Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 6, June : 2023

# A Genetic Algorithm-based Driving Decision Strategy (DDS) for an autonomous vehicle

**Mr. G.Ramamohanarao [1], Mr. M.Naveen [2]**

#1 Assistant professor in the department of AI and IT at DVR & DR.HS MIC College of Technology (Autonomous), Kanchikacherla, NTR (DT)..

#2 MCA student in the department of IT at DVR&Dr.HS MIC College of Technology, Kanchikacherla, NTR(District).

**Abstract_** A modern self-sustaining vehicle determines its driving method solely by considering exterior factors (Pedestrians, street conditions, etc.) without considering the vehicle's interior situation. To address the aforementioned issues, the author proposed a new strategy in this paper titled "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 project, the author introduced the DDS (Driving Decision Strategy) algorithm, which is based on a genetic algorithm to select optimal gene values that aid in making better decisions or predictions. DDS algorithm obtains sensor input and then passes it to genetic algorithm to select optimal value, resulting in faster and more efficient prediction.The performance of the proposed DDS with genetic algorithm is compared to that of existing machine learning algorithms such as Random Forest and MLP. (multilayer perceptron algorithm.). Propose DDS outperforms random forest and MLP in prediction accuracy..

## 1.INTRODUCTION

Technologies for advanced autonomous vehicles are being developed by businesses all over the world, and they are currently in the fourth stage of development. There are three levels to the self-driving car's operating principle: control, judgment, and recognition Vehicles are outfitted with a variety of sensors, including GPS, cameras, and radar, to aid in the recognition process. The judgment step decides on a driving strategy based on this information. After the driving environment is found, it is looked at, and the right driving plans and goals are made. After the control step is finished, the vehicle can drive itself. An autonomous vehicle repeats on its own the actions of recognition, judgment, and control in order to reach its destination.

As their performance improves, autonomous vehicles are becoming better at recognizing data. An increase in the number of these sensors may cause the vehicle's electrical system to become overloaded. Self-driving vehicle sensors collect data that is processed by in-vehicle computers. As the amount of computed data increases, the speed of judgment and control decreases as a result of overload. The vehicle's stability may be in jeopardy as a result of these issues. For the purpose of forestalling sensor over-burden, a few investigations have created equipment that can perform profound running tasks inside a vehicle, while others use distributed computing to figure sensor information.

**2.LITERATURE SURVEY**

**2.1 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**

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 despatched to the CAN or FlexRay protocol, whilst media facts acquired whilst using is despatched 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%.

**2.2 Yukiko Kenmochi, Lilian Buzer, 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 a 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..

**3.PROPOSED SYSTEM**

In this essay, the author discusses the idea of using internal vehicle information, such as steering and RPM levels, to predict different classes of behavior, such as speed (steering), lane changing, etc. All of the techniques currently in use focused on external data, such as the state of the roads and the number of pedestrians, but not on internal values. Thus, the author is analysing internal data to make an effective determination of steering condition and lane change. All internal data will be gathered from sensors, stored on the cloud, read by the application, and then subjected to machine learning algorithms to ascertain or forecast steering condition or lane change..

As we don't have any sensors to collect data, we are using historical vehicle trajectory dataset to carry out this project. If a driver slows down, a sensor value with the class label "lane changing" appears in the dataset. Similar to that, the dataset has different classes based on values. Using this dataset, a machine learning algorithm will be trained.

The algorithm then predicts the class for the test data after we apply test data to the trained model. The dataset details are listed below, and it is stored in the "DrivingDataset" folder.

.

**3.1 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**

20071010152332,2007-10-10T15:23:32.000000000,2007-10-10T15:32:59.000000000,2.21513818073,2.27421615004,2.85853043655,0.428624902772,-0.005093147516729999,-0.00230819670622,0.0647143832211,0.0377402391782,speed

20071011011520,2007-10-11T01:15:20.000000000,2007-10-11T01:22:10.000000000,3.71181007816,3.65065107266,6.35783373513,1.9271696164900003,-0.016218030061,-0.00147783417456,0.104789889519,0.09341315155410003,speed

20080628053717,2008-06-28T05:37:17.000000000,2008-06-28T05:46:42.000000000,4.65889245882,3.12829931751,13.0268086376,4.09914234541,0.00404703387141,0.0124246102197,2.11899984839,0.7521915347560001,steering_angle

20080628124807,2008-06-28T12:48:07.000000000,2008-06-28T12:57:16.000000000,1.71674094314,1.31398945454,18.5776836518,2.18497323244,-0.0312684175217,0.0308633583269,2.93888558793,0.7139256777420001,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.

### 3.2 MODULES

Upload Historical Trajectory Dataset: We gather all the sensor data from the kaggale website and upload to the proposed model

Generate Train & Test Model: We have to preprocess the gathered data and then we have to split the data into two parts training data with 80% and test data with 20%

Run Random Forest Algorithm: we have to train the RF with train data and test the RF with test data to get best result from the algorithm

Run MLP Algorithm: we have to train the MLP with train data and test the MLP with test data to get best result from the algorithm

Run DDS with Genetic Algorithm : we have to train the DDS with Genetic Algorithm with train data and test the DDS with Genetic Algorithm with test data to get best result from the algorithm

Accuracy Comparison Graph: we will find the best algorithm with highest accuracy in the form of graph

Predict DDS Type: Enter the test data to predict the direction of a car using the DDS with Genetic Algorithm.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 6, June : 2023

## 4.RESULTS AND DISCUSSIONS

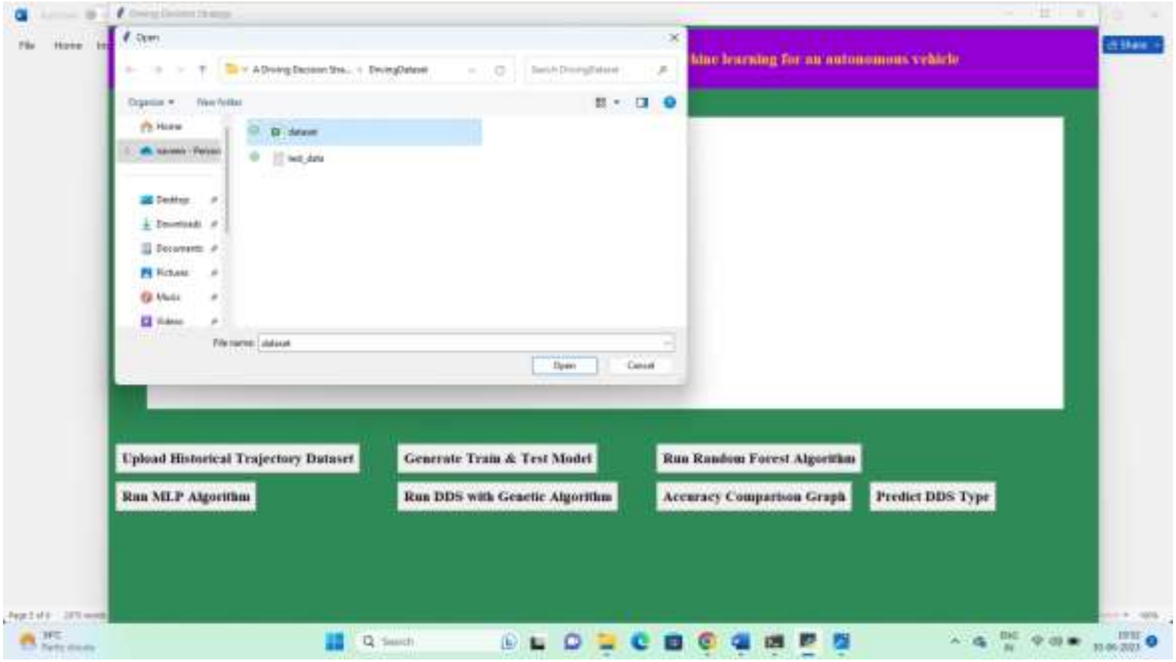


**Fig 4.1 Now select 'dataset.csv' file and click on 'Open' button to load dataset and to get below screen**


**Fig 4.2 In above screen 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 4.3 In above screen we calculated random forest accuracy, precision, recall and fmeasure and random forest got 67% prediction accuracy. Now click on 'Run MLP Algorithm' button to train MLP model and to calculate its accuracy**


**Fig 4.4 In above screen MLP got 48% prediction accuracy and in below screen we can see genetic algorithm code used for building propose DDS algorithm**

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 6, June : 2023


**Fig 4.5 In above screen propose DDS algorithm got 73% prediction accuracy and now click on 'Accuracy Comparison Graph' button to get below graph**

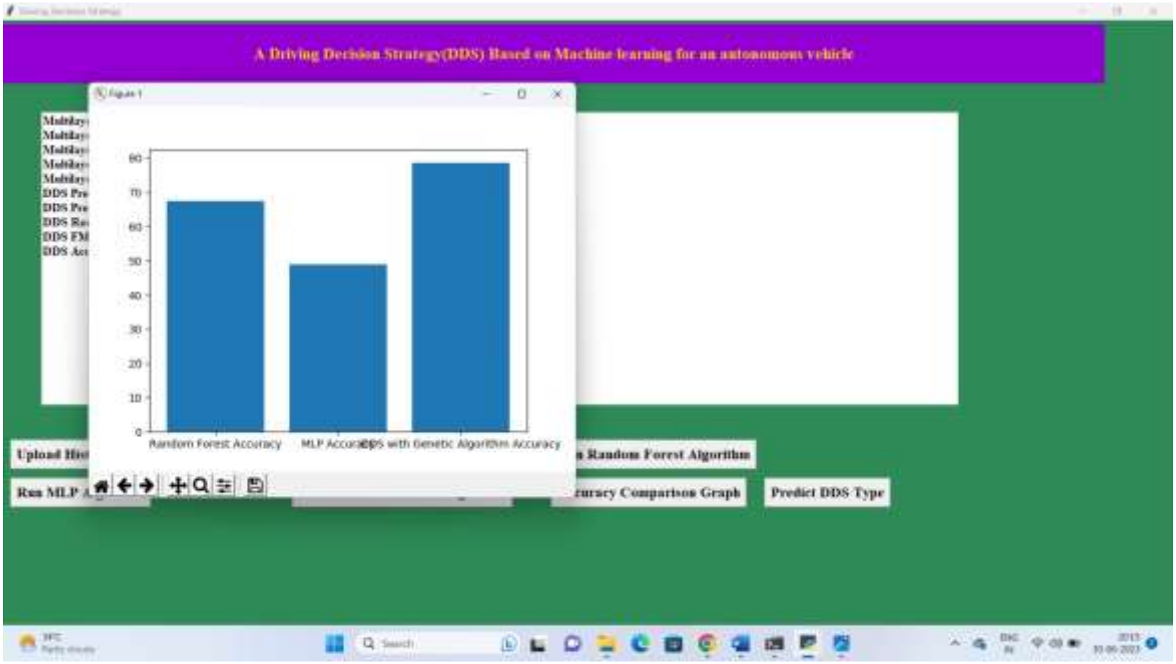


**Fig 4.6 In above graph x-axis represents algorithm name and y-axis represents accuracy of those algorithms and from above graph we can conclude that DDS is performing well compare to other two algorithms. Now click on 'Predict DDS Type' button to predict test data**

## 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 visualises 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.

Da 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 visualisation capabilities. A real-world test of DDS should be conducted in the future. Expert designers should also improve the visualisation 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, Lilian Buzer, 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