motor up to its maximum regenerative braking capacity. If the demand exceeds the front motor's capacity, the excess torque (within its own regenerative braking limit) is attempted to be met by the rear electric motor. If the total regenerative torque generated by the electric motors is insufficient to meet the driver's demand, the remaining difference is compensated by the mechanical braking system.

The main focus of the performance comparison is the change in battery state of charge (SOC) across different scenarios and, consequently, the maximization of energy recovered through regenerative braking. Four different test scenarios were designed to systematically evaluate the model's performance, starting from conditions similar to its training data (Table 2) and progressing to increasingly challenging conditions, including those outside the training range. These scenarios utilize standard driving cycles such as UDDS, NYCC, and WLTP Class 3. The UDDS represents typical urban driving conditions. The NYCC (New York City Cycle), unlike UDDS, represents an aggressive urban scenario with more frequent stop-and-go cycles and lower average speeds that allowed us to test the controller's adaptation under high-frequency braking demands. The WLTP Class 3 cycle is a modern, global standard, covering a much broader speed profile from low to extra-high. This is a perfect scenario for testing how well the model generalizes beyond the training data (extrapolation).

5.1 | Simulation Environment and Model Integration

All simulations were performed in the MATLAB/Simulink environment. The electric vehicle model used includes the fundamental components described in Section 2.

The integration of the developed hybrid control model into the vehicle simulation model was realized with the structure shown in Figure 17.

During the integration process, instantaneous input variables provided by the simulation model (total braking torque demand, SOC, speed, vehicle weight, road grade, adhesion coefficient, etc.) are are fed in parallel to the two components of the hybrid model. Before being sent to the DNN submodule, these inputs are normalized according to Equation (43) using the X_{min} and X_{max} values calculated during the training phase. Simultaneously, the original (unnormalized) form of these inputs is directly fed to the decision tree (DT) sub-models (one for EMf, one for EMr). The normalized predictions obtained from the DNN ($\hat{y}_{nn,\mathrm{EMf}}$, $\hat{y}_{nn,\mathrm{EMr}}$) and the predictions obtained from the decision trees ($\hat{y}_{tree,\mathrm{EMf}}$, $\hat{y}_{tree,\mathrm{EMr}}$) are multiplied by the optimized w_{nn} and w_{tree} weights, respectively, and summed to produce the final EMf and EMr torque commands. These commands are then sent to the electric motor sub-models in the simulation.

To ensure the model is active only during braking and to prevent unnecessary computational load, an "enabled subsystem" structure, shown in Figure 18, was utilized. This structure activates the hybrid model during vehicle braking and keeps it passive otherwise. The model's outputs, EMf and EMr torques, are also transmitted to the relevant motor control units only when braking is active, following a similar logic.

5.2 | Simulation Results and Comparative Analysis

In this section, the performance of the three defined control strategies (hybrid, fixed-ratio, baseline) will be compared based on SOC changes under four different scenarios detailed below. The scenarios are designed to test the model's adaptation and generalization capability, starting from conditions consistent with the hybrid model's training and progressing to parameter changes and conditions outside the training range.

17519578, 2025, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/itr2.70127 by Erkin Dinçmen - Yildiz Teknik University , Wiley Online Library on [21/12/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

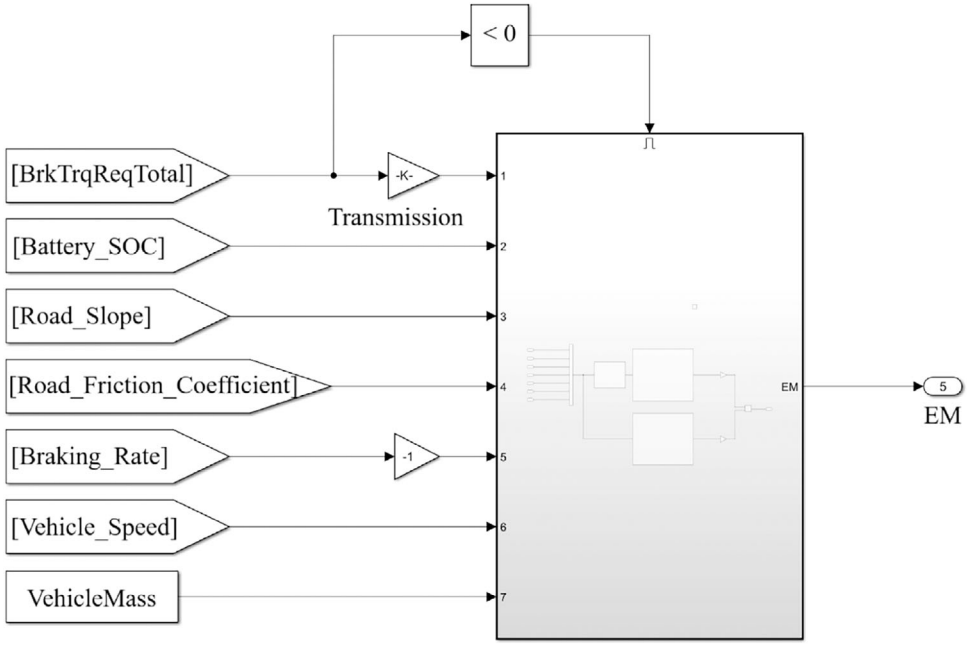


FIGURE 18 | Enabled subsystem logic used to activate the hybrid model only during braking conditions.

5.2.1 | UDDS Cycle Performance in Training Conditions

In this first scenario, the hybrid model's performance was evaluated under conditions very close to the parameter set used for its training. The UDDS cycle, representing urban driving, was used, and simulation parameters were set as follows:

- Driving cycle: UDDS
- Vehicle weight: 2400 kg
- Road grade: 0 degrees (0%)
- Road adhesion coefficient: 0.9
- Initial SOC: 60%

The purpose of this test is to verify how well the model performs compared to reference methods under the "ideal" conditions it has learned.

Figure 19 compares the SOC changes for the three control strategies in this scenario.

As can be seen from Figure 19, the hybrid model (red line) maintains a higher SOC level throughout the cycle compared to the other two strategies. Inset regions in the graph highlight the difference, especially during regenerative braking moments. For example, during braking around the 130-s mark, the increase in SOC for the hybrid model is greater than for the fixed ratio (green dashed) and baseline (blue dashed) methods. This visual observation is supported by the numbers in Table 5; the hybrid model achieved a 3.18% reduction in total SOC loss compared to the baseline at the end of the cycle. The fixed-ratio method showed a more modest 1.09% improvement. These results indicate that the model performs as expected under conditions consistent with its training data and optimizes energy recovery.

TABLE 5 | SOC results obtained at the end of Scenario 1 (UDDS, training conditions).

Control strategy	Final SOC (%)	SOC saved (%)	Improvement (%)
Hybrid model	58.9122	0.0357	3.18
Fixed ratio (70%–30%)	58.8887	0.0122	1.09
Baseline (default)	58.8765	—	—

5.2.2 | NYCC Cycle Performance With Different Vehicle Weight

The second scenario investigates the model's performance under a different driving cycle (NYCC, which includes more frequent stop-and-go conditions) and different vehicle speed and acceleration profiles compared to the UDDS scenario used in training. Other parameters were kept similar to training conditions.

- Driving cycle: NYCC
- Vehicle weight: 2000 kg
- Road grade: 0 degrees (0%)
- Road adhesion coefficient: 0.9
- Initial SOC: 60%

This scenario aims to demonstrate how well the model adapts to a different driving dynamic.

Figure 20 presents the SOC comparisons for this scenario.

Quantitative results are provided in Table 6.

17519578, 2025, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/itr2.70127 by Erkin Dinçmen - Yildiz Teknik University , Wiley Online Library on [21/12/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

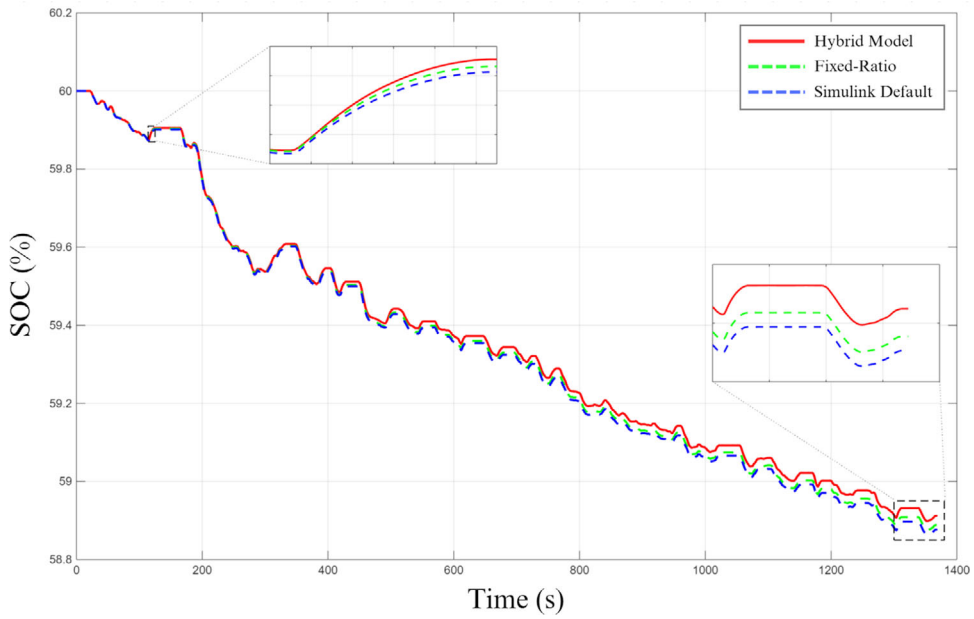


FIGURE 19 | Scenario 1 (UDDS, training conditions): SOC comparison of control strategies.

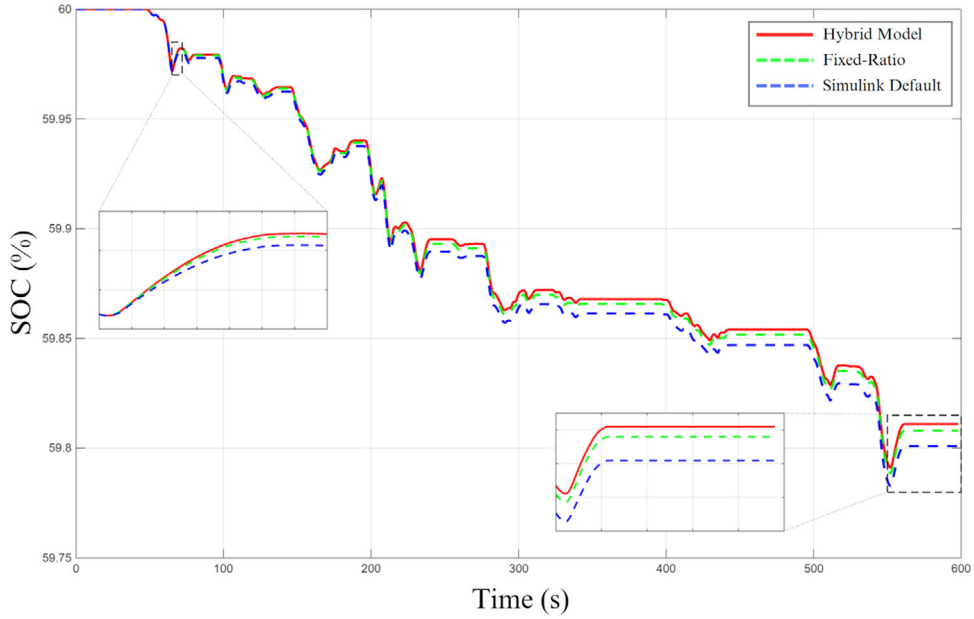


FIGURE 20 | Scenario 2 (NYCC, different weights): SOC comparison of control strategies.

The NYCC cycle involves more frequent and low-speed braking events compared to UDDS. As shown in Figure 20, the hybrid model maintains its superiority even under these intensive stop-and-go conditions. The magnified regions, especially around the 70-s mark and at the end of the cycle, demonstrate that the hybrid model more effectively increases SOC through regenerative braking than other methods. The results in Table 6 confirm this, revealing a significant 5.06% improvement of the hybrid model over the baseline. This rate is higher than the improvement in Scenario 1, suggesting that the developed model may have a greater advantage, especially in urban cycles involving frequent braking. The model's ability to adapt to a driving cycle with different vehicle speed and acceleration profiles is clearly observed in this scenario.

5.2.3 | NYCC Cycle With Different Parameters Within Training Range (Interpolation Test)

The third scenario tests the hybrid model's *interpolation* capability, that is, its performance in situations where parameters are within the minimum and maximum values used in training, although not directly present in the training dataset. For this, the NYCC cycle was again used, and the initial SOC was set to 50%.

17519578, 2025, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/itr2.70127 by Erkin Dinçmen - Yildiz Teknik University , Wiley Online Library on [21/12/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

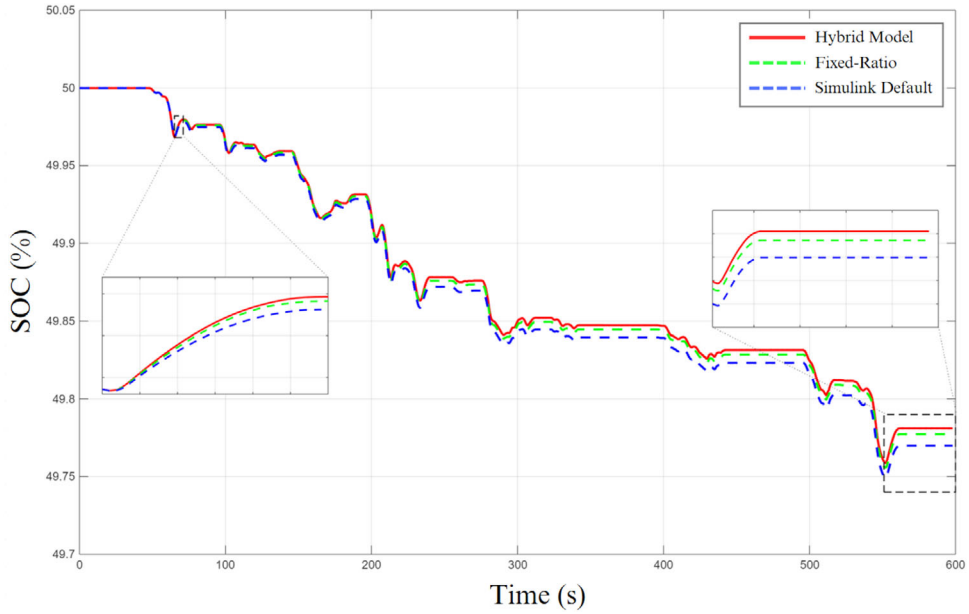


FIGURE 21 | Scenario 3 (NYCC, interpolation): SOC comparison of control strategies.

TABLE 6 | SOC results obtained at the end of Scenario 2 (NYCC, different weight).

Control strategy	Final SOC (%)	SOC saved (%)	Improvement (%)
Hybrid model	59.8110	0.0101	5.06
Fixed ratio (70%–30%)	59.8079	0.0070	3.54
Baseline (default)	59.8009	—	—

TABLE 7 | SOC results obtained at the end of Scenario 3 (NYCC, interpolation).

Control strategy	Final SOC (%)	SOC saved (%)	Improvement (%)
Hybrid model	49.7811	0.0112	4.88
Fixed ratio (70%–30%)	49.7773	0.0074	3.22
Baseline (default)	49.7698	—	—

- Driving cycle: NYCC
- Vehicle weight: 2300 kg
- Road grade: −5 degrees (approx. −8.75%)
- Road adhesion coefficient: 0.7
- Initial SOC: 50%

This test measures how successfully the model fills in the gaps in the training data and responds to unseen parameter combinations that are still reasonable. Figure 21 shows the SOC results for this interpolation scenario.

Table 7 presents the numerical results for this simulation.

In this scenario, multiple parameters such as vehicle weight, road grade, and adhesion coefficient were set to values not directly present in the training set but within its range. The graph in Figure 21 indicates that the hybrid model maintains its performance superiority despite these new conditions. Magnified sections again confirm that the hybrid model increases SOC more during braking than the others. The 4.88% improvement rate in Table 7 is close to Scenario 2 and still quite high. This demonstrates the model's ability to successfully interpolate between different points in the training data and generalize the learned control logic to new, reasonable situations not explicitly seen in the training set.

5.2.4 | WLTP Class 3 Cycle Outside Training Limits (Extrapolation Test)

The fourth and final scenario tests the model's performance under the most challenging conditions, namely its *extrapolation* capability. In this scenario, both the driving cycle (WLTP Class 3) is entirely different, and critical parameters (vehicle weight, road grade, adhesion coefficient) are outside the value ranges used in the model's training. Additionally, the initial SOC was set to 70%.

- Driving cycle: WLTP Class 3
- Vehicle weight: 2500 kg
- Road grade: −15 degrees (approx. −26.8%)
- Road adhesion coefficient: 1.0
- Initial SOC: 70%

17519578, 2025, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/itr2.70127 by Erkin Dinçmen - Yildiz Teknik University , Wiley Online Library on [21/12/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

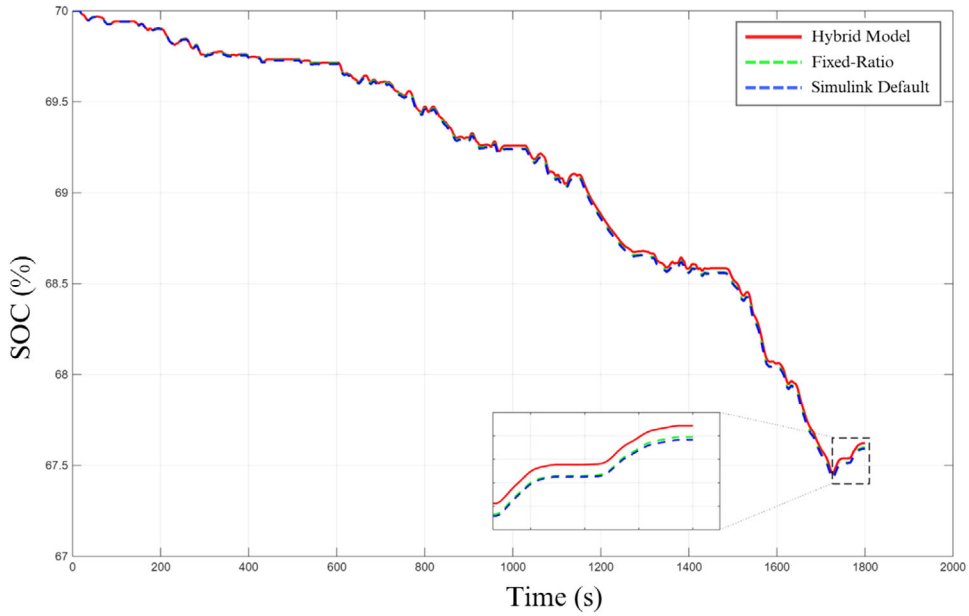


FIGURE 22 | Scenario 4 (WLTP, extrapolation): SOC comparison of control strategies.

TABLE 8 | SOC results obtained at the end of Scenario 4 (WLTP, extrapolation).

Control strategy	Final SOC (%)	SOC saved (%)	Improvement (%)
Hybrid model	67.6220	0.0303	1.26
Fixed ratio (70%–30%)	67.5977	0.0061	0.25
Baseline (default)	67.5916	—	—

This test is crucial for understanding how robust the model remains when encountering unexpected or extreme conditions and whether it can still provide some benefit.

Figure 22 presents the SOC comparisons for this challenging extrapolation scenario.

Numerical results for the simulation are summarized in Table 8.

This scenario demonstrates how the model behaves when pushed beyond its training data limits. Parameters such as an extremely steep negative grade (−15 degrees) and a high adhesion coefficient (1.0) represent conditions the model had not encountered before. Looking at Figure 22, it is noticeable that the difference between the three strategies decreases significantly compared to previous scenarios. However, particularly in the magnified region towards the end of the cycle (around 1700–1800 s), the hybrid model still provides a slight SOC advantage over the other two methods. Table 8 confirms this: the hybrid model achieved a 1.26% reduction in total SOC loss over the Baseline. Although this rate is lower than in previous scenarios, it is important as it shows that the model can still make a positive contribution even under these challenging conditions entirely outside its training limits. This indicates that the model does not fail catastrophically and still provides a net benefit, but its extrapolation performance is, as expected, lower than its interpolation performance, demonstrating its limitations. The improvement of the fixed ratio method in this scenario was quite low (0.25%), closely approaching the baseline.

6 | Discussion

The dynamic programming (DP)-based optimization, detailed in Section 3, established the theoretically achievable maximum energy recovery and the corresponding optimal regenerative braking torque distribution for a dual-motor electric vehicle (EV) under various driving scenarios. These optimal solutions served as the reference dataset for training the hybrid DNN-DT control model developed in Section 4.

The hybrid model's success in emulating DP-generated optimal torque commands was rigorously evaluated using various error metrics (RMSE, MAE) and the coefficient of determination (R^2). It demonstrated high precision (average $R^2 \approx 0.97$, as shown in Table 4) across the independent test scenarios.

As presented in Section 5, the validated hybrid control model, integrated into a full EV model, consistently outperformed two reference control strategies (fixed-ratio torque distribution and baseline/simulink default control) across four different test scenarios. Key findings include: (1) The hybrid model consistently demonstrated superior battery state of charge (SOC) preservation and energy recovery compared to reference methods across all tested scenarios. (2) The net reduction in SOC loss achieved by the hybrid model ranged between 1.26% (WLTP, extrapolation) and 5.06% (NYCC, 2000 kg), representing significant cumulative gains over an entire driving cycle compared to the baseline strategy. (3) The model successfully adapted to different parameter combinations (vehicle weight, road grade, road adhesion coefficient, initial SOC) within the training range (interpolation, Scenario 3)

17519578, 2025, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/itr2.70127 by Erkin Dinçmen - Yildiz Teknik University , Wiley Online Library on [21/12/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

and exhibited robustness under extreme extrapolation conditions (Scenario 4).

6.1 | Robustness of Hybrid Model Performance

The hybrid model's superior performance fundamentally stems from its effective learning and dynamic application of optimal control policies derived from DP. By learning complex, multi-dimensional relationships considering 7 characteristic variables, the model dynamically adapts to different driving conditions.

Furthermore, the model's robustness against potential sensor noise is inherently strengthened by its hybrid architecture. Although a standalone decision tree (DT) could be sensitive to high-frequency noise fluctuating around its sharp decision boundaries, this model weights the DT component's contribution at only 23.69%, while the dominant prediction (76.31%) originates from the DNN component. As the DNN is a continuous function approximator, its inherently smooth output acts as a filter, effectively damping potential 'chatter' from the DT component.

The resulting weighted-sum command is thus stable against minor input perturbations. The strong generalization performance observed in Scenario 3 (interpolation) and Scenario 4 (extrapolation)—where the model produced accurate outputs for entirely unseen input vectors—supports the conclusion that this architecture is inherently robust to small deviations in its inputs, whether they originate from new physical conditions or sensor noise.

6.2 | Comparative Evaluation With Reference Methods

The fixed-ratio distribution approach, despite its simplicity, proved suboptimal due to its inability to adapt to changing driving conditions and vehicle parameters, especially in scenarios divergent from training conditions (Scenarios 2, 3, 4). Its performance in the extrapolation scenario closely approached the baseline method, highlighting the limitations of static rules in unexpected real-world conditions. The baseline control logic, prioritizing the front electric motor's capacity, typically underutilized the rear motor's regenerative potential. In contrast, the hybrid model, by mimicking the optimal or near-optimal strategy learned from DP, more effectively balances motor efficiencies, battery charge acceptance limits, and instantaneous demands, thereby maximizing energy recovery across all scenarios.

6.3 | Practical Impact on Driving Range

The simulation results in Section 5 demonstrate a consistent reduction in net SOC loss, ranging from 1.26% to 5.06% compared to the baseline method. While statistically significant, it is important to translate this percentage into a practical, real-world impact on driving range. This impact can be quantified by calculating the vehicle's average energy consumption from the simulation data and then determining the equivalent range gained from the energy saved by the hybrid model.

The nominal total energy capacity of the battery (C_{bat}), which, based on the cell parameters in the simulation model (96 x 94 cells, 4.493 Ah/cell, and an average 3.67V OCV), is around 148.8 kWh.

Using this capacity, average energy consumption and the resulting range gained for each test scenario based on the baseline model's performance can be calculated as follows:

- Scenario 1 (UDDS, 12.07 km): The baseline model's net energy loss of 1.1235% SOC (1.672 kWh) equates to an average consumption of 13.85 kWh/100 km. The hybrid model's improvement of 0.0357% SOC (0.0531 kWh) translates to a gained range of 383 meters for a single cycle.
- Scenario 2 (NYCC, 1.89 km): The baseline model's net energy loss of 1.1991% SOC (1.784 kWh) in this scenario equates to an average consumption was 15.67 kWh/100 km. The hybrid model's improvement of 0.0101% SOC (0.0150 kWh) translates to a gained range of 96 meters.
- Scenario 3 (NYCC - interpolation, 1.89 km): Under these conditions, the baseline's average consumption was 18.15 kWh/100 km. The hybrid model's improvement of 0.0112% SOC (0.0167 kWh) translates to a gained range of 92 meters.
- Scenario 4 (WLTP, 23.25 km): The Baseline model's average consumption was 15.41 kWh/100 km. The hybrid model's improvement of 0.0303% SOC (0.0451 kWh) translates to a gained range of 293 meters.

While a gain of 90 to 380 meters per cycle may seem modest, this represents the savings from a single trip. The cumulative effect of this consistent energy recovery over thousands of urban driving cycles (e.g., UDDS, NYCC) during the vehicle's operational lifetime translates into a substantial improvement in overall energy efficiency and effective driving range.

6.4 | Computational Load and Real-Time Implementability

One of the advantages for developing a learning-based model is to find a real-time implementable strategy that approaches the high performance of optimization-based methods. Therefore, the computational intensity of the dynamic programming (DP) algorithm—used here to generate optimal training data—is not a drawback; rather, it represents a necessary offline investment to establish the "ground truth" optimal solution.

The primary computational challenge for online control comes from methods like model predictive control (MPC), which must solve a complex optimization problem at every time step. This high real-time computational load is a significant barrier to vehicle control units (VCUs). The proposed hybrid DNN-DT model, by contrast, moves this computational complexity entirely to the offline training phase.

The entire computational workflow was executed on a platform equipped with an Intel i7-1280P (2.00 GHz), an NVIDIA RTX 3050 (4GB), and 16GB of 4800MHz RAM. Data generation via the DP algorithm took approximately 2 h. Model training, includ-

17519578, 2025, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/itr2.70127 by Erkin Dinçmen - Yildiz Teknik University , Wiley Online Library on [21/12/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

ing the hyperparameter search via Bayesian optimization, was completed in a subsequent 30 min. Most importantly, the real-time implementability of the final hybrid model was validated by measuring its inference time (the time required to generate one set of torque commands from one set of sensor inputs). The average inference time for the hybrid model to calculate both EMf and EMr torques was measured to be 2.3 ms. This is significantly faster than the typical control cycle time of a vehicle control unit (VCU), which often operates at 10 ms or 20 ms intervals. This result confirms that the proposed hybrid model is lightweight and computationally efficient, making it fully suitable for real-time deployment on standard automotive hardware.

7 | Conclusion

A novel hybrid DNN–DT control strategy was proposed and validated to optimize regenerative braking torque distribution in dual-motor electric vehicles. An optimal reference dataset, consisting of 2253 samples derived from 7 characteristic variables, was generated by leveraging an offline dynamic programming (DP) algorithm across a comprehensive set of 32 distinct scenarios. This dataset was used to train the hybrid controller, which was then integrated into a 14-DOF simulation environment.

Comparative analysis against two reference methods (fixed-ratio and baseline) across four distinct driving cycles (UDDS, NYCC, WLTP) demonstrated that the DNN–DT controller successfully mimics the optimal DP strategy. It consistently achieved superior energy recovery, yielding a 1.26% to 5.06% reduction in net SOC loss, and proved computationally efficient (2.3 ms inference time) for real-time implementation. In conclusion, this study demonstrated that hybrid control techniques can be effectively utilized to solve complex control problems that aim to improve energy efficiency in electric vehicles.

Future research should focus on enhancing the model's generalization capabilities by expanding the training dataset to include a wider variety of standard driving cycles. Furthermore, the model's robustness could be improved through the exploration of advanced machine learning techniques, like physics-informed neural networks (PINNs), and the implementation of a sensitivity analysis to ensure better performance and reliability when extrapolating to untrained conditions. A key next step would be to move from simulation to physical validation, conducting experimental tests on a dynamometer test bench or a prototype vehicle to confirm the real-time computational performance and energy savings on an actual ECU.

Author Contributions

Omer Ergun: conceptualization, data curation, formal analysis, investigation, methodology, software, validation, visualization, writing – original draft, writing – review & editing. **Erkin Dincmen:** conceptualization, funding acquisition, investigation, methodology, project administration, supervision, writing – review & editing. **Ilyas Istif:** conceptualization, project administration, supervision, writing – review & editing.

Acknowledgements

The authors gratefully thank Prof. Dr. Taner Eskil and Imane Koulali for their advice during the development of the deep learning model. During the preparation of this work the author(s) used Google's Gemini LLM in order to improve readability and language of the work of manuscript. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

Funding

This work was supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK) under Grant No. 122M994.

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

1. D. A. Serrato, J. E. Tibaquirá, J. C. López, J. C. Castillo, M. Giraldo, and L. F. Quirama, "Evaluating Electric Vehicle and Emission Standards Improvement in a Latin American City," *Green Energy and Intelligent Transportation* 4 (2025): 100284.

2. H. Zhang, Y. Luo, N. Ding, et al., "Evaluation of eco-Driving Performance of Electric Vehicles Using Driving Behavior-Enabled Graph Spectrums: A Naturalistic Driving Study in China," *Green Energy and Intelligent Transportation* 4 (2025): 100246.

3. Q. Yuan, W. Hao, H. Su, G. Bing, X. Gui, and A. Safikhani, "Investigation on Range Anxiety and Safety Buffer of Battery Electric Vehicle Drivers," *Journal of Advanced Transportation* 2018 (2018): 8301209.

4. C. Vidal, O. Gross, R. Gu, P. Kollmeyer, and A. Emadi, "xEV Li-ion Battery low-Temperature Effects," *IEEE Transactions on Vehicular Technology* 68 (2019): 4560–4572.

5. J. Ruan, P. D. Walker, P. A. Watterson, and N. Zhang, "The Dynamic Performance and Economic Benefit of a Blended Braking System in a Multi-Speed Battery Electric Vehicle," *Applied Energy* 183 (2016): 1240–1258.

6. J. Liu, Q. Duan, L. Feng, M. Ma, J. Sun, and Q. Wang, "Capacity Fading and Thermal Stability of LiNixCoyMnzO2/graphite Battery After Overcharging," *Journal of Energy Storage* 29 (2020): 101397.

7. C. Luo, Y. Yang, and Z. Zhong, "Optimal Braking Torque Distribution of Dual-Motor Front-Rear Individually Driven Electric-Hydraulic Hybrid Powertrain Based on Minimal Energy Loss," *IEEE Access* 10 (2022): 134 404–134 416.

8. J. Huang, D. Qin, and Z. Peng, "Effect of Energy-Regenerative Braking on Electric Vehicle Battery Thermal Management and Control Method Based on Simulation Investigation," *Energy Conversion and Management* 105 (2015): 1157–1165.

9. P. Fajri, S. Lee, V. A. K. Prabhala, and M. Ferdowsi, "Modeling and Integration of Electric Vehicle Regenerative and Friction Braking for Motor/Dynamometer Test Bench Emulation," *IEEE Transactions on Vehicular Technology* 65 (2015): 4264–4273.

10. E. Louback, A. Biswas, F. Machado, and A. Emadi, "A Review of the Design Process of Energy Management Systems for Dual-Motor Battery Electric Vehicles," *Renewable and Sustainable Energy Reviews* 193 (2024): 114293.

11. X. Li, X. Zhang, and Y. Wang, "Regenerative Braking Control Strategies With Fixed Ratio and Variable Ratio Braking Forces Optimization Dis-

17519578, 2025, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/itr2.70127 by Erkin Dinçmen - Yildiz Teknik University , Wiley Online Library on [21/12/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

tribution for Electric Vehicles During Downhill Process," *International Journal of Automotive Technology* 23 (2022): 667–681.

12. S. Zhou, J. Liu, Z. Wang, and S. Sun, "Research on Design Optimization and Simulation of Regenerative Braking Control Strategy for Pure Electric Vehicle Based on emb Systems," *Transactions of FAMENA* 47 (2023): 33–49.

13. Z. Chen, R. Xiong, X. Cai, Z. Wang, and R. Yang, "Regenerative Braking Control Strategy for Distributed Drive Electric Vehicles Based on Slope and Mass co-Estimation," *IEEE Transactions on Intelligent Transportation Systems* 24 (2023): 14610–14619.

14. M. Tang and X. Zhang, "Optimal Regenerative Braking Control Strategy for Electric Vehicles Based on Braking Intention Recognition and Load Estimation," *IEEE Transactions on Vehicular Technology* 73 (2024): 3378–3392.

15. C. Li, L. Zhang, S. Lian, and M. Liu, "Research on Regenerative Braking Control of Electric Vehicles Based on Game Theory Optimization," *Science Progress* 107 (2024): 00368504241247404.

16. Q. Tang, Y. Yang, C. Luo, Z. Yang, and C. Fu, "A Novel Electro-Hydraulic Compound Braking System Coordinated Control Strategy for a Four-Wheel-Drive Pure Electric Vehicle Driven by Dual Motors," *Energy* 241 (2022): 122750.

17. W. Xu, H. Chen, H. Zhao, and B. Ren, "Torque Optimization Control for Electric Vehicles With Four in-Wheel Motors Equipped With Regenerative Braking System," *Mechatronics* 57 (2019): 95–108.

18. J. Zhang, Y. Yang, D. Qin, C. Fu, and Z. Cong, "Regenerative Braking Control Method Based on Predictive Optimization for Four-Wheel Drive Pure Electric Vehicle," *IEEE Access* 9 (2020): 1394–1406.

19. K. Kiddee and W. Khan-Ngern, "Performance Evaluation of Regenerative Braking System Based on a HESS in Extended Range BEV," *Journal of Electrical Engineering and Technology* 13 (2018): 1965–1977.

20. H. A. Yavasoglu, Y. E. Tetik, and H. G. Ozcan, "Neural Network-Based Energy Management of Multi-Source (Battery/UC/FC) Powered Electric Vehicle," *International Journal of Energy Research* 44 (2020): 12416–12429.

21. H. He, C. Wang, H. Jia, and X. Cui, "An Intelligent Braking System Composed Single-Pedal and Multi-Objective Optimization Neural Network Braking Control Strategies for Electric Vehicle," *Applied Energy* 259 (2020): 114172.

22. H. Tian, Z. Lu, X. Wang, X. Zhang, Y. Huang, and G. Tian, "A Length Ratio Based Neural Network Energy Management Strategy for Online Control of Plug-in Hybrid Electric City Bus," *Applied Energy* 177 (2016): 71–80.

23. C. T. Nguyen, B.-H. Nguyễn, J. P. F. Trovão, and M. C. Ta, "Torque Distribution Optimization for a Dual-Motor Electric Vehicle Using Adaptive Network-Based Fuzzy Inference System," *IEEE Transactions on Energy Conversion* 38 (2023): 2784–2795.

24. B. Prasanth, R. Paul, D. Kaliyaperumal, et al., "Maximizing Regenerative Braking Energy Harnessing in Electric Vehicles Using Machine Learning Techniques," *Electronics* 12 (2023): 1119.

25. X. Zeng, L. Liu, and T. Yuan, "Train Regenerative Braking Strategy Optimization Based on Reinforcement Learning," in *International Symposium for Intelligent Transportation and Smart City* (Springer, 2022), 55–66.

26. W. Yang, D. Luo, and W. Zhang, "Regenerative Braking Strategy Based on Deep Reinforcement Learning for an Electric Mining Truck," *Chinese Journal of Engineering* 46 (2024): 503–513.

27. S. Modi, J. Bhattacharya, and P. Basak, "Convolutional Neural Network–Bagged Decision Tree: A Hybrid Approach to Reduce Electric Vehicle's Driver's Range Anxiety by Estimating Energy Consumption in Real-Time," *Soft Computing* 25 (2021): 2399–2416.

28. I. Ullah, K. Liu, T. Yamamoto, M. Zahid, and A. Jamal, "Electric Vehicle Energy Consumption Prediction Using Stacked Generalization: An Ensemble Learning Approach," *International Journal of Green Energy* 18 (2021): 896–909.

29. A. Kermansaravi, S. S. Refaat, M. Trabelsi, and H. Vahedi, "Ai-Based Energy Management Strategies for Electric Vehicles: Challenges and Future Directions," *Energy Reports* 13 (2025): 5535–5550.

30. Z. Prakash, "Integration of ai and ml in Regenerative Braking for Electric Vehicles: A Review," *Frontiers in Artificial Intelligence* 8 (2025): 1626804.

31. T. Gillespie, *Fundamentals of Vehicle Dynamics*, rev. ed. (SAE International, 2021).

32. I. J. Besselink, A. J. Schmeitz, and H. B. Pacejka, "An Improved Magic Formula/Swift Tyre Model That Can Handle Inflation Pressure Changes," *Vehicle System Dynamics* 48 (2010): 337–352.

33. C. C. MacAdam, "An Optimal Preview Control for Linear Systems," *Journal of Dynamic Systems, Measurement, and Control* 102 (1980): 188–190, https://hdl.handle.net/2027.42/65011.

34. C. C. MacAdam, "Application of an Optimal Preview Control for Simulation of Closed-Loop Automobile Driving," *IEEE Transactions on Systems, Man, and Cybernetics* 11 (1981): 393–399.

35. C. C. MacAdam, "Development of Driver/Vehicle Steering Interaction Models for Dynamic Analysis. Final Technical Report," Tech. Rep. UMTRI-88-51 (The University of Michigan Transportation Research Institute, 1988), https://hdl.handle.net/2027.42/808.

36. United Nations, "UN Regulation no 13-h – Uniform Provisions Concerning the Approval of Passenger Cars With Regard to Braking [2023/401]," *Official Journal of the European Union* L 60/1 (2023), accessed June 20, 2025, https://eur-lex.europa.eu/eli/reg/2023/401/oj/eng.

37. International Organization for Standardization, "Road Vehicles - Determination of Centre of Gravity," ISO 10392:2011 (2011), accessed June 20, 2025, https://www.iso.org/standard/52978.html.

38. T. J. Barlow, S. Latham, I. S. McCrae, and P. G. Boulter, "A Reference Book of Driving Cycles for use in the Measurement of Road Vehicle Emissions," Tech. Rep. PPR354 (TRL Limited, 2009), http://worldcat.org/isbn/9781846088162.

39. S. Haykin, *Neural Networks: A Comprehensive Foundation*, 1st ed. (Prentice Hall PTR, 1994).

40. L. Breiman, J. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*, 1st ed. (Routledge, 2017).

41. J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian Optimization of Machine Learning Algorithms," in *Advances in Neural Information Processing Systems 25 (NIPS 2012)* (Curran Associates, Inc., 2012), https://proceedings.neurips.cc/paper_files/paper/2012/file/05311655a15b75fab86956663e1819cd-Paper.pdf.

42. V. Nair and G. E. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)* (Omnipress, 2010), 807–814, https://www.cs.toronto.edu/~hinton/absps/reluICML.pdf.

43. D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," arXiv preprint, arXiv:1412.6980 (2014).

44. Y. Yao, L. Rosasco, and A. Caponnetto, "On Early Stopping in Gradient Descent Learning," *Constructive Approximation* 26 (2007): 289–315.

17519578, 2025, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/itr2.70127 by Erkin Dinçmen - Yildiz Teknik University , Wiley Online Library on [21/12/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License