Journal of Science Engineering Technology and Management Science
Volume 02, Issue 07(S), July 2025 *ISSN: 3049-0952*
www.jsetms.com *DOI:10.63590/jsetms.2025.v02.i07(S).88-97*

AUTOPATHAI: AN INTELLIGENT NAVIGATION FRAMEWORK FOR AUTONOMOUS VEHICLES USING AI-DRIVEN PATH OPTIMIZATION

Dr. S. Sankar Ganesh, Shiva Kumar B, Shivanand G, Vamshi S
Department of Computer Science and Engineering (AIML), Kommuri Pratap Reddy Institute of Technology, Ghatkesar, Medchal, 500088

To Cite this Article
Dr. S. Sankar Ganesh, Shiva Kumar B, Shivanand G, Vamshi S, "Autopathai: An Intelligent Navigation Framework For Autonomous Vehicles Using Ai-Driven Path Optimization", Journal of Science Engineering Technology and Management Science, Vol. 02, Issue 07(S), July 2025,pp: 88-97, DOI: http://doi.org/10.63590/jsetms.2025.v02.i07(S).pp88-97
Submitted: 25-05-2025 *Accepted: 03-07-2025* *Published: 11-07-2025*

ABSTRACT
Autonomous vehicle navigation has emerged as a critical area of research, with global investments surpassing $80 billion in self-driving technologies. Studies indicate that over 90% of road accidents are caused by human error, emphasizing the need for automation. Additionally, navigation systems with advanced path-finding can reduce travel time by 15–25% in urban environments. However, existing manual analysis methods such as human visual monitoring, rule-based navigation, and static paper-based mapping suffer from subjectivity, lack of adaptability, and inefficiency in dynamic or unstructured environments. These techniques are time-consuming, prone to human error, and incapable of scaling for real-time autonomous decision-making. To address these limitations, the proposed system utilizes a structured preprocessing pipeline involving data cleaning, label encoding, and class distribution visualization. The input data, tailored to the application, undergoes null value imputation and label transformation using a label encoder to convert categorical direction labels into numerical form. A count plot is generated to assess class balance, providing insight into the need for SMOTE data augmentation. The final dataset is split into features and labels, ready for training with existing KNN and proposed MLP classifiers. This preprocessing stage ensures clean, balanced, and meaningful input that significantly enhances classification accuracy and real-time applicability in vehicle path-finding systems.

Key words: Autonomous Vehicle Navigation, Self-Driving Cars, Deep Reinforcement Learning, Edge AI for Vehicles, Trajectory Prediction

1. INTRODUCTION

The global autonomous vehicle market is rapidly expanding, with projections estimating it will surpass $2.3 trillion by 2030. According to the National Highway Traffic Safety Administration (NHTSA), 94% of serious crashes are due to human error, underscoring the urgent need for autonomous systems. Furthermore, with over 70% of urban traffic congestion linked to inefficient navigation and routing, path optimization remains a pivotal challenge in urban mobility management. Autonomous vehicles are designed to perceive their environment and make navigational decisions without human intervention. They rely heavily on sensors such as LiDAR, radar, ultrasonic detectors, GPS, and cameras to interpret their surroundings in real time. Advanced computational systems process this sensory data to identify road signs, lane markings, pedestrians, obstacles, and traffic signals. The complexity increases with dynamic environmental factors such as changing traffic conditions, weather variations, and the unpredictable behavior of nearby vehicles and pedestrians, making path-finding a crucial and ongoing challenge. Path planning and navigation in autonomous systems extend beyond basic point-to-point routing. It requires the integration of environmental perception, obstacle avoidance, motion prediction, and decision-making modules. Algorithms must ensure not just the shortest path but also safety, comfort, and legality of the route. These challenges are compounded in unstructured or semi-structured environments, such as rural areas, construction zones, or off-road terrains, where standard maps and lane-based navigation fail. Hence, real-time adaptive navigation remains one of the most researched and evolving aspects of autonomous systems. In industrial and commercial logistics, companies like Amazon and FedEx are leveraging autonomous navigation for warehouse robotics and last-mile delivery vehicles. These applications demand extremely accurate and adaptable navigation systems. In

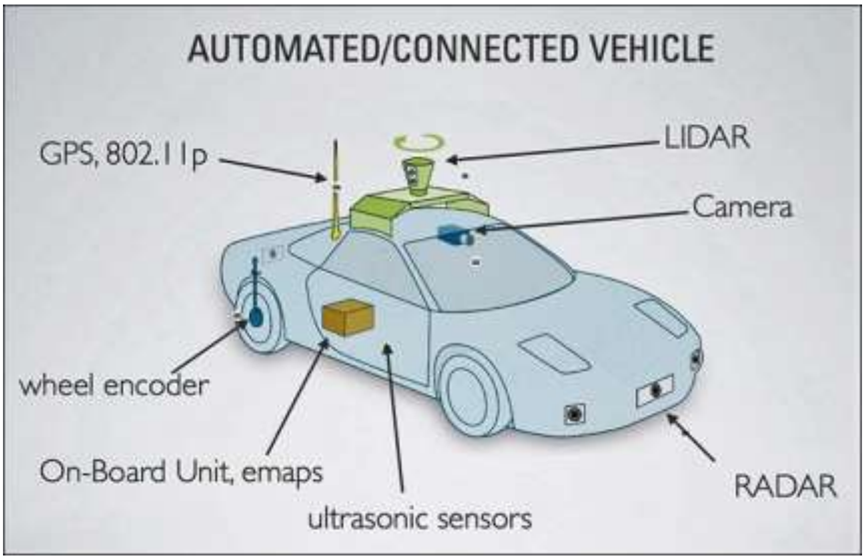

Fig 1. Autonomus navigation system

such high-throughput environments, even a minor error in navigation may lead to collisions or delivery delays. These enterprises are deeply dependent on data analysis to understand terrain characteristics, bottleneck zones, and optimal movement paths inside dynamic working environments. Urban municipalities and transportation networks are also integrating autonomous systems in public transit and surveillance. Cities like Singapore and Dubai are piloting autonomous taxis and shuttle buses to reduce traffic congestion and emissions. The core challenge for these vehicles lies in effective route planning in complex urban road networks. Data-driven navigation solutions, incorporating live traffic data, road conditions, pedestrian patterns, and environmental feedback, are essential for maximizing coverage and minimizing latency in vehicle routing.

2. LITERATURE SURVEY

Dang, Thai-Viet, and Ngoc-Tam Bui, Et. al[1] A real-time solution to the problem of obtaining hallway scenes from an exclusive image. The authors predicted a dense scene using a multi-scale fully convolutional network (FCN). The output was an image with pixel-by-pixel predictions that could be used for various navigation strategies. Additionally, a method for comparing the computational cost and precision of various FCN architectures using VGG-16 was introduced. The binary semantic segmentation and optimal obstacle avoidance navigation of autonomous mobile robots were two areas in which their method outperformed the methods of competing works. The authors successfully applied perspective correction to the segmented image in order to construct the frontal view of the general area, which identified the available moving area. The optimal obstacle avoidance strategy was comprised primarily of collision-free path planning, reasonable processing time, and smooth steering with low steering angle changes. Asano, Hikaru, Ryo Yonetani, et. al[2] A new problem was introduced: fair-delay multi-robot navigation, which aimed not only to enable such efficient, safe travels but also to equalize the travel delays among agents in terms of actual trajectories as compared to the best possible trajectories. The learning of a navigation policy to achieve this objective required resolving a nontrivial credit assignment problem with robotic agents having continuous action spaces. Hence, a new algorithm was developed called Navigation with Counterfactual Fairness Filter (NCF2). With NCF2, each agent performed counterfactual inference on whether it could advance toward its goal or should stay still to let other agents go. Doing so allowed the researchers to effectively address the aforementioned credit assignment problem and improve fairness regarding travel delays while maintaining high efficiency and safety. Monika, S., B. Annapurna, and T. Anuradha, et. al[3] “MEMS and voice Controlled Robotic Arm with Gesture”. The model of the system is segregated into robotic hand, a platform and accelerometer. With the help of ZigBee signals the robotic arm is controlled in a wireless way, which is of low cost and small in size. In the switch mode the accelerometer regulates the robotic arm which is affixed on an adjustable platform. Accelerometer is affixed on the hand, portraying its actions, then the arm of the robot travels appropriately. Around a casing, the platform and arm of the robot is co-ordinated through the signals & positions of the arm of the client. The unique actions accomplished by arm of the robot are: COLLECT & DEPOSIT / DUMP, LIFTING & DROPPING the things. The actions functioned by the model are: ONWARD, REARWARD, LEFT & RIGHT. The arrangement is done in such a way that the mounted camera can telecast the concurring video to any electronic appliance. Zhu, James, Anoushka Shrivastava, and Aaron M. Johnson et. Al[4] We synthesized works in law, engineering, and social science to present four actionable recommendations for how the robotics community could craft robots to mitigate the likelihood of self-defense situations arising. We established how current U.S. self-defense law could justify a human protecting themselves against a robot, discussed the then-current literature on human attitudes toward robots, and analyzed methods that had been produced to allow robots to operate close to humans. Finally, we presented hypothetical scenarios that underscored how current robot navigation methods could fail to sufficiently consider self-defense concerns and the need for the recommendations to guide improvements in the field.

Weerakoon, Kasun, Adarsh Jagan Sathyamoorthy, and Dinesh Manocha et. Al[5] We proposed a method to perceive a terrain’s geometric and other surface properties for efficient motion planning in outdoor environments. Our method incorporated two perceptron branches to identify the terrain’s elevation and roughness separately. The first branch used an elevation map created using LiDAR point clouds to compute a cost map that represented critical elevation changes. The second branch used a vision-based cost map trained using RGB images, IMU, and robot odometry. Then, least-cost waypoints were calculated on a combined cost map and were followed using the Dynamic Window Approach (DWA). Our planner navigated along the least-cost waypoints while adaptively varying the velocities to reduce vibration. We evaluated our method’s performance on a Husky robot in real-world environments. We observed that our method led to higher success rates and lower vibrations compared to state-of-the-art methods.

Jimenez, Mario F., Marcela Múnera, Carlos A. Cifuentes, and Anselmo Frizera et. Al[6] Advances in robotics and the constant growth of gait-related pathologies led to the development of different assistive devices. Smart walkers provide natural and intuitive strategies for gait assistance, such as path-following and guidance. Although these functionalities usually employ shared control approaches, the users' level of participation has yet to be assessed. This work presents the implementation of three modulation strategies for assisted navigation tasks. A path-following algorithm and a set of admittance-based controllers modulate the control authority between the user and the device. A group of 20 healthy subjects formed the validation group. Results showed a

kinematic estimation error of 0.13 m for the strategy that shared the control authority with the user. Statistical tests found significant differences regarding the naturalness of the proposed approach.
Zhao, Shengmin, and Seung-Hoon Hwang et. Al[7] An autonomous navigation robot platform named Owlbot was designed, which is equipped with a stepping motor as a mobile actuator. In addition, a stepping motor control algorithm was developed using polynomial equations, which can effectively convert speed instructions to generate control signals for accurately operating the motor. Using 2D LiDAR and an inertial measurement unit as the primary sensors, simultaneous localization, mapping, and autonomous navigation are realised based on the particle filtering mapping algorithm. The experimental results show that Owlbot can effectively map the unknown environment and realise autonomous navigation through the proposed control algorithm, with a maximum movement error being smaller than 0.015 m. Chen, Boyuan, Fei Xia, Brian Ichter, Kanishka Rao et.al[8] We develop NLMap, an open-vocabulary and queryable scene representation to address this problem. NLMap serves as a framework to gather and integrate contextual information into LLM planners, allowing them to see and query available objects in the scene before generating a context-conditioned plan. NLMap first establishes a natural language queryable scene representation with Visual Language models (VLMs). An LLM based object proposal module parses instructions and proposes involved objects to query the scene representation for object availability and location. An LLM planner then plans with such information about the scene. NLMap allows robots to operate without a fixed list of objects nor executable options, enabling real robot operation unachievable by previous methods. Cui, Can, Yunsheng Ma, Xu Cao, Wenqian Ye, Yang Zhou et. Al[9] LLMs have shown widespread attention in autonomous driving and map systems. Despite its immense potential, there is still a lack of a comprehensive understanding of key challenges, opportunities, and future endeavors to apply in LLM driving systems. In this paper, we present a systematic investigation in this field. We first introduce the background of Multimodal Large Language Models (MLLMs), the multimodal models development using LLMs, and the history of autonomous driving. Then, we overview existing MLLM tools for driving, transportation, and map systems together with existing datasets and benchmarks. Moreover, we summarized the works in The 1st WACV Workshop on Large Language and Vision Models for Autonomous Driving (LLVM-AD), which is the first workshop of its kind regarding LLMs in autonomous driving. To further promote the development of this field, we also discuss several important problems regarding using MLLMs in autonomous driving systems that need to be solved by both academia and industry. Tong, Pengfei, Xuerong Yang, Yajun Yang, Wei Liu, and Peiyi Wu et. Al[10] Collaborative visual positioning among multiple UAVs (UAV autonomous positioning and navigation, distributed collaborative measurement fusion under cluster dynamic topology, and group navigation based on active behavior control and distributed fusion of multi-source dynamic sensing information). Current research constraints are compared and appraised, and the most pressing issues to be addressed in the future are anticipated and researched. Through analysis and discussion, it has been concluded that the integrated employment of the aforementioned methodologies aids in enhancing the cooperative positioning and navigation capabilities of multiple UAVs during GNSS denial.

3. PROPOSED SYSTEM

The proposed approach offers several advantages, including improved directional accuracy through the use of the Multilayer Perceptron (MLP) classifier, which effectively captures nonlinear relationships in vehicle motion data. It overcomes the limitations of traditional methods like KNN by providing faster and more accurate predictions, especially in complex scenarios involving slight or ambiguous turns. The use of SMOTE ensures balanced learning across all direction categories, minimizing bias and enhancing model fairness. Preprocessing steps such as normalization and noise removal improve data quality and model stability. The system is scalable, adaptable to real-time applications, and suitable for deployment in various autonomous navigation systems ranging from urban vehicles to indoor robots, contributing to safer and more intelligent path-finding decisions.

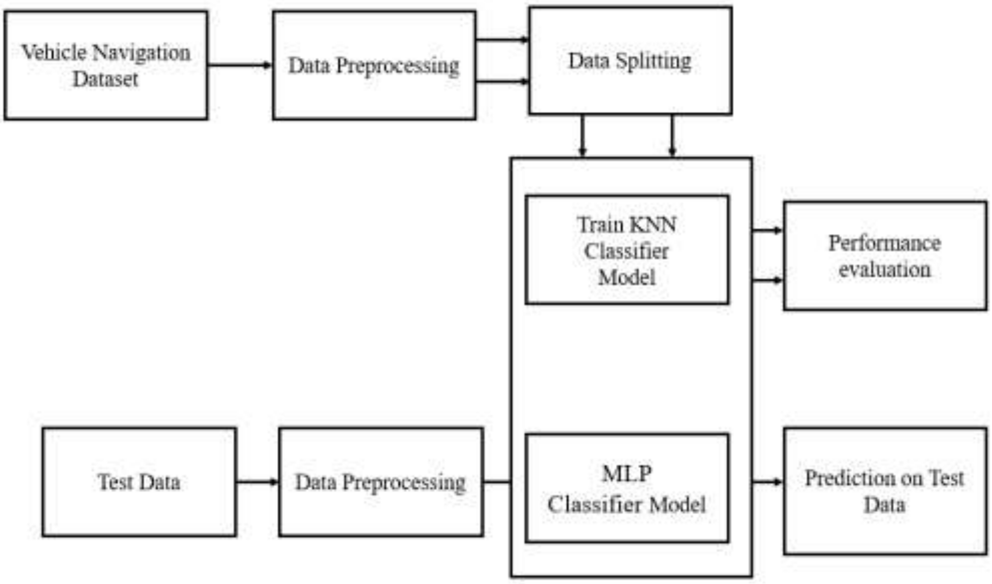

Fig 2. Architectural diagram of Proposed Methodology.

The study begins with the collection of a real-time vehicular motion dataset containing sensor readings such as gyroscope, accelerometer, GPS, and compass direction, with each record labeled for directional movements like left turn, right turn, or slight turn—captured manually or from driving simulations. Preprocessing involves cleaning noisy or incomplete data, applying smoothing techniques, standardizing features like angular velocity using z-score normalization, and encoding categorical direction labels into numerical form. To address class imbalance, particularly the underrepresentation of directional categories like slight turns, the SMOTE technique is employed to synthesize new minority class samples and achieve dataset balance. Following this, the dataset is split into training and testing subsets using a stratified approach, maintaining class distribution and enabling fair evaluation. As a baseline, the K-Nearest Neighbors (KNN) classifier is used to classify directions by majority voting among the nearest neighbors, though it struggles with scalability and sensitivity to irrelevant features. To overcome these limitations, a Multilayer Perceptron (MLP) classifier is proposed, which includes nonlinear hidden layers and a Softmax output layer for multi-class classification, trained using backpropagation to capture complex motion patterns. Model performance is evaluated through metrics such as accuracy, precision, recall, F1-score, and confusion matrix analysis, with MLP expected to outperform KNN, particularly in distinguishing subtle directional changes. Finally, the trained MLP model predicts vehicle movement directions on unseen test data, with results validated against actual labels, demonstrating the system's potential for real-time integration into autonomous vehicle navigation or driving simulations.

The MLP method is an advanced neural network architecture designed to detect complex, non-linear patterns within high-dimensional datasets, such as those used in internet bus Transport demand detection. In this specific application, bus Transport applications often contain subtle correlations between features that may not be captured by traditional models. The MLP architecture excels in learning such intricate relationships, making it suitable for distinguishing between genuine and demand applications. Its ability to automatically learn hierarchical feature representations through multiple layers gives it a significant advantage in demand detection scenarios where demand behaviors evolve and vary widely. Additionally, MLP models are highly scalable and adaptable, making them ideal for real-time, large-scale financial applications. Although today the Perceptron is widely recognized as an algorithm, it was initially intended as an image recognition machine. It gets its name from performing the human-like function of perception, seeing, and recognizing images. Interest has been centered on the idea of a machine which would be capable of conceptualizing inputs impinging directly from the physical environment of light, sound, temperature, etc. — the "phenomenal world" with which we are all familiar — rather than requiring the intervention of a human agent to digest and code the necessary information. Rosenblatt's perceptron machine relied on a basic unit of computation, the neuron.

Just like in previous models, each neuron has a cell that receives a series of pairs of inputs and weights as shown in Figure 4.6. The major difference in Rosenblatt's model is that inputs are combined in a weighted sum and, if the weighted sum exceeds a predefined threshold, the neuron fires and produces an output. The process begins with the **input layer**, where each node represents a feature from the preprocessed bus transport application dataset, including values like transaction amount, account type, and origin-destination identifiers. This data flows into multiple **Dense Layers**, where each neuron performs weighted transformations followed by non-linear **activation functions** such as ReLU, enabling the network to learn complex feature interactions and distinguish between legitimate and demand-based transactions. To prevent overfitting and enhance generalization, **dropout layers** are applied between dense layers by randomly disabling neurons during training, and **L2 regularization** may be used to penalize large weights. The **output layer**, typically using a sigmoid activation function for binary classification, produces a probability indicating the likelihood of a transaction being classified as demand. The model is trained using **backpropagation** and an optimizer like Adam, minimizing a loss function such as binary cross-entropy to iteratively adjust weights for better accuracy. Once training is complete, the MLP's performance is evaluated using metrics like **accuracy, precision, recall, and F1-score**, ensuring it can reliably detect demand in unseen bus transport application data, and the final model is deployed for real-time demand prediction.

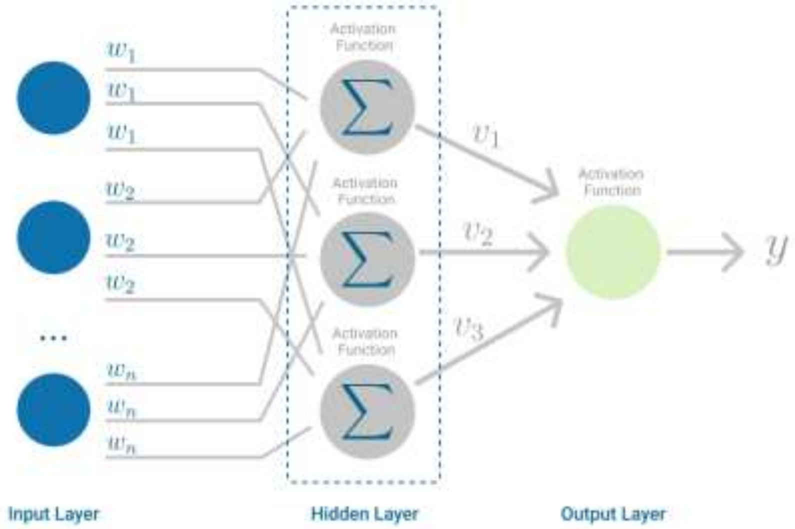

Fig 3. Architecture of MLP.

Perceptron uses SGD to find, the set of weight that minimizes the distance between the misclassified points and the decision boundary. Once SGD converges, the dataset is separated into two regions by a linear hyperplane. Although it was said the Perceptron could represent any circuit and logic, the biggest criticism was that it couldn't represent the XOR gate, exclusive OR, where the gate only returns 1 if the inputs are different. This was proved almost a decade later and highlights the fact that Perceptron, with only one neuron, can't be applied to non-linear data.

The MLP was developed to tackle this limitation. It is a neural network where the mapping between inputs and output is non-linear. A MLP has input and output layers, and one or more Dense

Layers with many neurons stacked together. And while in the Perceptron the neuron must have an activation function that imposes a threshold, like ReLU or sigmoid, neurons in a MLP can use any arbitrary activation function. MLP falls under the category of feedforward algorithms because inputs are combined with the initial weights in a weighted sum and subjected to the activation function, just like in the Perceptron. But the difference is that each linear combination is propagated to the next layer. Each layer is feeding the next one with the result of their computation, their internal representation of the data.

This goes all the way through the Dense Layers to the output layer. If the algorithm only computed the weighted sums in each neuron, propagated results to the output layer, and stopped there, it wouldn't be able to learn the weights that minimize the cost function. If the algorithm only computed one iteration, there would be no actual learning. This is where Backpropagation comes into play.

4. RESULTS

Figure 4 showcases the outcomes of the data preprocessing phase conducted as part of the research. Data preprocessing involves various tasks such as data cleaning, normalization, encoding categorical variables, handling missing values, and feature scaling. The figure was present visualizations or summary statistics illustrating the transformed dataset after preprocessing, demonstrating how the data has been prepared for subsequent analysis or modeling.

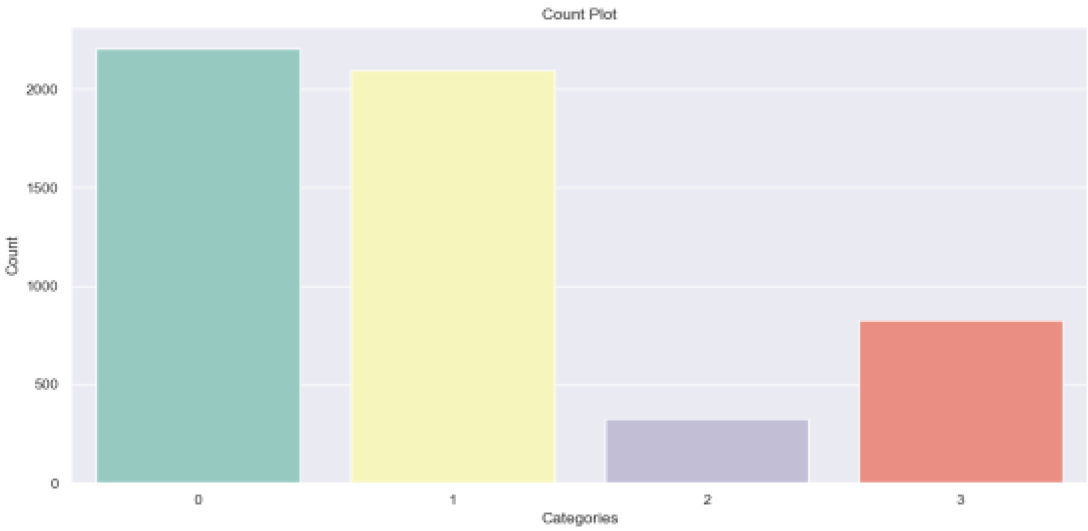

Fig 4. Result after data preprocessing

Figure 5 presents the results obtained after applying Synthetic Minority Over-sampling Technique (SMOTE) to address class imbalance issues in the dataset. SMOTE is a popular technique used in machine learning to artificially increase the number of instances in the minority class by generating synthetic samples. The figure was include visualizations or metrics indicating the impact of SMOTE on rebalancing the class distribution, potentially reducing bias in subsequent classification tasks.

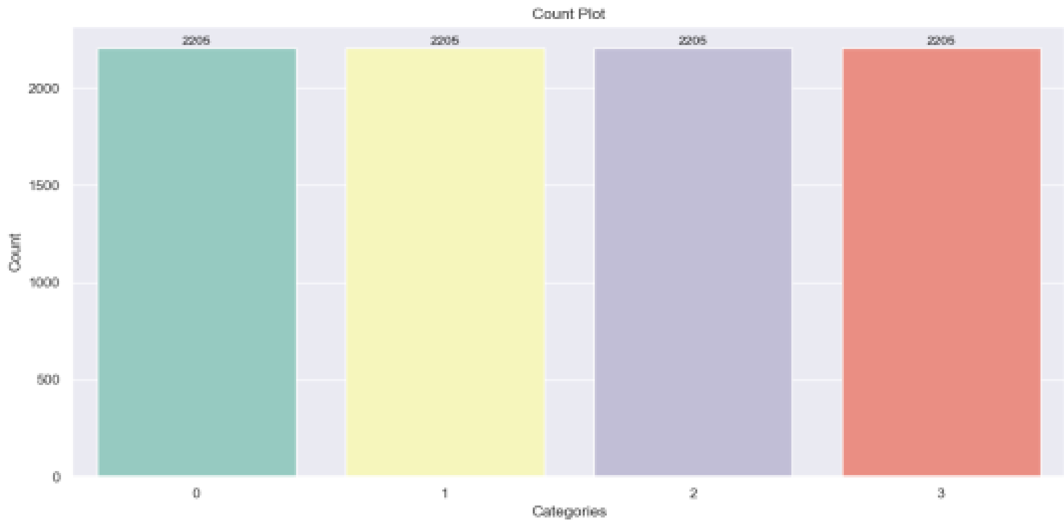

Fig. 5 Result after applying SMOTE

Figure 6 illustrates the confusion matrix resulting from the predictions of the proposed MLP classifier. Similar to Figure it offers a visual representation of the performance of the MLP model, allowing for the assessment of its classification accuracy and any misclassifications across different classes.

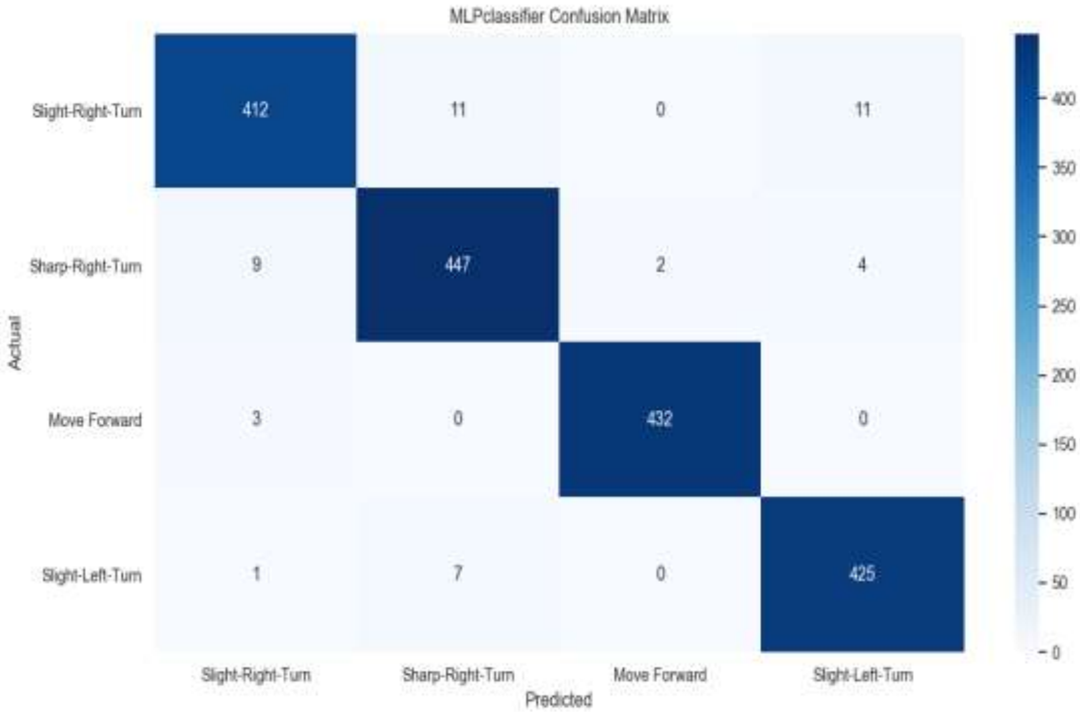

Fig 6. Proposed MLP Classifier confusion matrix

Table 1 details the performance of the Proposed MLP classifier on the same test set of 1764 samples, showing significant improvements over the KNN classifier. For the Slight-Right-Turn class (434 samples), the precision is 0.98, recall is 0.93, and F1-score is 0.95, indicating a substantial boost in correctly identifying this class compared to KNN. The Sharp-Right-Turn class (462 samples) achieves a precision of 0.95, recall of 0.97, and F1-score of 0.96, demonstrating strong performance. The Move Forward class (435 samples) excels with a precision of 1.00, recall of 0.99, and F1-score of 1.00, nearly perfect in both precision and recall. The Slight-Left-Turn class (433 samples) has a precision of 0.95, recall of 0.98, and F1-score of 0.97, showing high reliability. The overall accuracy is 0.97

(97%), with macro and weighted averages for precision, recall, and F1-score all at 0.97, indicating superior and consistent performance across all classes compared to KNN.

Table 1. Proposed MLP Classification Report

Class	Precision	Recall	F1-Score	Support
Slight-Right-Turn	0.98	0.93	0.95	434
Sharp-Right-Turn	0.95	0.97	0.96	462
Move Forward	1.00	0.99	1.00	435
Slight-Left-Turn	0.95	0.98	0.97	433
Accuracy			0.97	1764
Macro Avg	0.97	0.97	0.97	1764
Weighted Avg	0.97	0.97	0.97	1764

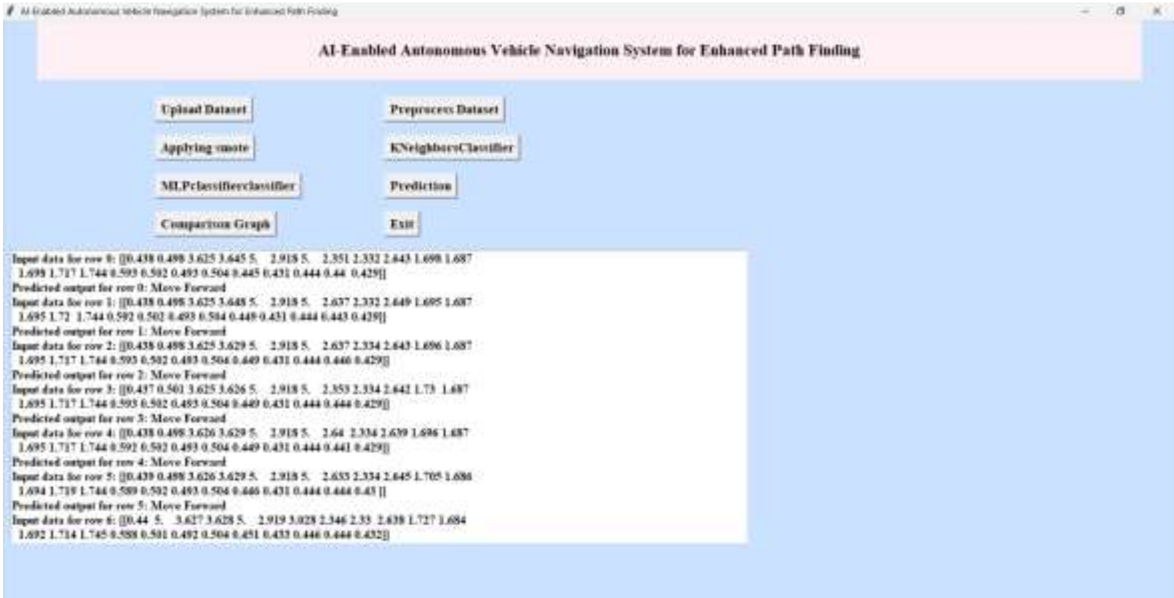

Fig 7. Prediction From Test Data.

5. CONCLUSION

The AI-Enabled Autonomous Vehicle Navigation System for Enhanced Path Finding, implemented using Tkinter and Python, effectively demonstrates the application of machine learning techniques to classify navigation commands for autonomous vehicles. The system integrates key functionalities, including dataset uploading, preprocessing, class balancing with SMOTE, and training of K-Nearest Neighbors (KNN) and Multi-Layer Perceptron (MLP) classifiers, achieving robust performance in predicting navigation actions such as Slight-Right-Turn, Sharp-Right-Turn, Move Forward, and Slight-Left-Turn. The MLP classifier outperforms the KNN model, delivering higher precision, recall, F1-score, and accuracy, which underscores its suitability for real-world autonomous navigation tasks requiring reliable decision-making. The user-friendly Tkinter GUI facilitates seamless interaction, allowing users to upload datasets, preprocess data, train models, make predictions, and compare classifier performance through an intuitive interface. The system's ability to handle test data predictions and display results in a clear, text-based format ensures accessibility for users with varying technical expertise. By leveraging libraries like pandas, scikit-learn, and seaborn, the application provides a comprehensive pipeline for data processing and model evaluation, making it a valuable tool for prototyping autonomous vehicle navigation systems. Overall, the application successfully achieves its objective of enhancing path-finding through AI-driven classification, offering a practical solution for autonomous vehicle control with high accuracy and usability.

REFERENCES

[1] Dang, Thai-Viet, and Ngoc-Tam Bui. "Obstacle Avoidance Strategy for Mobile Robot based on Monocular Camera." *Electronics* 12, no. 8 (2023): 1932.

[2] sano, H., Yonetani, R., Nishimura, M., & Kozuno, T. (2023). Counterfactual Fairness Filter for Fair-Delay Multi-Robot Navigation. *arXiv preprint arXiv:2305.11465*.

[3] Monika, S., B. Annapurna, and T. Anuradha. "Automatic robot navigation system for object picking." In *AIP Conference Proceedings*, vol. 2492, no. 1. AIP Publishing, 2023.

[4] Zhu, James, Anoushka Shrivastava, and Aaron M. Johnson. "Grounding Robot Navigation in Self-Defense Law." *arXiv preprint arXiv:2304.00347* (2023).

[5] Weerakoon, Kasun, Adarsh Jagan Sathyamoorthy, and Dinesh Manocha. "Terrain-aware autonomous robot navigation in outdoor environments." In *Open Architecture/Open Business Model Net-Centric Systems and Defense Transformation 2023*, vol. 12544, pp. 78-85. SPIE, 2023.

[6] Jimenez, Mario F., Marcela Múnera, Carlos A. Cifuentes, and Anselmo Frizera. "Haptic Human-Robot Collaboration for Walker-Assisted Navigation Based on Admittance Controllers." *IEEE Robotics and Automation Letters* 8, no. 5 (2023): 2622-2628.

[7] Zhao, Shengmin, and Seung-Hoon Hwang. "ROS-Based Autonomous Navigation Robot Platform with Stepping Motor." *Sensors* 23, no. 7 (2023): 3648.

[8] Chen, Boyuan, Fei Xia, Brian Ichter, Kanishka Rao, Keerthana Gopalakrishnan, Michael S. Ryoo, Austin Stone, and Daniel Kappler. "Open-vocabulary queryable scene representations for real world planning." In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11509-11522. IEEE, 2023.

[9] Cui, Can, Yunsheng Ma, Xu Cao, Wenqian Ye, Yang Zhou, Kaizhao Liang, Jintai Chen et al. "A survey on multimodal large language models for autonomous driving." In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 958-979. 2024.

[10] Tong, Pengfei, Xuerong Yang, Yajun Yang, Wei Liu, and Peiyi Wu. "Multi-UAV Collaborative Absolute Vision Positioning and Navigation: A Survey and Discussion." *Drones* 7, no. 4 (2023): 261.
.04851 (2023).