International Journal of Research in Engineering, Science and Management
Volume 4, Issue 10, October 2021
https://www.ijresm.com | ISSN (Online): 2581-5792

A Driving Decision Strategy (DDS) by Using Genetic Algorithm for an Autonomous Vehicle

Venkata Naga Rani Bandaru[1*], Venu Gopal Atchana[2], Suresh Kumar Garrugu[3], Sandeep Kumar Mude[4], V. R. J. Sastry Eemani[5]

[1,2,3,4,5]*Assistant Professor, Department of Information Technology, Vishnu Institute of Technology, Bhimavaram, India*

***Abstract*: A modern-day self-sustaining car determines its driving method by means of thinking about solely exterior factors (Pedestrians, street conditions, etc.) barring thinking about the interior circumstance of the vehicle. To overcome above problems, in this paper author proposed a new strategy i.e A Driving Decision Strategy(DDS) Based on Machine learning for an autonomous vehicle" Analysis of both external and internal factors determines the optimal strategy for an autonomous vehicle (consumable conditions, RPM levels etc.). To implement this, the project author has introduced and algorithm called DDS (Driving Decision Strategy) algorithm which is based on genetic algorithm to choose optimal gene values which helps in taking better decision or prediction. DDS algorithm obtained input from sensor and then passes to genetic algorithm to choose optimal value which helps in faster and efficient prediction. Propose DDS with genetic algorithm performance is comparing with existing machine learning algorithm such as Random Forest and MLP (multilayer perceptron algorithm.). Propose DDS shows better prediction accuracy compare to random forest and MLP.**

***Keywords*: Driving Decision Strategy (DDS), Multilayer Perceptron (MLP), Optimized Deep Learning Module (ODLM), Discrete aircraft, Segmentation.**

1. Introduction

Companies around the world are developing technologies for advanced autonomous vehicles, which are currently in the 4th stage of development. The principle of operation of self-driving cars can be classified into three levels: recognition, judgement and control. As part of the recognition process, vehicles are equipped with various sensors, including GPS, cameras, and radar. As a result of this information, the judgement step determines a driving strategy. When the driving environment is identified, it is analysed and appropriate driving plans are developed and the objectives. Vehicle starts driving on its own after the control step has been completed. In order to reach its destination, an autonomous vehicle performs a series of actions, repeating on its own the steps of recognition, judgement and control.

Autonomous vehicles are getting better at recognizing data as their performance improves. An increase in these sensors can lead to an overload of the vehicle's electrical system. In-vehicle computers compute data collected by sensors in self-driving vehicles. Due to overload, the speed of judgement and control decreases as the amount of computed data increases. These problems can jeopardise the vehicle's stability. As a means of preventing sensor overload, some studies have developed hardware that can perform deep-running operations inside a vehicle, while others use cloud computing to compute sensor data.

2. Literature Survey

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.

This paper proposes "An Integrated Self-diagnosis System (ISS) for an Autonomous Vehicle based totally on an Internet of Things (IoT) Gateway and Deep Learning," which collects records from an independent vehicle's sensors, diagnoses itself and the have an impact on between its components the usage of Deep Learning, and notifies the driver of the results. Three modules make up the ISS. The first In-Car Gateway Module (In-VGM) takes facts from in-vehicle sensors, such as media records from a black box, riding radar, and car manipulate messages, and sends every piece of statistics over every Controller Area Network (CAN) to the on-board diagnostics (OBD) or actuators with the aid of the, FlexRay, and Media Oriented Systems Transport (MOST) protocols. The statistics from in-vehicle sensors is dispatched to the CAN or FlexRay protocol, whilst media facts acquired whilst using is dispatched to the MOST protocol. A vacation spot protocol message kind is created from various kinds of messages that have been transferred. The 2nd Optimized Deep Learning Module (ODLM) generates the Training Dataset the usage of information obtained from in-car sensors and calculates the chance of car components and consumables, as nicely as the threat of different components influenced through a faulty part. to enhance the self-diagnosis velocity and decrease the device overhead, whilst a V2X primarily based Accident Notification Service (VANS) informs the adjoining motors and infrastructures of the self- diagnosis result analyzed with the aid of the OBD. This paper improves upon the simultaneous message transmission effectivity via the In-VGM by way of 15.25% and diminishes the mastering error price of a Neural Network algorithm via the ODLM through about 5.5%.

*Corresponding author: venkatanagarani.b@vishnu.edu.in

Yukiko Kenmochi, LilianBuzer, Akihiro Sugimoto, Ikuko Shimizu, “Discrete aircraft segmentation and estimation from a factor cloud the usage of neighborhood geometric patterns,” International Journal of Automation and Computing, Vol. 5, No. 3, pp. 246-256, 2008.

Using these days obtained discrete-geometry results, this work presents an approach for segmenting a 3D factor cloud into planar surfaces. A discrete airplane is described in discrete geometry as a set of grid factors that lie between two parallel planes separated by way of a tiny distance termed thickness. On discrete planes, not like in the non-stop case, there are a finite range of nearby geometric patterns (LGPs). Furthermore, such an LGP has a set of regular vectors as an alternative than a single regular vector. We reject non-linear factors from a factor cloud the usage of these LGP features, and then categorise non-rejected factors whose LGPs share frequent ordinary vectors into a planar-surface-point set.

Ning Ye, Yingya Zhang, Ruchuan Wang, Reza Malekian, “Vehicle trajectory prediction based totally on Hidden Markov Model,” The KSII Transactions on Internet and Information Systems, Vol. 10, No. 7, 2017.

Real-time, accurate, and dependable automobile trajectory prediction has excellent utility cost in Intelligent Transportation Systems (ITS), logistics distribution, and cell e-commerce. Vehicle trajectory prediction can now not solely furnish correct location-based services, however it can additionally screen and predict visitor’s stipulations in advance of time, and then suggest the exceptional route for customers. In this research, we first mine the double layers of hidden states of historic automobile trajectories, and then use historic records to calculate the parameters of the HMM (hidden Markov model). Second, we use the Viterbi approach to locate the hidden country sequences with double layers that correspond to the simply pushed trajectory. Finally, based totally on the hidden Markov mannequin with double layers hidden states, we provide a new algorithm (DHMTP) for automobile trajectory prediction, which predicts the nearest neighbour unit of function facts of the following ok stages. In phrases of predicting the following okay phases' trajectories, the experimental findings exhibit that the recommended algorithm's prediction accuracy is raised by using 18.3 percentage when in contrast to the TPMO approach and through 23.1 percentage when in contrast to the Naive algorithm, mainly when visitors glide is greater, such as this time from weekday morning to evening. Moreover, the time overall performance of DHMTP algorithm is additionally definitely elevated in contrast with TPMO algorithm.

3. Proposed System

In this paper author is describing concept for driving decision strategy by observing vehicle internal data such as steering and RPM level to predict various classes such as speed (steering), changing lane etc. All existing technique was concentrate on external data such as road condition and pedestrians etc. but not concentrate on internal values. So to take efficient determination of steering condition and changing lane author is analyzing internal data. All internal data will be collected from sensor and then store on cloud and then application will read data from cloud and then apply machine learning algorithms to determine or predict steering condition or changing lane.

To implement this project, we are using historical vehicle trajectory dataset as we don’t have sensors to collect data so we are using trajectory dataset. In dataset if user is slowing down vehicle then it has some sensor value with class label as ‘lane changing’. Similarly based on values we have different classes in dataset. Machine learning algorithm will be trained on such dataset and then when we apply test data on trained model then algorithm will predict class for that test data. Below are the dataset details and this dataset saved inside ‘DrivingDataset’ folder.

A. Data set

trajectory_id,
start_time,
end_time,
rpm_average,
rpm_medium,
rpm_max,
rpm_std,
speed_average,
speed_medium,
speed_max,
speed_std,
labels

20071010152332,2007-10-
10T15:23:32.000000000,2007-10-
10T15:32:59.000000000,2.21513
818073,2.27421615004,2.858530
43655,0.428624902772,-
0.005093147516729999,-
0.00230819670622,0.0647143832
211,0.0377402391782,speed
20071011011520,2007-10-
11T01:15:20.000000000,2007-10-
11T01:22:10.000000000,3.71181
007816,3.65065107266,6.357833
73513,1.9271696164900003,-
0.016218030061,-
0.00147783417456,0.1047898895
19,0.09341315155410003,speed
20080628053717,2008-06-
28T05:37:17.000000000,2008-06-
28T05:46:42.000000000,4.65889
245882,3.12829931751,13.02680
86376,4.09914234541,0.0040470
3387141,0.0124246102197,2.118
99984839,0.7521915347560001,s
teering_angle 20080628124807,2008-06-
28T12:48:07.000000000,2008-06-
28T12:57:16.000000000,1.71674
094314,1.31398945454,18.57768
36518,2.18497323244,-
0.0312684175217,0.03086335832

69,2.93888558793,0.7139256777
420001,steering_angle 20080825044741,2008-08-25T04:47:41.000000000,2008-08-25T05:05:12.000000000,2.38238360506,1.5371758264500002,20.919113327999998,2.865359735,-0.00720368601786,-0.000910857743471,2.01833073218,0.471527016571, lane_change

In above dataset all bold names are the dataset column names and below it are the dataset values. In dataset we can see sensor report each record with trajectory id, date, time and with speed and rpm details. In last column we can see labels as LANE_CHANGE, STEERING ANGLE and SPEED and with above dataset values and with label we will train machine learning algorithm and calculate accuracy.

Below are the test data which will not have any class label and it will have only sensor values and by applying sensor values on trained model we can predict or determine driving decision.

trajectory_id,
start_time,
end_time,
rpm_average,
rpm_medium, rpm_max,
rpm_std,
speed_average,
speed_medium,
speed_max,
speed_std

20080823105259,2008-08-23T10:52:59.000000000,2008-08-23T11:03:41.000000000,1.871265931,1.50554575041,31.326428333800006,2.51544461011,0.039840794139,0.0126100556557,10.1724891367,0.90256325184
20080821073812,2008-08-21T07:38:12.000000000,2008-08-21T08:30:53.000000000,4.17415377139,2.13114534045,22.3494958748,4.85923705089,0.00675714954958,0.0031868308583600001,2.76052942367,0.469073794101
20080913092418,2008-09-13T09:24:18.000000000,2008-09-13T09:24:36.000000000,3.03831788365,2.6180090273700003,5.81633341636,1.6937811468,0.0559180233599,0.163687128621,1.43391460095,0.997515549234

In above test data we can see only test values are there but not class label and after applying above test data on machine learning trained model we can predict/determine driving strategy such as going on speed, changing lane or steering angle.

4. Results and Discussion

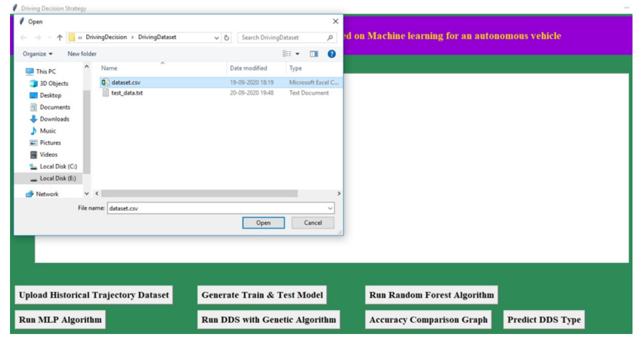

Fig. 1. Select 'dataset.csv' file and click on 'Open' button to load dataset and to get below screen

Fig. 2. Dataset contains 977 total trajectory records and application using 781 (80% of dataset) records for training and 196 (20% of dataset) for testing. Now both training and testing data is ready and now click on 'Run Random Forest Algorithm' button to train random forest classifier and to calculate its prediction accuracy on 20% test data

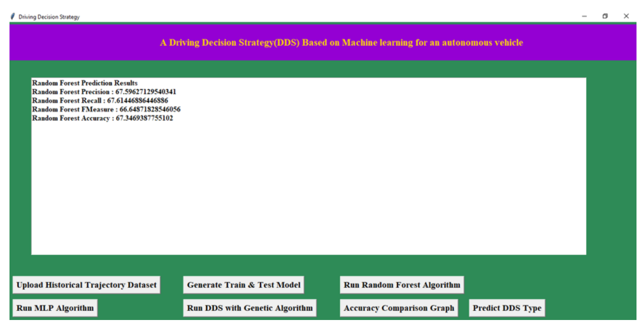

Fig. 3. Calculated random forest accuracy, precision, recall and f measure and random forest got 67% prediction accuracy. Now click on 'Run MLP Algorithm' button to train MLP model and to calculate its accuracy

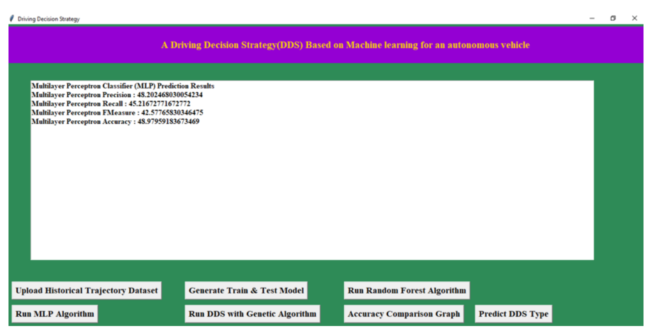

Fig. 4. MLP got 48% prediction accuracy and in below screen we can see genetic algorithm code used for building propose DDS algorithm

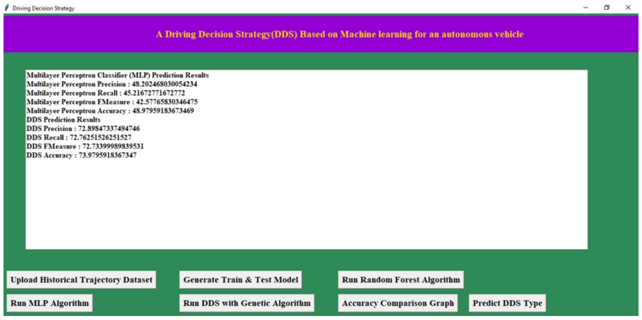

Fig. 5. Proposed DDS algorithm got 73% prediction accuracy and now click on ‘Accuracy Comparison Graph’ button to get below graph

Fig. 6. x-axis represents algorithm name and y-axis represents accuracy of those algorithms and from above graph we can conclude that DDS is performing well compare to other two algorithms. Now click on ‘Predict DDS Type’ button to predict test data

5. Conclusion

A Driving Decision Strategy was proposed in this paper. It uses a genetic algorithm based on gathered data to establish the vehicle's ideal driving strategy based on the slope and curve of the road it is travelling on, and it visualizes the autonomous vehicle's driving and consumables circumstances to provide drivers. To demonstrate the validity of the DDS, experiments were conducted to determine the optimal driving strategy by evaluating data from an autonomous vehicle. The DDS finds the best driving strategy 40 percent faster than the MLP, despite having similar accuracy. DDS also has a 22 percent higher accuracy than RF and calculates the best driving strategy 20 percent faster than the RF system. When accuracy and real-time are required, the DDS is the best choice.

The DDS sends only the data needed to identify the vehicle's optimal driving strategy to the cloud, and analyses it using a genetic algorithm, it is faster than other methods. These tests were carried out in a virtual environment using PCs, which had inadequate visualization capabilities. A real-world test of DDS should be conducted in the future. Expert designers should also improve the visualization components.

References

[1] 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.

[2] Yukiko Kenmochi, LilianBuzer, Akihiro Sugimoto, Ikuko Shimizu, “Discrete plane segmentation and estimation from a point cloud using local geometric patterns,” International Journal of Automation and Computing, vol. 5, no. 3, pp. 246-256, 2008.

[3] Ning Ye, Yingya Zhang, Ruchuan Wang, Reza Malekian, “Vehicle trajectory prediction based on Hidden Markov Model,” The KSII Transactions on Internet and Information Systems, vol. 10, no. 7, 2017.

[4] 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.