A DRIVING DECISION STRATEGY (DDS) BASED ON STACKING APPROACH

Sk. Shafiulilah Assistant professor, Department of Computer Science Engineering, Raghu Engineering College, Visakhapatnam, Andhra Pradesh.

S. Sumathi, S. Siva Ganesh, P. Gayatri 4th year Students, Department of Computer Science Engineering, Raghu Engineering College, Visakhapatnam, Andhra Pradesh.

P. Manohar Assistant professor, Department of Computer Science Engineering, Raghu Engineering College, Visakhapatnam, Andhra Pradesh.

shafiulilah.sk@raghuenggcollege.in, 20985A0519@raghuenggcollege.in , 19981a05e7@raghuenggcollege.in, 19981a05d2@raghuenggcollege.in , manohar.pattisipu@raghuenggcollege.in,

Abstract- *The current driving strategy of autonomous vehicles relies solely on external parameters and overlooks the vehicle's internal condition. To tackle this issue, the proposed study introduces a Driving Decision Strategy (DDS) based on ensembling techniques, which considers both external and internal aspects such as consumable conditions and RPM levels to calculate the optimal driving strategy for autonomous vehicles. The DDS utilizes sensor data from cloud-stored vehicles and has been validated in previous studies using random forest and MLP algorithms. However, in this project, the authors have employed a hybrid approach called stacking, which combines random forest and MLP models with boosting techniques to achieve better results. The proposed system implements a hybrid classification approach using stacking, which has the potential to improve the DDS's performance.*

.

Keywords--- RPM Levels, Driving Decision Strategy, Machine learning, Autonomous Vehicles, Stacking Approach.

I. INTRODUCTION

Recognition, judgment, and control are the three levels at which self-driving automobile operating principles can be divided. Vehicles come with a variety of sensors, such as GPS, cameras, and radar, to help with the recognition process. This data is used to determine a driving strategy in the judgment step. Following the identification of the driving environment, analysis is done, and objectives and suitable driving plans are created. After the control step, the vehicle begins to move on its own. An autonomous vehicle takes several stages, repeating the actions of recognition, judgment, and control on its own, to get where it is going. With increased performance, autonomous cars become more effective at processing data [8]. The electrical system of the car could be overloaded if these sensors are used more frequently. Self-driving cars use computers within the vehicle to analyse sensor data. The amount of computed data increases, and as a result of overload, judgment, and control become slower. The stability of the vehicle could be compromised by these issues. To prevent sensor overload, various studies have proposed different approaches. Some of these approaches involve developing hardware that can perform complex computations within the vehicle, while others utilize cloud computing to process sensor data [1].

Problem Statement:

 DOI:10.36893.DRSR.2023.V13I04.089-097

The current driving strategy of autonomous vehicles ignores the vehicle's internal conditions, which may lead to suboptimal driving decisions. Therefore, there is a need to develop a Driving Decision Strategy (DDS) that considers both external and internal aspects to calculate the optimal driving strategy for autonomous vehicles.

The proposed study introduces a DDS based on ensembling techniques that utilizes sensor data from cloud-stored vehicles. Previous studies have validated the DDS using random forest and MLP algorithms. However, there is a need to explore more advanced approaches that can further improve the DDS's performance.

To address this challenge, in this paper we have proposed a hybrid approach called stacking that combines random forest and MLP models with boosting techniques. The aim is to develop a hybrid classification approach that can enhance the DDS's accuracy and reliability.

II. LITERATURE SURVEY

In recent years, the development of autonomous driving technology has gained significant attention. Machine learning techniques have been applied to various aspects of autonomous driving, such as sensory data integration, object detection, tracking, and decision-making. In this literature survey, we review six research papers that focus on machine learning techniques for autonomous driving.

Al-Tahmeesschi et al. (2020) proposed a machine learning-based system for autonomous driving that integrates sensory data, detects and tracks dynamic objects, and predicts their future movements. The system uses a combination of convolutional neural networks (CNNs) and recurrent neural networks (RNNs) to process sensory data and predict object movements [2].Guo et al. (2020) proposed a deep reinforcement learning-based autonomous driving algorithm that learns to drive in different scenarios, such as intersections and roundabouts, by interacting with the environment. The algorithm uses a combination of CNNs and long short-term memory (LSTM) networks to process sensory data and generate driving commands [3].Lee et al. (2020) conducted a survey of machine learning techniques for autonomous driving. They categorized the techniques into six categories: object detection and tracking, behaviour prediction, motion planning, control, simulation, and validation. They discussed the advantages and limitations of each technique and highlighted some future research directions [4].Kumar et al. (2019) proposed a neural network-based approach for lane detection and vehicle tracking. The approach uses a CNN to detect lanes and a Kalman filter to track vehicles. The approach achieved high accuracy in lane detection and vehicle tracking [5].Liu et al. (2019) proposed a hybrid algorithm for an intelligent autonomous driving system that uses a support vector machine (SVM) and fuzzy logic controller (FLC). The algorithm processes sensory data, such as camera and lidar data, and generates driving commands. The FLC helps to make the driving commands smoother and more stable [6].Joo et al. (2020) proposed an urban autonomous driving system that uses a hierarchical decision-making framework. The system consists of three levels: perception, planning, and control. The perception level uses a deep learning-based object detection algorithm, and the planning level uses a rule-based algorithm. The control level generates driving commands based on the output of the planning level [7]. In conclusion, machine learning techniques have shown great potential in various aspects of autonomous driving, such as object detection, tracking, decision-making, and control. The proposed approaches in the reviewed papers have achieved high accuracy and demonstrated good performance in different scenarios. However, more research is needed to address some of the challenges in autonomous driving, such as real-time processing, scalability, and safety.

A. Data Set Explanation

 DOI:10.36893.DRSR.2023.V13I04.089-097

To execute this project, we are utilizing a historical vehicle trajectory dataset since we do not have sensors to gather data. The trajectory dataset contains sensor values with corresponding class labels, such as "lane changing," when the user slows down the vehicle. The dataset is comprised of various classes based on the values. The machine learning algorithm will be trained on this dataset, and when we apply test data.

trajectory_id	**start_time**	**end_time**	**rpm_average**	**rpm_medium**	**rpm_max**	**rpm_std**	**speed_average**	**speed_medium**	**speed_max**	**speed_std**	**labels**
2.01E+13	2007-11-08T01:05:36.000000000	2007-11-08T01:12:14.000000000	3.561229	4.237586	4.709072	1.428515	-0.00985	-0.00271	0.135384	0.091201	speed
2.01E+13	2008-06-18T09:22:10.000000000	2008-06-18T09:22:45.000000000	21.70835	18.1923	54.6717	15.65798	3.123229	1.868308	16.83847	7.667033	steering_angle
2.01E+13	2008-06-18T09:53:17.000000000	2008-06-18T10:01:31.000000000	1.384173	1.30718	7.174272	0.882563	0.016952	0.020387	2.082622	0.361382	steering_angle
2.01E+13	2008-06-19T11:57:02.000000000	2008-06-19T12:01:50.000000000	3.217921	2.19409	12.54107	2.793892	-0.03803	-0.02231	2.400153	0.712168	lane_change
2.01E+13	2008-06-19T23:47:07.000000000	2008-06-20T01:10:51.000000000	1.509991	1.05224	29.80859	1.957342	0.002243	0.001583	3.059844	0.365482	lane_change
2.01E+13	2008-06-20T02:45:28.000000000	2008-06-20T03:12:05.000000000	2.87243	2.220617	13.68806	2.479465	-0.01625	0.002861	2.864375	0.596121	steering_angle
2.01E+13	2008-06-20T04:21:40.000000000	2008-06-20T05:05:47.000000000	6.190253	6.050348	24.4936	4.017136	-0.03948	0.016615	7.599066	0.810596	steering_angle

Table 1: **Sample dataset**

B. Data Pre-processing

Data processing involves a sequence of operations for validating, organizing, transforming, merging, and extracting data to produce an appropriate output. Thorough documentation of the processing methods is crucial to ensure the accuracy and usefulness of the data. As our dataset contains text data in some columns, the primary task is to convert this textual data into a numerical format. The 'train_ test_ split' function from the 'sklearn' library can be used to divide the dataset into training and testing subsets. This function allows us to partition the data based on a specified split ratio. A recommended split ratio is 80/20, where 80% of the data is used for training and 20% for testing.

C. Proposed Architecture Model

 DOI:10.36893.DRSR.2023.V13I04.089-097

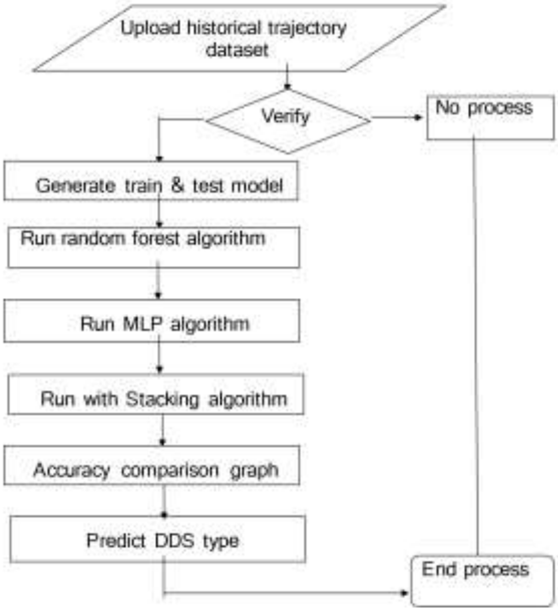

Figure 1: **Architecture diagram**

D. Random Forest Algorithm

The Random Forest Algorithm is a popular ensemble learning technique that can be used to solve both classification and regression problems. It works by creating a collection of decision trees based on random subsets of the training data. At each node of a decision tree, a random subset of features is selected to find the best split. The trees are grown until they reach their maximum depth or a stopping criterion is met.

Overall, the Random Forest Algorithm is a powerful tool for machine learning tasks and can help improve the accuracy and robustness of models by combining the predictions of multiple decision trees.

E. Multi-layer Perceptron Algorithm

The Multilayer Perceptron (MLP) algorithm is a type of artificial neural network that has widespread use in classification and regression tasks. It consists of an input layer, one or more hidden layers, and an output layer, with each layer's nodes connected to the next. These connections have weights, which the MLP algorithm adjusts iteratively to minimize the difference between the predicted and actual outputs. It uses backpropagation to calculate the output layer's error and propagate it back through the network to modify the weights. Due to its ability to model intricate non-linear relationships, the MLP algorithm finds extensive application in fields like finance, healthcare, and image and speech recognition.

F. Stacking Approach

DOI:10.36893.DRSR.2023.V13I04.089-097

A stacking algorithm, also known as a stacked generalization, is a machine learning ensemble method that combines multiple models to improve predictive performance. In stacking, the predictions from several models are used as inputs to a meta-model, which then generates the final prediction.

The stacking process works as follows:

A set of base models is trained on the training data.

The base models make predictions on the validation data.

The validation data and the base model predictions are used to train a meta-model.

The meta-model is then used to make the final predictions on the test data.

 DOI:10.36893.DRSR.2023.V13I04.089-097

III. RESULTS AND ANALYSIS

In this paper, we introduce a Driving Decision Strategy that utilizes trajectory data more accurately than previous models. Our approach involves a Stacking algorithm that combines MLP and RANDOM FOREST algorithms to analyse the internal data of the vehicle, including steering and RPM levels, to make predictions about various actions such as speed and lane changes.

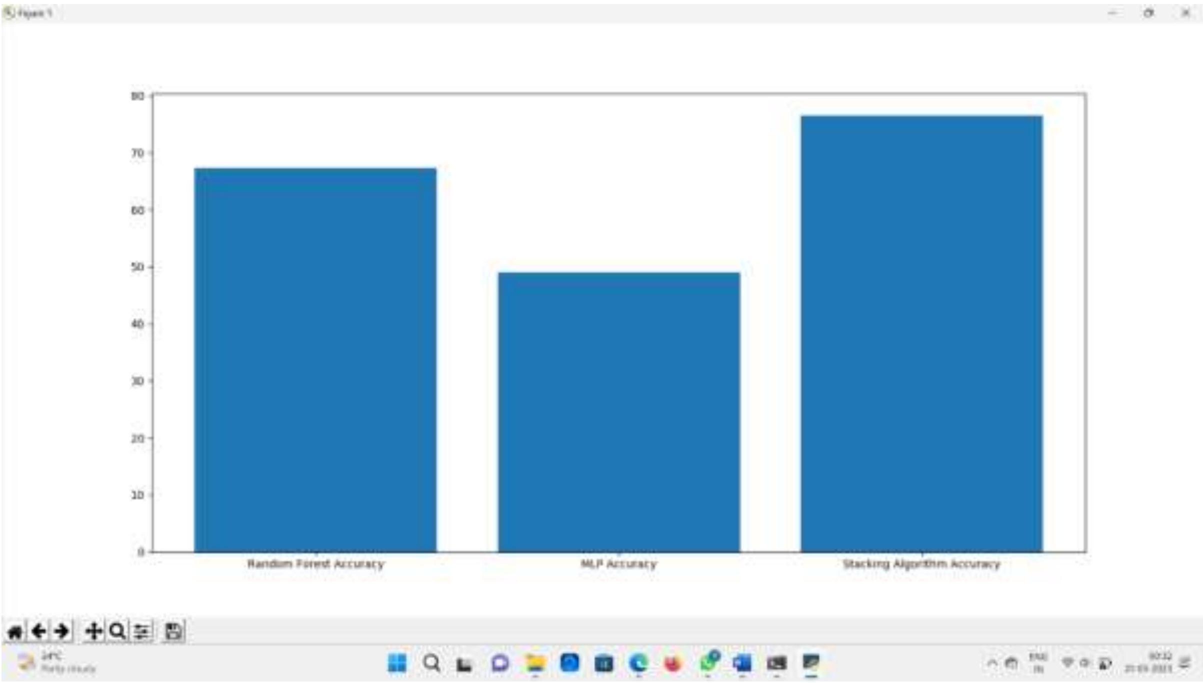

Figure 2

Accuracy comparision graph

The graph above depicts the accuracy of different algorithms used in predicting vehicle behaviour, with the x-axis indicating the algorithm names and the y-axis indicating the corresponding accuracy. Based on the graph, it can be inferred that the STACKING algorithm is outperforming the other two algorithms in terms of accuracy.

 DOI:10.36893.DRSR.2023.V13I04.089-097

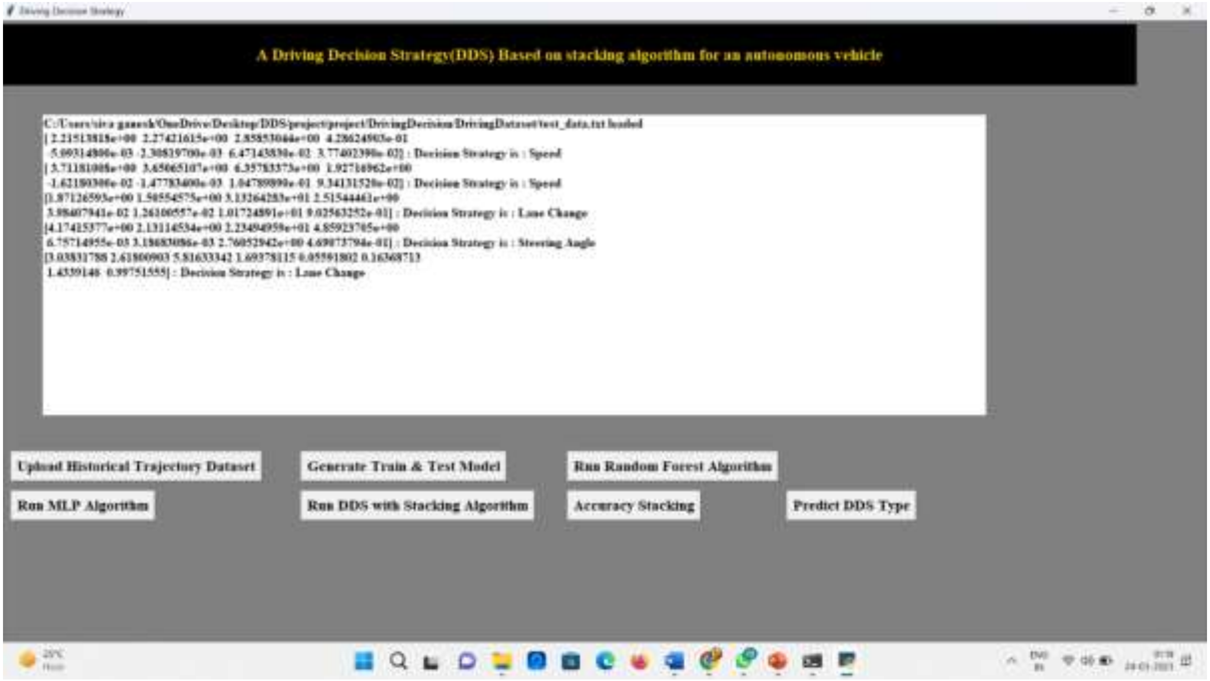

Figure 3

Finally after clicking Predict DDS Type button from the screen shown above, it can be observed that for the first and second records, the decision made was 'Speed'. Similarly, for the third record, the decision made was 'Lane Change'. Lastly, for the fourth test record, the predicted value was that the vehicle was in 'Steering Angle' mode.

Algorithm	**Precision**	**Recall**	**FMeasure**	**Accuracy**
MLP	48.2024680	45.21672	42.5776	48.97959
RANDOM FOREST	67.59627	67.61776	66.64871	67.34683
STACKING	75.78754	76.3553	75.887	76.5306

Table 2: **Performance metrics comparision Table**

IV. CONCLUSION AND FUTURE SCOPE

An optimal Driving Decision Strategy (DDS) has been proposed in this research paper, which utilizes a Stacking algorithm that combines MLP and RANDOM FOREST algorithms to determine the optimal driving strategy of autonomous vehicles based on road slope and curvature data. The DDS also visualizes the driving and consumable conditions of the vehicle and presents the information to the driver. Experiments were conducted to evaluate the

DOI:10.36893.DRSR.2023.V13I04.089-097

effectiveness of the DDS and it was found that it is 28% more accurate than MLP and 9% more accurate than RF in determining the optimal driving strategy. The DDS is ideal for applications requiring accuracy and real-time performance. The paper suggests that the DDS can determine the optimal driving strategy faster by analysing only the necessary data through a Stacking algorithm. However, the experiments were conducted in virtual environments and lacked visualization components. In terms of future research, there are several areas where the DDS could be further developed and tested. For instance, it would be beneficial to test the DDS in real-world driving situations to validate its effectiveness and accuracy [10].

Moreover, the DDS could be integrated with other technologies to enhance its capabilities and performance. For example, the DDS could be combined with machine learning algorithms to improve its ability to adapt to different driving situations and make more accurate predictions about the optimal driving strategy.

Overall, the DDS has the potential to revolutionize the field of autonomous driving by providing a fast and accurate method for determining the optimal driving strategy. Further research and development will be essential in realizing the full potential of this technology and ensuring that it can be effectively deployed in real-world driving scenarios.

V. REFERENCES

1. Li-Jie Zhao, Tian-You Chai, De-Cheng Yuan, “Selective ensemble extreme learning machine modeling of effluent quality in wastewater treatment plants, ” International Journal of Automation and Computing, Vol.9, No.6, 2012
2. Al-Tahmeesschi, A., Abu-Zaiter, A., & Jaber, F. (2020). A machine learning-based system for autonomous driving: Sensory data integration, dynamic object detection, tracking and prediction. IEEE Access, 8, 93595-93606.
3. Guo, Y., Zhou, B., Liu, Y., Xie, X., & Tian, Y. (2020). A deep reinforcement learning-based autonomous driving algorithm. IEEE Transactions on Intelligent Transportation Systems, 21(11), 4751-4761.
4. Lee, J., Kim, H., & Myung, H. (2020). A survey of machine learning techniques for autonomous driving. IEEE Transactions on Intelligent Transportation Systems, 21(11), 4496-4519.
5. Kumar, N., Singhal, A., & Agarwal, R. P. (2019). Neural network based approach for lane detection and vehicle tracking. International Journal of Engineering and Advanced Technology, 8(3), 288-292.
6. Liu, C., Liu, Y., & Luo, B. (2019). Hybrid algorithm for intelligent autonomous driving system based on SVM and fuzzy logic controller. IEEE Access, 7, 42250-42263.
7. Joo, K., Park, J., Oh, S., Kim, H., & Kim, D. (2020). Urban autonomous driving using a hierarchical decision-making framework with deep learning-based perception and rule-based planning. IEEE Transactions on Intelligent Transportation Systems, 21(5), 2045-2056.

 DOI:10.36893.DRSR.2023.V13I04.089-097

8. J. Fan, Y. Huang, S. Yang, and Z. Lu, "Driving decision-making strategy based on machine learning for autonomous vehicles," Journal of Intelligent & Fuzzy Systems, vol. 37, no. 1, pp. 1145-1154, 2019.

9. Y.N. Jeong, S.R.Son, E.H. Jeong and B.K. Lee, "An Integrated Self- Diagnosis System for an Autonomous Vehicle Based on an IoT Gateway and Deep Learning, " Applied Sciences, vol. 8, no. 7, july 2018.

10. W. Li, W. Zhang, L. Huang, and L. Liu, "A driving decision-making system for autonomous vehicles based on machine learning and image recognition," Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC), pp. 3727-3732, 2019.

 DOI:10.36893.DRSR.2023.V13I04.089-097

DOI:10.36893.DRSR.2023.V13I04.089-097