

Constraint optimization and key factor analysis based vehicle emergency braking strategy generator

Received: 26 December 2025

Accepted: 23 February 2026

Published online: 27 February 2026

Cite this article as: Xu R., Xu S., Jiang P. *et al.* Constraint optimization and key factor analysis based vehicle emergency braking strategy generator. *Sci Rep* (2026). <https://doi.org/10.1038/s41598-026-41679-w>

Rui Xu, Shijie Xu, Peng Jiang, Xiangfen Zhang, Feiniu Yuan, Xiqiang Guan, Ziyu Chen, Ruifeng Zhai, Yixuan Wang, Tianyi Lu & Wenhao Lu

We are providing an unedited version of this manuscript to give early access to its findings. Before final publication, the manuscript will undergo further editing. Please note there may be errors present which affect the content, and all legal disclaimers apply.

If this paper is publishing under a Transparent Peer Review model then Peer Review reports will publish with the final article.

ARTICLE IN PRESS

Constraint Optimization and Key Factor Analysis based Vehicle Emergency Braking Strategy Generator

Rui Xu¹, Shijie Xu¹, Peng Jiang¹, Xiangfen Zhang^{1,*}, Feiniu Yuan¹, Xiqiang Guan^{1,*}, Ziyou Chen², Ruifeng Zhai¹, Yixuan Wang¹, Tianyi Lu¹, Wenhao Lu¹

¹College of Information, Mechanical and Electrical Engineering, Shanghai Normal University, Shanghai, China; ²King's University College, University of Western Ontario, London, Canada; ³Department of Electrical and Computer Engineering, University of Alberta, Alberta, Canada;

* Correspondence: Zhang: xiangfen@shnu.edu.cn Guan: xqguan@shnu.edu.cn

Abstract

Enhancing vehicle emergency braking performance is crucial for vehicular safety and reliability. We have observed that the traditional vehicle dynamics based model predictive control (MPC) algorithm which is used to produce emergency braking strategy fails to achieve the minimization of emergency braking distance. To address this issue, this article employed a simulation based approach to improve emergency braking performance via machine learning based algorithms. We design a data optimization model which optimizes the longitudinal forces and slip ratios of the four wheels based on back-propagation neural network (BPNN) and constraint optimization algorithm to reduce emergency braking distance. In addition, a Balltree nearest neighbor search based generator is proposed to produce superior emergency braking strategies for real-time purpose. The experimental results demonstrate that the emergency braking strategy which we optimized achieves an average enhancement of 13% in emergency braking performance compared to the traditional MPC algorithm. Unlike other machine learning based algorithms that have a delay of over 0.01 seconds, the proposed generator has an execution time of 0.0008 seconds, meeting the delay requirements under emergency braking conditions.

Keywords: constraint optimization, emergency braking, key factor analysis, model predictive control, nearest neighbor search

1. Introduction

With the continuous increase of the number of vehicles and highway infrastructures, the importance of active vehicle safety technology has significantly increased [1-4]. Vehicles must come to a complete stop from the high speed within a short distance while maintaining stability to prevent unstable phenomena such as side-slip and spin [5]. The design of traditional braking system often relies heavily on empirical knowledge

and engineering testing [6]. While this approach provides a fundamental level of safety, there remains considerable room for improvement under highly variable and complex road conditions [7]. In recent years, with the rapid development in the field of machine learning, the application of machine learning toward enhancing vehicle safety and reliability has emerged as a prominent research trend [8-10].

The number of data dimensions that can be processed by machine learning algorithms far exceeds that of the algorithms based on vehicle dynamics models [11]. However, most machine learning algorithms require substantial computational resources, which means machine learning algorithms have difficulties in satisfying the real-time requirements, that is, the algorithms need to generate the output in an extremely short period [12].

To address these challenges, a simulation based approach was adopted to produce a comprehensive dataset under various driving scenarios, where CarSim 2019 was utilized to generate diverse conditions by systematically varying configuration parameters and the simulated data was recorded by sampling throughout each simulation run. The simulated data was then used to train and validate machine learning algorithms aimed at enhancing emergency braking performance. We propose a data optimization model and an emergency braking strategy generator. The proposed model can optimize emergency braking strategy data to achieve a shorter emergency braking distance and the proposed generator is able to produce superior emergency braking strategy by utilizing the data optimized by the proposed model.

The main contributes of this article are: (1) We design a machine learning based model to optimize emergency braking data points. The data points after the optimization will contribute to generating emergency braking strategy with superior emergency braking performance. (2) We propose an emergency braking strategy generator which employs Balltree nearest neighbor search algorithm to generate emergency braking strategy in an extremely short period. (3) We construct a dataset and the data points in this dataset include optimized emergency braking strategies.

The remainder of this article is structured as follows: Section II discusses the limitations of current algorithms and presents the algorithm proposed in this article. In Section III of this article, a constraint optimization algorithm is used to optimize braking effect and an algorithm is proposed to identify the key factors influencing a label through neural networks. Section IV is experiments, including data collection, experiments related to model training, and experimental results. In Section V, the advantages of the proposed algorithms are summarized and discussed.

1.1. Key Terms

The terms which will be frequently mentioned in the following article will be listed separately in this part.

1. Braking system [13]

The braking system refers to the vehicle's integrated control system, responsible for regulating and generating braking forces to ensure safe deceleration and stopping.

2. Emergency braking [14]

Emergency braking refers to the rapid and forceful deceleration of a vehicle in response to an unexpected or hazardous situation, usually to avoid a collision. It can be initiated manually by the driver applying maximum brake force or automatically by an Autonomous Emergency Braking (AEB) system when an imminent collision is detected.

3. Emergency braking strategy [15]

Emergency braking strategy defines the control logic and decision-making process that determines how braking forces are applied during an emergency event. In this article, we employed the longitudinal force of the left front, left rear, right front, and right rear wheels (F_{x_L1} , F_{x_L2} , F_{x_R1} and F_{x_R2}) and the longitudinal slip ratios of the left front, left rear, right front, and right rear wheels ($KappL_L1$, $KappL_L2$, $KappL_R1$ and $KappL_R2$) as the parameters that the emergency braking strategy need to control.

4. Emergency braking performance/emergency braking effect [14]

Emergency braking performance/emergency braking effect refers to the quantitative evaluation of the effectiveness of the emergency braking system in avoiding or mitigating collision. In this article, we employed the emergency braking distance to quantize emergency braking performance under stable conditions.

5. Emergency braking distance [16]

Emergency braking distance refers to the total distance a vehicle travels from the initiation of emergency braking to a complete stop.

6. Vehicle slip ratio [17]

Vehicle slip ratio quantifies the relative difference between the wheel's rotational speed and the vehicle's actual linear speed. It is defined in the formula (1). In this formula, $V_{vehicle}$ stands for the vehicle's linear velocity and V_{wheel} stands for the tangential velocity at the tire circumference.

$$Vehicle\ slip\ ratio = \frac{V_{vehicle} - V_{wheel}}{V_{vehicle}} \quad (1)$$

7. Model predictive control (MPC) algorithm [18]

Model predictive control (MPC) algorithm refers to a control strategy that predicts future system behavior using a dynamic model and optimizes control inputs over a finite horizon to achieve desired performance while satisfying constraints.

8. Vehicle dynamics model [15]

Vehicle dynamics model refers to the mathematical representation of a vehicle's motion behavior, describing the relationships among forces, slip ratios, and vehicle responses such as acceleration.

9. Back-propagation Neural Networks (BPNN) [19]

Back-propagation Neural Network (BPNN) is a supervised learning model characterized by its backward error propagation mechanism. During training, errors from the output layer are propagated backward through the network's hidden layers, enabling the calculation of gradients for updating connection weights via gradient descent to reduce overall prediction error.

10. Balltree Nearest Neighbor Algorithm [20]

Balltree nearest neighbor algorithm refers to a tree-based spatial indexing method that organizes data points into nested hyperspheres (balls) based on Euclidean distance to accelerate nearest-neighbor search, especially in high-dimensional spaces.

Table 1 will list the symbols and notations that are defined by this article.

2. Related Work

Some traditional research on braking system optimization have developed braking strategies through mathematical modeling [21-23]. Parseh, M. et al. [21] propose a motion-planning framework for autonomous vehicles that incorporates predicted post-impact vehicle motion into collision mitigation decisions. However, the study is limited by certain assumptions. A fuzzy-logic controlled antilock braking system (ABS) is proposed by reference [22]. The simulations show a shorter stopping distance under road surface with high road adhesion coefficient but a smooth enhancement in road surface that has low road adhesion coefficient such as slippery road surface. Labh, S. K. et al. [23] shows a classical proportion integration differentiation (PID) controller to regulate wheel slip for ABS.

While these methods are effective under certain conditions, they often fail to fully utilize the braking capacity in extreme situations or when encountering sudden obstacles [24]. Furthermore, minimizing emergency braking distance imposes specific requirements on vehicle slip ratio and braking force output which refers to the magnitude of decelerating force generated by the braking system. However, the vehicle stability which describes the ability of a vehicle to resist unwanted conditions should be maintained through precise torque distribution. The two objectives, minimizing the emergency braking distance and maintaining vehicle stability, one objective will exhibit strong nonlinear change when linear change happens on another objective, which is defined as nonlinear coupling [25,26]. According to the references [25,26], braking deviation and the loss of steering ability of front wheels will occur due to the nonlinear characteristics of tire force under emergency braking conditions. Mathematical modeling methods often involve important approximations, which can be difficult to accurately characterize this strongly nonlinear coupling relationship [21,27].

In recent years, machine learning, especially deep learning and reinforcement learning, has had a significant impact on the field of intelligent transportation [28-30]. Methods for optimizing braking strategies based on neural network are able to learn meaningful braking-related features from large volumes of driving data [31-34]. Basso, F. et al. [31] proposed a new image-inspired architecture to capture microscopic scene of vehicle behavior, which allows to develop real-time collision prediction models through image processing methods such as convolutional neural network and deep convolutional generative adversarial networks. Teng, X. et al. [32] introduced DTTCNet, a transformer model designed to improve the accuracy of time-to-collision (TTC) prediction, which outperforms existing methods in terms of accuracy. Numerous studies have explored the application of deep reinforcement learning algorithms to optimize vehicle braking control [33,34]. Among these, Chen, D. et al. [34] proposed a novel reward function for

edge cases and a longitudinal control policy for leading and following vehicles, which successfully prevented potential chain collisions in simulated high-risk scenarios.

With the advancement of sensor technology [35], the aforementioned deep learning algorithms are now capable of conducting a holistic analysis of vehicle and road information, thereby enhancing the reliability of emergency braking systems and intelligent driving technologies. These studies not only demonstrate the superiority of machine learning algorithms in optimizing emergency braking systems, but also underscore their significant potential for future applications in this domain.

While current machine learning algorithms have been employed in certain aspects of braking, they lack general applicability in emergency braking because the execution time of the algorithm from receiving input data to outputting results must be strictly limited. As a direct result of this constraint, machine learning algorithms have limited application in the model predictive control (MPC) algorithm [12]. The existing MPC algorithms which are mostly based on vehicle dynamics models have acceptable execution time. However, the vehicle dynamics models lead to suboptimal emergency braking distance because of the inherent inaccuracies of vehicle dynamics models [36].

To overcome these limitations, this article proposes Vehicle Emergency Braking Strategy Data Optimization Model (VEBSDOM) and Vehicle Emergency Braking Strategy Generator (VEBSG). VEBSDOM is used to reduce the emergency braking distance of emergency braking strategy data while ensuring the stability of vehicles. VEBSG is able to produce emergency braking strategies in much shorter time than the other present algorithms.

3. Design of Algorithms

As a new attempt, this article proposes machine learning based modules VEBSDOM and VEBSG to generate emergency braking strategy. The overall architecture of these modules is shown in Fig. 1. We built the Emergency Braking Comprehensive Parameters Dataset (EBCP Dataset) from CarSim 2019. EBCP Dataset comprises comprehensive and structured vehicle and road surface parameters that are recorded at a sampling rate of 40 Hz in emergency braking process. In this dataset, we employed longitudinal acceleration (A_x) and lateral acceleration (A_y) as output parameters (labels).

In vehicle dynamics, A_x and A_y are fundamental indicators of the vehicle's motion states. A_x directly reflects the rate of speed change of the vehicle. A larger A_x value indicates stronger braking deceleration and shorter emergency braking distance [37]. Therefore, A_x serves as the metric for evaluating emergency braking effect. On the other hand, A_y is closely related to the vehicle's lateral stability and cornering behavior. Large A_y value may lead to loss of tire adhesion or vehicle skidding [38]. Thus, A_y is sufficient for assessing the stability performance, as it characterizes the lateral dynamic response and the risk of instability under emergency braking conditions.

This article focuses on cutting down on the emergency braking distance to enhance emergency braking effect. The input parameters in the EBCP Dataset are fed to the key analysis algorithm to select influential factors. The Emergency Braking Core Features Dataset (EBCF Dataset) is constructed using these influential factors, indicating that this

dataset comprises only vehicle and road surface parameters which are worthy of consideration during emergency braking.

Then, a constraint optimization algorithm is used to process the data points in the EBCF Dataset and enhance the emergency braking effect of the emergency braking strategies included in the data points. After processing, we construct Optimized Emergency Braking Strategy Dataset (OEBS Dataset) as the database of VEBSG. The OEBS Dataset comprises input data (parameters representing the current state of vehicle and road surface), decision variables (parameters standing for the optimized emergency braking strategy) and labels (labels before and after optimization). VEBSG can generate the emergency braking strategy in real-time based on the input data.

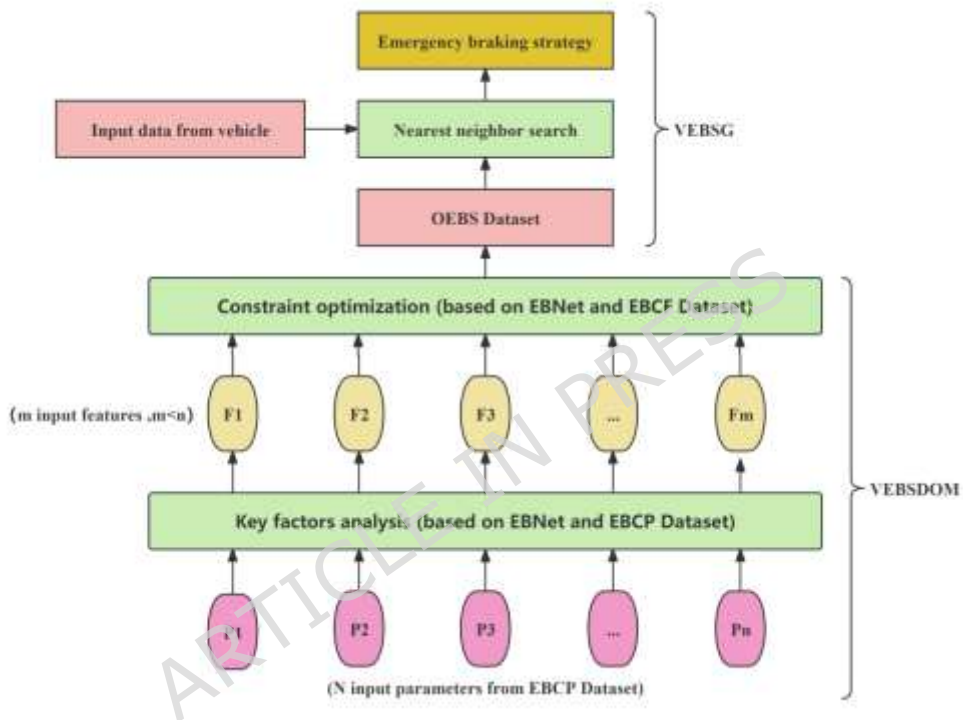


Fig. 1. Overall architecture of VEBSDOM and VEBSG

3.1. Design of VEBSDOM Model

VEBSDOM comprises two key algorithms: key factor analysis algorithm and constraint optimization algorithm. These algorithms allow VEBSDOM to focus on influential factors and produce the emergency braking strategy which demonstrates superior braking performance.

The key factor analysis (KFA) algorithm proposed in this article is to address the issue of data dimension redundancy of the EBCP Dataset. The KFA algorithm is a parameters selection method used to select the input parameters which are useful to predict the output parameters in a prediction model. A considerable number of input parameters in the EBCP Dataset failed to provide a substantial improvement in predicting A_x and A_y and datasets with hundreds of dimensions would lead to a high latency in nearest neighbor search based VEBSG. However, the sampling time or control latency of most

vehicle electronic control units (ECU) are set to approximately 0.01 seconds, while the vehicle undergoes significant dynamic changes within a millisecond timescale. Therefore, we believe that our designing should generate the emergency braking strategy within 0.01seconds based on the input vehicle parameters [39,40].

To address these challenges, this article proposes KFA algorithm to select the input parameters that have significant impact on predicting A_x and A_y from EBCP Dataset as the input parameters of EBCF Dataset. The detailed architecture of the algorithm is illustrated in Fig. 2.

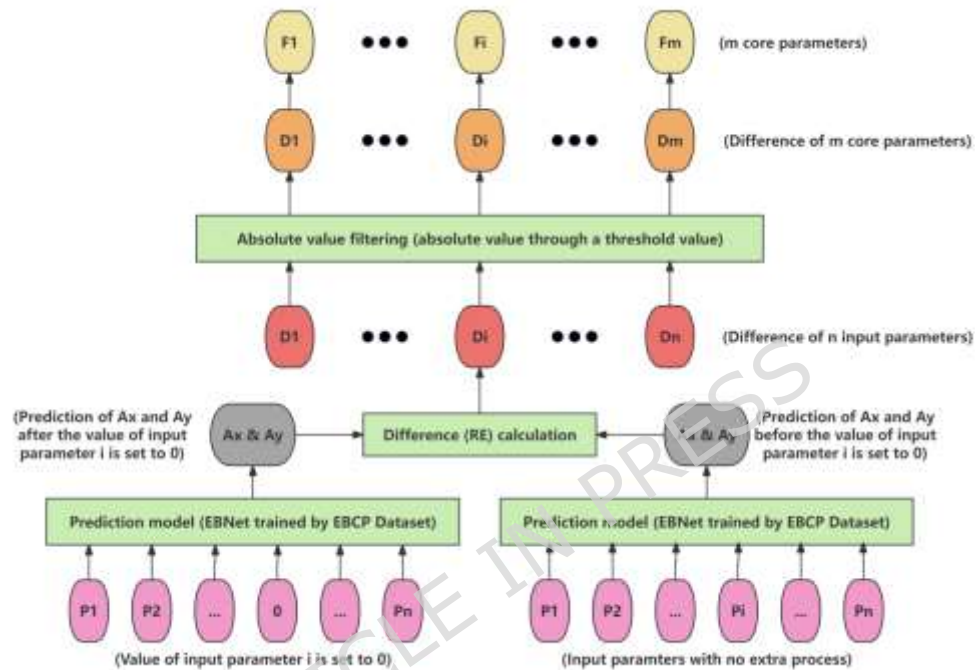


Fig. 2. Architecture of key factor analysis algorithm

As illustrated in Fig. 2, we successively set the value of one input parameter in the EBCP Dataset to 0. For each setting, we predict the value of the output (A_x and A_y). Then we calculate the difference between the prediction before and after the setting. The mean relative error (RE) is employed as a metric to measure the difference. Among each time to set the value of an input parameter to 0, if the difference of A_x or A_y exceeds a threshold value, the input parameter is identified as one key factor. In this article, we design a Back-propagation Neural Networks (BPNN) [19]—Emergency Braking Neural Network (EBNet). EBNet is a BPNN that receives the vehicle and road surface parameters as input to predict A_x and A_y . EBNet trained by the EBCP Dataset serves as the prediction model of the algorithm.

Compared to the principal components analysis (PCA) algorithm [41] which is a dimensionality reduction method that projects high-dimensional data onto a lower-dimensional subspace while preserving as much data variance as possible, the KFA algorithm selects features from the input parameters instead of compressing the input into a lower-dimensional space, which means the KFA algorithms need not an additional dozen milliseconds to process the data and can satisfy the 0.01 seconds

latency requirement.

Moreover, PCA algorithm is to retain as many features as possible with non-specific feature extraction, whereas KFA algorithms are able to rectify this shortcoming by retaining the features which are useful to predict the output parameters.

EBNet in this article serves as a prediction model to predict A_x and A_y through the vehicle and road surface features within the EBCP Dataset or EBCF Dataset. The schematic diagram of the EBNet architecture is presented in Fig. 3 [42], where 'k' denotes the kernel size, 's' represents the stride (defaulting to 1), and the padding method is set to 'same'. Both the input and output data are standardized by using the standard normalizer.

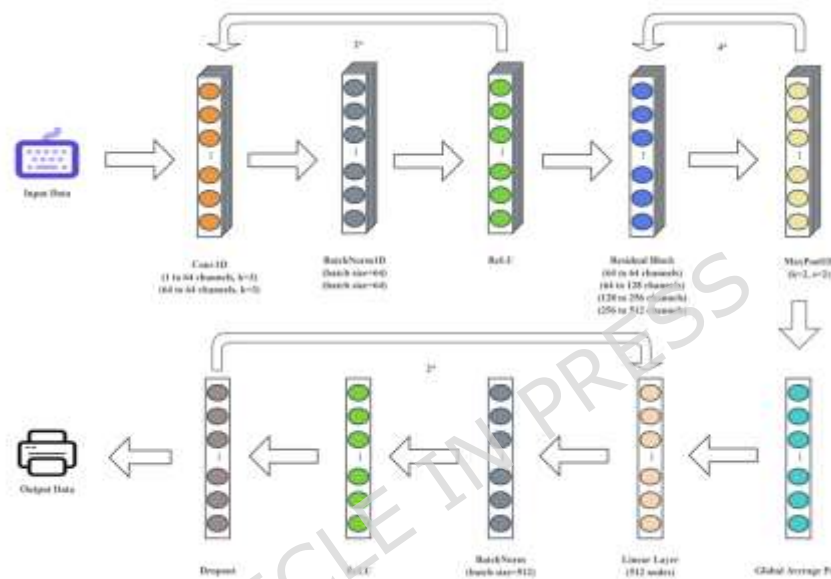


Fig. 3. Architecture of EBNet (The "n*" denotes that the layer blocks encompassed by the arrow are replicated n times)

The detailed structure of the residual block [43] in EBNet is provided in Fig. 4. All convolutional layers within this block share the same number of output channels as the residual block itself.

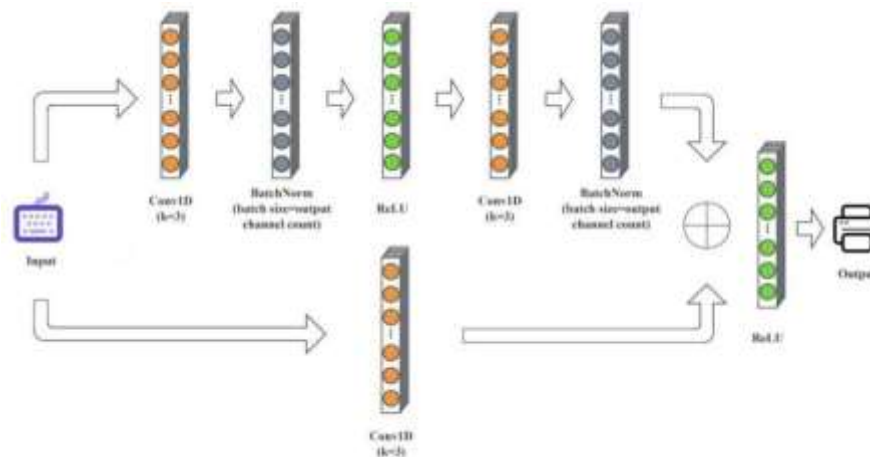


Fig. 4. Residual block of EBNet

The KFA algorithm only allows the model to predict the braking effect and stability. However, the value of VEBSDOM focuses on reducing emergency braking distance instead of merely predicting A_x and A_y . Thus, an optimization for the EBCF Dataset is necessary.

This article employs constraint optimization algorithm [44] to optimize data points in the EBCF Dataset. Constraint optimization algorithm is a computational method designed to find the optimal feasible solution to an optimization problem by maximizing or minimizing the objective function while satisfying a set of given constraints. We focus on enhancing A_x to cut down on the emergency braking distance and improve emergency braking effect. The objective function which needs to be minimized could be given by equation (2).

$$\min\{-|Ax|, Ax \in S\}, S = \{Ax \in \chi: g_i(Ax, Ay) \leq 0, h_k(Ax, Ay) = 0\} \quad (2)$$

Where $g_i(Ax, Ay)$ ($i=1, 2, 3, \dots, m$) stand for the inequality constraints and $h_k(Ax, Ay)$ ($k=1, 2, 3, \dots, n$) stand for the equality constraints.

Due to the high dimensionality of the data and large number of decision variables, this article designs function $F(Ax, Ay)$ which takes into account the constraints and the objective function for the efficient computer processing. The expression of $F(Ax, Ay)$ is given by equation (3). In this formula, a penalty is imposed when constraints are violated and λ_i, α_k stand for the penalty weight.

$$F(Ax, Ay) = -|Ax| + \sum_{i=1}^m \lambda_i \cdot \max(g_i(Ax, Ay), 0) + \sum_{k=1}^n \alpha_k \cdot |h_k(Ax, Ay)| \quad (3)$$

To minimize $F(Ax, Ay)$, we employed gradient descent algorithm. A_x and A_y are not the decision variables, but the decision variables can be linked to A_x and A_y via EBNet. Thus, A_x and A_y can be expressed as $A_x(P_1, P_2, P_3, \dots, P_s)$ and $A_y(P_1, P_2, P_3, \dots, P_s)$ respectively with P being the decision variables. Thereby $F(Ax, Ay)$ can be expressed as $F(A_x(P_1, P_2, P_3, \dots, P_s), A_y(P_1, P_2, P_3, \dots, P_s))$. To facilitate understanding, we represent $F(A_x(P_1, P_2, P_3, \dots, P_s), A_y(P_1, P_2, P_3, \dots, P_s))$ using decision variables P and a new mapping H , which is denoted as $H(P_1, P_2, P_3, \dots, P_s)$.

The partial derivatives of $F(Ax, Ay)$ with respect to each decision variable are calculated and we approximate the partial derivative using equation (4).

$$\frac{d F(Ax, Ay)}{d P_i} \approx \frac{H(\dots (P_i + \Delta w) \dots) - H(\dots (P_i - \Delta w) \dots)}{2 * \Delta w} \quad (4)$$

Where Δw is a small quantity.

During every epoch, each P will be tuned according to the partial derivatives and a learning rate λ . The formula (5) can be expressed as follows:

$$P_i \rightarrow P_i - \lambda * \frac{d F(Ax, Ay)}{d P_i} \quad (5)$$

In the calculation of the gradient descent, we use the approximation of $\frac{d F(Ax, Ay)}{d P_i}$

instead of $\frac{d F(Ax, Ay)}{d P_i}$.

The operation of the constraint optimization algorithm can be divided into two stages. The first stage is the main optimization stage, which is designed to optimize the braking effect under the constraints that ensure the optimized emergency braking strategies are physically implementable. After a certain number of epochs, the main optimization stage is completed. If the optimized emergency braking strategies given in the first stage violate the constraints, the algorithm will proceed to the additional optimization stage i.e. the second stage. The constraints will be placed greater emphasis in the second stage.

Based on the EBCF Dataset and the results of the constraint optimization, this article constructed the OEBS Dataset.

3.2. Design of VEBSG Generator

VEBSG is a nearest neighbor search algorithm based generator, being able to search the data point in the OEBS Dataset which is nearest to the input data from vehicle sensors in term of distance. MPC algorithms used for emergency braking strategy generating can approximately tolerate a latency of 0.01 s, which implies that the vast majority of machine learning based models are almost rendered inapplicability [45]. To satisfy the latency constraint, this article leverages the Balltree nearest neighbor search algorithm [20] to produce emergency braking strategy. The nearest neighbor search algorithm retrieves the data point in the OEBS Dataset which has the smallest Euclidean distance with the current input data. According to the scikit-learn nearest neighbors documentation [46], Euclidean distance enables to provide a satisfactory effectiveness and becomes a common choice in the design of nearest neighbor algorithms.

The Balltree nearest neighbor search algorithm includes two steps. The first step is to establish a binary tree model. The dataset is recursively divided into nested hyperspheres. In every dividing, we select one data point which has the maximum Euclidean distance to the center of the hypersphere and another data point which has the maximum Euclidean distance to the first data point. These two data points serve as the center of two sub-hyperspheres. Each sub-hypersphere will become the hypersphere that need to be divided in the next dividing unless the number of data points in a hypersphere is less than a threshold value and this kind of hyperspheres serve as the leaf nodes of the model.

The second step is searching for the nearest neighbor of the query point. The minimum possible Euclidean distances between the hyperspheres and the query point are calculated. We continuously select the sub-hypersphere with smallest minimum possible Euclidean distance until a leaf node is selected. After the search reaches a leaf node, we calculate the Euclidean distances between the query data point and all data points and select the one which has minimum Euclidean distance as the nearest neighbor.

4. Experiments

This section elaborates on the experimental setup for the datasets and proposed modules, including the data processing and the validation of modules.

4.1. Data Acquisition

4.1.1. Simulation Scenario Settings

The EBCP Dataset was collected in the simulation environment — CarSim 2019. We sampled the emergency braking processes at 40 Hz, from receiving the braking signal to the moment when the vehicle's longitudinal speed (V_x) drops below 1 km/h, since the data points with V_x below 1 km/h is practically meaningless. In the simulation, we set a straight road with a gradient to be 0 degree and an MPC algorithm with default parameters.

During the simulations, we repeatedly set the emergency braking scenario with the set of initial V_x , road adhesion coefficient, wheel parameters and the type of vehicle. The value or value range of each parameter is provided in Table 2. The wheel parameters include tire section width, rim diameter, effective rolling radius and tire flatness ratio. We utilized the parameters vehicle height, vehicle weight, length from the front wheel to the tail, sprung mass and wheelbase to describe the types of vehicle.

4.1.2. Dataset

The experiment was conducted using CarSim 2019, with data acquisition performed throughout the entire emergency braking process at a sampling interval of 0.025 s. The dataset (EBCP Dataset) comprises 447 input parameters and 2 output parameters (A_x , A_y).

Although the emergency braking data acquired by CarSim can accurately capture vehicle dynamics behavior, the EBCP Dataset still contains a certain amount of anomalous data. We firstly eliminated grossly abnormal and redundant data. Then, the entire dataset was fed into an anomaly detection algorithm. Within this algorithm, the expected proportion of outliers was set to 10%. The EBCP Dataset contains approximately 100,000 samples after the data processing.

In the experiment, the KFA algorithm identified 40 input parameters with high RE as the influential parameters for each output parameter. The EBCF Dataset was derived from the EBCP Dataset by remaining the influential parameters and parameters highly relevant to the influential parameters. For instance, if the longitudinal force of the left front wheel (F_{x_L1}) is considered to be an influential parameter, the longitudinal force of the other three wheels (F_{x_L2} , F_{x_R1} , F_{x_R2}) will also be retained.

The following eight parameters— F_{x_L1} , F_{x_L2} , F_{x_R1} and F_{x_R2} , which respectively signify the longitudinal forces of the left front, left rear, right front, and right rear wheels; along with $KappL_L1$, $KappL_L2$, $KappL_R1$ and $KappL_R2$, representing the longitudinal slip ratios of the left front, left rear, right front, and right rear wheels—are considered as the parameters that need to be tuned within the framework of the constraint optimization algorithm. These parameters also represent emergency braking strategy. In the experiment, we removed the input parameters which are highly correlated with

the 8 parameters. After that, we obtained the EBCF Dataset that comprises 92 input features and 2 output parameters.

The OEBS Dataset was constructed based on the results of constraint optimization and the EBCF Dataset. The composition of the OEBS Dataset is detailed in Table 3

4.2. Performance Verification of VEBSDOM Model

This part of the experiment was implemented in Python using TensorFlow. The performance verification of VEBSDOM model was carried out on a workstation equipped with an NVIDIA RTX 5060 with 8GB of video memory. The experiment validated the performances of the VEBSDOM model through two steps. Firstly, we evaluated and compared the accuracy of EBNet based on the EBCP Dataset and EBCF Dataset to demonstrate the accuracy of EBNet and the effectiveness of the KFA algorithm in terms of accuracy. Then we showed the details in constraint optimization algorithm and evaluated the enhancement of emergency braking performance after optimization.

EBNet serves as a prediction model for the KFA algorithm and constraint optimization algorithm. The training results of the EBNet are shown in Table 4. When training EBNet using the EBCP Dataset, we randomly splitted the dataset into a training set (80%), a validation set (10%) and a test set (10%), while the EBCF Dataset was splitted into a training set (45%), a validation set (10%) and a test set (45%) in the training of EBNet using the EBCF Dataset. The test set (45%) were used at the step of constraint optimization and to construct the OEBS Dataset. In Table 4, we compared EBNet trained by the EBCP Dataset which has not been processed by the KFA algorithm and the EBCF Dataset which has been processed by the KFA algorithm.

The performance indicators include mean absolute error (MAE), mean square error (MSE), RE and high RE proportion. It is worth noting that the experiment evaluated the high RE proportion of A_x and A_y . In emergency braking situations, constraints related to A_x and A_y , such as friction circle constraints, should be taken into account at each time interval to prevent instability. Therefore, we can use this metric to determine the margin of the constraints imposed on the model error. Taking into account both the reduction in emergency braking effectiveness caused by this margin and the satisfaction of the constraints, we set the threshold to be 10% and use this as the basis for setting the margin of the constraints.

The comparison of the results from training EBNet using the EBCP Dataset (with KFA algorithm) and EBCF Dataset (without KFA) reveals that the KFA algorithm successfully improves the accuracy of EBNet as well as reduces the input dimensions to cut down on the latency of VEBSG.

The experiment enhanced the absolute value of A_x to improve emergency braking effect through tuning F_{x_L1} , F_{x_L2} , F_{x_R1} , F_{x_R2} , K_{appL_L1} , K_{appL_L2} , K_{appL_R1} and K_{appL_R2} while ensuring the constraints. We optimized each data point in the EBCF Dataset through tuning the 8 parameters and utilizing EBNet based on the EBCF Dataset for predicting and measuring the optimization effect.

We designed 2 essential constraints to ensure the physical realizability and safety of the optimized emergency braking strategies.

One constraint is called friction circle constraint, that is, the sum of the squares of A_x and A_y must not exceed the friction circle which can be expressed as $(\mu \cdot g)^2$. Where μ stands for road adhesion coefficient and g stands for the acceleration of gravity. Friction circle shows the maximum resultant adhesive force available between the tire and the road surface, which means there will be unstable phenomena happen if the tire adhesive force exceeds $(\mu \cdot g)^2$. To prevent error-induced fluctuations in the neural network from leading to violations of the friction circle constraint, a scaling factor of 0.98 was introduced in the experiment and applied to $(\mu \cdot g)^2$. The formula is as follows:

$$A_x^2 + A_y^2 - 0.98(\mu \cdot g)^2 \leq 0 \quad (6)$$

The other constraint is called A_y constraint, that is, the absolute value of A_y must not exceed a presetting maximum value. In the experiment, the determination of the presetting maximum value took into account the value range of A_y in the EBCF Dataset as well as the error of EBNet based on the EBCF Dataset. The formula is as follows:

$$|A_y| - k \leq 0 \quad (7)$$

Where k stands for the preset maximum. We set k to 2.8 m/s^2 in the experiment.

We set an objective function and minimized the objective function through gradient descent. The design of the objective function is based on the optimizing object and constraints. The function is as follows:

$$F(A_x, A_y) = -|A_x| + \lambda_1 \cdot \max(A_x^2 + A_y^2 - 0.98(\mu \cdot g)^2, 0) + \lambda_2 \cdot \max(|A_y| - k, 0) \quad (8)$$

Where λ_1 and λ_2 stand for the penalty weights of the constraints.

In the experiment, we configured the parameters in the constraint optimization algorithm and the configured parameters are listed in Table 5.

The entire set of data points in the EBCF Dataset has satisfied the constraints in the experiment. Statistical results of A_x before and after optimizing are shown in Table 6. The unit of A_x in the table is g that stands for 9.8 m/s^2 .

According to the average value in the Table 6, the optimized A_x has increased by about 13% on average after optimization

4.3. Performance Verification of VEBSG Generator

This part of experiment was conducted in Python, employing the scikit-learn library for the validation of the generator VEBSG. VEBSG generates the emergency braking strategy based on the input data points collected from the vehicle sensors, and the OEBS Dataset serves as the database of Balltree nearest neighbor search algorithm in VEBSG.

Among the 92 input parameters in the OEBS Dataset, 84 parameters (excluding the 8 parameters which need to be tuned) were monitored in the nearest neighbor search algorithm. The 84 parameters were utilized to calculate the Euclidean distance between the data points in the OEBS Dataset and the input data acquired from the vehicle sensors, and then we searched for the data point that minimizes the Euclidean distance. The experiment produced the emergency braking strategies based on the emergency braking strategy of this data point.

Due to the subtle differences between the data point in the OEBS Dataset and the input data, we recommend to introduce minor variations into the emergency braking strategies in OEBS Dataset to ensure that the constraints are still complied. For instance, Fx_L1, Fx_L2, Fx_R1 and Fx_R2 were fine-tuned to ensure the stability of the vehicles in the experiment.

In the experiment, 30,000 data points from the OEBS Dataset served as the database of the nearest neighbor search algorithm and about 10,000 other data points were used to simulate the input data. The results showed 0.00087s in execution time from receiving the input parameters to finish generating the output parameters (emergency braking strategy), which satisfies the requirement of latency under emergency braking conditions.

A comparison between the average emergency braking effect of the emergency braking strategies generated by VEBSG and the traditional MPC algorithm is presented in Table 3. The emergency braking strategies in the OEBS Dataset stands for the emergency braking strategies generated by VEBSG. The emergency braking strategies in the EBCF Dataset stem from the EBCP Dataset that was collected in CarSim 2019, which means the emergency braking strategies in EBCF Dataset are generated by vehicle dynamics based MPC algorithms within the vehicle model in CarSim 2019. For the execution time, we measure the normalization of the input data point and the searching of the data point in the database. The unit of Ax in Table 7 is g that stands for 9.8 m/s².

As can be seen from Table 7, the emergency braking strategies generated by VEBSG achieves about 13% improvement of Ax over the emergency braking strategies produced by vehicle dynamics based MPC algorithm on average.

If we approximately regard emergency braking as a uniformly accelerated motion with the initial value of Vx be V₀, we can get the distance formula and duration time formula of this kind of motion, which are defined as in formula (9) and formula (10) respectively.

$$L = V_0 * t - 0.5 * t^2 * Ax \quad (9)$$

$$t = V_0 / Ax \quad (10)$$

Where t stands for the duration time of emergency braking process and L stands for the distance.

By combining formula (9) and (10), we obtain the formula (11):

$$L = 0.5 * V_0^2 / Ax \quad (11)$$

From formula (11), we can get formula (12) which describes the relationship between the increment of Ax and the decrement of L:

$$\left(\frac{1}{1 + x\%} \right) * L = 0.5 * V_0^2 / [Ax * (1 + x\%)] \quad (12)$$

Where x% stands for the increment percentage of Ax.

Thus, We can calculate and conclude that a 13% improvement of Ax will lead to an approximately 11% reduction of emergency braking distance. Furthermore, VEBSG is able to generate strategy in 0.00087s, which is much faster than the traditional vehicle dynamics based MPC algorithm.

5. Discussion

In this article, we propose a parameter selection method (KFA algorithm), a data optimization model (VEBSDOM) and a emergency braking strategy generator (VEBSG). Their effectiveness are evaluated in the experiment.

For the KFA algorithm, it enables us to select the input parameters of a prediction model which are useful for predicting a label or output parameter. This method was originally used to reduce the data dimensions to cut down on the execution time of the VEBSG. With fewer dimensions of data, VEBSG is able to calculate the Euclidean distance faster. At the same time, we found that the KFA algorithm improves the prediction accuracy of the neural networks (EBNet), which is possibly because the neural networks are able to pay more attention on useful features in the KFA algorithm.

For the VEBSG, it generates the emergency braking strategy based on the database constructed through VEBDOM and presents an 13% improvement of emergency braking effect (A_x) and 0.00087s of execution time, which is able to satisfy the real-time engineering requirements and provide a safer and shorter emergency braking distance. Compared with other deep learning algorithms, these algorithms could struggle to meet the real-time requirements of MPC algorithms. Depending on the short execution time, VEBSG enhances the safety of emergency braking from another perspective.

To sum up, our study improves vehicles safety through continuously generating emergency braking strategies after the vehicles turning into emergency braking conditions.

However, there are limitations in our study. Firstly, we only consider about the current state of the vehicle and road while the past state may be useful for prediction. Secondly, the VEBSG is based on the nearest neighbor algorithm, which means it can only provide a fundamental level of prediction accuracy as the execution time is extremely short.

For the further study, we will focus on predicting the future vehicle state by utilizing the past and current vehicle and road state, which further improves the accuracy of prediction. Moreover, we believe that there is possibility to predict vehicle state in the coming several hundred milliseconds, enabling to spare more time to generate emergency braking strategies with more advanced and large machine learning based model.

6. Conclusion

The proposed VEBSG model can generate an optimized emergency braking strategy in 0.0008 s and showed an 13% average improvement in emergency braking effect compared with the emergency braking strategies produced by the traditional MPC

algorithm based on vehicle dynamics model. According to the experiment results, the primary advantages of the VEBSG model are twofold. On one hand, the data optimization model VEBSDOM exhibits a high accuracy, with high-error data points being virtually absent. This ensures both the practical realizability and precision. On the other hand, compared to the deep learning based algorithms and the traditional MPC algorithm, the nearest neighbor search algorithm allows VEBSG module to achieve shorter execution time.

To the best of our knowledge, VEBSG is the first try to deploy machine learning on the MPC for the formulation of emergency braking strategies. The superior performances of VEBSDOM and VEBSG strategies are evidenced by its high accuracy and computational efficiency. The OEBS Dataset is also able to be loaded into the memory to reduce the required time and be expanded by offline training.

Funding Declaration

This work was partially supported by the National Natural Science Foundation of China (62171285), Natural Science Foundation of Shanghai Municipality (20ZR1440500) and General Research Fund of Shanghai Normal University (KF2021100 and Sk201220).

Author Contributions

Rui Xu, Shijie Xu contributed to the conception and design of this study. Rui Xu, Xiangfen Zhang and Feiniu Yuan performed the experiment and evaluated the model performance. Peng Jiang, Rui Xu collected and processed the datasets. Xiqiang Guan, Tianyi Lu, Ruifeng Zhai, Yixuan Wang contributed to the theoretical research. Xiangfen Zhang, Rui Xu, Ziyou Che and Wenhao Lu contributed significantly to result analysis and manuscript preparation. Fig. 1, Fig. 2, Fig. 3 were drawn by Rui Xu and Xiangfen Zhang. Fig. 4 was drawn by Shijie Xu. Rui Xu, Xiangfen Zhang and Feiniu Yuan contributed to the manuscript revision.

Data Availability Statement

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

Competing Interests Statement

No potential conflict of interest was reported by the author(s).

Reference

- [1] Lutanto, A., Fajri, A., Nugroho, K. C. & Falah, F. Advancements in automotive braking technology for enhanced safety: A review. *MIRAE* **1**, 6–21 (2025). doi:10.70935/6n21wx31.
- [2] Sokolovskij, E. & Žuraulis, V. Advances in vehicle dynamics and road safety: Technologies, simulations, and applications. *Appl. Sci.* **14**, 3735 (2024). doi:10.3390/app14093735.
- [3] Furlan, A. D. et al. Advanced vehicle technologies and road safety: a scoping review of

- the evidence. *Accid. Anal. Prev.* **147**, 105741 (2020). doi:10.1016/j.aap.2020.105741.
- [4] Zhao, J., Yan, X., Zhou, Z. & Zhang, Z. Rule-adaptive lane-changing trajectory planning method for autonomous vehicles driven by dynamic risk information. *Sci. Rep.* **15**, 31514 (2025). doi:10.1038/s41598-025-15382-1.
- [5] Meléndez-Useros, M., Jiménez-Salas, M., Viadero-Monasterio, F. & López Boada, B. L. Tire slip H_∞ control for optimal braking depending on road condition. *Sensors* **23**, 1417 (2023). doi:10.3390/s23031417.
- [6] Ricciardi, V., Travagliati, A., Schreiber, V., Klomp, M., Ivanov, V., Augsburg, K. & Faria, C. A novel semi-empirical dynamic brake model for automotive applications. *Tribol. Int.* **146**, 106223 (2020). doi:10.1016/j.triboint.2020.106223.
- [7] El-Gindy, M., El-Sayegh, Z. Road Vehicle Braking Performance. In: Road and Off-Road Vehicle Dynamics. *Springer, Cham.* (2023) doi:10.1007/978-3-031-36216-3_7
- [8] Qiu, C., Tang, H., Yang, Y. et al. Machine vision-based autonomous road hazard avoidance system for self-driving vehicles. *Sci. Rep.* **14**, 12178 (2024). doi:10.1038/s41598-024-62629-4.
- [9] Henning-Günther, T., Grujic, D., Werner, T., Weber, L., Neurohr, B. & Möhlmann, E. Verifying safety of safety-critical systems with rare events via optimistic optimization. *IEEE Open J. Intell. Transp. Syst.* **6**, 1569–1579 (2025). doi:10.1109/OJITS.2025.3638166.
- [10] Chae, H., Kang, C. M., Kim, B., Kim, J., Chung, C. C. & Choi, J. W. Autonomous braking system via deep reinforcement learning. Proc. IEEE Intell. Transp. Syst. Conf. 1–6 (2017).
- [11] Hashemi, A., Orzechowski, G., Mikkola, A. & McPhee, J. Multibody dynamics and control using machine learning. *Multibody Syst. Dyn.* **58**, 397–431 (2023). doi:10.1007/s11044-023-09884-x.
- [12] Sun, Y., Shi, W., Huang, X., Zhou, S. & Niu, Z. Edge learning with timeliness constraints: Challenges and solutions. *IEEE Commun. Mag.* **58**(12), 27–33 (2020). doi:10.1109/MCOM.001.2000382
- [13] Mirzaeinejad, H. & Mirzaei, M. A novel method for non-linear control of wheel slip in anti-lock braking systems. *Control Engineering Practice* **18**, 918–926 (2010). doi: 10.1016/j.conengprac.2010.03.015.
- [14] Deng, G. & Li, D. Safety Distance Analysis of Automatic Emergency Braking System. In *Selected Contributions of 2024 2nd International Conference on Electric Vehicle and Vehicle Engineering* (eds Wong, P. K. & Xu, J.) *Lecture Notes in Electrical Engineering*, vol. 1424, 452–458 (Springer, Singapore, 2025). doi:10.1007/978-981-96-6827-4_47.
- [15] Gounis, K. & Bassiliades, N. Intelligent momentary assisted control for autonomous

- emergency braking. *Simulation Modelling Practice and Theory* **115**, 102450 (2022). doi:10.1016/j.simpat.2021.102450.
- [16] Zhang, R. & Pourkand, A. Emergency-braking Distance Prediction using Deep Learning. *arXiv preprint arXiv 2112.01708* (2021). doi:10.48550/arXiv.2112.01708.
- [17] Shewale, N. S. & Deivanathan, R. Modelling and simulation of anti-lock braking system. *International Journal of Engineering and Technical Research* **7**(1) (2017).
- [18] García, C. E., Prett, D. M. & Morari, M. Model predictive control: Theory and practice—a survey. *Automatica* **25**, 335–348 (1989). doi:10.1016/0005-1098(89)90002-2.
- [19] Rumelhart, D., Hinton, G. & Williams, R. Learning representations by back-propagating errors. *Nature* **323**, 533–536 (1986). doi:10.1038/323533a0
- [20] Rakotondrasoa, H. M., Bucher, M. & Sinayskiy, I. Quantitative comparison of nearest neighbor search algorithms. *arXiv preprint arXiv: 2307.05235* (2023).
- [21] Parseh, M., Nybacka, M. & Asplund, F. Motion planning for autonomous vehicles with the inclusion of post-impact motions for minimising collision risk. *Veh. Syst. Dyn.* **61**, 1707–1733 (2023). doi:10.1080/00423114.2022.2088396.
- [22] Qasem, G. A. A., Abdullah, M. F., Farid, M. & Bakhuraisa, Y. A. Enhancing anti-lock braking system performance using fuzzy logic control under variable friction conditions. *Symmetry* **17**, 1692 (2025). doi:10.3390/sym17101692.
- [23] Labh, S. K., Khanal, R. & Dahal, C. Comparative analysis of performance of antilock braking system using PID and fuzzy controllers. *OODBODHAN* **8**, 135–142 (2025). doi:10.3126/oodbodhan.v8i1.81260.
- [24] Zuhilmi, I. M., Peeie, M. H., Asyraf, S. M., Sollehudin, I. M. & Ishak, I. M. Experimental study on the effect of emergency braking without anti-lock braking system on vehicle dynamics behaviour. *Int. J. Automot. Mech. Eng.* **17**, 7832–7841 (2020). doi:10.15282/ijame.17.2.2020.02.0583.
- [25] Wang, X., Li, W., Li, Z. & Li, L. Effect of braking torque on vehicle nonlinear dynamics. *Meccanica* **58**, 1267–1289 (2023). doi:10.1007/s11012-023-01683-0.
- [26] Eberhart, M., Plöchl, M., Unterreiner, M. & Edelmann, J. Insights into stability and control of the powerslide motion with variable drive torque distribution applied to a driver assistance system. *Veh. Syst. Dyn.* 1–21 (2025). doi:10.1080/00423114.2025.2457433.
- [27] Saraiev, O. & Gorb, Y. A mathematical model of the braking dynamics of a car. *SAE Tech. Pap.* (2018). doi:10.4271/2018-01-1893.
- [28] Rajesh, N., Zheng, Y. & Shyrokau, B. Comfort-oriented motion planning for automated vehicles using deep reinforcement learning. *IEEE Open J. Intell. Transp. Syst.*

4, 348–359 (2023). doi:10.1109/OJITS.2023.3275275.

[29] Lee, M. An analysis of the effects of artificial intelligence on electric vehicle technology innovation using patent data. *World Pat. Inf.* **63**, 102002 (2020). doi:10.1016/j.wpi.2020.102002.

[30] Stefanidou, A., Politi, E. & Dimitrakopoulos, G. Foundation models in autonomous driving: A review of current tasks and applications. *IEEE Open J. Intell. Transp. Syst.* **6**, 1522–1538 (2025). doi:10.1109/OJITS.2025.3633871.

[31] Basso, F., Pezoa, R., Varas, M. & Villalobos, M. A deep learning approach for real-time crash prediction using vehicle-by-vehicle data. *Accid. Anal. Prev.* **162**, 106409 (2021). doi:10.1016/j.aap.2021.106409.

[32] Teng, X., Xu, S., Guo, D., Guo, Y., Meng, W. & Zhang, X. DTTCNet: Time-to-collision estimation with autonomous emergency braking using multi-scale transformer network. *IEEE Trans. Mob. Comput.* **23**, 14903–14917 (2024). doi:10.1109/TMC.2024.3454122.

[33] Folkers, A., Rick, M. & Büskens, C. Controlling an autonomous vehicle with deep reinforcement learning. *Proc. IEEE Intell. Veh. Symp.* 2025–2031 (2019). doi:10.1109/IVS.2019.8814124.

[34] Chen, D., Gong, Y. & Yang, X. Advanced longitudinal control and collision avoidance for high-risk edge cases in autonomous driving. *arXiv preprint arXiv:2504.18931* (2025).

[35] Guerrero-Ibáñez, J., Zeadally, S. & Contreras-Castillo, J. Sensor technologies for intelligent transportation systems. *Sensors* **18**, 1212 (2018). doi:10.3390/s18041212.

[36] Chen, J., Xu, X. & Yang, J. Adaptive model predictive control for autonomous vehicle trajectory tracking. *Vehicles* **7**, 114 (2025). doi:10.3390/vehicles7040114.

[37] Kudarauskas, N. Analysis of emergency braking of a vehicle. *Transport* **22**(3), 154–159 (2007). doi:10.3846/16484142.2007.9638118.

[38] Sun, W. & Wang, S. Research on lateral acceleration of lane changing. in *Frontier Computing* (eds Hung, J., Yen, N. & Hui, L.) Lecture Notes in Electrical Engineering, vol. 542, 950–960 (Springer, Singapore, 2019). doi:10.1007/978-981-13-3648-5_120.

[39] Seo, M., Yoo, C., Park, S.-S. & Nam, K. Development of wheel pressure control algorithm for electronic stability control (ESC) system of commercial trucks. *Sensors* **18**, 2317 (2018). doi:10.3390/s18072317

[40] Han, K., Lee, B. & Choi, S. B. Development of an antilock brake system for electric vehicles without wheel slip and road friction information. *IEEE Trans. Veh. Technol.* **68**, 5506 – 5517 (2019). doi:/10.1109/TVT.2019.2911687

[41] Hotelling, H. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology* **24**(6), 417 – 441 (1933). doi:10.1037/h0071325.

- [42] Nair, V. & Hinton, G. E. Rectified Linear Units Improve Restricted Boltzmann Machines. *Proc. 27th Int. Conf. Mach. Learn.* **2010**, 807–814.
- [43] He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 770–778 (2016).
doi:10.1109/CVPR.2016.90.
- [44] Solis, C. U., Clempner, J. B. & Poznyak, A. S. Continuous-time gradient-like descent algorithm for constrained convex unknown functions: Penalty method application. *J. Comput. Appl. Math.* **355**, 268–282 (2019). doi:10.1016/j.cam.2019.01.023
- [45] Ning, Z., Iradukunda, H. N., Zhang, Q. & Zhu, T. Benchmarking machine learning: how fast can your algorithms go? *arXiv preprint arXiv 2101.03219* (2021).
- [46] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., & Duchesnay, E.. Nearest Neighbors — scikit-learn: Machine Learning in Python. (2011)
Retrieved from <https://scikit-learn.org/stable/modules/neighbors.html>

ARTICLE IN PRESS