The Princeton Journal of Interdisciplinary Research
ISSN 3069-8200

Predicting Drag Coefficient from Automotive Design Parameters Using Machine Learning

Aarush Mohan (author)[1]

[1] Cambridge Centre of International Research, Bangalore, Karnataka, India

Abstract

This study explores the application of machine learning (ML) to predict the drag coefficient (Cd) of vehicles using design parameters, offering an alternative to conventional wind tunnel and computational fluid dynamics (CFD) methods. Using the publicly available DrivAerNet dataset, which contains over 4,000 vehicle configurations, the research focused on the 15 most influential geometric parameters identified through feature importance analysis. Multiple ML models were implemented, beginning with linear and polynomial regression to establish interpretable baselines, and extending to neural networks for capturing nonlinear aerodynamic relationships. Results show that while linear regression provided quick and transparent estimates, polynomial regression achieved moderate improvements, and tuned neural networks consistently delivered the highest predictive accuracy. The findings underscore the potential of ML to serve as a scalable, cost-effective, and efficient tool for aerodynamic assessment during the early stages of vehicle design. By reducing reliance on expensive simulations and physical testing, ML models enable engineers to

accelerate design workflows, optimize aerodynamics, and advance the development of energy-efficient vehicles. Future work may involve integrating 3D car mesh data and advanced deep learning methods such as graph neural networks or transformers to further enhance prediction accuracy and applicability.

Introduction

Aerodynamic efficiency is a key part of modern car design. It directly affects a vehicle's fuel efficiency, high-speed stability, and overall performance. The drag coefficient, or Cd, measures aerodynamic resistance. Car manufacturers have long aimed to keep this number as low as possible, especially with the growing demand for electric and fuel-efficient vehicles. Even small reductions in Cd can lead to significant fuel savings over time. For instance, lowering Cd by just 0.01 can improve highway fuel economy by 0.2 mpg in a mid-size car. Companies invest millions to optimize aerodynamics. A full-scale wind tunnel can cost over $50 million to construct and operate. Many prototypes have been scrapped due to poor drag performance. Early versions of the BMW i8 were significantly reworked to meet efficiency targets.

Traditional methods for measuring the drag coefficient rely on a wind tunnel or computational fluid dynamics (CFD). While these methods are accurate, they are also expensive, time-consuming, and often not practical during the early stages of design.

As the automotive industry becomes more digital, machine learning (ML) offers a way to streamline and speed up design workflows. Machine learning, a type of artificial intelligence, allows computers to identify patterns in data and make predictions without needing specific programming for every case. In vehicle design, ML models can quickly

analyze the relationship between geometric parameters and aerodynamic performance. This helps engineers make early design decisions without lengthy simulations or tests. ML is faster and cheaper, and it allows for real-time adjustments and improvements across various configurations. This flexibility is essential in today's fast-paced automotive design world.

This research looks at using machine learning models to predict the drag coefficient based on car geometric parameters found in the publicly available DrivAerNet dataset. The DrivAerNet dataset includes over 4,000 vehicle configurations along with their aerodynamic measurements, providing a strong foundation for training and testing ML models.

Literature Review

Recent years have seen a significant rise in research on the use of machine learning (ML) to predict aerodynamic performance metrics, such as drag coefficient (Cd), efficiently. This approach serves as an alternative to traditional computational fluid dynamics (CFD) and wind tunnel tests. For instance, Li et al. (2023) compared multiple linear regression, SVR (Support Vector Regression), and BP neural networks for predicting vehicle drag. They used PSO-tuned SVR with common car angles and found that SVR outperformed simpler methods in terms of prediction accuracy and generalization. The study emphasized the potential of ML to capture correlations between geometric parameters and aerodynamic performance.

Additionally, Dube and Hiravennavar (2020) evaluated several other ML models, including Random Forests, Kriging, Decision Trees, and Neural Networks, to predict drag contributions from underbody components like belly pans and airdams. Their models were trained using data from CFD simulations. The results demonstrated that even basic ML metamodels could estimate Cd with enough accuracy for engineering decisions.

A recent review in MDPI's Computers journal (2023) by Kalinja Naffer-Chevassier, Florian De Vuyst and Yohann Goardou, analysed surrogate models for predicting both drag and lift coefficients. It utilized XGBoost, Random Forest, SVR, and KNN. The authors concluded that XGBoost and Random Forest consistently achieved R^2 values over 0.8 for drag prediction, outperforming SVM and KNN, which struggled with more complex geometries. The review highlights the advantage of combined methods in aerodynamic surrogate modeling.

In another recent study, He et al. (2025) introduced the DrivAer Transformer, a point cloud transformer model trained on the DrivAerNet++ dataset, which includes 8,000 car shapes. This method predicts Cd and flow-field properties directly from 3D meshes, showing high precision and real-time inference, significantly reducing the dependence on traditional CFD during early-stage design.

While ML surrogate models show great promise, challenges remain. Overfitting is a frequent issue, especially with high-degree polynomials or deep networks trained on limited data. SVR models often need careful tuning of hyperparameters (e.g., PSO optimization), and while ensemble models can achieve strong accuracy, they may be less interpretable. Moreover, although transformer-based models like DAT perform well on large datasets with mesh inputs, they require significant computational resources and expertise to train.

Overall, the literature indicates that ML models, particularly PSO SVR, Random Forest, XGBoost, and transformer-based neural networks, can predict aerodynamic drag with minimal errors. However, there are trade-offs between interpretability, computational efficiency, generalizability, and data requirements.

The Princeton Journal of Interdisciplinary Research
ISSN 3069-8200

Methodology

Dataset Overview

This study utilized the DrivAerNet Parametric Dataset, an open-source collection of synthetic aerodynamic data generated for research on vehicle drag prediction. The dataset consists of 4,165 unique car body configurations, each with a respective average drag coefficient (Cd) value. Each car configuration was defined by combinations of more than 30 geometric design parameters such as car length, roof height, diffuser angle, spoiler size, and mirror position.

The dataset was well suited to train and test machine learning models due to its structure and the variations in parametric data that simulate real-world design variability provided a solid basis to evaluate prediction accuracy and generalization capability for vehicle aerodynamic prediction tasks.

Preprocessing

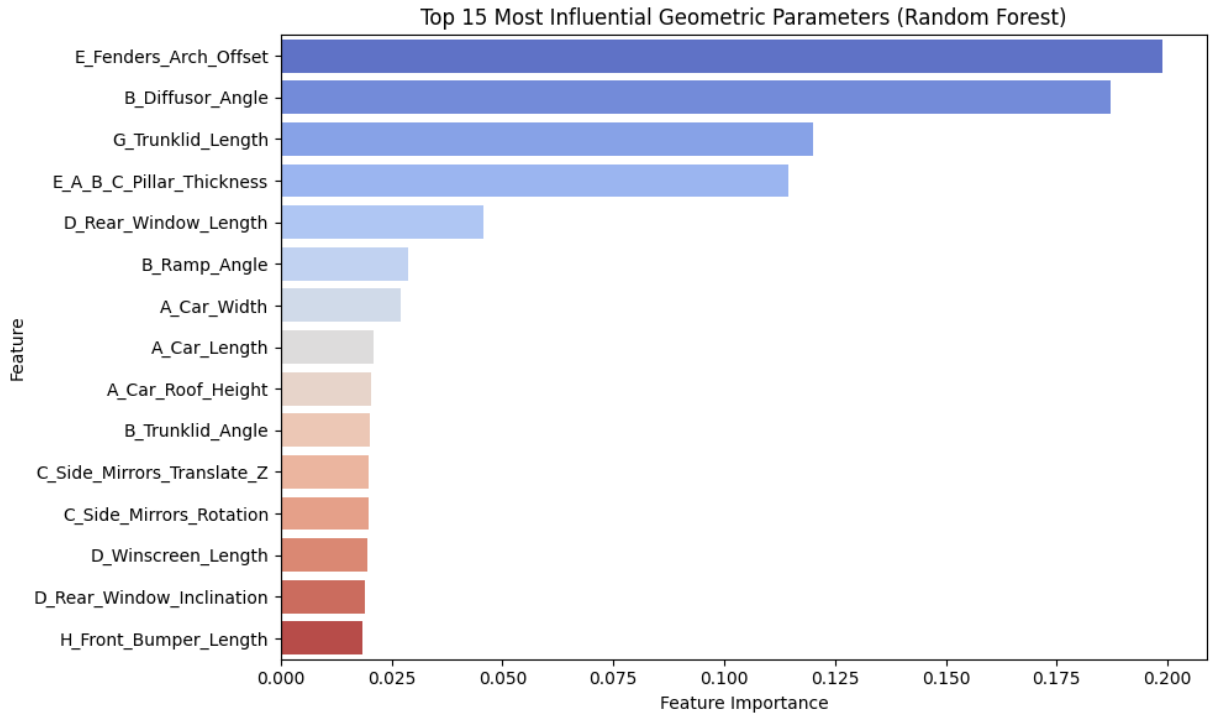
To keep the model simple and lower the risk of overfitting, the dataset was narrowed down to the 15 most influential geometric parameters based on feature importance scores from a Random Forest Regressor as shown in **Figure 1**. Choosing only the top 15 features ensures the model captures most of the predictive power without adding unnecessary complexity. Using all available parameters would significantly increase the required computational power. It could also introduce redundant or weakly correlated inputs, which might weaken the model's learning and hurt performance. On the other hand, picking too few features could lead to underfitting, where the model fails to capture important patterns in the

data. Fifteen features struck a balance. This number kept enough geometric variety to show aerodynamic influence while keeping the model efficient and able to generalize.

Before training, all feature values were normalized using standard scaling to ensure that no single variable dominated due to larger units. The dataset was then split into two parts, 80% for training and 20% for testing, to evaluate the models' performance on unseen data.

Figure 1

15 Most Influential Design Parameters Determined by Random Forest Regressor.



Model Selection

To predict the Cd based on a combination of vehicle design features, this study used regression models and neural networks, which are popular methods in machine learning for predicting continuous outcomes. These methods were combined to transition from simpler, more interpretable models to more complex ones that can capture nonlinear behavior.

Regression models were initially chosen because they provide a clear mathematical understanding of how each input affects the output. Starting with linear regression as a baseline due to its simplicity and ease of implementation, this model assumes a direct, proportional relationship between input variables and Cd, making it a good foundation for comparing models. However, because aerodynamic performance is affected by complex interactions among various geometric features, linear models may not capture all the details of the dataset.

To account for non-linearity and more complex relationships, polynomial regressions were used. Several polynomial degrees were tested, ultimately selecting one that balanced predictive performance with the risk of overfitting. Higher-degree polynomials can improve training accuracy but often overfit the data, leading to poor performance on unseen test samples.

Given the nonlinear nature of this multivariable problem, neural networks were also explored. These models can automatically learn complex relationships from the data without requiring manual feature engineering. Initially, a neural network was built using the PyTorch library due to its flexibility and user control in model construction. However, to enhance performance and automate hyperparameter tuning, Keras was chosen for the final neural network model. Its integration with KerasTuner allowed for an efficient exploration of various

layer configurations, neuron counts, and learning rates. This made Keras a more practical option for fine-tuning the model and ensuring it could generalize well.

Results and Analysis

The linear regression model was chosen as a starting point, as it is both simple and easy to interpret. The linear regression model returned an MSE of approximately 0.01589, and R^2 score of approximately 0.50499, indicating it could explain ~50% of the variance of drag coefficient. The linear regression model was extremely fast and transparent; however, it could not sufficiently capture the complex, non-linear interactions that existed between the 15 input variables. The residual plot, **Figure 3**, shows a random scatter with obvious spread within it, primarily concentrated below the baseline suggesting there was limited accuracy in the individual predictions. However, the linear regression model did provide an effective baseline benchmark to compare the more complex methods.

Figure 2

Linear Regression: Actual vs. Predicted Cd.

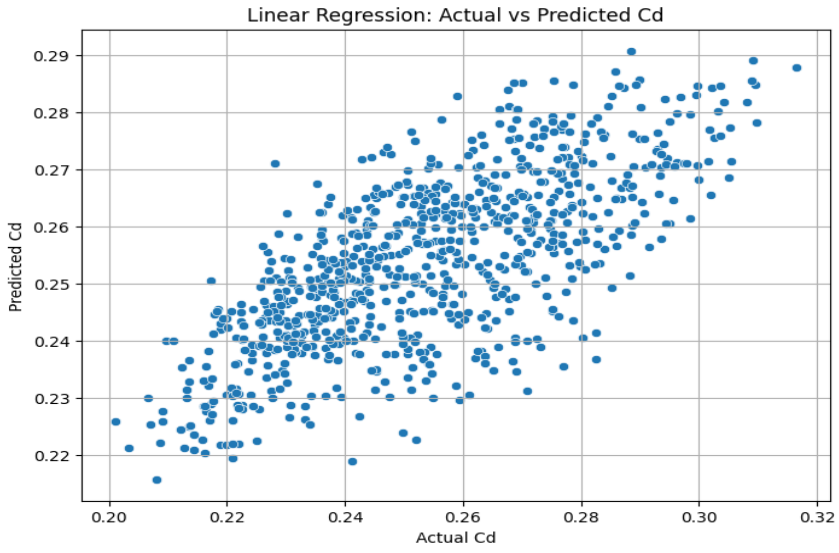
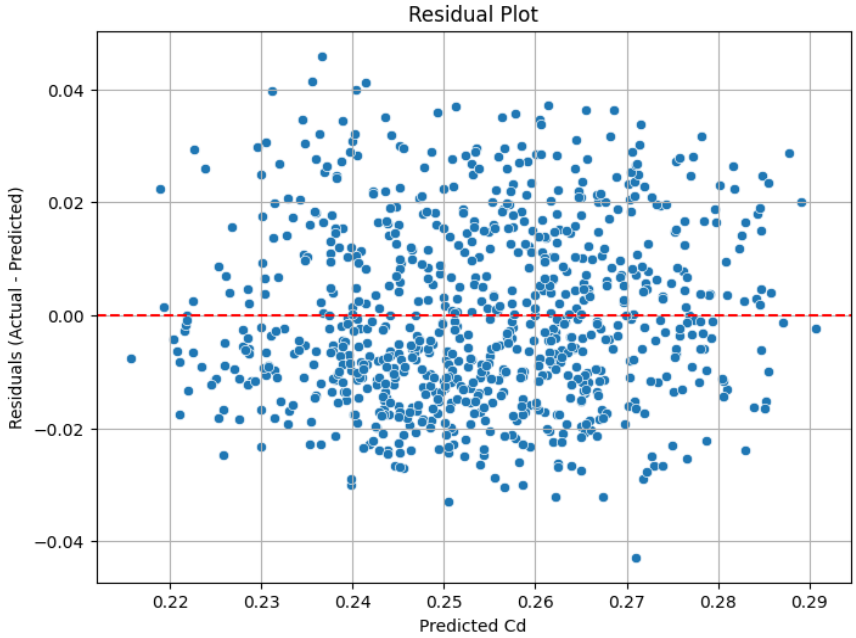


Figure 3

Linear Regression: Residual Plot.



To take this a step further, a polynomial regression model was implemented. By manually testing various polynomial degrees using trial and error, it was found that the 2nd degree provides the best results for standard regressions. This version achieved an improved MSE of approximately 0.01495 and an R^2 score of 0.5621. It was also noted that degree three onwards led to extreme overfitting (R^2 values were negative on the test set), as the model was starting to memorize the training data rather than learning generalizable patterns. This behaviour is common for polynomial models where the degree is increased too much or there is too much complexity involved.

Next, a neural network was built using the PyTorch library to learn more complex non-linear relationships. This model showed further improvements with an MSE of 0.0003, R^2 score

of 0.4585, and Mean Absolute Error of 0.0130. While the R^2 score was slightly lower than both regression models, these models produced more consistent results in terms of their predicting capabilities. However, the distribution of residuals**,** showed a tilt towards negative values, indicating that, on average, the model tends to under-predict drag coefficients. To improve the results, hyperparameter tuning was necessary. Therefore, another neural network was created, this time using the Keras library with KerasTuner. Before tuning, the model achieved an MSE of 0.0003, R^2 of 0.4669, and MAE of 0.0133 on the test dataset. These initial results were quite similar to the model built using PyTorch.

After implementing hyperparameter tuning, performance improved significantly. The optimized model reached an MSE of 0.0002, R^2 of 0.5226, and MAE of 0.0123. This shows a decrease in average prediction error and a higher degree of explained variance, meaning the model was better at capturing the true influence of input parameters on drag coefficient (Cd). Though the numerical differences may seem small, they represent a meaningful improvement in both accuracy and consistency, especially when working with high-resolution engineering data.

The visual outputs support this further. **Figure 6** shows tighter clustering of points compared to **Figure 4**, indicating more accurate predictions of the tuned model. Furthermore, the histogram of residuals, **Figure 7,** is more symmetrically distributed around zero, contrasting with that of the pre-tuned model, **Figure 5**, where residuals were more negatively skewed. This balance suggests that the tuned model reduced systematic bias, making it a more dependable predictor. Overall, these improvements highlight the importance of hyperparameter tuning in neural network performance. Even with the same underlying structure, adjusting factors like learning rate and hidden layer size led to a more robust and general model.

Figure 4

Keras Neural Network (Before Tuning): Actual vs. Predicted Cd.

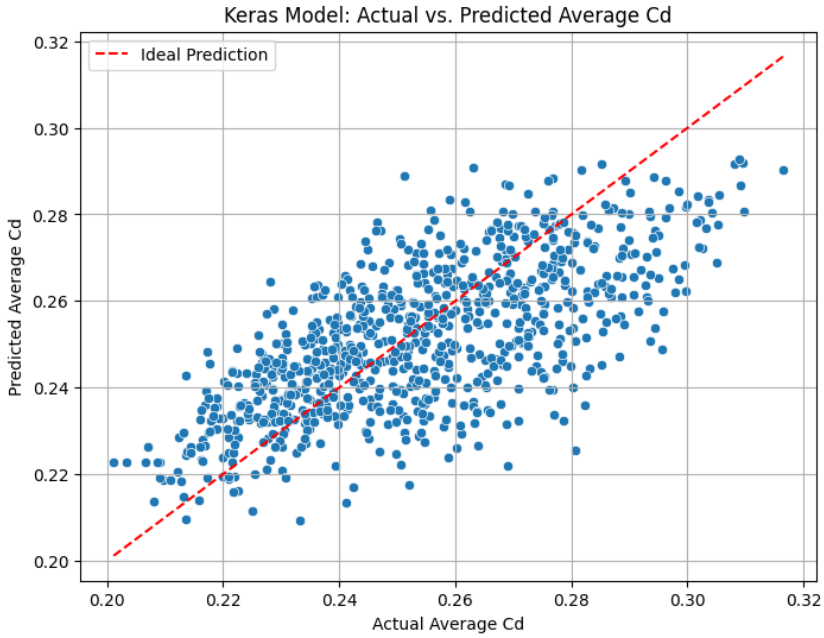


Figure 5

Keras Neural Network (Before Tuning): Histogram of Residuals

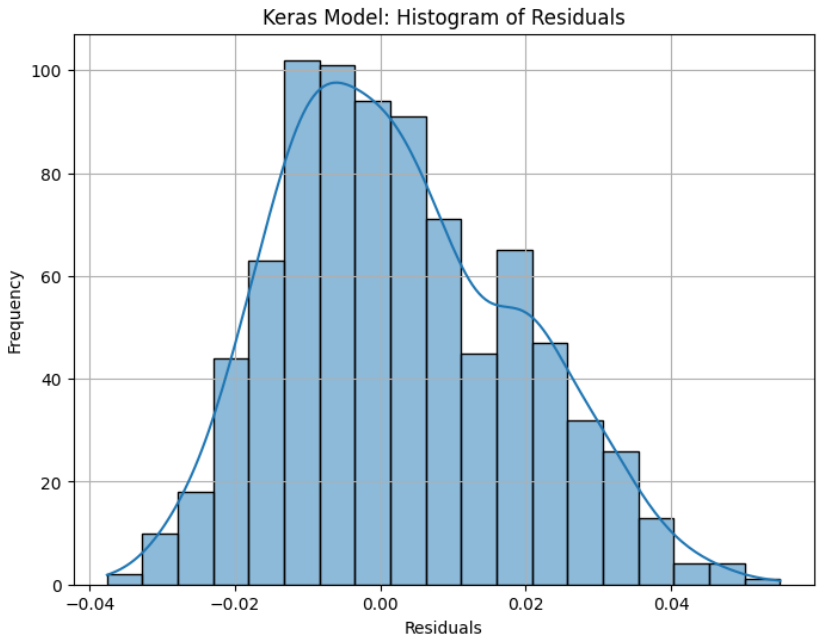


Figure 6

Keras Neural Network (After Tuning): Actual vs. Predicted Cd.

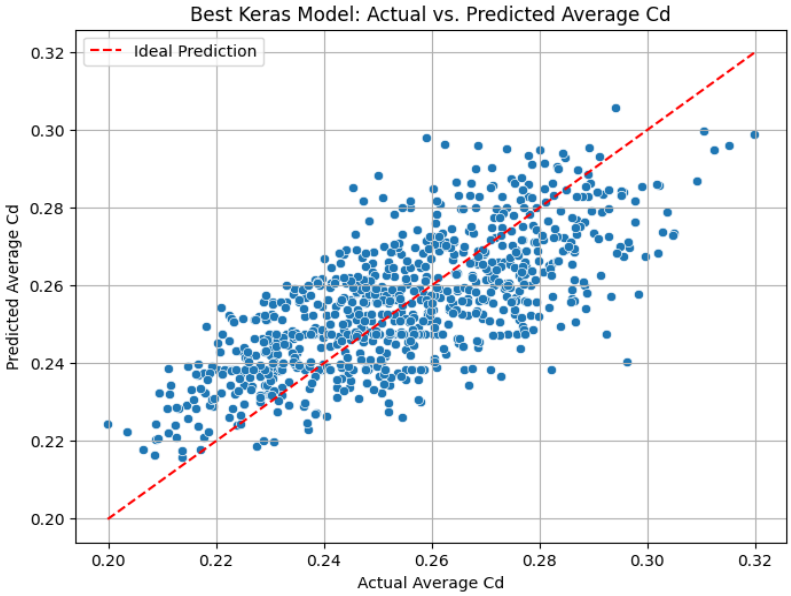
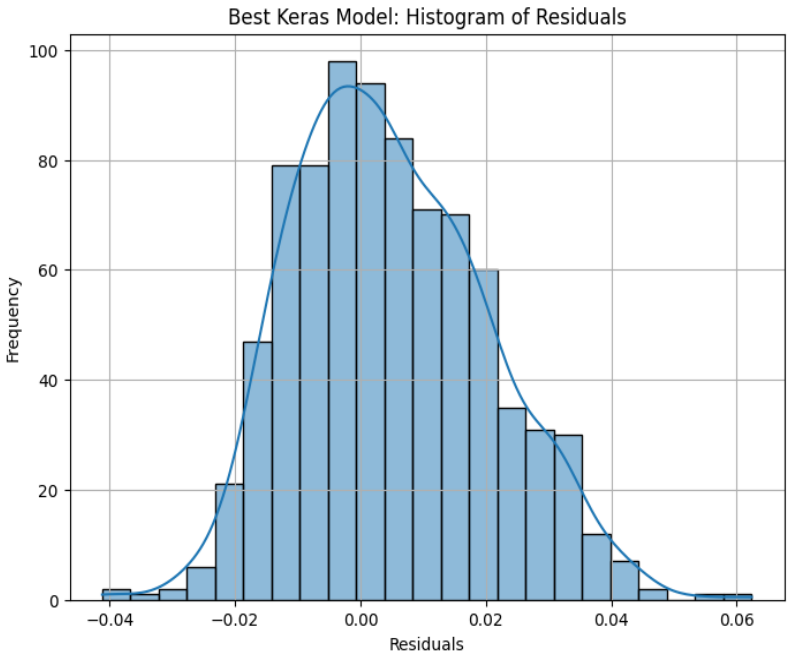


Figure 7

Keras Neural Network (After Tuning): Histogram of Residuals



The Princeton Journal of Interdisciplinary Research
ISSN 3069-8200

Conclusion

The study sought to evaluate machine learning models for estimating the aerodynamic drag coefficient (Cd) of car designs based on geometric parameters. This research examined over 4,000 simulated vehicle configurations in the DrivAerNet dataset. They focused on the 15 geometric parameters that had the most impact on model predictions. This analysis helped assess the predictive capability of various regression models, from simple linear models to more complex neural networks.

The results indicated that linear regression served as a reasonable baseline model that was quick to compute. However, it did not capture many nonlinear relationships in the designs. Polynomial regression performed slightly better, especially at degree 2, but higher-degree polynomial models may have overfitted the data. Neural networks provided the best overall results with consistency. The tuned Keras model produced balanced and fairly accurate outcomes. There were strong positive correlations between the predicted and actual Cd values, and the residuals showed little systematic bias.

This paper highlights that machine learning, including deep learning, can act as a scalable and cost-effective alternative to computational fluid dynamics (CFD) simulations in the early phases of vehicle design. By focusing on existing design parameters, one can quickly evaluate the aerodynamic performance of different vehicle designs, speeding up the process to create more efficient and sustainable models.

Looking ahead, there are several interesting avenues to explore. One option is to use the DrivAerNet++ dataset, which includes full 3D CAD meshes for each car configuration. These meshes can be analyzed with libraries like Open3D or Trimesh. This would enable the

development of deep learning models that learn directly from shape geometry, potentially utilizing 3D convolutional neural networks (3D-CNNs) or graph neural networks (GNNs). This approach would eliminate the need for manual feature engineering and allow the model to automatically learn important geometric features that impact drag.

Additionally, researchers could explore transfer learning techniques by pretraining models on synthetic simulation data and refining them with real-world wind tunnel or CFD datasets, enhancing their practical application. Another path is to create surrogate models that incorporate machine learning predictions into traditional design workflows, providing real-time feedback during 3D modeling. This could make it easier to conduct live aerodynamic assessments during the design of concept vehicles or prototypes.

As the automotive industry adopts digitization and AI-driven design, further integrating machine learning models into design and validation processes could significantly reduce time, cost, and reliance on physical testing. This shift would support the development of more aerodynamic and energy-efficient vehicles.

The Princeton Journal of Interdisciplinary Research
ISSN 3069-8200

References

American Psychological Association. (2017). Stress in America: The state of our nation. https://www.apa.org/news/press/releases/stress/2017/state-nation.pdf

Chen, Q., Elrefaie, M., Dai, A., & Ahmed, F. (2025). TripNet: Learning large-scale high-fidelity 3D car aerodynamics with triplane networks [Preprint]. arXiv. https://arxiv.org/abs/2503.17400

Chen, Q., Song, B., Zhang, Y., Xu, L., & Dai, A. (2024). Surrogate model for drag prediction of 3D car meshes using triplane networks. Designs, 12(10), 207. https://www.mdpi.com/2079-3197/12/10/207

Dube, P., & Hiravennavar, S. (2020). Machine learning based approach to predict aerodynamic drag force acting on the underbody components of a vehicle. SAE Technical Paper 2020-01-0684. https://doi.org/10.4271/2020-01-0684

Elrefaie, M., Dai, A., & Ahmed, F. (2024). DrivAerNet: A parametric car dataset for data-driven aerodynamic design and prediction [Preprint]. arXiv. https://arxiv.org/abs/2403.08055

Elrefaie, M., Morar, F., Dai, A., & Ahmed, F. (2024). DrivAerNet++: A large-scale multimodal car dataset with computational fluid dynamics simulations and deep learning benchmarks [Preprint]. arXiv. https://arxiv.org/abs/2406.09624

He, J., Luo, X., & Wang, Y. (2025). DrivAer transformer: A high-precision and fast prediction method for vehicle aerodynamic drag coefficient based on the DrivAerNet++ dataset [Preprint]. arXiv. https://arxiv.org/abs/2504.08217

Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., & Talwalkar, A. (2018). Hyperband: A novel bandit-based approach to hyperparameter optimization. Journal of Machine Learning Research, 18(1), 1–52. http://jmlr.org/papers/v18/16-558.html

Li, X., Wang, Y., Liu, Z., Chen, J., & Huang, S. (2025). Real-time drag coefficient prediction for platooning vehicles using ensemble machine learning. Processes, 13(7), 2056. https://www.mdpi.com/2227-9717/13/7/2056

Song, B., Xu, L., Dai, A., & Ahmed, F. (2023, June 9). Surrogate modeling of car drag coefficient with depth and normal renderings [Preprint]. arXiv. https://arxiv.org/abs/2306.06110

TensorFlow. (n.d.). Hyperparameter tuning with Keras tuner. https://www.tensorflow.org/tutorials/keras/keras_tuner