Reliability Engineering and System Safety 230 (2023) 108950

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

journal homepage: www.elsevier.com/locate/ress

Machine learning approaches to estimate suspension parameters for performance degradation assessment using accurate dynamic simulations

Yongjun Pan [a,*], Yu Sun [a], Zhixiong Li [b,c], Paolo Gardoni [d]

[a] College of Mechanical and Vehicle Engineering, Chongqing University, Chongqing, 400044, China
[b] Faculty of Mechanical Engineering, Opole University of Technology, Opole, 45758, Poland
[c] Yonsei Frontier Lab, Yonsei University, Seoul, 03722, Republic of Korea
[d] Department of Civil and Environmental Engineering, University of Illinois at Urbana-Champaign, Champaign, IL 61820, United States of America

ARTICLE INFO

Keywords:
Parameter estimation
Machine learning
Vehicle suspension
Stiffness and damping coefficients
Dynamic simulation

ABSTRACT

The suspension is one of the most vital systems in a vehicle. Its performance degrades over time due to road conditions. The suspension parameters of a moving vehicle are difficult or sometimes impossible to measure within the desired level of accuracy due to high costs and other associated impracticalities. In this work, we comprehensively investigate various machine learning (ML) methods to estimate the suspension parameters for assessing performance degradation. These methods include particle swarm optimization backward propagation, radial basis function neural network, generalized regression neural network, deep belief network, wavelet neural network, Elman neural network, extreme learning machine, and fuzzy neural network. During the training process, the vehicle states, calculated using a semi-recursive multibody model, are used as the inputs to predict the stiffness and damping coefficients of the suspensions. The semi-recursive multibody model considers the dynamic properties of all the components, which enables accurate vehicle states and characteristics. In addition, we compare the performance of the ML methods by using the reference data (multibody model data). The results show that the ML approaches can estimate accurate stiffness and damping coefficients in real-time.

1. Introduction

The suspension elastically connects the chassis frame with the wheels to absorb vibrations and impacts from the ground. It directly influences the vehicle riding and handling performance. When a vehicle travels on severe road profiles, the suspension performance degrades over time, affecting the vehicle system's reliability. If the suspension performance is seriously impaired, the vehicle's safety is considerably affected. Considering heavy-duty truck as an example, the vehicle is prone to roll on bumpy roads or under other worse road conditions after the suspension performance decreases. In severe cases, the vehicle will roll over [1,2]. Therefore, the suspension performance affects the reliability and driving safety of the vehicle. The parameters estimation of the vehicle suspension is of great significance for measuring the vehicle system's reliability and the driving safety.

Various critical parameters affect a vehicle's performance, such as suspension's stiffness and damping coefficients. The suspension parameters vary due to road roughness and mechanical fatigue. The real-time or online parameter estimation of suspension has attracted a considerable attention [3,4]. The conventional estimation methods are usually signal- or model-based. The signal-based methods considerably rely on the preciseness of the relevant devices, thus suffering from uncertainties to some extent [5]. Generally, it is difficult or sometimes impossible to measure moving vehicle's suspension parameters within the desired accuracy level due to high costs and other associated impracticalities [6].

In contrast, computation precision and efficiency are mutually exclusive in model-based methods. Always expensive and time-consuming, accurate parameter estimation via complex algorithms. In literature, accurate parameter estimation typically requires either a high computational power or a low sampling rate [7,8]. In this regard, the machine learning (ML) approaches present a novel and practical avenue for estimating the suspension parameters. The ML-based methods are capable of evaluating extremely nonlinear relationships between the vehicle states and suspension parameters. Compared to the purely signal- and model-based estimation methods, the ML methods reduce the influence of uncertainties or disturbances and avoid developing accurate and complicated physical models [9,10].

Recently, the ML techniques have achieved considerable progress in complex engineering problems, e.g., parameter estimation, intelligent

[*] Corresponding author.
E-mail addresses: yongjun.pan@cqu.edu.cn (Y. Pan), z.li@po.edu.pl (Z. Li), gardoni@illinois.edu (P. Gardoni).

https://doi.org/10.1016/j.ress.2022.108950
Received 22 April 2022; Received in revised form 26 October 2022; Accepted 30 October 2022
Available online 4 November 2022
0951-8320/© 2022 Elsevier Ltd. All rights reserved.

control, and reliability and safety analysis. Mazumder et al. investigated the feasibility of using ML algorithms as an alternative to the physical-based approaches in pipeline failure prediction [11]. Lu et al. developed a methodological framework that integrates the ML techniques with path analysis to quantify the behavioural pathways in bicycle–motor vehicle crashes [12]. Xu et al. extended the domain of ML applications to the helicopter accidents, and demonstrated the type of analysis they enable and the novel insights that can be obtained from the best-in-class classifier to support safety improvement [13]. The most well-known ML methods include particle swarm optimization backward propagation (PSO-BP), radial basis function neural network (RBFNN), generalized regression neural network (GRNN), deep belief network (DBN), wavelet neural network (WNN), Elman neural network (ElmanNN), extreme learning machine (ELM), and fuzzy neural network (FNN) [14–16]. Drehmer et al. used PSO and sequential quadratic programming algorithms to optimize the suspension parameters under different road profiles and vehicle-speed conditions [17]. Xie et al. proposed a method for forecasting the short-term power load that combines the ElmanNN and PSO algorithms [18]. Lopez-Sánchez et al. designed a robust controller using the GRNN algorithm to address the issue of trajectory tracking and provide effective disturbances rejection [19]. Lin et al. presented an intelligent control system based on RWNN for tracking the desired trajectory of an unmanned aerial vehicle [20]. In order to enhance vehicle handling and riding performances, Huang et al. proposed an intelligent robust control method using an improved ELM algorithm for simultaneously improving tracking accuracy and chattering suppression [21]. Taghavifar et al. designed a sliding mode interval fuzzy type-2 neural network controller to suppress the vibrations generated from a rough terrain imposed to the nonlinear vehicle suspensions [22]. Zhu et al. developed a DBN information fusion fault diagnosis method to identify the continuously changing and uncertain thruster fault patterns of a deep-sea human-occupied vehicle [23]. Jiao et al. developed a novel fault monitoring framework based on Gap-DBN and support vector data description for prognostics under multiple fault modes [24]. Jing et al. introduced an adaptive RBFNN algorithm to approximate the performance function and conduct the structural reliability analysis [25].

The literature review indicates that the ML applications of vehicle suspension performance and related parameter estimation problems are very challenging and remain open. In this work, we explore the estimation capabilities of different ML methods in estimating the suspension parameters for performance degradation assessment. We use several relevant vehicle states as the inputs, including the position of the vehicle's centroid, the vertical velocity, the vertical acceleration, and the pitch angle. The models' outputs include suspension parameters such as stiffness and damping. We employ the following machine learning (ML) techniques: PSO-BP, RBFNN, GRNN, DBN, WNN, ElmanNN, ELM, and FNN. These methods precisely describe the complex nonlinear relationship between input parameters and output responses. Furthermore, they have been widely used in parameter identification and estimation. These models are used for ML modelling based on the same data samples, which are obtained from a semi-recursive vehicle multibody model. Please note that unlike other vehicle models, such as simplified models, surrogate models, or nonlinear dynamics models, the multibody method considers and models the dynamic properties of all system components. The multibody model can accurately reconstruct the time-domain signals of vehicle suspensions, thus providing an appealing alternative to the signal-based methods. This enables accurate modelling of dynamics and data acquisition [26–28]. The highlights of this work are as follows:

- The vehicle states obtained from accurate multibody dynamic simulations are used for estimating the suspension parameters.
- A series of ML methods is developed to assess the suspension performance degradation using vehicle states as the inputs.
- We explore the estimation capabilities of different ML methods in front and rear suspension parameters.

The rest of this work is organized as follows. Section 2, uses an efficient semi-recursive multibody formulation method to develop a vehicle dynamics model. The dynamic simulations are performed, and the obtained vehicle states are used for training the ML models. In Section 3, the ML methods, including PSO-BP, RBFNN, GRNN, DBN, WNN, ElmanNN, ELM, and FNN, are described mathematically in terms of parameter estimation. In Section 4, the results are analyzed in detail, and the model accuracy is investigated and quantified. The capacity of different ML methods to estimate suspension parameters is studied, and appropriate ML models are suggested for this suspension example. In Section 5, we conclude this work.

2. Vehicle multibody model and dynamic simulation

2.1. Multibody modelling of vehicle and suspensions

In this section, we use a semi-recursive multibody method for accurately modelling vehicle dynamics. It is noteworthy that the method based on the double-velocity transformation is highly efficient [29]. The first velocity transformation uses relative (joint) coordinates $\mathbf{z}$ to describe the Cartesian coordinates and dynamic equations of the system. As a result, it is possible to achieve dimensionality reduction of the multibody model and higher computational efficiency.

The first velocity transformation can be achieved easily for an open-loop system. The three-dimensional configuration of the open-loop vehicle system is described by the relative (joint) coordinates. Please note that the universal and spherical joints are broken down into several revolute and prismatic joints with auxiliary massless bodies to fully use recursive kinematics and dynamics. The equations of motion for an open-loop vehicle system are mathematically expressed as follows [30]:

$$\mathbf{R}_{\mathrm{d}}^{\mathrm{T}}\bar{\mathbf{M}}^{\Sigma}\mathbf{R}_{\mathrm{d}}\ddot{\mathbf{z}} = \mathbf{R}_{\mathrm{d}}^{\mathrm{T}}(\bar{\mathbf{Q}}^{\Sigma} - \bar{\mathbf{M}}^{\Sigma}\dot{\mathbf{R}}_{\mathrm{d}}\dot{\mathbf{z}}) = \mathbf{R}_{\mathrm{d}}^{\mathrm{T}}\left(\bar{\mathbf{Q}}^{\Sigma} - \bar{\mathbf{P}}^{\Sigma}\right) \tag{1}$$

where, $\mathbf{R}_{\mathrm{d}}$ represents the first velocity transformation matrix, $\bar{\mathbf{M}}^{\Sigma}$, $\bar{\mathbf{P}}^{\Sigma}$, and $\bar{\mathbf{Q}}^{\Sigma}$ represent the accumulated mass matrix, velocity-dependent inertial forces, and external forces. $\dot{\mathbf{z}}$ and $\ddot{\mathbf{z}}$ represent the relative velocities and accelerations of the vehicle system, respectively.

A closed-loop vehicle multibody system can be transformed into an open-loop structure by cutting kinematic joints or eliminating slender rods. The loop-closure-constrained equations are generated by using joint-cut or rod-removal techniques [31,32]. When applying the rod-removal techniques, the rod-related inertial and external forces, such as the spring force, damping force, and gravity force, should be calculated and accumulated in the recursive system. The second velocity transformation is performed using the constrained equations and their corresponding time derivatives. In this process, the relative (joint) coordinates are expressed by independent relative coordinates [33]. Please note that the length of the independent-relative-coordinate vector is equal to the number of degrees of freedom of the closed-loop vehicle system. This shows that the dimensions of the dynamic equations can be further reduced. By performing the double-velocity transformation, the equations of motion of a closed-loop vehicle multibody system are expressed as follows [27]:

$$\mathbf{R}_{\mathbf{z}}^{\mathrm{T}}\mathbf{R}_{\mathrm{d}}^{\mathrm{T}}\bar{\mathbf{M}}^{\Sigma}\mathbf{R}_{\mathrm{d}}\mathbf{R}_{\mathbf{z}}\ddot{\mathbf{z}}^{\mathrm{i}} = \mathbf{R}_{\mathbf{z}}^{\mathrm{T}}\mathbf{R}_{\mathrm{d}}^{\mathrm{T}}\left[\bar{\mathbf{Q}}^{\Sigma} - \mathbf{T}^{\mathrm{T}}\bar{\mathbf{M}}\frac{d\left(\mathbf{T}\mathbf{R}_{\mathrm{d}}\mathbf{R}_{\mathbf{z}}\right)}{dt}\dot{\mathbf{z}}^{\mathrm{i}}\right] \tag{2}$$

where, $\mathbf{T}$ represents the path matrix, $\bar{\mathbf{M}}$ represents the composite mass matrix, and $\mathbf{R}_{\mathbf{z}}$ represents the second velocity transformation matrix. $\dot{\mathbf{z}}^{\mathrm{i}}$ and $\ddot{\mathbf{z}}^{\mathrm{i}}$ represent the independent relative velocities and accelerations of the closed-loop vehicle system, respectively.

The expression presented in Eq. (2) can be rewritten in a compact form as follows:

$$\hat{\mathbf{M}}\ddot{\mathbf{z}}^{\mathrm{i}} = \hat{\mathbf{F}} \tag{3}$$

where, $\hat{\mathbf{M}}$ represents the generalized mass matrix of the vehicle system, and $\hat{\mathbf{F}}$ denotes the generalized inertial and external forces. To investigate the response of suspension towards vibrations, when a vehicle

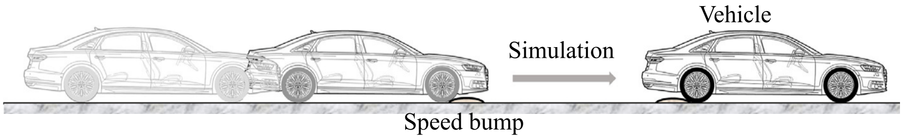


Fig. 1. The dynamic simulation on the bumpy road.

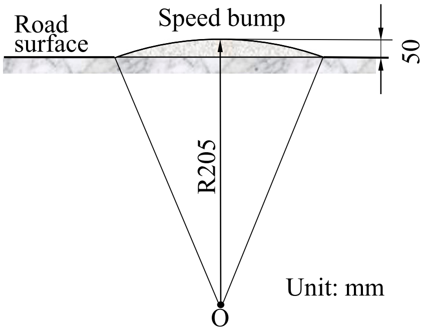


Fig. 2. The profile of the speed bump.

passes through a bumpy road, we expand the expression in Eq. (3) as follows. These matrices represent the motion of suspensions, the chassis frame, and the wheels, respectively:

$$\begin{bmatrix} \hat{\mathbf{M}}_{ch} & \hat{\mathbf{M}}_{cs} & \hat{\mathbf{M}}_{cw} \\ \hat{\mathbf{M}}_{cs}^{\mathrm{T}} & \hat{\mathbf{M}}_{su} & \hat{\mathbf{M}}_{sw} \\ \hat{\mathbf{M}}_{cw}^{\mathrm{T}} & \hat{\mathbf{M}}_{sw}^{\mathrm{T}} & \hat{\mathbf{M}}_{wh} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{z}}_{ch}^{\mathrm{i}} \\ \ddot{\mathbf{z}}_{su}^{\mathrm{i}} \\ \ddot{\mathbf{z}}_{wh}^{\mathrm{i}} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{F}}_{ch} \\ \hat{\mathbf{F}}_{su} \\ \hat{\mathbf{F}}_{wh} \end{bmatrix} \tag{4}$$

where, $\hat{\mathbf{M}}_{ch} \in \mathbb{R}^{6\times 6}$, $\hat{\mathbf{M}}_{su} \in \mathbb{R}^{4\times 4}$, and $\hat{\mathbf{M}}_{wh} \in \mathbb{R}^{4\times 4}$ represent the substructures of the generalized mass matrix corresponding to chassis, suspensions, and wheels, respectively. $\hat{\mathbf{M}}_{cs} \in \mathbb{R}^{6\times 4}$, $\hat{\mathbf{M}}_{cw} \in \mathbb{R}^{6\times 4}$, and $\hat{\mathbf{M}}_{sw} \in \mathbb{R}^{4\times 4}$ represent the substructures of the generalized mass matrix corresponding to the couplings of chassis, suspensions, and wheels, respectively. $\ddot{\mathbf{z}}_{ch} \in \mathbb{R}^{6}$, $\ddot{\mathbf{z}}_{su} \in \mathbb{R}^{4}$, and $\ddot{\mathbf{z}}_{wh} \in \mathbb{R}^{4}$, represent the independent relative accelerations of chassis, suspensions, and wheels, respectively. $\hat{\mathbf{F}}_{ch} \in \mathbb{R}^{6}$, $\hat{\mathbf{F}}_{su} \in \mathbb{R}^{4}$, and $\hat{\mathbf{F}}_{wh} \in \mathbb{R}^{4}$, represent the substructures of the generalized inertial and external forces corresponding to chassis, suspensions, and wheels, respectively.

The vertical accelerations of the suspensions and the chassis on a bumpy road are computed as follows:

$$\ddot{\mathbf{z}}_{ch}^{\mathrm{i}} = \hat{\mathbf{M}}_{ch} \backslash \left(\hat{\mathbf{F}}_{ch} - \hat{\mathbf{M}}_{cs} \ddot{\mathbf{z}}_{su}^{\mathrm{i}} - \hat{\mathbf{M}}_{cw} \ddot{\mathbf{z}}_{wh}^{\mathrm{i}} \right) \tag{5}$$

$$\ddot{\mathbf{z}}_{su}^{\mathrm{i}} = \hat{\mathbf{M}}_{su} \backslash \left(\hat{\mathbf{F}}_{su} - \hat{\mathbf{M}}_{cs}^{\mathrm{T}} \ddot{\mathbf{z}}_{ch}^{\mathrm{i}} - \hat{\mathbf{M}}_{sw} \ddot{\mathbf{z}}_{wh}^{\mathrm{i}} \right). \tag{6}$$

where, $\ddot{\mathbf{z}}_{su}^{\mathrm{i}}$ contains the vertical accelerations of the suspensions, and $\ddot{\mathbf{z}}_{ch}^{\mathrm{i}}$ contains the accelerations along the X-axis, Y-axis, and Z-axis, and pitch, roll, and yaw accelerations of the chassis. These vehicle responses (states) are obtained based on accurate multibody dynamic simulations and are used for ML modelling to estimate the suspension parameters.

The expressions presented in Eqs. (2) to (6) use only a small set of independent relative accelerations $\ddot{\mathbf{z}}^{\mathrm{i}}$. Thus this result in higher computational efficiency compared to various commercial software packages, such as ADAMS and RecurDyn. Furthermore, the ordinary differential form enables the users to employ high-order numerical integrators to perform stable and accurate simulations, particularly in longer simulations. The Runge–Kutta and the Adams–Bashforth–Moulton integrators have been widely used for the real-time simulation of full vehicles [34]. The proposed vehicle multibody modelling method considers and models the nonlinear dynamic properties of a vehicle system's components, including the chassis, suspensions, and tires. Moreover, it is suitable for accurately calculating vehicle dynamics in various driving conditions and road environments.

2.2. Dynamic simulation and data acquiring

We consider a sedan vehicle consisting of MacPherson strut suspensions in the front axle and multi-link suspensions in the rear axle. The general parameters of the vehicle system and the necessary parameters of the vehicle components are presented in Tables 1 and 2, respectively. The dynamic tire forces are calculated using the Pacejka tire model and imposed on the vehicle's multibody system as external forces [35]. The vehicle multibody model is developed based on the semi-recursive multibody method. In order to obtain the required vehicle states, including pitch angle, the position of the vehicle's centroid, vertical velocity, and acceleration, we perform dynamic simulations on a predefined bumpy road. The performance degradation of the suspension on the bumpy road is more severe than on a flat road. The vehicle states and suspension parameters present very obvious nonlinear characteristics. The vehicle state obtained in this case is more conducive for developing the ML models for suspension parameter estimation.

The vehicle's movement is presented in Fig. 1, and the profile of the bumpy road is illustrated in Fig. 2. The initial speed of the vehicle is 20 m/s, and the driving torque on the front wheels is 500 N m. The height of the speed bump usually ranges between 30 mm and 60 mm, and the width is usually between 300 mm and 600 mm. In this work, we consider the height and width of speed bump to be 50 mm and 300 mm, respectively, as they depict better speed-controlling effects. The suspensions absorb shocks and vibrations when the vehicle travels on a bumpy road. The time step is 1 ms, and the simulation time is 3 s. Thus, 3000 integrations are used for the numerical calculation and simulation. This accurate simulation can contain effective shock-vibration response data.

The responses of vehicle corresponding to different suspension parameters are calculated using the multibody dynamics model. A schematic diagram depicting the degrees of freedom of the vehicle model is presented in Fig. 3. The vehicle multibody system has 14 degrees of freedom, including the rotations of the four wheels, the vertical movements of the four suspensions, the movements of the chassis along the X-, Y-, and Z-axis, and its pitch, roll, and yaw angles. The performance degradation of the front and rear suspension is reflected by the changes in the suspension parameters, affecting the shock absorber's spring and damping forces. As part of the external forces on the vehicle, the spring and damping forces are calculated and added to the right-hand side of Eq. (2), thus affecting the entire vehicle states. In addition, the alternations of external forces on the front and rear suspensions due to the randomness of the road surface heavily affect the vehicle states. Therefore, the relationship between the suspension parameters and vehicle states is highly nonlinear. We aim to model the complex nonlinear relationship by using ML methods [36]. Afterwards, this modelled relationship is used to estimate the stiffness and damping coefficients of the suspensions by using realistic vehicle states.

In order to acquire the data samples for subsequent ML-based modelling, we decrease the stiffness and damping coefficients of the front and rear suspensions to imitate performance degradation. The stiffness of the front suspensions is reduced to 35,500 N/m from 40,000 N/m. The stiffness of the rear suspensions is reduced to 30,500 N/m from 35,000 N/m. The front and rear suspensions' damping coefficient is reduced to 1800 N s/m from 1980 N s/m. In addition, the changing intervals of the stiffness and damping coefficients are −500 N/m and

Table 1
Important parameters of the vehicle system.

Parameter	Value	Parameter	Value
Multibody system mass	1210 kg	Tire rolling radius	0.4673 m
Distance from centroid to front axle	2.0791 m	Centroid height	0.5373 m
Distance from centroid to rear axle	2.0791 m	Wheelbase	2.8 m
Vehicle torque	500 N m	Vehicle speed	10 m/s 20 m/s 30 m/s
Stiffness of front suspensions	40000 N/m	Damping of front suspensions	1980 N s/m
Stiffness of rear suspensions	35000 N/m	Damping of rear suspensions	1980 N s/m
Pitching moment of inertia	775 kg m^2	Sprung mass	1068 kg
Unsprung mass of the front axle	66 kg	Unsprung mass of the rear axle	76 kg

Table 2
Parameters of some pivotal components of the vehicle model.

Component		Mass (kg)	Centroid coordinates (m)	Moment of inertia (kg m^2)
Chassis		1000	(0, 0, 0.5373)	diag(300, 600, 600)
Steering knuckle	Left	10	(2.1286, 1.1518, 0.1808)	diag(0.01, 0.01, 0.01)
	Right	10	(2.1286, −1.1518, 0.4808)	diag(0.01, 0.01, 0.01)
Suspension	Dampener_L	2	(2.1066, 1.1008, 1.0257)	diag(0.0001, 0.0001, 0.0001)
	Dampener_R	2	(2.1066, −1.1008, 1.0257)	diag(0.0001, 0.0001, 0.0001)
	Swing arm_L	2.5	(2.1648, 0.9299, 0.3373)	diag(0.02, 0.02, 0.02)
	Swing arm_R	2.5	(2.1648, −0.9299, 0.3373)	diag(0.02, 0.02, 0.02)
Front wheel	Left	23	(2.0791, 1.1967, 0.4673)	diag(0.4, 0.8, 0.4)
	Right	23	(2.0791, −1.1967, 0.4673)	diag(0.4, 0.8, 0.4)
Rear wheel	Left	23	(−2.0791, 1.1967, 0.4673)	diag(0.4, 0.8, 0.4)
	Right	23	(−2.0791, −1.1967, 0.4673)	diag(0.4, 0.8, 0.4)

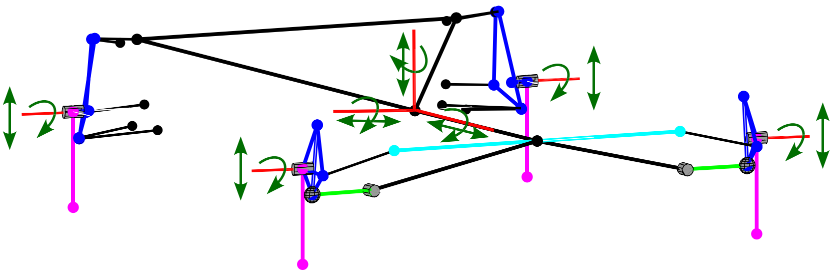


Fig. 3. The schematic diagram of vehicle degrees of freedom.

−20 N s/m. Finally, we collect 10,000 data samples by performing dynamic simulations on the bumpy road.

We use the layered fixed ratio method to select 1000 data samples from the 10,000 data samples for ML modelling. In addition to estimating the suspension parameters subject to performance degradation, we also analyze the capabilities of different ML methods under a small amount of data. Among the 1000 data samples, 70% of samples are used as the training set, 15% of samples are used as the verification set, and 15% of samples are used as the test set. These subsets are formed by random sampling. Please note that normalizing data samples are necessary as effective normalization improves the model accuracy and convergence speed. In addition, it also enables the models to avoid gradient explosions during the training process. In this work, we use the min–max normalization method, also known as the dispersion standardization method. It uses a linear transformation and maps the original value between 0 and 1. The min–max normalization method is mathematically expressed as follows:

$$x_i^* = \frac{x_i - min(x_i)}{max(x_i) - min(x_i)} \tag{7}$$

where, x_i^* denotes the normalized value, x_i denotes the actual value of the ith initial array, and $min()$ and $max()$ refer to the minimum and maximum values in the sample array, respectively.

3. ML modelling for estimating suspension parameters

Recently, the ML methods have been widely used in parameter estimation and fault diagnosis of vehicles. With the availability of abundant data and increasing computing power, the performance of ML methods has improved significantly. In this work, the PSO-BP, RBFNN, GRNN, DBN, WNN, ElmanNN, ELM, and FNN are used to estimate the suspension parameters. The inputs of each ML model is set as $\mathbf{X} = [V\_\theta \quad V\_p \quad V\_v \quad V\_a]^{\mathrm{T}}$, and outputs $\mathbf{Y} = [F\_k \quad F\_c \quad R\_k \quad R\_c]^{\mathrm{T}}$. They represent four vehicle states and four suspension parameters, respectively. In the end, we also present the general ML modelling procedure.

- **PSO-BP**. The hybrid algorithm consists of particle swarm optimization (PSO) and backward propagation neural network (BPNN). The PSO algorithm is initialized from a group of random particles. These particles continuously update themselves based on the individual and global extremum in an iterative manner to reach a global optimal solution. The BPNN is a multi-layer feedforward neural network, which is trained based on the back-propagation algorithm. The layers of this model consist of an input layer, several hidden layers, and an output layer. Each layer contains multiple neurons. The output of a single neuron is mathematically expressed as:

$$y_j = f\left(\sum_{i=1}^{n} \omega_{jk} \times x_k + b\right) \tag{8}$$

where, x_k and y_j represent the input and output activation, respectively. In the first hidden layer of the networks, x_k represents the parameters of the input layer. In this vehicle example, k equals 4, and x_k contains vehicle states V_θ, V_p, V_v, and V_a.

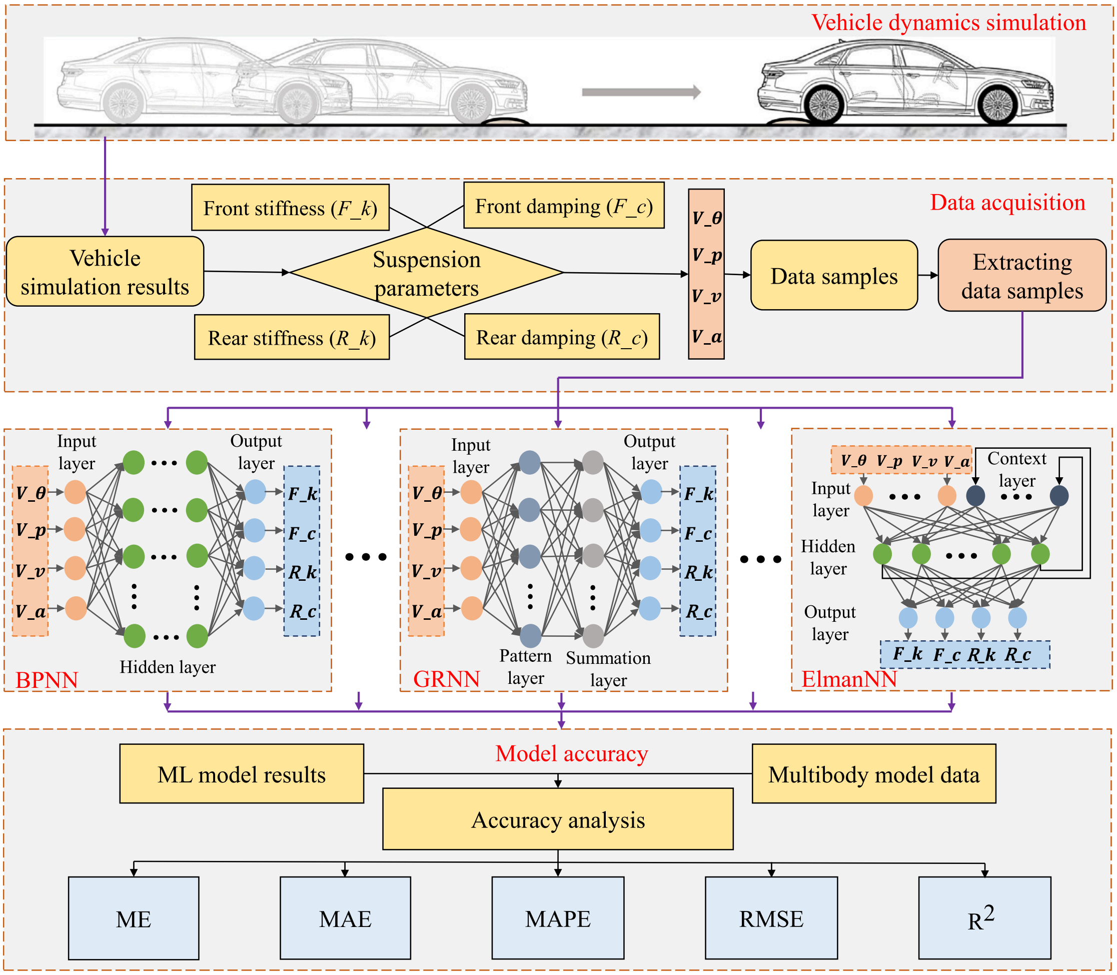


Fig. 4. ML modelling procedure for parameter estimation of vehicle suspensions.

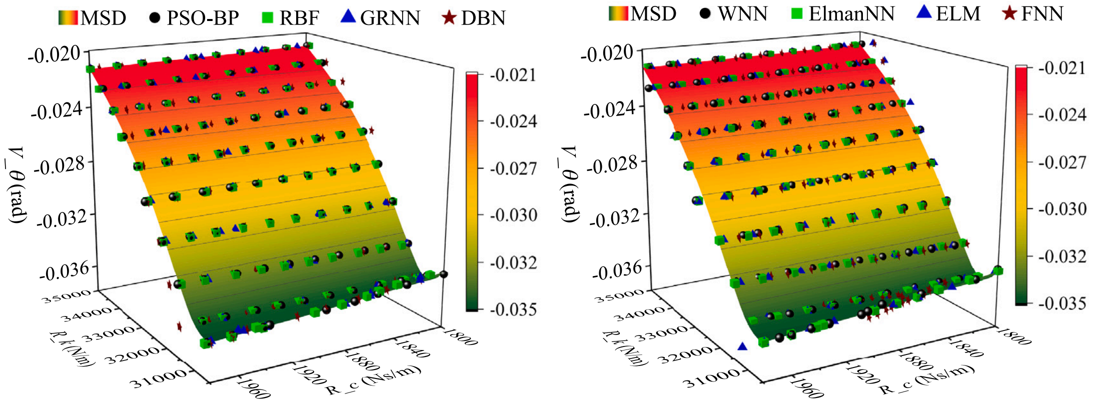


Fig. 5. The ML models and multibody model results: pitch angle.

Vector y_j represents the output of the first hidden layer. The output of the last hidden layer of the network ($\hat{\mathbf{Y}}$) represents the predicted suspension parameters ($\mathbf{Y}$). ω_{jk} and b represent the weight and the bias, respectively. $f(\cdot)$ represents the activation function. The PSO-BP algorithm is an attractive choice as it combines the strong local searching ability of the BP algorithm and the strong global searching ability of the PSO algorithm [37].

- **RBFNN**. This algorithm obtains a solution by considering the problem as a curve-fitting problem in a high dimensional space [38]. It tends to find a radial basis function to provide the best fit for the training data. The most common radial basis function is the Gaussian radial basis function, also referred to as the Gaussian kernel function. It is mathematically expressed as:

$$\varphi\left(x_i, c_j\right) = e^{-\frac{\|x_i - c_j\|^2}{2\sigma^2}} \tag{9}$$

where, c_j represents the center point of the jth neuron, σ represents the width of the Gaussian kernel, and $\|x_i - c_j\|$ represents the Euclidean distance from the sample to the center point. The parameter x_i denotes the input of the network, which is also the input of the first hidden layer. It represents the vehicle states ($\mathbf{X}$)

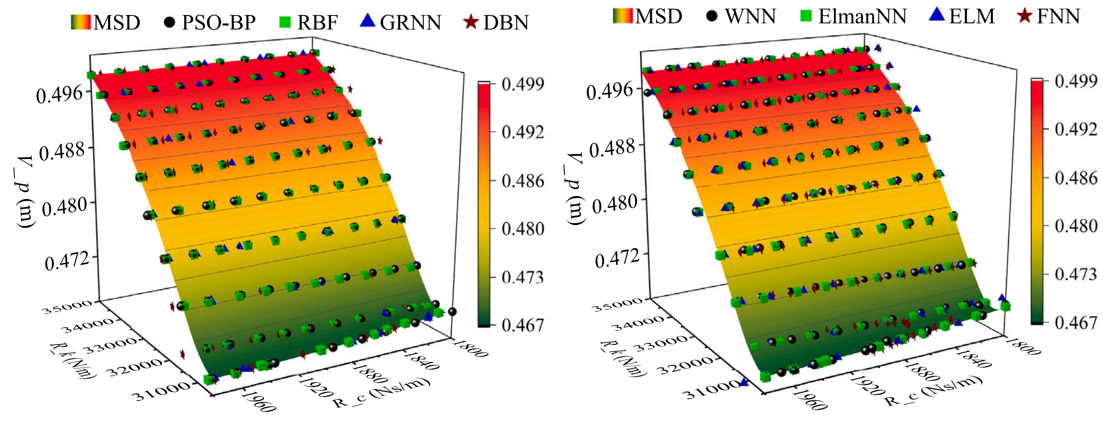


Fig. 6. The ML models and multibody model results: position of the vehicle's centroid.

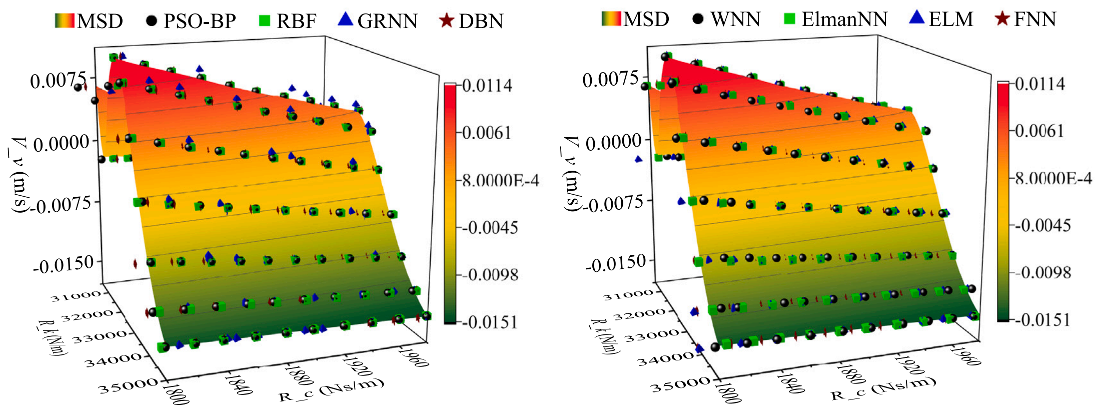


Fig. 7. The ML models and multibody model results: vertical velocity.

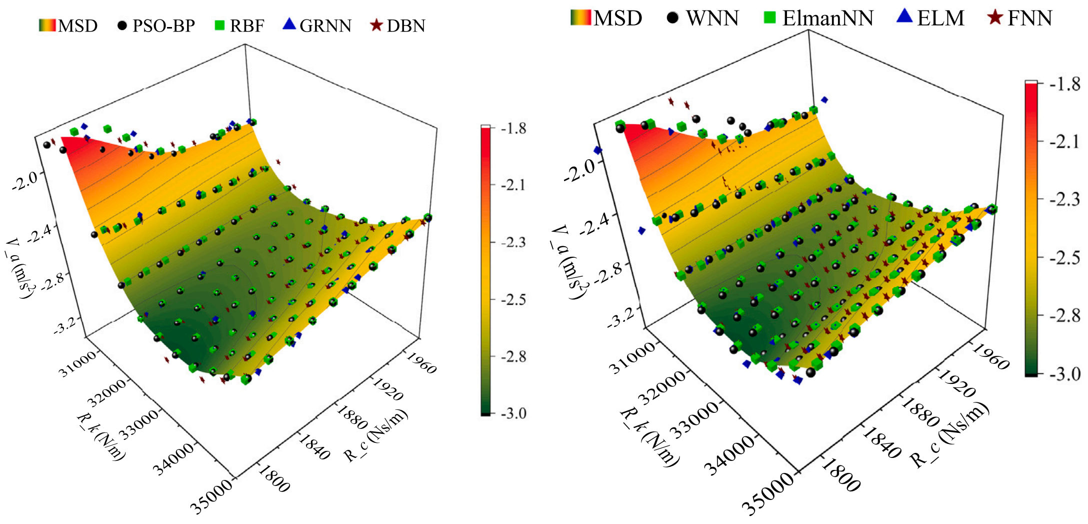
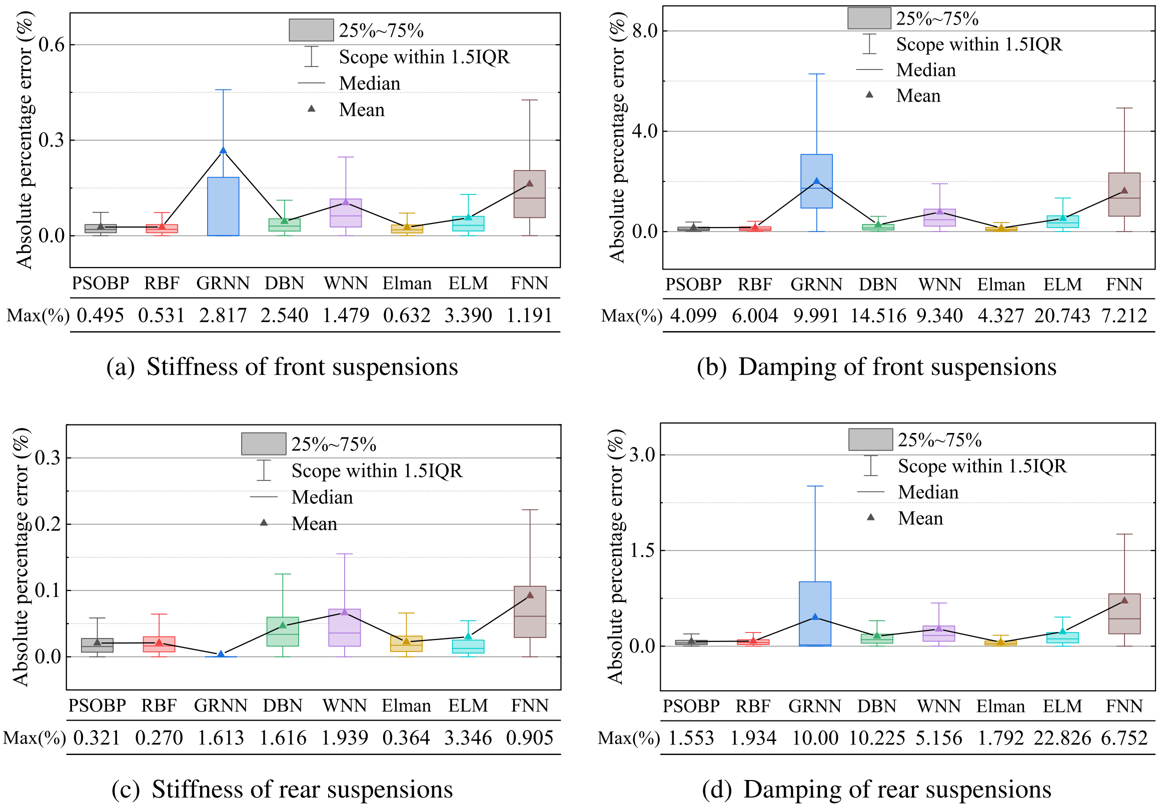


Fig. 8. The ML models and multibody model results: vertical acceleration.



	PSOBP	RBF	GRNN	DBN	WNN	Elman	ELM	FNN
Max(%)	0.495	0.531	2.817	2.540	1.479	0.632	3.390	1.191

(a) Stiffness of front suspensions

	PSOBP	RBF	GRNN	DBN	WNN	Elman	ELM	FNN
Max(%)	4.099	6.004	9.991	14.516	9.340	4.327	20.743	7.212

(b) Damping of front suspensions

	PSOBP	RBF	GRNN	DBN	WNN	Elman	ELM	FNN
Max(%)	0.321	0.270	1.613	1.616	1.939	0.364	3.346	0.905

(c) Stiffness of rear suspensions

	PSOBP	RBF	GRNN	DBN	WNN	Elman	ELM	FNN
Max(%)	1.553	1.934	10.00	10.225	5.156	1.792	22.826	6.752

(d) Damping of rear suspensions

Fig. 9. Box plots of percentage absolute error: suspension parameters.

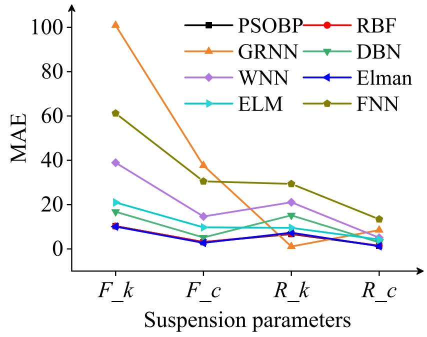


Fig. 10. MAE comparison of all ML models.

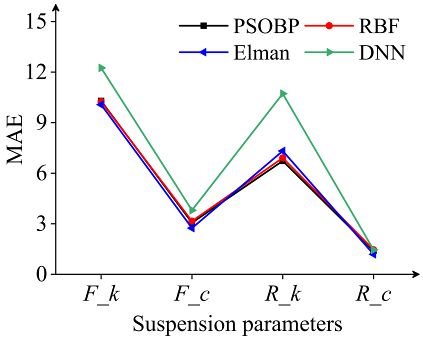


Fig. 11. MAE comparison of the best three ML models and DNN model.

in this vehicle example. The RBFNN algorithm is mathematically expressed as:

$$f(x) = \sum_{j=1}^{n} \omega_j \cdot \varphi\left(x, c_j\right) \tag{10}$$

where, ω_j represents the weight of the jth neuron. The mean square error is used for computing the error. The BP algorithm is used to back-propagate the error, and the gradient descent method is used to obtain the direction of RBFNN parameter optimization for minimizing the error function.

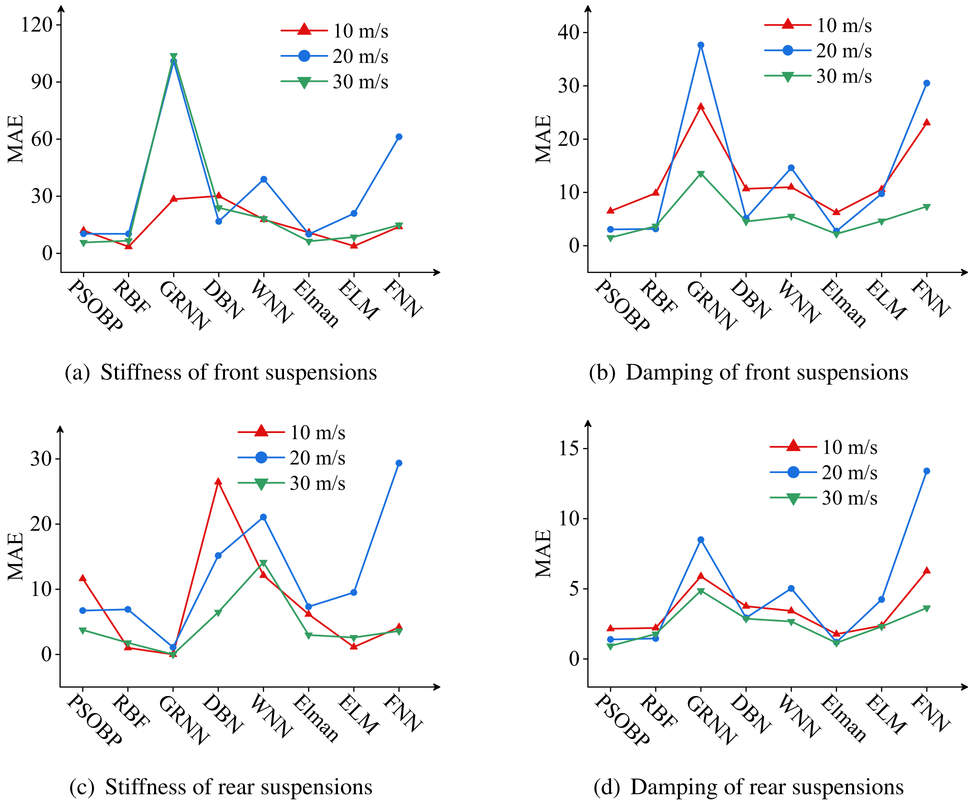
- **GRNN**. The GRNN has a similar structure to the RBFNN algorithm. It consists of four layers, including the input layer, mode layer, summation layer, and output layer. The GRNN algorithm is based on nonlinear regression.

$$\hat{Y} = E(y/X) = \frac{\int_{-\infty}^{\infty} y f(X, y)\, dy}{\int_{-\infty}^{\infty} f(X, y)\, dy} \tag{11}$$

where, X represents the observed value of the variable x that is the element of $\mathbf{X}$, and y represents the scalar random variables, i.e., the element of $\mathbf{Y}$. $f(X, y)$ represents joint continuous probability density function, and $\hat{Y}$ represents the predicted output under the input X. Given $f(X, y)$, the output of the network is computed as follows:

$$\hat{Y}(X) = \frac{\sum_{i=1}^{n} Y_i \exp\left[-\frac{(X-X_i)^{\mathrm{T}}(X-X_i)}{2\sigma^2}\right]}{\sum_{i=1}^{n} \exp\left[-\frac{(X-X_i)^{\mathrm{T}}(X-X_i)}{2\sigma^2}\right]} \tag{12}$$

where, σ represents the standard deviation of the Gaussian kernel function, also known as the smoothing factor. Y_i represents the



(a) Stiffness of front suspensions

(b) Damping of front suspensions

(c) Stiffness of rear suspensions

(d) Damping of rear suspensions

Fig. 12. MAE of the 8 ML models in 10 m/s, 20 m/s, and 30 m/s vehicle speeds.

sample observation of y, and n represents the total number of sample observations.

- **DBN**. The DBN algorithm contains a logistic regression layer and several unsupervised learning parts [39]. The logistic regression layer is used to perform the predictions. The unsupervised learning parts are developed using the restricted Boltzmann machine (RBM) method. The RBM model based on energy is defined as:

$$E(v,h) = -\sum_{i=1}^{n}\sum_{j=1}^{m}\omega_{ij}h_i v_j - \sum_{j=1}^{m} b_j v_j - \sum_{i=1}^{n} c_i h_i \tag{13}$$

where, v and h represent the visible layer and the hidden layer, respectively, and ω_{ij} represents the connection weight between the explicit layer unit i and the hidden layer unit j. The minimum value of the connection weight is defined by employing an effective energy function. Then, the probabilities of the visible and hidden layers are determined. The parameters of the RBM can be set to the likelihood estimation parameters of the joint probability of v and h. The likelihood function of the observation data is maximized to acquire the parameters of RBMs.

- **WNN**. The WNN shares a similar structure with the RBFNN algorithm, except that the wavelet basis function replaces the radial basis function. For the input x_i, i.e., vehicle states $\mathbf{X}$, the output of the hidden layer is calculated as follows:

$$h(j) = h_j\left(\frac{\sum_{i=1}^{k}\omega_{ij}x_i - b_j}{a_j}\right) \qquad j = 1,2,\ldots,l \tag{14}$$

where, $h(j)$ represents the output value of the jth node in the hidden layer, ω_{ij} represents the connection weight between the input layer and the hidden layer, and h_j represents the wavelet basis function. b_j and a_j represent the translation factor and the expansion factor of the h_j, respectively. Finally, the output layer is expressed as:

$$y(k) = \sum_{i=1}^{l}\omega_{ik}h(i) \qquad k = 1,2,\ldots,m \tag{15}$$

where, ω_{ik} represents the weight from the hidden layer to the output layer, $h(i)$ represents the output of the ith hidden layer node, and l and m represent the number of hidden layer nodes and output layer nodes, respectively.

- **ElmanNN**. The ElmanNN is a feedforward recurrent network. It consists of four layers: the input layer, hidden layer, context layer, and output layer. The connections between the input layer, the hidden layer, and the output layer are regarded as feedforward networks, which are analogous to a traditional multilayer neural network. The neurons in the context layer are regarded as the memory units. They provide the feedback internally, thus significantly improving the dynamic information procession. The input and the output of the network are $\mathbf{X}$ and $\mathbf{Y}$, respectively. The input and output vectors' dimensions are the same, which equals 4 in this vehicle example. The external input, context, and output weight matrices are defined as $W^{h,i}$, $W^{h,c}$, and $W^{o,h}$, respectively. Due to the presence of the context layer, the complete input vector x_l contains the elements of $\mathbf{X}$ and the hidden layer output vector. The complete input weight matrix is defined as $W^h = \left[W^{h,i}\ W^{h,c}\right]$. Thus, the corresponding output is expressed as:

$$h_j = f\left(a_j^h\right) = \frac{1}{1+\exp\left(-a_j^h\right)} \qquad j = 1,2,\ldots,m \tag{16}$$

$$a_j^h = \sum_{l=1}^{k} W_{jl}^h * x_l \qquad j = 1,2,\ldots,m \tag{17}$$

where, m and k represent the number of neurons in the hidden and complete input layers, respectively. The output vector is computed by using the following equations:

$$\widehat{\mathbf{Y}} = f\left(a_i^o\right) = \frac{1}{1+\exp\left(-a_i^o\right)} \quad i = 1, 2, \ldots, n \tag{18}$$

$$a_i^o = \sum_{j=1}^{m} W_{ji}^{o,h} * h_j \quad i = 1, 2, \ldots, n \tag{19}$$

$\widehat{\mathbf{Y}}$ represents the predicted value of $\mathbf{Y}$. The optimal weight matrices are obtained using continuous iterations of backward propagation.

- **ELM**. The ELM is a single-hidden layer feedforward neural network. It consists of three layers, namely the input layer, hidden layer, and output layer, where the hidden layer needs not be neuron alike. The output function of the ELM for single-hidden layer feedforward networks is described as:

$$\mathbf{F}(\mathbf{X}) = \sum_{i=1}^{L} h_i(\mathbf{X})\,\boldsymbol{\beta}_i^{\mathrm{T}} = \mathbf{h}(\mathbf{X})\,\boldsymbol{\beta} \tag{20}$$

where, $\boldsymbol{\beta} = \left[\boldsymbol{\beta}_1, \quad \cdots \quad , \boldsymbol{\beta}_L\right]^{\mathrm{T}}$ represents the output weight matrix between the hidden layer and the output layer. $\mathbf{h}(\mathbf{X}) = \left[h_1(\mathbf{X}), \quad \cdots \quad , h_L(\mathbf{X})\right]$ represents ELM nonlinear random feature mapping. $\mathbf{h}(\mathbf{X})$ represents the output of the ith hidden node, which is expressed as:

$$h_i(\mathbf{X}) = G\left(\mathbf{w}_i, b_i, \mathbf{X}\right) \tag{21}$$

where, $G\left(\mathbf{w}_i, b_i, \mathbf{X}\right)$ is a nonlinear piecewise continuous function that satisfies the ELM general approximation capability theorems. Specifically, $\mathbf{X}$ is the input vector, b_i is the bias weight of the ith hidden node, and $\mathbf{w}_i$ represents the input weight vector connecting the input layer with the ith hidden node. The weights of this network are set based on the Moore–Penrose generalized inverse, instead of being updated based on the gradient-based backpropagation algorithms. The ELM algorithm is computationally efficient and has a generalized performance.

- **FNN**. FNN comprises of three distinct layers, including the input layer, fuzzy rules layer, and output layer, which realizes a fuzzy inference system. The input variables of the FNN are the outputs of the input layer, which can be expressed as $\mathbf{X} = [V\_\theta \quad V\_p \quad V\_v \quad V\_a]^{\mathrm{T}}$ in this vehicle example. Each neuron of the fuzzy rules layer represents a fuzzy rule and computes a membership value for the input sample based on the relevant multivariate Gaussian fuzzy membership function. The input of each neuron in this layer is the input vector X, and the output of each neuron is the membership value of the input sample. The membership function of each fuzzy rule is defined as follows:

$$\varphi_i(X) = \exp\left(-d_i^2(X)\right) \tag{22}$$

$$d_i^2(X) = \left(X - M_i\right)^{\mathrm{T}} \sum\nolimits_i^{-1} \left(X - M_i\right) \tag{23}$$

where, $\varphi_i(X)$ is the ith fuzzy rule membership function. M_i and $\sum_i$ are the mean vector and the covariance matrix of the ith fuzzy rule, respectively. $d_i^2(X)$ is the weighted distance (Mahalanobis distance) between the input vector X and the center of ith fuzzy rule's patch M_i. The neuron with linear activation function is used to provide the network's final output. The final output of the network, $\hat{Y}$, is computed as follows:

$$\hat{Y} = \phi W = \sum_{i=1}^{n} w_i \varphi_i \tag{24}$$

$$\phi = \left[\varphi_1 \quad \cdots \quad \varphi_n\right] \tag{25}$$

Therefore, a general ML modelling procedure can be developed for estimating the suspension parameters. The input layer of this model comprises four neurons, corresponding to the pitch angle (V_θ), the position of the vehicle's centroid (V_p), the vertical velocity (V_v), and the vertical acceleration (V_a). The output layer of the model also contains four neurons, which correspond to the stiffness and damping coefficients of the front suspensions (F_k, F_c) and the rear suspensions (R_k, R_c). We use 1000 data samples for training and testing the ML model. The selection of learning rate, number of neurons, and other parameters is performed based on experiments [40]. For instance, due to its efficiency, the Levenberg–Marquardt algorithm is used for backpropagation. The overall modelling procedure for suspension parameter estimation is presented in Fig. 4.

4. Results and discussion

4.1. Results of ML models

We use eight ML mentioned above models for estimating the suspension parameters. In order to visualize the predicted results, the nonlinear relationship estimated between the model's inputs (vehicle states) and the outputs (suspension parameters) is presented in Figs. 5 to 8. The multibody model results are also included as a reference for performing comparison. The stiffness and damping coefficients of the rear suspensions are set as the ordinate and abscissa in the figures, respectively. Note that the stiffness and damping coefficients of the front suspensions are fixed at 40,000 N/m and 1980 N s/m, respectively, for convenience of plotting. It is evident from Figs. 5 to 8 that the predicted results of each model are consistent with the reference data. Please note that only a fraction of the results are presented for better visualization.

In order to quantify the differences in the predicted results of different ML models, we evaluate and compare the performance of the models in terms of several evaluation metrics, including the coefficient of determination (R^2), maximum absolute error (ME), mean absolute error (MAE), mean absolute percentage error (MAPE), and root mean square error (RMSE). ME and MAE represent the maximum and mean absolute errors between the predicted data and the reference data (multibody model results), respectively. MAPE represents the mean absolute percentage error and RMSE measures the corresponding deviation. The comparison of results is presented in Table 3.

It is evident from Table 3 that the performance of the eight ML as mentioned earlier methods is different from each other in terms of ME, MAE, MAPE, and RMSE. Please note that the incorrect numerical values in Table 3 are marked with a cross. The MAEs of the front stiffness obtained using the GRNN and ELM models are 1000 N/m and 1203.52 N/m, respectively. The MAE of the rear stiffness obtained using the ELM model is 1020.56 N/m. Contrarily, other ML models result in relatively small MAE values. In addition, we observe that the MAPE of the front damping coefficients obtained by using the GRNN and FNN models are 2.00% and 1.615%, respectively. These values are relatively large and are not acceptable for accurate parameter estimation. The R^2 of the front suspensions' damping-coefficient obtained using the GRNN and FNN models are 0.27 and 0.55, respectively. This shows that these models are unable to estimate the damping coefficients accurately. The R^2 of the front and rear damping-coefficients obtained using the WNN and FNN models are 0.83 and 0.86, respectively. This shows that their estimation errors are relatively large and possibly unacceptable. The R^2 of the other ML models is close to 1 (larger than 0.9), which means that the suspension parameters can be accurately estimated by using these ML models (see Figs. 6 and 7).

4.2. Discussion

To effectively visualize the accuracy of ML models, we use box plots to describe the absolute percentage error between the predicted data and the reference data. The comparative results are presented in Fig. 9. The elements of the box plot include the median, the average, the lower quartile, the upper quartile, and the maximum value. The quartile

Table 3
The accuracy analysis of the ML models.

ML models	Parameters	ME	MAE	MAPE (%)	RMSE	R^2
PSO-BP	F_k	198.15	10.29	0.027	15.71	0.99
	F_c	81.17	3.06	0.162	5.85	0.99
	R_k	99.56	6.73	0.021	9.54	0.99
	R_c	27.95	1.39	0.074	2.30	0.99
RBF	F_k	193.77	10.25	0.027	15.21	0.99
	F_c	118.87	3.15	0.166	6.04	0.99
	R_k	82.29	6.91	0.021	9.09	0.99
	R_c	34.81	1.46	0.078	2.20	0.99
GRNN	F_k	1000 (×)	100.92	0.266	211.66	0.98
	F_c	180	37.66	2.00 (×)	49.06	0.27 (×)
	R_k	500	1.09	0.004	22.05	0.99
	R_c	180	8.50	0.45	15.51	0.93
DBN	F_k	901.78	16.78	0.045	34.45	0.99
	F_c	287.41	5.17	0.273	12.44	0.95
	R_k	501.11	15.16	0.047	26.01	0.99
	R_c	184.05	2.93	0.156	6.54	0.99
WNN	F_k	525.02	38.92	0.103	64.11	0.99
	F_c	174.47	14.62	0.78	26.63	0.83 (×)
	R_k	591.41	21.07	0.066	42.11	0.99
	R_c	92.80	5.03	0.267	8.46	0.98
ElmanNN	F_k	249.74	10.07	0.027	16.07	0.99
	F_c	85.68	2.74	0.145	5.20	0.99
	R_k	110.91	7.32	0.022	9.89	0.99
	R_c	35.49	1.19	0.063	1.90	0.99
ELM	F_k	1203.52 (×)	20.96	0.056	45.22	0.99
	F_c	410.71	9.75	0.517	16.79	0.91
	R_k	1020.56 (×)	9.51	0.030	33.69	0.99
	R_c	415.43	4.24	0.225	12.60	0.95
FNN	F_k	440.56	61.12	0.162	85.57	0.99
	F_c	142.80	30.50	1.615 (×)	38.62	0.55 (×)
	R_k	280.45	29.36	0.092	44.79	0.99
	R_c	121.52	13.40	0.708	21.50	0.86 (×)

represents the divided value after the entire series is divided into four equal parts. The difference between the upper and lower quartile is called the interquartile range (IQR). Fig. 9 presents the discreteness and bias of the results. The center position and spreading range of the data distribution are also presented. Please note that the height of the box reflects the fluctuations in the data to a certain extent.

We use MAE as an example to investigate the accuracy of the ML models. The MAE results are presented in Fig. 10. The results show that except GRNN and FNN, the MAEs of other ML models, such as PSO-BP, RBF, and ElmanNN are relatively small. These models are potentially the best three ML models for estimating the suspension parameter. In our previous work [41], we also developed a DNN model for estimating the suspension parameter based on the same data. We compare the performance of DNN with PSO-BP, RBF, and ElmanNN in terms of MAE. The comparative results are presented in Fig. 11. It is evident from the results that the performance of the PSO-BP, RBF, and ElmanNN models is better as compared to the DNN model. The accuracy gain in terms of MAE reaches 10%.

Due to its depth, the computational complexity of DNN is relatively high. In addition, it falls into the local minima in some cases. On the other hand, the PSO algorithm assists the neural network avoiding the local minimum and enhances its searching ability. Therefore, the PSO-BP model shows better performance as compared to the DNN model. However, this also means th PSO-BP requires a greater number of iterations and longer time during the training process. The numerical results and quantitative analysis show that the ElmanNN algorithm is also an appealing alternative. As ElmanNN is based on the BP network, adding a context layer to the hidden layer ensures that the BP network has a short-term memory function and improves the accuracy of the ML model.

Furthermore, we also develop ML suspension parameter identification models under the initial vehicle speeds of 10 m/s and 30 m/s. The MAE of the predicted suspension stiffness and damping coefficients in the cases of 10 m/s, 20 m/s, and 30 m/s is presented in Fig. 12. It is evident from Fig. 12 that the MAE trends for the three initial speed cases are similar. The GRNN and FNN models are least accurate. The PSO-BP, RBF, and Elman models are the most accurate. The MAEs of the front suspension stiffness coefficients predicted by the PSO-BP model are 11.99 N/m, 10.29 N/m, and 5.71 N/m, which correspond to the speeds of 10 m/s, 20 m/s, and 30 m/s, respectively. These values are minimal. The MAEs of the rear suspension damping coefficients obtained using the PSO-BP model are 2.15 N s/m, 1.39 N s/m, and 0.93 N s/m, corresponding to the speeds of 10 m/s, 20 m/s, and 30 m/s, respectively. These values are extremely small. This proves that the PSO-BP, RBF, and Elman methods are robust in high and low vehicle speeds. In addition, it is evident from Fig. 12 that ML-predicted results under the 30 m/s initial speed are more accurate than the results of the 10 m/s and 20 m/s speeds. This is because the nonlinear relationship between the suspension parameters and vehicle states is clearer at higher vehicle speed, which is beneficial for developing an accurate ML models. In the case of low speed, it may be a feasible way to add more vehicle states as the inputs for improving the prediction accuracy.

5. Conclusions

In this work, a series of ML models is used to estimate the stiffness and damping coefficients of suspensions due to performance degradation. The dynamic simulations on a bumpy road are performed by using an efficient multibody model to capture the critical states of a vehicle. It is noteworthy that the vehicle multibody model considers the dynamic properties of all components, such as chassis, suspensions, tires, etc., resulting in accurate vehicle states. The vehicle states and suspension parameters are used as the inputs and outputs of the ML models, respectively. The accuracies of the eight ML models are investigated in terms of evaluation metrics, including ME, MAE, MAPE, RSME, and R^2. The comparative results show that the PSO-BP, RBF, and ElmanNN models obtain the most accurate estimation. The RBF model uses fewer parameters and obtains a higher computational efficiency. The ElmanNN model is more robust as compared to other ML models. Overall, a comprehensive investigation of ML methods for estimating the suspension parameters based on accurate dynamic simulations is performed, and appropriate ML models are recommended. There are two limitations to the presented suspension-parameter estimation approach. First, precise vehicle multibody models require dynamic principal component parameters, which are sometimes difficult to obtain. Second, the selected initial parameters of the ML models may affect the estimated results. A actual vehicle test will be carried out for further validation in the near future.

CRediT authorship contribution statement

Yongjun Pan: Writing – review & editing, Supervision, Project administration, Methodology, Funding acquisition, Conceptualization. **Yu Sun:** Writing – original draft, Visualization, Validation, Software, Investigation, Data curation. **Zhixiong Li:** Writing – review & editing, Methodology, Conceptualization. **Paolo Gardoni:** Supervision, Resources.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

This work was funded by the National Natural Science Foundation of China (No. 12072050 and No. 12211530029), the Research Project of State Key Laboratory of Mechanical System and Vibration, China (No. MSV202216), the Fundamental Research Funds for the Central Universities, China (No. 2021CDJQY-032), and the Norwegian Financial Mechanism 2014-2021 under Project Contract No 2020/37/K/ST8/02748.

References

[1] Alrejjal A, Ksaibati K. Impact of mountainous interstate alignments and truck configurations on rollover propensity. J Saf Res 2022;80:160–74.

[2] özmen B, Topaç M. Effect of damping rate on fatigue failure tendency of a topology-optimised swing arm for a heavy commercial truck cab suspension. Eng Fail Anal 2022;137:106276.

[3] Yan S, Sun W, He F, Yao J. Adaptive fault detection and isolation for active suspension systems with model uncertainties. IEEE Trans Reliab 2019;68(3):927–37.

[4] Chen K, He S, Xu E, Tang R, Wang Y. Research on ride comfort analysis and hierarchical optimization of heavy vehicles with coupled nonlinear dynamics of suspension. Measurement 2020;165:108142.

[5] Chen X, Bose N, Brito M, Khan F, Thanyamanta B, Zou T. A review of risk analysis research for the operations of autonomous underwater vehicles. Reliab Eng Syst Saf 2021;216:108011.

[6] Singh KB, Arat MA, Taheri S. Literature review and fundamental approaches for vehicle and tire state estimation. Veh Syst Dyn 2019;57(11):1643–65.

[7] Bogdanski K, Best MC. Kalman and particle filtering methods for full vehicle and tyre identification. Veh Syst Dyn 2018;56(5):769–90.

[8] Yan R, Dunnett S, Jackson L. Model-based research for aiding decision-making during the design and operation of multi-load automated guided vehicle systems. Reliab Eng Syst Saf 2022;219:108264.

[9] Xu Z, Saleh JH. Machine learning for reliability engineering and safety applications: Review of current status and future opportunities. Reliab Eng Syst Saf 2021;211:107530.

[10] Sharma N, Gardoni P. Mathematical modeling of interdependent infrastructure: An object-oriented approach for generalized network-system analysis. Reliab Eng Syst Saf 2022;217:108042.

[11] Mazumder RK, Salman AM, Li Y. Failure risk analysis of pipelines using data-driven machine learning algorithms. Struct Saf 2021;89:102047.

[12] Lu W, Liu J, Fu X, Yang J, Jones S. Integrating machine learning into path analysis for quantifying behavioral pathways in bicycle-motor vehicle crashes. Accid Anal Prev 2022;168:106622.

[13] Xu Z, Saleh JH, Subagia R. Machine learning for helicopter accident analysis using supervised classification: Inference, prediction, and implications. Reliab Eng Syst Saf 2020;204:107210.

[14] Hossain MS, Ong ZC, Ismail Z, Noroozi S, Khoo SY. Artificial neural networks for vibration based inverse parametric identifications: A review. Appl Soft Comput 2017;52:203–19.

[15] Theissler A, Pérez-Velázquez J, Kettelgerdes M, Elger G. Predictive maintenance enabled by machine learning: Use cases and challenges in the automotive industry. Reliab Eng Syst Saf 2021;215:107864.

[16] Saraygord Afshari S, Enayatollahi F, Xu X, Liang X. Machine learning-based methods in structural reliability analysis: A review. Reliab Eng Syst Saf 2022;219:108223.

[17] Drehmer LRC, Casas WJP, Gomes HM. Parameters optimisation of a vehicle suspension system using a particle swarm optimisation algorithm. Veh Syst Dyn 2015;53(4):449–74.

[18] Xie K, Yi H, Hu G, Li L, Fan Z. Short-term power load forecasting based on Elman neural network with particle swarm optimization. Neurocomputing 2020;416:136–42.

[19] Lopez-Sanchez I, Rossomando F, Pérez-Alcocer R, Soria C, Carelli R, Moreno-Valenzuela J. Adaptive trajectory tracking control for quadrotors with disturbances by using generalized regression neural networks. Neurocomputing 2021;460:243–55.

[20] Lin C-M, Tai C-F, Chung C-C. Intelligent control system design for UAV using a recurrent wavelet neural network. Neural Comput Appl 2014;24(2):487–96.

[21] Huang W, Zhao J, Yu G, Wong PK. Intelligent vibration control for semiactive suspension systems without prior knowledge of dynamical nonlinear damper behaviors based on improved extreme learning machine. IEEE/ASME Trans Mechatronics 2021;26(4):2071–9.

[22] Taghavifar H, Mardani A, Hu C, Qin Y. Adaptive robust nonlinear active suspension control using an observer-based modified sliding mode interval type-2 fuzzy neural network. IEEE Trans Intell Veh 2020;5(1):53–62.

[23] Zhu D, Cheng X, Yang L, Chen Y, Yang SX. Information fusion fault diagnosis method for deep-sea human occupied vehicle thruster based on deep belief network. IEEE Trans Cybern 2021;1–14.

[24] Jiao R, Peng K, Dong J, Zhang C. Fault monitoring and remaining useful life prediction framework for multiple fault modes in prognostics. Reliab Eng Syst Saf 2020;203:107028.

[25] Jing Z, Chen J, Li X. RBF-GA: An adaptive radial basis function metamodeling with genetic algorithm for structural reliability analysis. Reliab Eng Syst Saf 2019;189:42–57.

[26] Funes FJ, García de Jalón J. An efficient dynamic formulation for solving rigid and flexible multibody systems based on semirecursive method and implicit integration. J Comput Nonlinear Dyn 2016;11(5):051001.

[27] Pan Y, Callejo A, Bueno JL, Wehage RA, García de Jalón J. Efficient and accurate modeling of rigid rods. Multibody Syst Dyn 2017;40(1):23–42.

[28] Rodríguez AJ, Sanjurjo E, Pastorino R, Naya MÁ. State, parameter and input observers based on multibody models and Kalman filters for vehicle dynamics. Mech Syst Signal Process 2021;155:107544.

[29] García de Jalón J, Álvarez E, de Ribera F, Rodríguez I, Funes F. A fast and simple semi-recursive formulation for multi-rigid-body systems. In: Ambrósio J, editor. Advances in computational multibody systems. Computational methods in applied sciences, vol. 2, Springer Netherlands; 2005, p. 1–23.

[30] von Schwerin R. Multibody system simulation, numerical methods, algorithms and software. Springer; 1999.

[31] Hidalgo AF, García de Jalón J. Real-time dynamic simulations of large road vehicles using dense, sparse, and parallelization techniques. J Comput Nonlinear Dyn 2015;10(3):031005.

[32] Pan Y, Dai W, Xiong Y, Xiang S, Mikkola A. Tree-topology-oriented modeling for the real-time simulation of sedan vehicle dynamics using independent coordinates and the rod-removal technique. Mech Mach Theory 2020;143:103626.

[33] García de Jalón J, Callejo A, Hidalgo AF. Efficient solution of Maggi's equations. J Comput Nonlinear Dyn 2012;7(2):021003.

[34] Pan Y, Dai W, Huang L, Li Z, Mikkola A. Iterative refinement algorithm for efficient velocities and accelerations solutions in closed-loop multibody dynamics. Mech Syst Signal Process 2021;152:107463.

[35] Pacejka H. Tire and vehicle dynamics. Elsevier; 2005.

[36] He X, Wang Z, Li Y, Khazhina S, Du W, Wang J, et al. Joint decision-making of parallel machine scheduling restricted in job-machine release time and preventive maintenance with remaining useful life constraints. Reliab Eng Syst Saf 2022;222:108429.

[37] García Nieto P, García-Gonzalo E, Sánchez Lasheras F, de Cos Juez F. Hybrid PSO–SVM-based method for forecasting of the remaining useful life for aircraft engines and evaluation of its reliability. Reliab Eng Syst Saf 2015;138:219–31.

[38] Jiang Q, Zhu L, Shu C, Sekar V. An efficient multilayer RBF neural network and its application to regression problems. Neural Comput Appl 2022;34:1–18.

[39] Kir Savaş B, Becerkli Y. Behavior-based driver fatigue detection system with deep belief network. Neural Comput Appl 2022;34:14053–65.

[40] Tian H, Wang P, Tansey K, Zhang S, Zhang J, Li H. An IPSO-BP neural network for estimating wheat yield using two remotely sensed variables in the Guanzhong plain, PR China. Comput Electron Agric 2020;169:105180.

[41] Pan Y, Sun Y, Min C, Li Z, Gardoni P. Maneuver-based deep learning parameter identification of vehicle suspensions subjected to performance degradation. Veh Syst Dyn 2022;1–17.