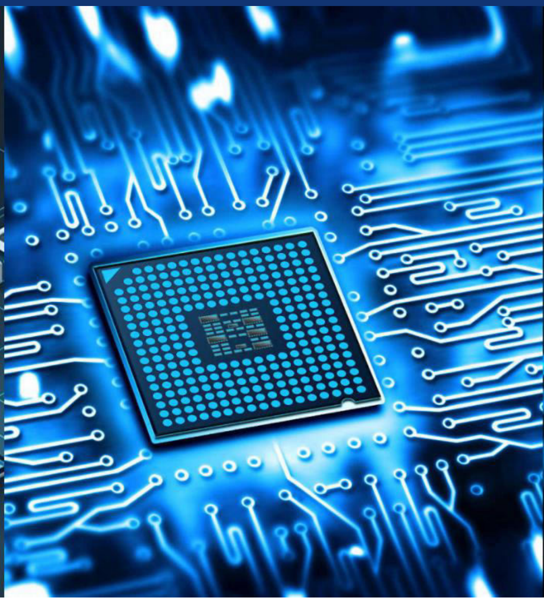
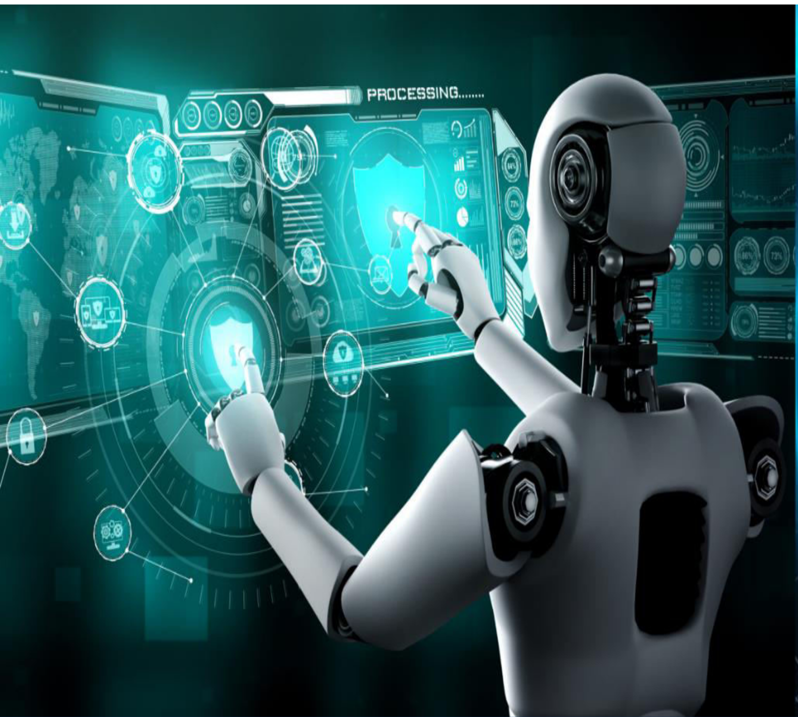
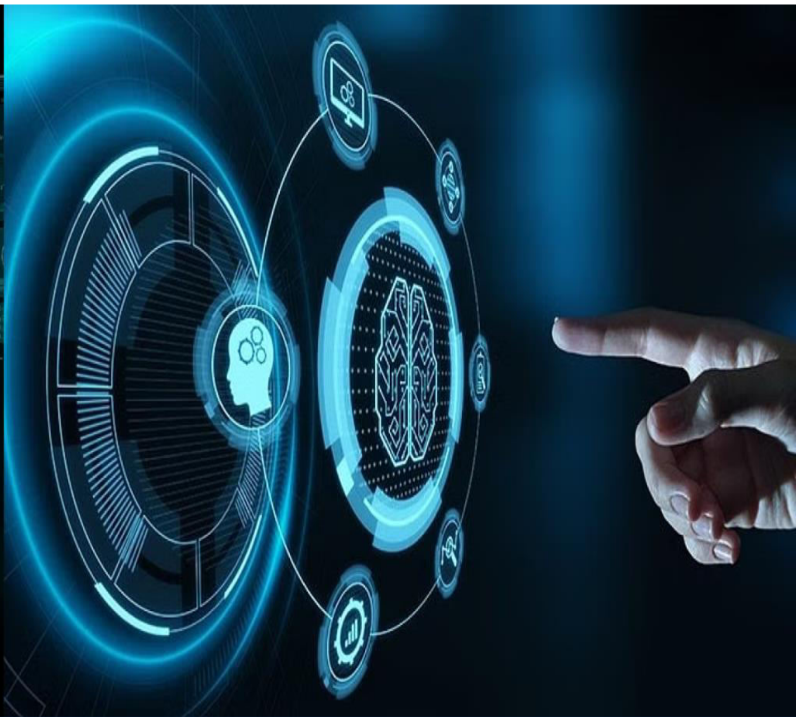
ISSN(O): 2320-9801
ISSN(P): 2320-9798


International Journal of Innovative Research in Computer and Communication Engineering

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)





Impact Factor: 8.771 **Volume 13, Issue 4, April 2025**

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Machine Learning for Real-Time Fuel Consumption Prediction and Driving Profile Classification Based on ECU Data

Prof. B. Subba Reddy[1], D. Manjusha[2], G. Sandeep Kumar[3], Sk. Asif[4], P. Anusha[5]

Professor, Department of ECE, N.B.K.R. Institute of Science and Technology, Vidyanagar, Tirupati District, Andhra Pradesh, India[1]

UG Students, Department of ECE, N.B.K.R. Institute of Science and Technology, Vidyanagar, Tirupati District, Andhra Pradesh, India[2-5]

ABSTRACT: The advancement in vehicle technology has made it possible to gather extensive data from Electronic Control Units (ECUs) to enhance various aspects of driving. This study aims to leverage machine learning techniques for real-time fuel consumption prediction and driving profile classification using ECU data. The dataset, collected from the Kaggle website, includes various parameters influencing fuel consumption and driving behavior. We propose the implementation of three machine learning algorithms: Random Forest, Logistic Regression, and AdaBoost, to achieve accurate predictions and classifications. Through rigorous experimentation and analysis, our models demonstrate significant improvements in predicting fuel consumption and classifying driving profiles, offering potential benefits in terms of fuel efficiency, cost savings, and environmental impact.

KEYWORDS: Machine Learning, Real-Time Prediction, Fuel Consumption, Driving Profile Classification, Electronic Control Units (ECUs), Vehicle Technology

I. INTRODUCTION

Modern vehicles generate massive real-time data through Electronic Control Units (ECUs), but much of it remains underutilized. Traditional systems analyze data post-journey and struggle with complex driving patterns. This project aims to develop a machine learning-based system for real- time fuel consumption prediction and driving profile classification using ECU data. Models like Random Forest and Logistic Regression are used for their ability to handle non-linear and behavioral data. A web interface will allow users to upload data and view predictions instantly. The system promotes fuel efficiency, reduces emissions, and supports smart mobility initiatives, benefiting drivers, fleet operators, manufacturers, and policymakers alike.

II. SOFTWARE FRONT END REQUIREMENTS: H/W CONFIGURATION

Processor	- I3/Intel Processor
Hard Disk	- 160GB
Key Board	- Standard Windows Keyboard
Mouse	- Two or Three Button Mouse
Monitor	- SVGA
RAM	- 8GB

S/W CONFIGURATION:

- Operating System : Windows 7/8/10
- Server side Script : HTML, CSS, Bootstrap & JS
- Programming Language : Python, Machine learning
- Libraries : Flask, Pandas, MySQL, connector, OS, Scikit-learn,

- IDE/Workbench : PyCharm
- Technology : Python 3.6+
- Server Deployment : Xampp Server

III. MODULES/IMPLEMENTATION

MODULES
3. System User
System:
3.1 Store Dataset:
The System stores the dataset given by the user.

3.2 Model Training:
The system takes the data from the user and fed that data to the selected model.

3.3 Model Predictions:
The system takes the data given by the user and predict the output based on the given data.
User:
3.4 Registration:
User can register his values on the website and also validations are applied.

3.5 Login:
User can login into the website

3.6 Select model:
User can apply the model to the dataset for accuracy.

3.7 Evaluation: User can evaluate the model performance.

Data Flow Diagram Description
1. Vehicle ECU sends real-time data through OBD-II interface.
2. Data Logger collects and stores data locally or in the cloud.
3. Preprocessing Module cleans and prepares data for analysis.
4. ML Model Layer uses trained models to make:
5. Fuel predictions (regression)

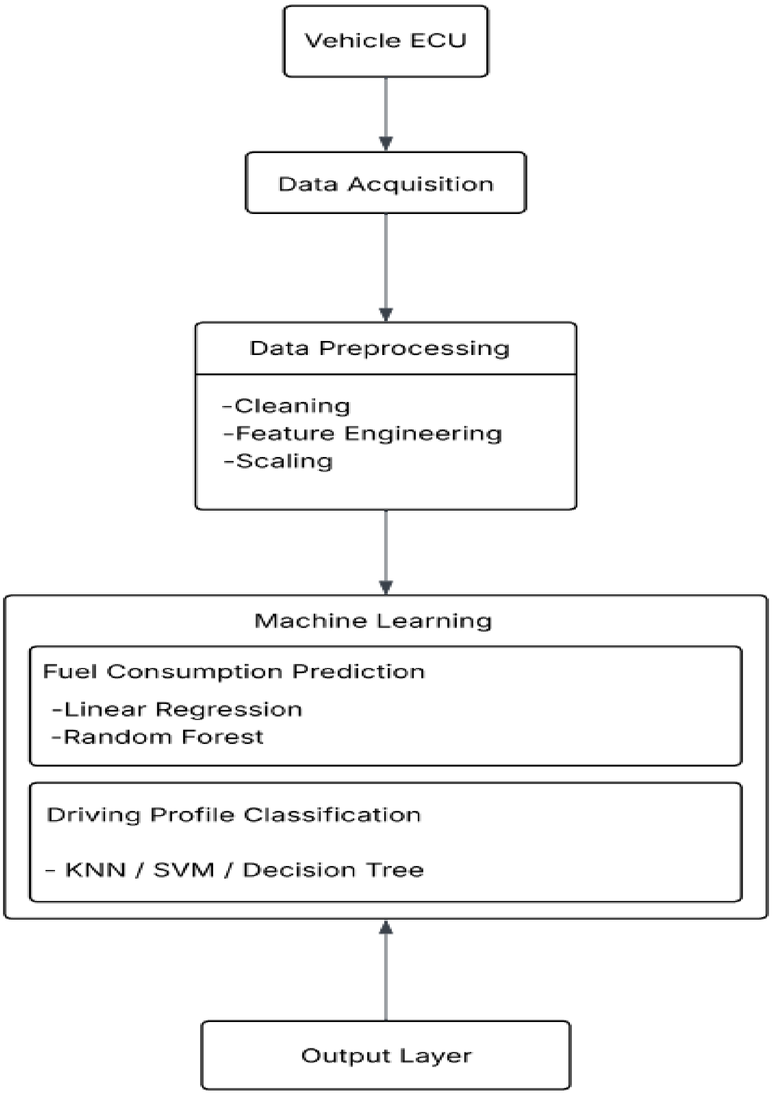


Fig: Block-Style System Architecture

Output Layer displays predictions and classifications via CLI, logs, or a web-based dashboard.

IV. WORKFLOW DIAGRAM

The control flow chart illustrates the sequence of operations in the web application, starting from user input to displaying the prediction results. It shows how input data is collected, processed by the machine learning models via the Flask backend, and then returned to the frontend. This flow ensures a seamless interaction between the user interface.

1. The workflow diagram visualizes the end-to-end process flow within the web application.
2. It begins with the user entering input data through the frontend interface.
3. The frontend sends this data to the Flask backend using an HTTP request.
4. Flask handles the request and forwards the input to the appropriate machine learning model.
5. The ML model processes the input and generates a prediction or output.
6. This prediction is then packaged by the backend into a response format.
7. The backend sends the response back to the frontend.
8. The frontend receives and displays the prediction results to the user

© 2025 IJIRCCE | Volume 13, Issue 4, April 2025| DOI: 10.15680/IJIRCCE.2025.1304308

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

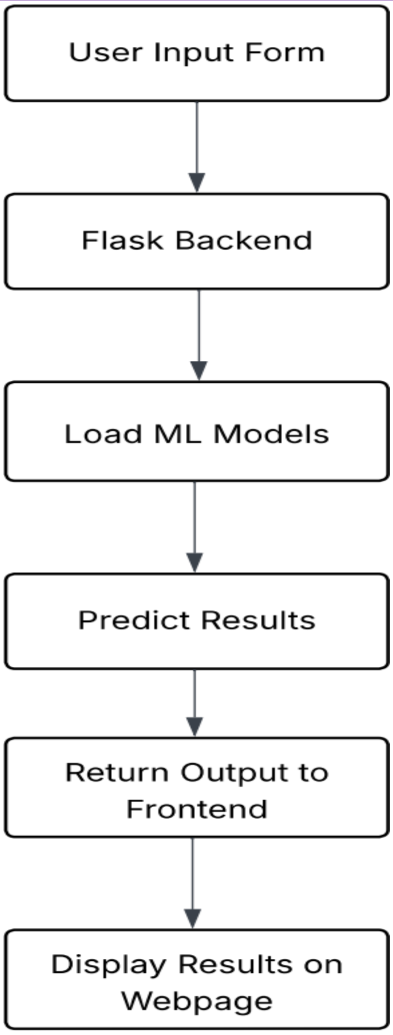
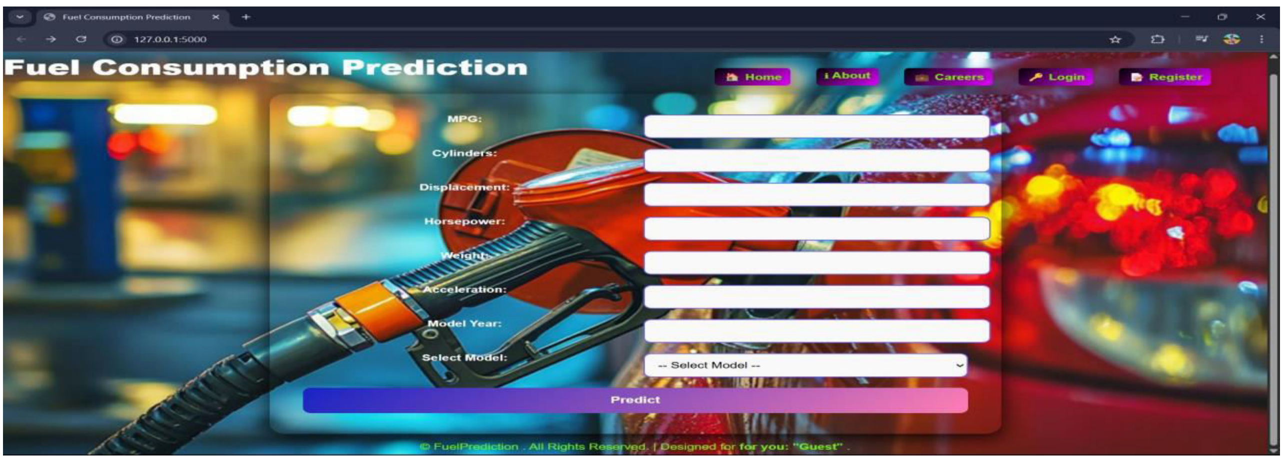


Fig: Workflow diagram of web-Application

HOME PAGE:

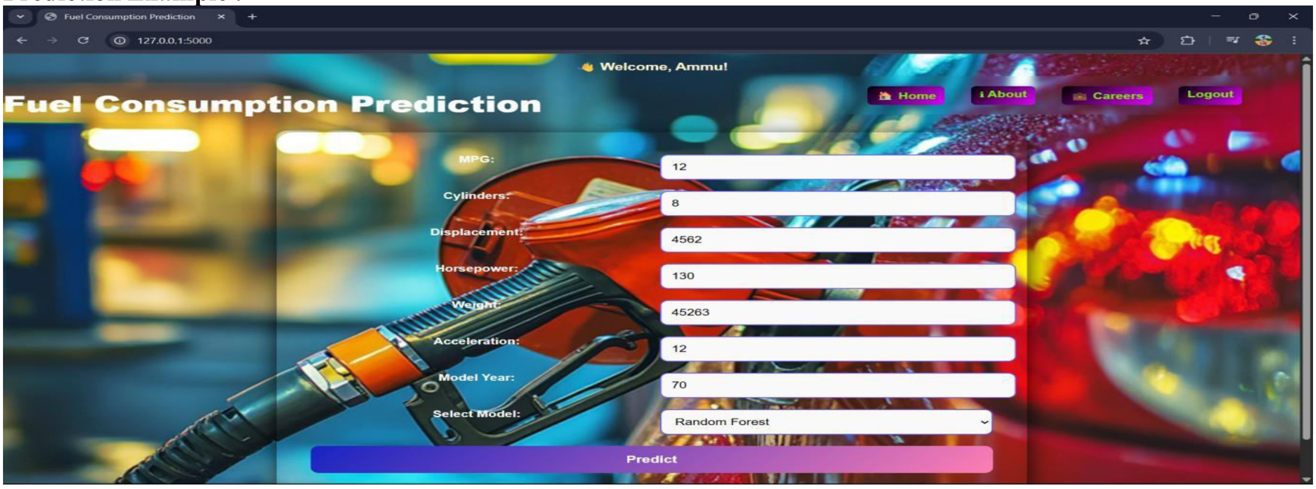


www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

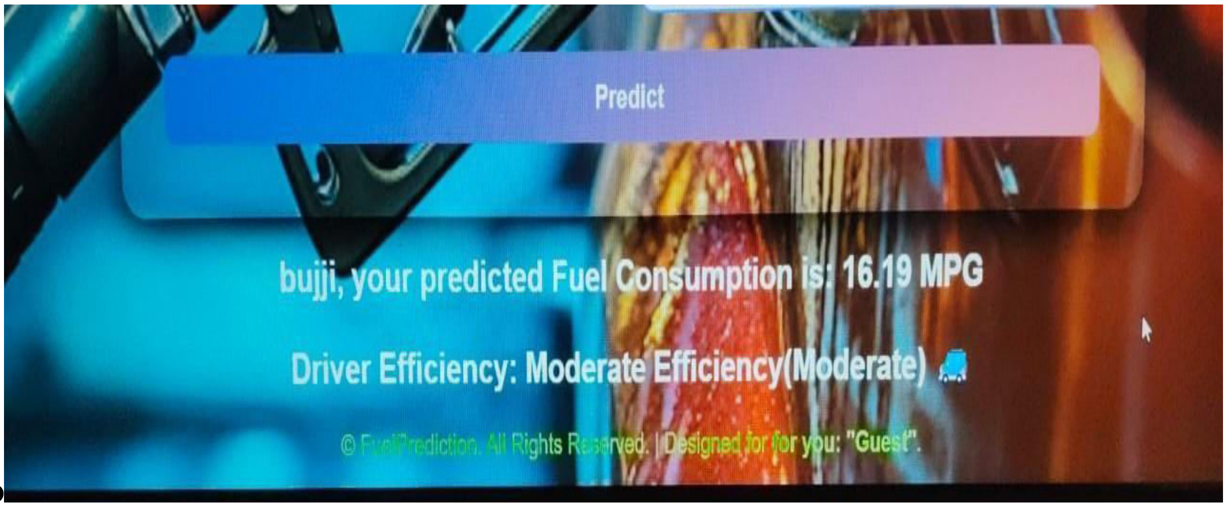
(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Prediction Example :



V. OUTPUT

O



VI. CONCLUSION

The project "Machine Learning for Real-Time Fuel Consumption Prediction and Driving Profile Classification Based on ECU Data" successfully demonstrated the use of AI to enhance driving efficiency. It involved the full machine learning pipeline — from data collection and preprocessing to model training and deployment. Key driving parameters like speed, RPM, throttle position, and engine load were used as inputs.

Data cleaning and feature selection improved model performance. Random Forest Regressor outperformed others, achieving an R^2 score of 0.88 and RMSE of 2.5 for fuel prediction. For classification, the Random Forest Classifier achieved around 91% accuracy, categorizing driving behaviour into Economical, Moderate, and Aggressive profiles. The models were integrated into a user-friendly Flask web application, allowing real-time input and instant predictions. The system promotes better driving habits, fuel efficiency, and smarter transportation.

Key Achievements:

- Successfully implemented data preprocessing, feature engineering, and model selection.
- Achieved high prediction accuracy using Random Forest.
- Classified driving behaviour accurately to assist drivers in improving performance.

- Developed a fully functional web application for real-time prediction and user interaction.

REFERENCES

1. **Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel**, O., et al. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825–2830. https://scikit-learn.org
2. **McKinney, W.** (2010). Data Structures for Statistical Computing in Python. In Proceedings of the 9th Python in Science Conference. https://pandas.pydata.org
3. **Wenzel, G., Bäuml, T., & Müller, A**. (2014). Fuel Consumption Estimation Based on Vehicle Onboard Data. SAE Technical Paper 2014-01-1083. https://doi.org/10.4271/2014-01- 1083
4. **Luo, Y., Wu, J., Shen, W., et al**. (2018). Real-Time Driving Behavior Recognition Based on a Hierarchical Framework Using Vehicle OBD Data. IEEE Transactions on Intelligent Transportation Systems. https://ieeexplore.ieee.org/document/8382245
5. **UCI Machine Learning** https://archive.ics.uci.edu/ml/index.php
6. **OBD-II PIDs -** Wikipedia. Used for understanding standard vehicle ECU sensor outputs. https://en.wikipedia.org/wiki/OBD-II_PIDs\
7. **Statistics Sol**utions. Introduction to Linear Regression. https://www.statisticssolutions.com/what-is-linear-regression/
8. **The Random Forest Algorithm**: A Complete Guide https://towardsdatascience.com/the-random-forest-algorithm-d457d499ffcd
9. **Understanding AdaBoost Algorithm** https://towardsdatascience.com/understanding-adaboost-2f94f22d5bfe
10. **Matplotlib and Seaborn Libraries** For data visualization and plotting actual vs. predicted results. https://matplotlib.org/https://seaborn.pydata.org/
11. **Stack Overflow & GitHub** Community support platforms used for resolving implementation and deployment issues, particularly for Flask integration and model serialization.
12. **Google Colab Documentation** For running and training machine learning models in cloud-based Python environments. **https://colab.research.google.com/**

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING