VNUHCM Journal of Science and Technology Development 2026, 29(3):4205-4213

Open Access Full Text Article

Research Article

IoT-Integrated Car Rental System with AI-Based Driving Behavior Evaluation Using an OBD-II Interface

Van Nhanh Nguyen[1], Huy Q. Tran[2,*], Vinh Khac Le[1]

ABSTRACT

In this paper, we present the design and development of an Internet of Things (IoT)-integrated car rental system that collects and monitors data remotely in real time using an OBD-II interface and a mobile application. The proposed system gathers critical information about both vehicles and users, including driver identity, trip time, driving speed, fuel consumption, and vehicle location. All data is transmitted through an IoT platform and stored on a cloud-based mobile application, enabling easy and transparent access for both users and service providers.

Beyond vehicle monitoring, the system integrates an optimized AdaBoost algorithm using GridSearchCV to evaluate driving behavior. This model analyzes data collected from the OBD-II interface to identify unsafe driving behaviors, such as frequent sudden braking, abrupt acceleration, and unstable speed control.

Experimental results reveal significant differences in the performance of the evaluated machine learning algorithms. Decision tree and logistic regression models yielded the lowest results, with accuracy values of approximately 69% and 68%, respectively, and their precision, recall, and F1-score metrics remained below 70%. In contrast, the random forest, support vector machine, and k-nearest neighbors models achieved better and relatively similar performances, with accuracy values around 73%. Among all models, our optimized AdaBoost algorithm delivered the best classification results, with accuracy, precision, recall, and F1-score all reaching approximately 80%.

Based on observed behaviors, driver profiles will be created and stored. These profiles can enhance service quality, support risk management, and enable more personalized rental policies. Test results confirm that the system is feasible for real-time operations and provides valuable insights into driver behavior.

Key words: car rental systems, car management, IoT, AI, driver behavior analysis, adaptive boosting

[1]*Institute of Engineering, HUTECH University, Ho Chi Minh City, Vietnam*

[2]*Robotics and Mechatronics Research Group, Faculty of Engineering and Technology, Nguyen Tat Thanh University, Vietnam*

Correspondence

Huy Q. Tran, Robotics and Mechatronics Research Group, Faculty of Engineering and Technology, Nguyen Tat Thanh University, Vietnam

Email: tqhuy@ntt.edu.vn

History

- Received: 09-03-2026
- Revised: 13-04-2026
- Accepted: 25-04-2026
- Published Online: 28-08-2026

DOI : https://doi.org/10.32508/vnuhcmj-std.v29i3.4683

INTRODUCTION

Currently, GPS technology is widely used in traffic management for applications such as real-time vehicle location tracking. This approach allows for the efficient use of available resources, facilitates route planning, and allows for timely coordination when needed, all at a low cost while ensuring safety and user-friendliness [1]. In cars, the engine control unit (ECU) manages and optimizes engine performance. However, repairing ECUs at small garages is often challenging due to limited access to proprietary diagnostic tools. The OBD-II communication interface provides a viable solution to this issue [2,3].

During OBD-II operation, real-time processing limitations may occur during RS232 serial communication. [4] developed a C# application to address this issue: the proposed system incorporates devices such as the STN1110 board and standard OBD cable. This model is then applied to various vehicles to collect fault codes, engine speed (RPM), and other sensor data.

In practice, we can fully monitor and evaluate vehicle performance and fuel consumption by analyzing data on temperature, airflow, and engine RPM. This enables accurate assessment of engine performance and early detection of abnormalities. Fuel consumption data collected via OBD-II is also used to evaluate driving behavior. Evaluations can be conducted with the same vehicle model under different traffic conditions along the same route [5]. These faults are categorized by vehicle type and year of manufacture to expedite the diagnostic process and enhance maintenance efficiency [6,7]. For example, [8] employed an integrated IoT system for real-time fault detection using a Raspberry Pi and an OBD-II device. This approach improves traffic safety through early warning systems and automated communication with relevant authorities.

Combining IoT systems with smartphone applications improves convenience and practicality. Recent studies have proposed intelligent systems for the real-time monitoring of bus operations, improving passenger experience, and streamlining manage-

Cite this article : Nguyen V N, Tran H Q, Le V K. **IoT-Integrated Car Rental System with AI-Based Driving Behavior Evaluation Using an OBD-II Interface**. *VNUHCMJ. Sci. Technol. Dev.* 2026; 29(3): 4205-4213.

ment. Notable features include emergency buttons, route tracking, and updates to arrival and departure times [9,10].

Researchers in previous studies [11,12] developed rental management applications for e-commerce platforms using mobile and IoT technologies. The system achieved 100% test coverage and exhibited high satisfaction levels. User and car owner satisfaction was reported at 96% and 100%, respectively. Beyond these standard features, the application also detects accidents and provides real-time monitoring. In practice, short-term rental cars often exhibit higher fuel consumption and emissions due to poor driving habits [13]. Consequently, monitoring driver behavior is essential to ensure the safety of both drivers and vehicles [14,15].

Despite extensive research on vehicle management and IoT technology applications in transportation, fully integrated and synchronized solutions remain scarce. Notably, systems capable of collecting and managing critical data on rental vehicles, such as speed, fuel consumption, location, and driver behavior, within a single comprehensive platform are still lacking. Consequently, developing an intelligent system to assist vehicle owners with management is crucial. This system should not only store vehicle records and track financial and legal information before and after rental, but also support the real-time monitoring of vehicle operational status.

To address this need, our research proposes the application of an optimized AdaBoost algorithm to enhance the efficiency of analyzing and evaluating data collected from IoT devices, thereby improving vehicle monitoring and management in rental systems. This study contributes to the field of smart rental car systems by innovatively integrating IoT, AI, and mobile technologies. The key contributions of this work include:

- Integrating a data collection and management model that combines OBD-II, IoT, and a cloud platform enables the acquisition of real-time vehicle data, such as location, fuel consumption, speed, and driver identity.
- Developing a mobile application to offer transparent access to vehicle sta-tus and driving data for both users and service providers, which enables re-mote fleet management and improves renter accountability.
- Implementing optimized AdaBoost with GridSearchCV to analyze OBD-II data for detecting unsafe driving behaviors, allowing for real-time driving risk assessment.

The rest of this paper is organized as follows: Section 2 introduces the proposed model and mobile application; Section 3 details the machine learning-based driver behavior assessment; Section 4 describes the experimental setup and presents key results; and Section 5 offers conclusions and suggests future work.

SYSTEM OVERVIEW

This section outlines the hardware architecture of the data collection system, with a focus on its IoT components and their integration with the mobile application.

System Hardware Design

In this system, a vehicle (e.g., a 2019 Toyota Vios) serves as the primary data source (Figure 1). The IoT system connects directly to the car's OBD-II port to gather operational data, including vehicle speed, fuel consumption, engine status, and diagnostic trouble codes. Data from the OBD-II port is read using the ELM327 module, which allows for data collection through the car's OBD-II system and transmits it wirelessly to a microcontroller via Wi-Fi.

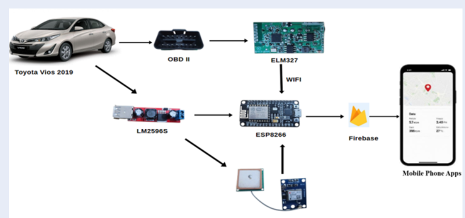

Figure 1: IoT-Based car rental monitoring system architecture [Source: Authors].

The ESP8266 functions as the system's central processor and communication module. It receives data from the ELM327, performs preliminary processing, and transmits the data to the cloud computing platform. To maintain a stable power supply for electronic components, the system incorporates an LM2596S step-down module. This module converts and stabilizes voltage from the vehicle's power source to levels suitable for microcontrollers and sensors. Concurrently, the NEO-6M GPS module acquires the vehicle's geographic coordinates and real-time location data, enabling route tracking and remote monitoring of the vehicle's location.

All data collected from the sensors and modules will be synchronized by the ESP8266 and transmitted to Google's Firebase cloud platform for storage, management, and processing. The information will then be displayed visually in a mobile application developed by the team. This application enables users or service providers to easily monitor the vehicle's operating status, current location, and other important parameters. The complete experimental model is shown in Figure 2. Based on the provided system description and block diagram, Table 1 summarizes the main hardware and software components along with their relevant technical specifications. The system also integrates AI algorithms to analyze driving behavior (e.g., harsh braking or sudden acceleration) based on OBD-II data, automatically issue risk alerts, and build personalized driver profiles.

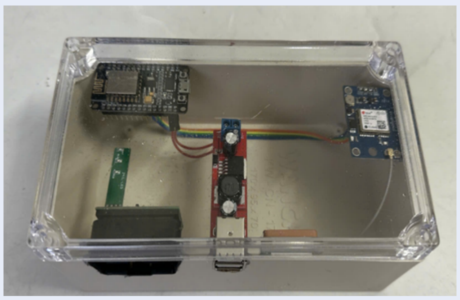

Figure 2: IoT-Based car rental monitoring model [Source: Authors].

Based on the PID code, data are obtained from the OBD-II port as described in Table 2. The ESP8266 can easily access relevant data. Specifically, code 0x0D retrieves vehicle speed, while code 012F retrieves fuel consumption level.

Car Management App

Car Management is a vehicle rental management application developed by our team using Flutter software. This app is designed to meet the practical needs of rental businesses with an intuitive, user-friendly interface. It features login functionality for employees and managers and includes vehicle management tools.

The interface displays complete information about all vehicle types, including manufacturer and model details. The "Add New Vehicle" feature allows users to add new cars along with relevant information. The application also integrates a feature to look up old contracts, which is useful when signing contracts with existing customers. This feature also helps track current rental statuses and manages the vehicle return process by contract code.

The administration dashboard provides all transaction information, supports reporting, and analyzes business performance. The Car Management application automates the entire management process, from vehicle inventory management to contract monitoring. In addition, the application includes features that can manage rental vehicle information and monitor driver behavior, as illustrated in Figure 3.

Before using the service, users are clearly informed about the types of data collected and must provide consent through a rental agreement. The service provider also commits to ensuring data privacy and security in accordance with established policies.

DRIVER BEHAVIOR ANALYSIS USING AN ADABOOST ALGORITHM

In practice, abnormal driving behaviors can pose significant risks, including accidents during operation, engine damage, or, less severely, increased fuel consumption. To assess a driver's attitude and safety level while driving, three behavioral indicators are used: speeding, sudden braking, and sudden acceleration.

In Vietnam, speeding is defined as exceeding the permitted speed by 5 km/h or more. [14] defines sudden braking as negative acceleration exceeding -3.0 m/s^2 over a short period. When drivers exceed these thresholds, it indicates reckless driving and an unstable state of mind. Table 3 details high-risk driving behaviors, their detection criteria, and safety implications.

The dataset used in this study was collected from approximately 50 drivers under real-world driving conditions, resulting in a total of 1,000 samples. Each sample includes overspeed and acceleration values obtained from the OBD-II system. In this work, driver behavior assessment is formulated as a multi-class classification problem, where the input features are overspeed and acceleration, and the output labels include three categories: good, normal, and poor. These labels are defined based on the number of detected violations, including speeding, sudden braking, and sudden acceleration.

The dataset captures variability among drivers with differing driving styles and conditions. In practice, class distribution is slightly imbalanced, as good and normal driving behaviors occur more frequently than poor behavior. This imbalance mirrors real-world scenarios but may affect classification performance. However, the AdaBoost algorithm addresses this issue by assigning higher weights to misclassified samples, encouraging the model to prioritize challenging and minority cases. Furthermore, measurement noise is incorporated to simulate real operating conditions, enhancing the proposed model's robustness.

Table 1: System componentsand technical specifications [Source: Authors].

Component	Specification
Toyota Vios 2019	Supports Communication protocol: ISO 15765-4 (CAN) OBD-II port voltage: 12 V
ELM327 OBD-II Interface	OBD-II / EOBD compliant Wireless communication: Wi-Fi Reads PIDs codes Operating voltage: 12 V
ESP8266	MCU: Tensilica L106 32-bit Clock frequency: 80–160 MHz Integrated Wi-Fi 802.11 b/g/n Operating voltage: 3.3 V
LM2596 DC–DC Buck Converter	Input voltage: 4–40 V Output voltage: 1.25–35 V (adjustable) Maximum output current: 3 A Conversion efficiency: up to 92%
NEO-6M GPS Module	Chipset: u-blox NEO-6M Positioning accuracy: ~2.5 m Communication interface: UART Operating voltage: 3.3–5 V
Firebase Cloud Platform	Realtime Database / Firestore Real-time data synchronization REST API support
Mobile Application	Platform: Android / iOS Displays vehicle location, speed, and system status Receives risk and safety alerts
AI-based Analytics Module	Detection of harsh braking and sudden acceleration Driver behavior profiling and risk assessment Input data: OBD-II parameters and GPS data

Table 2: Car PIDs codes obtained from OBD-II [Source: Authors].

Parameters	PIDs Code
Fuel consumption level	012F
Speed	0X0D

Table 3: High-risk driving behaviors and their detection criteria [Source: Authors].

Driving Behavior	Detection Criterion
Speeding	Vehicle speed exceeds the legal limit by $\geq$ 5 km/h
Hard braking	Longitudinal acceleration ≤ -3.0 m/s^2 within 0.5–1.0 s
Sudden acceleration	Longitudinal acceleration $\geq +3.0$ m/s^2

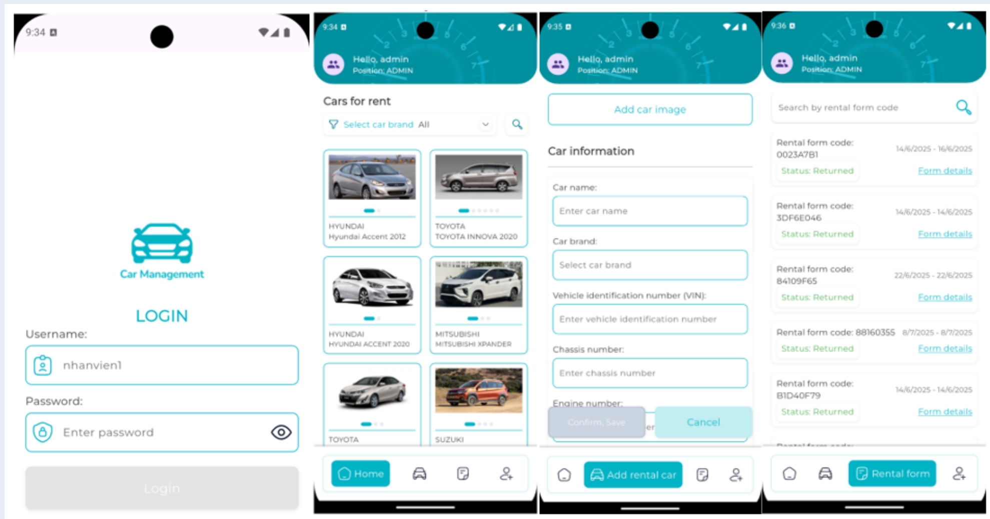

Figure 3: Car management mobile application [Source: Authors].

AdaBoost is an advanced machine learning algorithm that combines weak models to improve prediction accuracy. This study utilizes an optimized AdaBoost algorithm that incorporates a decision tree. Decision trees are a simple machine learning algorithm that is well-suited for evaluating driving behavior. Specifically, the model detects unsafe driving patterns, including sudden acceleration, abrupt braking, and speeding. This model is integrated with a real-time IoT system that continuously updates the vehicle's status during the rental period. The system's results and analyses allow vehicle owners to make informed decisions, such as reminding drivers or adjusting rental fees based on contractual agreements. In addition, misclassified models were iteratively refined to improve deep analysis accuracy. These adjustments make our proposed model highly effective for processing large data streams from multiple connected devices.

To perform the driver behavior assessment process, all collected data were formatted as follows, where n represents the number of samples:

$$D = \{(x1, y1), (x2, y2), \ldots, (xn, yn)\} \tag{1}$$

where $x_i = [Overspeed_i, Acceleration_i]$ and $y_i \in \{poor, normal, good\}$.

The data were then normalized as follows:

$$x' = \frac{x - \mu}{\tau}$$

where $\mu = mean(x)$ and $\tau = std(x)$.

The AdaBoost algorithm constructs a strong classifier by combining multiple weak learners. The key principle of this method is to increase the weights of misclassified samples in subsequent iterations, thereby directing the model's focus toward them. The training process involves several boosting iterations. In the first iteration, each sample's weight is calculated as follows:

$$w_i^{(1)} = \frac{1}{N} \tag{2}$$

where $i = 1, 2, \ldots, N$.

At each iteration $\boldsymbol{t}$, a weak classifier $h_t(x)$ is trained using the weighted dataset. This study employs a shallow decision tree as the weak classifier to preserve the simplicity of the model. The weak classifier has the following structure[16]:

$$h_t(x) : X \rightarrow \{1, 2, \ldots, K\} \tag{3}$$

where:

- $h_t(x)$is the prediction of the weak classification at iteration t.
- $\boldsymbol{K}$ is the number of classes.

After training the weak classifier, the model error at iteration t is calculated based on the total weight of the misclassified samples[16].

$$\varepsilon_t = \sum_{i=1}^{N} w_i^{(t)} I(y_i \neq h_t(x_i)) \tag{4}$$

where:

- $\varepsilon_t$is the error of the classifier at iteration t

- $w_i^{(t)}$ is the weight of sample i at iteration t
- $I(\cdot)$ is the indicator function:

$$I(condition)\begin{cases} \mathbf{1}\ if\ condition\ is\ True \\ \mathbf{0}\ if\ condition\ is\ False \end{cases} \quad (5)$$

In this study, weights are assigned to individual samples rather than specific driving behaviors. Misclassified samples receive higher weights in subsequent iterations, enabling the model to prioritize challenging cases and enhance overall performance. The contribution of each weak classifier to the final model is determined by the coefficient α_t. This coefficient reflects the reliability of the classifier at iteration $\boldsymbol{t}$:

$$\sigma_t = \eta \frac{1}{2} \ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right) \quad (6)$$

In Equation 6, the learning rate coefficient η adjusts the influence of each weak classifier.

After each iteration, the weights of the training samples are adjusted to increase the importance of misclassified samples. The weight update formula is defined as follows [16]:

$$w_i^{(t+1)} = w_i^t e^{-\sigma^t y_i h_t(x_i)} \quad (7)$$

After $\boldsymbol{T}$ boosting loops, the final model is constructed by aggregating all weak classifiers using their respective weights [17].

$$F(x) = \sum_{t=1}^{T} \sigma^t\ h_t\ (x) \quad (8)$$

The final classification decision is determined by the highest value of the composite function [17].

$$H(x)\ =\ arg\ max_K\ \sum_{t=1}^{T} \sigma^t I(h_t\ (x) = k) \quad (9)$$

Table 4 summarizes the key configuration parameters of the optimized AdaBoost model. It details the primary settings of the ensemble model, such as the base learner structure, learning rate, and number of estimators.

RESULTS AND DISCUSSION

In this study, we collected real-world data from 50 different car renters to evaluate the effectiveness of the proposed AI model. Data collection and evaluation were performed using an IoT device directly connected to the vehicles via OBD-II. We linked this device to a mobile application, allowing car owners to receive quick, accurate, and comprehensive updates on vehicle information. Driver behavior was categorized based on the number of violations for each specific offense: no violations indicated good driving behavior, one violation corresponded to average behavior, and two or more violations were classified as poor behavior.

The collected data were updated and displayed directly on the mobile application. Figure 4 shows the actual vehicle speed recorded by the mobile application via the IoT model developed for this study. The results illustrate changes in vehicle speed over time. This approach allows us to easily observe different phases of movement, including steady driving as well as acceleration and deceleration. In addition, instances of sudden deceleration or acceleration by the driver can be identified.

Figure 4: Real car speed chart [Source: Authors].

The confusion matrix was used to evaluate the effectiveness of the AdaBoost algorithm in assessing driving behavior. Behaviors were categorized into three groups: Poor, Normal, and Good (Figure 5). The confusion matrix indicates that the model made 156 correct predictions out of 200 samples, resulting in an overall accuracy of approximately 78%. The "Good" class was most accurately identified, with 106 correct predictions out of 122 actual samples. The "Poor" class also performed relatively well, with 12 out of 16 samples correctly classified and only a few misclassified as "Normal" or "Good". The "Normal" class had the lowest identification rate, with only 38 out of 62 samples correctly predicted, including 22 samples mistakenly classified as "Good". This is indicative of an overlap between the "Normal" and "Good" classes, making it difficult for the model to distinguish between these two states. This confusion can be attributed to the fact that "Normal" and "Good" driving behaviors often exhibit similar characteristics, es-

Table 4: Main AdaBoost parameters [Source: Authors].

Parameter	Value
Base estimator	Decision tree classifier
Base estimator depth	max_depth = 2
Number of estimators	n_estimators = 50
Learning rate	learning_rate = 1.0
Boosting algorithm	SAMME.R
Loss function	Exponential loss
Random state	42
Input features	2
Output classes	3

pecially when the number of violations is low. Consequently, their feature distributions tend to overlap, making it challenging for the model to clearly separate these two classes. In addition, the use of only two input features (overspeed and acceleration) may limit the model's ability to capture more subtle differences in driving behavior. Variations in individual driving styles and real-world noise in the collected data may also contribute to misclassification.

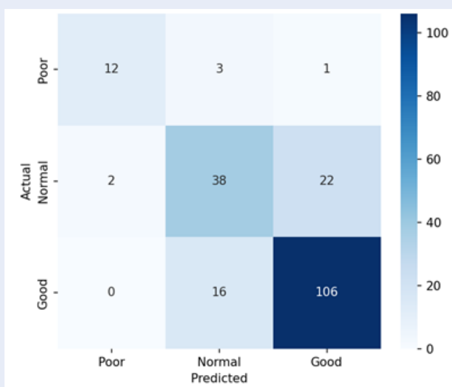

Figure 5: Confusion matrix [Source: Authors].

The proposed method's effectiveness was assessed by comparing its performance metrics—including accuracy, precision, F1-score, and recall—with those of classic machine learning algorithms: random forest, decision tree, support vector machine (SVM), k-nearest neighbors (kNN), and logistic regression.

Figure 6 shows that the decision tree and logistic regression algorithms yielded relatively poor results, with accuracy rates of approximately 69% and 68%, respectively. The precision, recall, and F1-score metrics for these two models were also below 70%. In contrast, the random forest, SVM, and kNN algorithms performed better and exhibited a high degree of similarity across their metrics. Specifically, the random forest model exhibited an accuracy of approximately 73%, while the SVM and kNN models both reached around 73.5% accuracy. The precision, recall, and F1-score values for these three models also hovered around 73%. However, AdaBoost demonstrated the best classification performance, with all four metrics reaching values of approximately 80%. This indicates that combining the boosting mechanism with the optimized parameters of the AdaBoost model significantly improved its predictive capabilities.

To assess the effectiveness of the proposed model and minimize bias from random data partitioning, we performed cross-validation as detailed in Table 5. Specifically, we used a 5-fold cross-validation method; i.e., the entire dataset of 1000 samples was divided into five equal parts. In each iteration, four parts (80%) were designated for training, while the remaining part (20%) served as the test set. This process was repeated five times, with each data segment used as a test set once. The final result represents the average of these five evaluations. Experimental results indicated that AdaBoost achieved the highest cross-validation score (0.778). In comparison, SVM and kNN both reached approximately 0.752; random forest attained around 0.717; while decision tree and logistic regression had the worst scores at 0.698 and 0.714, respectively. These findings demonstrate that the AdaBoost model exhibits a more stable performance than the other classification algorithms assessed in this study.

This study aims not only to evaluate driving behavior but also to gather data on vehicle location, fuel consumption, and travel routes. This information will

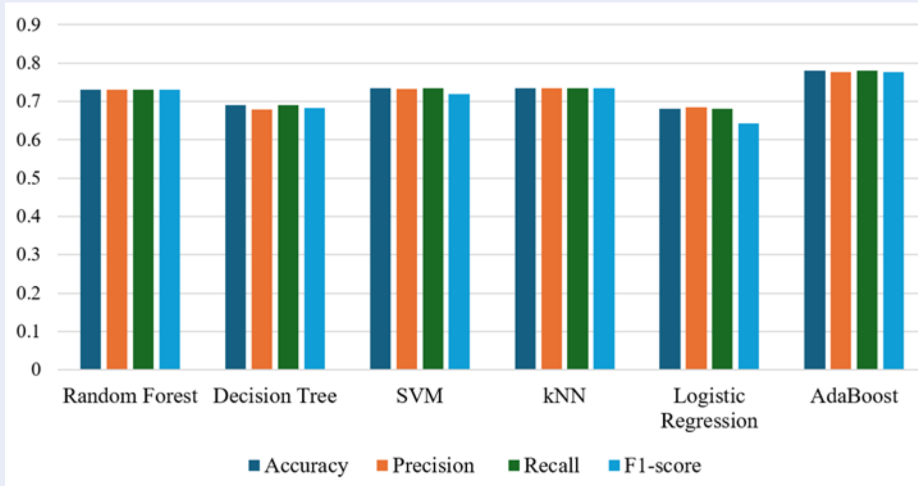

Figure 6: Performance comparison between AdaBoost and other classification algorithms [Source: Authors].

Table 5: Cross-validation accuracy of classification models [Source:Authors].

Model	Cross-Validation Accuracy (5-Fold)
Random Forest	0.717
Decision Tree	0.698
SVM	0.752
kNN	0.752
Logistic Regression	0.714
AdaBoost	0.778

streamline and enhance the reliability of rental car management processes. Specifically, this data will be fully recorded to support penalty processing for traffic violations during vehicle operation. In addition, the database will be used to adjust reward points as well as guide future rental fees for individual customers.

While the system performs effectively on a small scale, scaling this system to hundreds of vehicles presents several challenges, including increased data load, potential delays, and more complex data storage requirements. To mitigate these issues, future work will involve processing some data directly on the device, thereby reducing the burden on the cloud. In addition, the system will adopt more flexible cloud architectures for easier scalability. Enhanced communication methods, such as MQTT or Kafka, will be implemented to improve data transmission efficiency. The system will also be redesigned into smaller modules to facilitate management and expansion. Finally, the AI model will be optimized to enable faster real-time processing.

CONCLUSION AND FUTURE WORK

In this study, we report on the successful development of a comprehensive model, encompassing both software and hardware, for data collection and monitoring in intelligent self-driving car rental systems. The hardware component employed IoT technology to facilitate remote signal transmission and connectivity. Integrating the data collection system with OBD-II and IoT allows for continuous data acquisition and real-time performance tracking. A mobile application also supported continuous, thorough data collection and management. We gathered data on renter identity, vehicle type, current vehicle status, driving speed, fuel consumption, and travel route throughout each trip. Finally, AI was leveraged to accurately and impartially evaluate driver behavior, providing valuable insights for future rental decision-making and pricing strategies.

In the future, we aim to update and include features such as vehicle maintenance reminders, vehicle insurance status updates, and integration with government databases for automated fine processing. These improvements are crucial for optimizing operational efficiency and fostering trust between service providers and renters.

ABBREVIATIONS

Adaboost: Adaptive boosting
AI: Artificial Intelligence
CAN: Controller Area Network
DC–DC: Direct Current to Direct Current
ECU: Engine Control Unit
EOBD: European On-Board Diagnostics
GPS: Global Positioning System
GridSearchCV: Grid Search Cross-Validation
IoT: Internet of Things
kNN: k-Nearest Neighbors
MCU: Microcontroller Unit
OBD-II: On-Board Diagnostics II
PIDs: Parameter IDs
REST API: Representational State Transfer Application Programming Interface
RPM: Round Per Minute
SAMME.R: Stagewise Additive Modeling using a Multiclass Exponential loss function
SVM: Support Vector Machine
UART: Universal Asynchronous Receiver/Transmitter

COMPETING INTERESTS

The authors hereby declare there are no conflicts of interest associated with this work.

AUTHORS' CONTRIBUTIONS

Van Nhanh Nguyen: Conceptualization, Investigation, Methodology, Data Analysis, Writing – original draft. Huy Q. Tran: Supervision, Investigation, Methodology, Writing –review & editing. Vinh Khac Le: Software, Data curation, Data Analysis.

ACKNOWLEDGEMENTS

We would like to thank Nguyen Tat Thanh University, Ho Chi Minh City, Vietnam for the support this study.

REFERENCES

1. Verma R, Singh BK, Zahidi F. Management of GPS tracking systems in transportation. Singapore: Springer; 2024. Available from: https://doi.org/10.1007/978-981-97-0515-3_11.
2. Sim AX, Sitohang B. OBD-II standard car engine diagnostic software development. In: Proceedings of the 2014 International Conference on Data and Software Engineering (ICODSE). Bandung, Indonesia: IEEE; 2014. p. 1–5. Available from: https://doi.org/10.1109/ICODSE.2014.7062704.
3. Rybitskyi O, Golian V, Golian N, Dudar Z, Kalynychenko O, Nikitin D. Using OBD-2 technology for vehicle diagnostic and using it in the information system. Bulletin of National Technical University "KhPI". Series: System Analysis. Control and Information Technologies. 2023;2023(1(9)):97–103. Available from: https://doi.org/10.20998/2079-0023.2023.01.15.
4. Yoo C, Ko Y. Development of the vehicle diagnosis program using OBD-II. Transactions of the Korean Society of Automotive Engineers. 2015;23(3):271–8. Available from: https://doi.org/10.7467/ksae.2015.23.3.271.
5. Rimpas D, Papadakis A, Samarakou M. OBD-II sensor diagnostics for monitoring vehicle operation and consumption. Energy Reports. 2020;6:55–63. Available from: https://doi.org/10.101/egyr.2019.10.018.
6. Jhou JS, Chen SH, Tsay WD, Lai MC. The implementation of OBD-II vehicle diagnosis system integrated with cloud computation technology. In: Proceedings of the Second International Conference on Robot, Vision and Signal Processing (RVSP 2013). Kitakyushu, Japan: IEEE; 2013. p. 9–12. Available from: https://doi.org/10.1109/RVSP.2013.55.
7. Salih TA, Younis NK. Designing an intelligent real-time public transportation monitoring system based on IoT. OAlib. 2021;8(10):1–14. Available from: https://doi.org/10.4236/oalib.1107985.
8. Singh SK, Singh AK, Sharma A. OBD II based intelligent vehicular diagnostic system using IoT. In: Jain S, Groppe S, editors. Proceedings of the International Semantic Intelligence Conference 2021 (ISIC 2021), CEUR Workshop Proceedings. vol. 2786; 2021. p. 511–515.
9. Sutar SH, Koul R, Suryavanshi R. Integration of smartphone and IoT for development of smart public transportation system. In: Proceedings of the 2016 International Conference on Internet of Things and Applications (IOTA). Pune, India: IEEE; 2016. p. 73–78. Available from: https://doi.org/10.1109/IOTA.2016.7562698.
10. Nagaraju CH, Kondagandla R. IoT-based live monitoring public transportation security system by using Raspberry Pi, GSM GPS. Cham: Springer; 2022. Available from: https://doi.org/10.1007/978-3-030-96634-8_43.
11. Zainuddin AA, et al.. Integration of car rental system with mobile app management and IoT for optimised resources and safety in IIUM.; 2024. Available from: https://doi.org/10.1063/5.0212464.
12. Ogbiti JT, Aaron W. Development of a web-based car rental management system. TheScientificWorldJournal. 2024;19(3):797–804. Available from: https://doi.org/10.4314/swj.v19i3.27.
13. Adamczak M, Kovacic-Lukman R, ToboA, Tórz M, Cyplik P. Role of IoT solutions in reducing CO2 emission and road safety in car rental and car sharing market. Cham: Springer; 2020. Available from: https://doi.org/10.1007/978-3-030-24355-5_12.
14. Osafune T, Takahashi T, Kiyama N, Sobue T, Yamaguchi H, Higashino T. Analysis of accident risks from driving behaviors. International Journal of Intelligent Transportation Systems Research. 2017;15(3):192–202. Available from: https://doi.org/10.1007/s13177-016-0132-0.
15. Zhou Y, Fu C, Jiang X, Yu Q, Liu H. Who might encounter hard-braking while speeding? Analysis for regular speeders using low-frequency taxi trajectories on arterial roads and explainable AI. Accident; Analysis and Prevention. 2024;195. Available from: https://doi.org/10.1016/j.aap.2023.107382.
16. Freund Y, Schapire RE. A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences. 1997;55(1):119–39. Available from: https://doi.org/10.1006/jcss.1997.1504.
17. Zhu J, Zou H, Rosset S, Hastie T. Multi-class AdaBoost. Statistics and its Interface. 2009;2(3):349–60. Available from: https://doi.org/10.4310/SII.2009.v2.n3.a8.