Dynamic Stability Analysis of Vehicles at Superelevated Horizontal Curves Using Machine Learning

Bülent Yıldırım
Faculty of Technology, Civil Engineering
Isparta University of Applied Sciences
Isparta, Türkiye

***Abstract*— The stability of vehicles traveling in different road conditions varies depending on road geometry and vehicle characteristics. To evaluate the stability of vehicles negotiating horizontal curves, which are an important component of highway alignment, certain critical factors, especially rollover and skidding speeds, need to be determined in advance. These parameters express the speed at which the vehicle can safely turn around curves for the current road conditions and how much it can remain balanced on inclined surfaces.**

In this context, the study examines the stability of vehicles traveling on horizontal curves. Within the scope of the study, data sets were created for different vehicle types, road conditions and road characteristics. These datasets cover a variety of scenarios to understand how different types of vehicles behave in different conditions. Various models have been developed using machine learning algorithms on these data sets, and predictions have been made with the models developed for different scenarios. As a result of the analysis, it was determined that the Polynomial Regression model, one of the developed machine learning models, provided better results. In addition, the effects of superelevation on horizontal curves for different vehicle types and road conditions have been examined in detail.

The analyses showed that especially the curve radius of 200 meters was found to be a critical value. As a result, it has been understood that with the use of machine learning techniques, inferences regarding vehicle stability and safety can be obtained more quickly and can contribute to the development of driver assistance systems conventional and autonomous vehicles in the future.

***Keywords*— dynamic stability; skidding; rollover; superelevated horizontal curves; machine learning**

I. Introduction

Horizontal curves are of vital importance in highway design in terms of road safety and effective traffic flow. Horizontal curves are designed to allow vehicles to change direction while, at the same time, allowing drivers to travel safely. Vehicle stability is one of the main factors taken into consideration in horizontal curves. Vehicle stability refers to the ability of a vehicle to remain balanced at a certain speed, and this feature is of great importance in horizontal curves. At high speeds or sudden maneuvers, vehicles must remain balanced and adverse situations such as skidding or tipping over must be prevented.

The curves used in the design of horizontal alignment must be determined adequately to ensure that vehicles remain stable at a given speed [1]. The radius of these curves should be compatible with the vehicle speed and allow drivers to use curves easily. In addition, the smoothness of the road surface is also important for vehicle stability, because an uneven or problematic road surfaces can make it difficult to control vehicles and cause accidents.

Superelevation is the slope given to the road in the transverse direction. It has a significant impact on vehicle stability, especially in horizontal curves [2-4]. Along horizontal curves vehicles lean towards outside when rotating. These types of curves have a certain advantage in terms of balancing the centrifugal force of the vehicle and maintaining its stability. On the other hand, curves without superelevation are curves where the road surface has a crown. Vehicles have difficulty in terms of stability as they tend to lean more when turning on curves without superelevation.

Another factor that affects vehicle stability in horizontal curves is the vehicle type. For example, a truck, which is a high-speed vehicle over the ground, may be more prone to rollover on the curves without superelevation, which can negatively affect stability. In contrast, a fast sports vehicle with low ground clearance may have features such as a lower center of gravity and a larger tire contact area, which can provide better stability on curves without superelevation. As a result, the differences between curves without and with superelevation have a significant impact on vehicle stability. Proper design of both types of curves is of great importance for the safety of vehicle occupants and regulation of traffic flow.

Within the scope of the study, firstly, information about horizontal curves and vehicle dynamics is given, then the machine learning approach is introduced, after the machine learning section, the speed estimation models developed for different vehicle types on curves with and without superelevation are explained and the analyses performed are explained. The findings obtained are presented in the results section.

II. Vehicle Dynamics in Horizontal Curves

Some physical forces act on vehicles traveling on horizontal curves, and the stability of the vehicle varies depending on the impact of these forces. Vehicle dynamics equations are equations that mathematically describe the motion of a vehicle. With the help of these equations, it is possible to understand the dynamic behavior of vehicles and predict their situation under different conditions. The vehicle's speed, acceleration, curve stability and other parameters can be calculated using these equations. Vehicle dynamics equations are important in different fields such as civil engineering, automotive design and driver assistance systems. These equations are used to evaluate the performance of vehicles in different road conditions, increase traffic safety and improve vehicle design. According to basics of physics and dynamics, the forces acting on a vehicle at horizontal curves are shown in Fig. 1 [1].

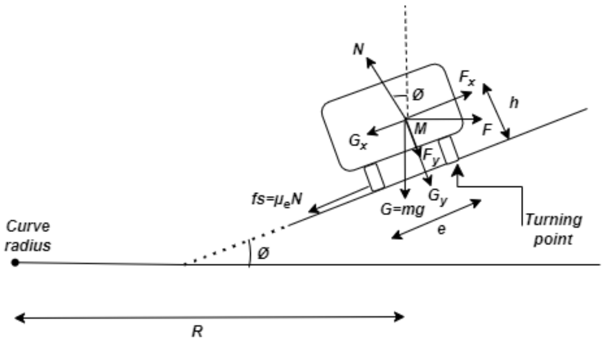


Fig. 1. Forces acting on a vehicle travelling along a horizontal curve

The symbols used in the figure are

F: The centrifugal force (N)

f_s: Lateral friction force

μ_e: Coefficient of lateral friction

N: Normal force

M: Centre of gravity of the vehicle

G: Weight of the vehicle

h: Height of the vehicle's centre of gravity from the road (m)

e: Wheelbase of the vehicle (m)

m: Mass of the vehicle (kg)

R: Curve radius (m)

If the turning radius decreases or the vehicle speed increases, the centrifugal force increases. This may cause the vehicle to be exposed to more external forces while turning, thus reducing its stability. In order for the vehicle to skid, be $> f_s$. Considering factors such as the vehicle's turning radius, speed and mass, the value of the centrifugal force can be calculated as follows.

$$\sum F_x = 0 \tag{1}$$

$$F_x - G_x - f_s = 0 \tag{2}$$

$$F_x = G_x + f_s \tag{3}$$

$$\frac{G}{g} * \frac{(V/3.6)^2}{R} * \cos\emptyset = G * \sin\emptyset + \mu_e * \left(\frac{G}{g} * \frac{(V/3.6)^2}{R} * \sin\emptyset + G * \cos\emptyset\right) \tag{4}$$

Rearranging (4), the following equation is obtained to give the critical skidding speed.

$$V_{skid.} = 11.3 * \sqrt{\frac{R*(tan\emptyset+\mu_e)}{(1-\mu_e*tan\emptyset)}} \tag{5}$$

The expression $\frac{(V/3.6)^2}{R}$ here represents the value of centrifugal acceleration (also called the centrifugal ratio). The value of the friction force is calculated as

$$f_s = \mu_e * N \tag{6}$$

Apart from sliding (skidding), depending on vehicle type and some other parameters, vehicles may also overturn (also known as rollover) at horizontal curves. The rollover formula for a vehicle is calculated by writing the moment equilibrium of all the forces acting on the vehicle with respect to the pivot point, as shown in Fig. 1 [4-6].

The forces considered in Fig. 1 and the moment balance for the pivot point are defined below.

$$F_x * h = G_y * e/2 + F_y * e/2 + G_x * h \tag{7}$$

$$\frac{G}{g} * \frac{(V/3.6)^2}{R} * \cos\emptyset * h = G * \cos\emptyset * \frac{e}{2} + \frac{G}{g} * \frac{(V/3.6)^2}{R} * \sin\emptyset * \frac{e}{2} + G * \sin\emptyset * h \tag{8}$$

the rollover speed is then obtained as

$$V_{rol} = 11.3 * \sqrt{\frac{R*\left(\frac{e}{2}+tan\emptyset*h\right)}{\left(h-tan\emptyset*\frac{e}{2}\right)}} \tag{9}$$

As seen in the equation, the wheelbase of the vehicle, the height of the centre of gravity from the road and the horizontal curve radius affect the rollover speed of the vehicle[7-9].

III. Models and Analyses

Extracting patterns from high dimensional data is a challenging task because visualizing the data distribution gets harder as the number of dimensions in the data, i.e. data features, increases. To understand the distribution of such data and extract meaning from them, we employ various machine learning algorithms. The performance of the models depends on the data distribution that they are trained on. Moreover, the sparsity of the data is another significant factor in the model selection process. The best-performing model should be able to generalize the data distribution well so that it can perform robustly on unseen data.

To decide which machine learning algorithm performs best on the data we work on, the common method is to split the dataset into three subsets named training, validation, and test datasets. First, we train the candidate models on the training dataset. Since various hyperparameters are defined in these machine learning algorithms, we try possible combinations of their values. To measure how well the models performed with these hyperparameter values, we take inference of these trained models on the validation dataset. As the ground truth values are known in the dataset, we select the best-performing model according to the prediction errors on this dataset.

As the process of selecting the best model includes the validation dataset, it cannot be used to report the performance of the final models. The dataset used for the final comparison should be unseen data. Therefore, we use the test set as the unseen data to calculate the performance of the final models for comparison. The performances are reported by calculating how close the predicted models are to the ground truth values. According to this comparison, we select the best model among the candidates.

Researchers have been working on various types of applications of machine learning on the autonomous driving task [10, 11, 12]. In this work, we extend the analysis of [13] on machine-learning algorithms trained for prediction of skidding and rollover speed by considering the angle of superelevation of the horizontal curves. The details of the analyses are provided in this section, which consists of the subsections named dataset generation, machine learning algorithms, and model analyzes.

A. *Dataset Generation*

In Section 1, the equations for calculating the speed of skidding and rollover had been explained in detail. For calculating the skidding speed on a horizontal curve with superelevation, horizontal curve radius (R), side-friction coefficient (μ_e), and angle of the superelevation $(\emptyset)$ are required. On the other hand, horizontal curve radius (R), angle of the superelevation $(\emptyset)$, wheelbase (e), and height of the centre of gravity above ground (h) are needed to calculate the speed of rollover on a horizontal curve with superelevation. We create two different datasets for our analyses. For this purpose, we generate data samples for each variable defined in the calculations. We select the range of these variables from the real-life examples.

For the predicting the speed of skidding, all possible combinations of the following sets are used as the variable values:

- Horizontal curve radius (R): {30, 60, 90, 120, 200, 300, 400, 500}
- Side-friction coefficient (μ_e): {0.4, 0.38, 0.36, 0.34, 0.32, 0.3, 0.28, 0.26, 0.24, 0.22, 0.2, 0.18, 0.16, 0.14, 0.12, 0.1, 0.08, 0.06, 0.04}
- tan(∅): {0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.10}

In total, the speed of skidding dataset consists of 1368 samples. A different approach is used in generating the speed of the rollover dataset. There are 4 different vehicle types defined in the dataset. For all types, the following variable values are sampled from the same sets, which are defined as follows:

- Horizontal curve radius (R) in metres: {30, 40, 50, 60, 70, 80, 90, 100, 110, 120, 130, 140, 150, 160, 170, 180, 190, 200, 225, 250, 275, 300, 325, 350, 375, 400, 425, 450, 475, 500}
- tan(∅): {0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.10}

In the dataset, the vehicle types are selected as car, minibus, heavy truck, and bus. Inspired from the real-life examples, the wheelbases (e) of these categories are defined as 1.736, 2.059, 2.482, and 2.200, respectively (in metres). The height of the centre of gravity above ground (h) is determined from real-life examples as minimum and maximum values. For cars, minibuses, and heavy trucks, the range is defined as [0.508, 0.5842], [0.762, 1.016], and [1.524, 2.159], respectively (in metres). As an exception, an approximate value is used for the bus, which is 1.302 m. Simple random sampling method is utilized to sample 10 possible h values for each category having a value range. Similar to the speed of skidding dataset, all possible combinations of the value sets are used as the data samples. As a result, the speed of rollover dataset consists of 270 samples for the bus category and 2700 samples for each of the rest. In total, there are 8370 samples in the dataset.

B. *Machine Learning Algorithms*

Three different machine learning algorithms are used to analyse their prediction performance on the speed of skidding and rollover. Since the speed values are continuous variables, the best candidates for the analysis are selected as linear regression, polynomial regression, and multi-layer perceptron models. These models are trained with the supervised training approach as the ground-truth values of the target speeds are defined in the datasets.

1) Linear Regression: The linear regression method assumes that the output values have a linear relation with the input values. To find such a relation, coefficients β_{1-n} are assigned to the input values and a bias term β_0 is defined. The equation of this relation is as follows:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n \quad (10)$$

The linear relation represents a hyperplane, where the input value coefficients define the orientation of the hyperplane while the bias term determines the displacement of the hyperplane in space relative to the origin. In the equation, x_{1-n} are the input values while y stands for the predicted value. The coefficients and the bias term β_{0-n} are the learnable parameters in this method. These parameters are found by minimizing the error between predicted and ground-truth values. The most common approach used as the objective is to minimize the mean square error.

2) Polynomial Regression: The training process of the polynomial regression method is similar to the linear regression; however, it takes input values with a different approach. Unlike linear regression, polynomial regression finds coefficients for the power of inputs from 1 to k to be able to learn non-linear relations. As an example, for k=3 and two input variables x_1 and x_2, the equation of the learned function is as follows:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1^2 + \beta_4 x_2^2 + \beta_5 x_1 x_2 + \beta_6 x_1^3 + \beta_7 x_2^3 + \beta_8 x_1^2 x_2 + \beta_9 x_1 x_2^2 \quad (11)$$

k is the hyperparameter of the model. The best value for minimizing the error depends on the data distribution of the dataset. It can be determined by calculating the error on the validation dataset for different *k* values and selecting the one that minimizes the error.

3) Multi-layer Perceptron (MLP): Neural networks are one of machine learning algorithms that are a function composed of multiple non-linear functions, called neurons. The way of combining these functions defines if a model is single-layer or multi-layer. The multi-layer version of the neural networks is called a multi-layer perceptron in the literature. Each layer processes data coming from the previous layer and feeds the results to the next layer. The data processing is done by the pre-defined non-linear functions, neurons, that have learnable parameters in them. The equation of a neuron is as follows:

$$\mathrm{a_j} = \mathrm{g}\left(\sum_{\mathrm{i=1}}^{\mathrm{n}} \mathrm{w_{ij}} \mathrm{x_i} + \mathrm{b_j}\right) \quad (12)$$

In the equation, $\mathrm{a_j}$ stands for the value generated for the next layer by jth neuron in the current layer. For finding this value, first, we multiply all the input values $\mathrm{x_i}$, provided by the n neurons in the previous layer, with the corresponding weights $\mathrm{w_{ij}}$ and sum the results together with the bias $\mathrm{b_j}$, where the weights and the bias are learnable parameters. Later, we pass the results of the sum, linear operation, through the pre-defined activation function g so that the relations are mapped to a non-linear space. This mapping allows the network to learn more complex data distributions.

The simplest MLP model consists of three layers that are input, hidden, and output layer. The number of neurons in the input layer is determined by the number of dimensions of the input data, which is equal to the number of features defined in the dataset. Unlike the input layer, the hidden layer consists of k neurons, which is a hyperparameter that needs to be tuned. As the predicted value is the final output of the network, the number of neurons in the output layer has to be equal to the number of predicted values. In our analysis, it is equal to one because only a speed value is predicted on both datasets. Also, the number of hidden layers is fixed to one and the number of neurons in the hidden layer k is tuned on the validation dataset.

The process starting from feeding the input data to the network to obtaining the output value is called forward propagation. After the prediction is obtained, the learnable parameters in the neural network are updated by using the backpropagation method [14]. This method simply updates the parameters by minimizing an error function [15-16]. The derivatives of the error with respect to the parameters are used to update the learnable parameters. By iteratively updating these parameters with small steps, the model converges to a set of parameters that minimizes the error on the training dataset. The mean square error is the error function that is commonly used for training the regression models.

C. Model Analysis

For the implementation of the models, we employed scikit-learn [17], and PyTorch [18] libraries. Furthermore, the comparison of the ground truth and the predicted data are visualized with the help of Matplotlib [19] library. As explained in Section 2, the training and evaluation processes require three different dataset splits. For this purpose, the samples in the datasets are divided into training, validation, and test datasets, which contain 64%, 16%, and 20% of the dataset samples, respectively. The samples are shuffled before creating the dataset splits to ensure randomness. Before training, the input and output values are normalized to the [0,1] range. After finalizing the dataset pre-processing, the models are trained on the training dataset with the means square error loss. During training, the performance of the model is monitored by evaluating them on the validation dataset with different hyperparameters to select the best-performing model. The prediction errors on the validation dataset are reported based on normalized values. For the performance comparison between the models, comparing the best models selected per algorithm, prediction errors on the test dataset are reported for both normalized and unnormalized values.

Only a single model was trained with linear regression as there is no hyperparameter defined in this algorithm. The error values of the trained model on the ground-truth values are provided in Table 1. Moreover, the ground truth and the predicted values on the test dataset are visualized in Fig. 2. In the table, (N) stands for normalized values.

During the training process of the polynomial regression model, various polynomial degrees *k* are tried for hyperparameter tuning. The error values of the model on ground-truth data reached during these trials are provided in Table 2, and the ground-truth and the predicted values on the test dataset are visualized in Fig. 3.

The best-performing models for both datasets are found by selecting the value of *k* that reached to minimum error on the validation dataset, which is (k=10) for both datasets. Later, errors of the best-performing models on the test dataset are reported for the final comparisons.

In the MLP training process, the number of epochs is fixed to 10,000, ReLU is used as the activation function, and the combinations of the following hyperparameters are utilized for hyperparameter tuning:

- Number of neurons in the hidden layer: {4, 8, 16, 32, 64, 128, 256}
- Learning rate: {0.05, 0.01, 0.005, 0.001}

During training, the best models are selected by monitoring the validation error. For both datasets, the best hyperparameters found are 0.01 for the learning rate and 256 for the number of neurons in the hidden layer. The error values of the model on ground-truth data reached during these trials are provided in Table 3, and the ground-truth and the predicted values on the test dataset are visualized in Fig. 4.

Table 1 Mean square errors of predicted values by the linear regression model.

Prediction Type	Validation (N)	Test (N)	Test
Skidding Speed	0.00305	0.00322	89.88647
Rollover Speed	0.00225	0.00227	240.19882

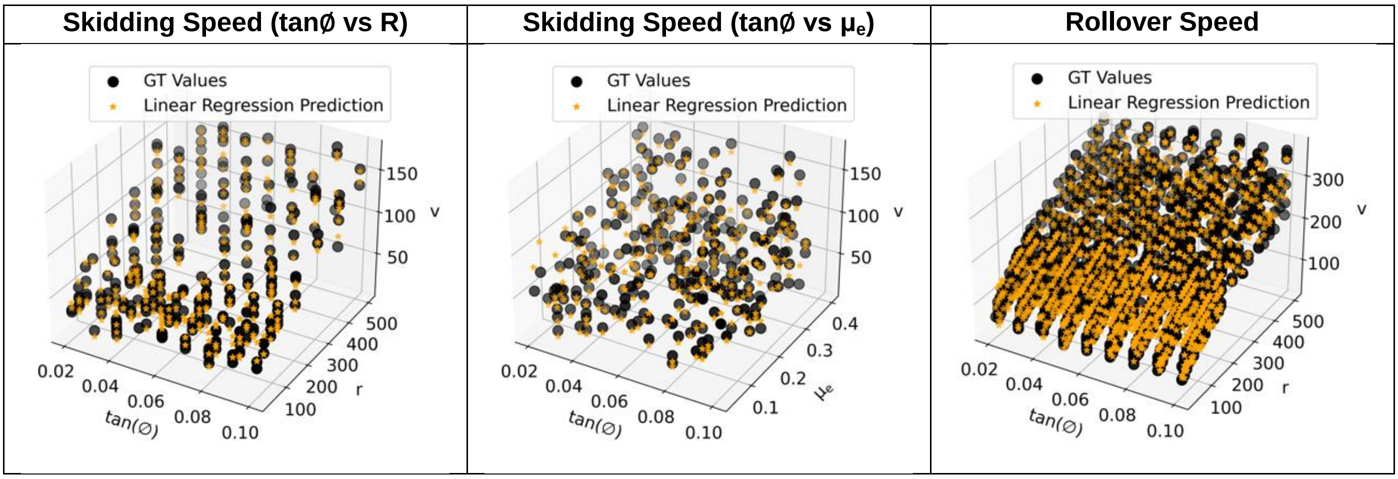
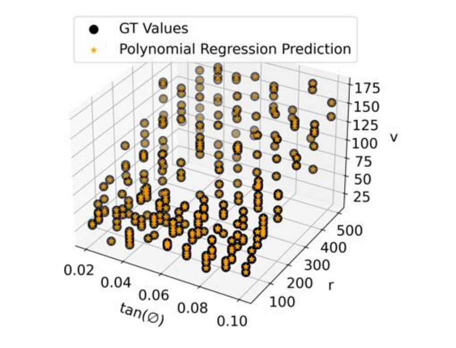


Fig. 2 Performance of the linear regression model in predicting skidding and rollover speeds on the test dataset.

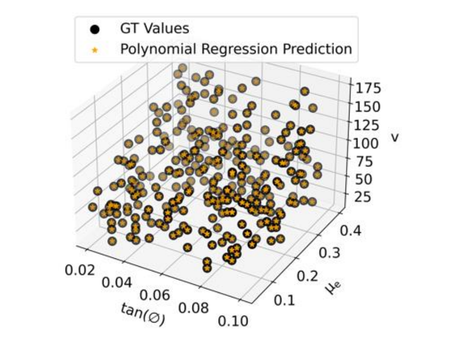
Table 2 Mean square errors of predicted values by the polynomial regression model.

Prediction Type	Degree of Polynomial (k)	Validation (N)	Test (N)	Test
Skidding Speed	1	3.05e-03	-	-
	2	2.12e-04	-	-
	3	2.93e-05	-	-
	4	4.24e-06	-	-
	5	2.83e-06	-	-
	6	7.17e-08	-	-
	7	3.34e-09	-	-
	8	3.90e-10	-	-
	9	8.29e-11	-	-
	10	**1.82e-11**	**9.32e-12**	**2.61e-07**
Rollover Speed	1	2.25e-03	-	-
	2	1.37e-04	-	-
	3	1.80e-05	-	-
	4	3.41e-06	-	-
	5	7.39e-07	-	-
	6	1.64e-07	-	-
	7	4.54e-08	-	-
	8	1.19e-08	-	-
	9	3.03e-09	-	-
	10	**8.56e-10**	**7.16e-10**	**7.57e-05**

Skidding Speed (tanØ vs R)



Skidding Speed (tanØ vs μ_e)



Rollover Speed

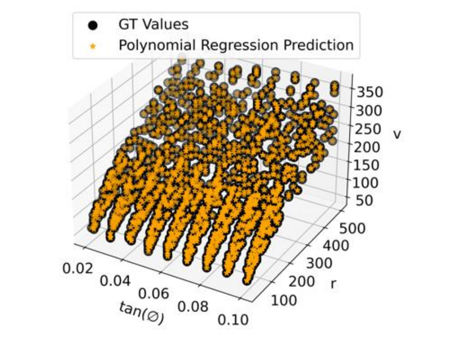
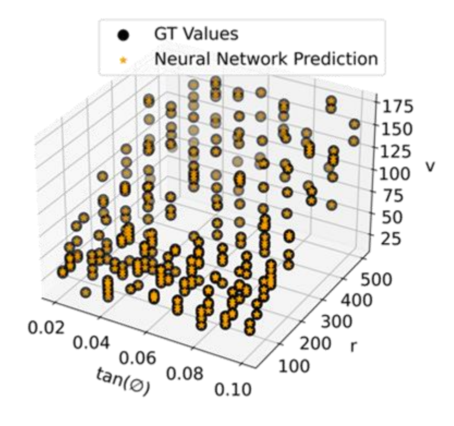


Fig. 3 Performance of the polynomial regression model in predicting skidding and rollover speeds on the test dataset.

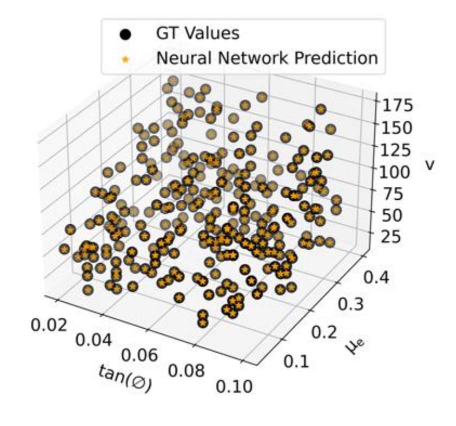
Table 3 Mean square errors of predicted values by the MLP model.

Prediction Type	Validation (N)	Test (N)	Test
Skidding Speed	8.62e-07	6.24e-07	1.74e-02
Rollover Speed	1.11e-06	9.57e-07	1.01e-01

Skidding Speed (tanØ vs R)



Skidding Speed (tanØ vs μ_e)



Rollover Speed

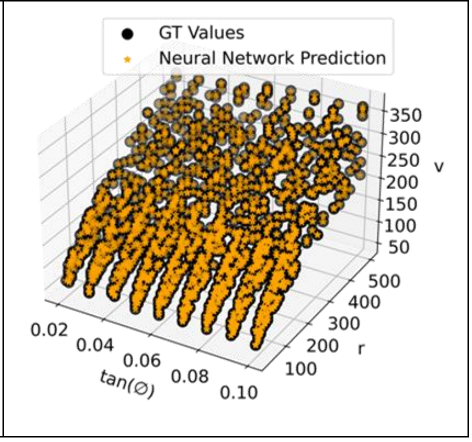


Fig. 4 Performance of the MLP model in predicting skidding and rollover speeds on the test dataset

As seen in Table 4, it was understood that the rollover speed of the vehicles increased as the curve radius and the amount of superelevation increased. This finding revealed that superelevation in the horizontal curve design has a positive effect on vehicle stability. At the same time, increasing the horizontal curve radius will increase traffic safety and vehicle stability. As can be seen from sample findings given in the table, results are very close to reality were obtained with the polynomial regression model on the data sampled from both training and test datasets. Accordingly, it has been concluded that the polynomial regression model, one of the machine learning approaches, can be used to model vehicle stability.

Table 4 Comparison of the polynomial regression model's rollover speed predictions and ground truth values using the mean square error across different vehicle types and radius of horizontal curve (R) values.

Vehicle Type	Radius of Horizontal Curve, R (m)	Tangent of Angle of the superelevation, (tanØ)	Predicted Rollover Speed (km/sa)	Ground Truth Rollover Speed (km/sa)
Bus	100	0.02	105.9784	105.9865
Bus	100	0.06	110.3159	110.3225
Bus	100	0.10	114.7869	114.7964
Bus	200	0.02	149.8802	149.8875
Bus	200	0.06	156.0101	156.0195
Bus	200	0.10	162.3404	162.3466
Car	100	0.02	151.1666	151.1763
Car	100	0.06	156.6875	156.6974
Car	100	0.10	166.8858	166.8956
Car	200	0.02	213.7868	213.7956
Car	200	0.06	224.3290	224.3383
Car	200	0.10	236.0161	236.0260
Minibus	100	0.02	134.1302	134.1385
Minibus	100	0.06	140.0167	140.0254
Minibus	100	0.10	146.3544	146.3652
Minibus	200	0.02	189.6943	189.7004
Minibus	200	0.06	198.0177	198.0258
Minibus	200	0.10	206.9843	206.9917
Heavy Truck	100	0.02	104.0604	104.0654
Heavy Truck	100	0.06	108.3338	108.3396
Heavy Truck	100	0.10	112.7309	112.7374
Heavy Truck	200	0.02	147.1626	147.1707
Heavy Truck	200	0.06	153.2061	153.2153
Heavy Truck	200	0.10	159.4258	159.4347

IV. Results and Discussion

In this study, vehicle stability on horizontal curves was examined and skidding and rollover speeds were estimated for different curve geometries with a machine learning approach. In this context, machine learning approaches such as linear regression, polynomial regression and multilayer perceptron approaches were taken into consideration. Superelevation rates were changed for different curve radii and vehicle types, and training and test data sets were created accordingly. With training and test data, the validity of machine learning models was examined and the methods were compared.

In general, when the error values and prediction graphs of the models on the test set are compared, it has been observed that the capacity of the linear regression model is not sufficient to learn the relationships, while the capacity of the polynomial regression and multilayer perceptron models is successful in predicting the real distribution. As the polynomial degree in the polynomial regression model was increased, the prediction capacity of the model increased and a performance increase was observed up to a point. The best models selected from polynomial regression models showed better performance than multilayer perceptron models.

When the results obtained were evaluated in terms of vehicle dynamics, it was understood that the rollover speeds of different vehicle types varied for the same curve radius, and that this change was directly related to the vehicle size (gauge). According to these results, it has been determined that it will be beneficial for traffic safety for large-vehicles (buses, trucks, etc.) to pay attention to the speed limits, especially on small radius curves, and that safer navigation can be achieved for all vehicle types if the curve radius is 200 m and above has been made.

References

[1] Yayla N, Karayolu Mühendisliği, Birsen Yayınevi, İstanbul, 2009.

[2] Gunay, B. Sliding and Rollover on Highways - Subtleties to Note, The Turkish Chamber of Civil Engineers (Teknik Dergi), vol.33(4), pp. 12329-12334, 2022. https://doi.org/10.18400/tekderg.766631

[3] Lamm R, Psarianos B & Mailaende T. Highway Design and Traffic Safety Engineering Handbook. 848 p, NY: McGraw-Hill, New York, 1999.

[4] Rogers, M. Highway Engineering. Oxford, UK, Blackwell Science, 2003

[5] Tipler P A, Mosca G. Physics for Scientists and Engineers. 6th ed. W. H. Freeman and Company, New York, 2007.

[6] Meriam J L, Kraige L G. Engineering Mechanics: Dynamics. 6th ed. John Wiley & Sons, Hoboken, 2007.

[7] O'Flaherty CA. Highways. Traffic Planning and Engineering. 3rd ed. Edward Arnold, London, UK, 1986.

[8] Gillespie T. D. Fundamentals of Vehicle Dynamics. Society of Automotive Engineers, Warrendale, PA, 1992.

[9] Jin Z, Li B, Li J. Dynamic Stability and Control of Tripped and Untripped Vehicle Rollover. Synthesis Lectures on Mechanical Engineering.Morgan & Claypool Publishers, San Rafael, CA, 2019.

[10] Zou Q, Jiang H, Dai Q, Yue Y, Chen L, Wang Q. Robust lane detection from continuous driving scenes using deep neural networks. IEEE Transactions on Vehicular Technology, 69(1), 41–54, 2020

[11] Tabelini L, et al. PolyLaneNet: Lane Estimation via Deep Polynomial Regression. 25th International Conference on Pattern Recognition (ICPR), pp. 6150-6156, 2020.

[12] Liu Q, Li X, Yuan S, Li, Z. Decision-Making Technology for Autonomous Vehicles: Learning-Based Methods, Applications and Future Outlook. In 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), 30–37, IEEE Press, 2021.

[13] Yıldırım, B. Yatay Kurplarda Taşıt Stabilitesinin Makine Öğrenmesi ile Modellenmesi. Uluslararası Sürdürülebilir Mühendislik ve Teknoloji Dergisi, 8 (1), 74-86, 2024

[14] Hornik K, Stinchcombe M, White H. Multilayer feedforward networks are universal approximators. Neural Networks, 2(5), 359–366, 1989.

[15] Murat Y.S. Prediction of Traffic Volumes in Bosphorus Bridge using Artificial Neural Networks. 11th Mini-EURO Conference on Artificial Intelligence in Transportation Systems and Science. Helsinki University of Technology.2-6 August 1999. Helsinki-Finland, 1999.

[16] Murat Y.S. Comparison of Fuzzy Logic and Artificial Neural Networks Approaches in Vehicle Delay Modeling. Transportation Research Part C-Emerging Technologies, 14(1), 316–334, 2006.

[17] Pedregosa F, et al. Scikit-learn: Machine learning in Python, Journal of Machine Learning Research, 12(Oct), 2825–2830, 2011.

[18] Paszke A, et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Advances in Neural Information Processing Systems 32, 8024–8035, Curran Associates, Inc. 2019.

[19] Hunter J. Matplotlib: A 2D graphics environment. Computing in Science & Engineering, 9(3), 90–95, 2007.