IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Optimizing Path Planning For Mobile Robots: An ANN-Based Approach For Static Obstacle Navigation

[1]Muhammad Essa M, [2]Pravin Balaji, [3]Anbarasi MP

[1-2]UG Student, [3]Assistant Professor (Sl. Gr)
[1]Department of Robotics and Automation Engineering,
[1]PSG College of Technology, Coimbatore, India

Abstract: This paper details the development of an Artificial Neural Network (ANN) model designed to optimize the path planning algorithms chosen for autonomous robots in environments populated with obstacles, where the obstacles are represented as circles with various radii as input features to the model. The goal of the ANN is to predict the best algorithm for path planning out of a set of candidates, namely APF, SAPF, DWA, and VAPF. The model has been trained using regression analysis to determine the performance of the algorithm with minimum traversal time by choosing the one that takes the minimum time efficiency of speed for each configuration. Therefore, this method is for enhancing autonomous navigation by dynamically selecting the fastest and safest path planning strategy in real time.

Index Terms **- Path Planning Algorithms, Classifier Neural Networks, Static Path Avoidance, Path Selection, Potential Field Algorithms.**

I. Introduction

Autonomous robotic systems are increasingly of high importance in a wide range of applications, from fully automated industries to space and exploration, for their ability to work independently and efficiently. However, one of the significant challenges these systems face relates to path planning in cluttered environments with safety and efficiency. In robotics, path planning is required by such machines to be able to prevent potential collisions, minimize travel time, and ensure that some tasks are properly completed. This challenge has been approached by many algorithms, each with advantages under different conditions. Among these is the Artificial Potential Field (APF) algorithm developed by Khatib. This algorithm is highlighted for its simplicity and effectiveness in local path planning. The APF algorithm generates attractive forces toward the goal and repulsive forces away from obstacles to guide the robot throughout the environment [1]. However, APF faces limitations, such as local minima, where the robot can become trapped without reaching its goal [2].

Over these disadvantages, many researchers proposed different versions of the APF. One of the most popular enhancements is the Safe Artificial Potential Field (SAPF) algorithm, where a safety factor has been added by preventing the robot from passing or passing very closely to the obstacles during navigation [3]. This variant overcomes the inherent limitations of the classical APF, including unsafe closeness to obstacles and collisions in cluttered environments. Additionally, DWA emphasizes dynamic settings where the robot as well as obstacles may be moving. DWA determines the trajectory to be taken by examining the possible velocities within a dynamic window to ensure that the chosen path is collision-free and keeps moving towards the goal [4,5]. This approach proves particularly useful in scenarios in which static algorithms such as APF struggle due to unpredicted movement of obstacles. Besides this, Vortex Artificial Potential Field (VAPF) extends on APF principles and introduces the concept of vortices aiding the robot to move through the narrow and complex spaces, and hence increases maneuverability in cluttered environments [2].

Despite these advancements, selecting the most suitable algorithm in real-time remains a significant challenge. Each algorithm performs optimally under specific conditions: APF excels in simple, static environments; SAPF is ideal for safety-critical tasks; DWA is effective in dynamic settings; and VAPF is advantageous in complex, narrow spaces. The variability in obstacle configurations and robot dynamics makes it difficult to predetermine the best path-planning algorithm for every situation [16]. This requires an intelligent system that can select the right algorithm in real time, according to the conditions of the environment.

In response to this need, by designing an ANN model that is capable of autonomously selecting the optimal path-planning algorithm for any scenario. ANNs have proven to be powerful tools for decision-making and pattern recognition in robotics [13, 14]. In this method, the ANN learned to predict the optimal algorithm given the static layout of the obstacles. This setting was a static start and end point and the static circles with a constant radius for obstacles. Using MATLAB to simulate the paths generated by different algorithms, and trained the ANN to learn patterns in the environment that align with the fastest, most efficient path [6]. The ANN uses the data from all of the path-planning algorithms, measuring how long it takes each one to accomplish the goal of driving through environments with obstacles. The model had been trained on a variety of simulated environments and had achieved up to an 85% accuracy in predicting the optimum algorithm for a given scenario [8, 9]. This suggests that the ANN can considerably boost the real-time capabilities of decision-making in autonomous robots. Indeed, in practical applications it allows for the dynamic switching between path-planning algorithms depending on the layout of the obstacles by ensuring the robot moves safely and efficiently across a wide variety of environments [7, 10].

The significance of this approach lies in its adaptability and ability to enhance both safety and efficiency in robotic navigation. The ANN-based system reduces the determination of the appropriate algorithm in such systems from a tedious and error-prone process of manual tuning or preselecting algorithms. In fact, adaptability in difficult, cluttered environments is particularly important, where the obvious optimal path may not be clear and conditions can change rapidly. The integration of neural networks with path-planning algorithms provides a promising approach to enhance the flexibility and decision-making abilities of autonomous systems [17, 18].

II. Path Planning Algorithms

The path planning problem can be grouped as follows: with respect to the definition of a mobile robot and its working environment, a mobile robot searches for an optimal or suboptimal path in association with a certain performance criterion between any initial state and its target state. Good path planning technology of mobile robot can not only reduce time consumption, but also reduce the wear and capital investment of mobile robot.

Local Path Planning, also referred as online path planning or dynamic path planning, refers to the situation in which a robot undertakes real-time motion planning while remaining partially or fully unknown to its surroundings. Local path planning has the advantage of being highly flexible; on the other hand, it may be locally optimal but not globally guaranteed, or the goal may not even be accessible.

A. Artificial Potential Field

The APF approach simulates the interaction of electrostatic forces within the environment in order to model path planning. In this approach, the robot's initial position is supposed to share the same charge as the obstacles, while the goal possesses an opposite charge. In general, the initial position is considered to have a higher potential, and the goal is assigned a lower potential. Therefore, the hindrances exert a repulsive force and the objective wields an attractive force, both of these forces are governed by the electrostatic force model.

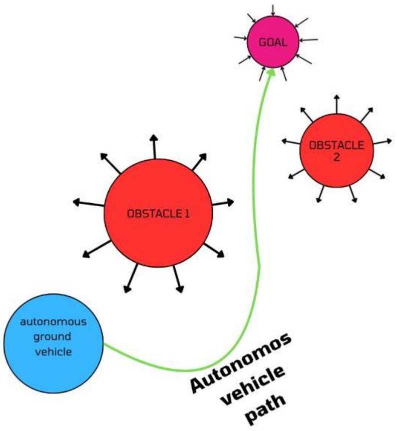

Fig 1: Potential Field Visualization

The potential field is mathematically modeled using either conical or quadratic potential functions in order to avoid potential singularities that could develop when the robot gets close to the goal. Let's designate d as the distance between two points and U as the potential field. Let the attractive potential field be given by,

$$\square_{\square\square\square}(\square) = \begin{cases} \frac{1}{2}\zeta d^2(q, q_{goal}), & d(q, q_{goal}) \le d^*_{goal} \\ d^*_{goal}\zeta d^2(q, q_{goal}) - \frac{1}{2}\zeta d^2(q_{goal}), & d(q, q_{goal}) > d^*_{goal} \end{cases} \quad (1)$$

In environments with multiple obstacles, the total repulsive force is computed as the sum of individual repulsive forces from each obstacle:

$$\square_{\square\square\square}(\square) = \begin{cases} \frac{1}{2}\eta\left(\frac{1}{d(q)} - \frac{1}{Q*}\right), & d(q) \le Q* \\ 0, & d(q) > Q* \end{cases} \quad (2)$$

The total potential field is the sum of the attractive and repulsive fields:

$$\square = \square_{\square\square\square} + \square\square_{\square\square\square} \quad (3)$$

The resultant force is derived from the gradient of the potential field:

$$\mathrm{F} = \nabla\mathrm{U} = \mathrm{grad(U)} \quad (4)$$

$$\mathrm{F} = \mathrm{F_{att}} + \Sigma\mathrm{F_{rep}} \quad (5)$$

This would ensure the obtaining of a collision free path in an unknown environment. One significant drawback of the Artificial Potential Field approach is the possibility of encountering local minima and saddle points. These occur when the robot becomes trapped in regions where the potential forces cancel each other out, preventing further movement. Additionally, in narrow or tightly packed environments, the method can result in oscillations or deviations near obstacles, which may lead to unstable or suboptimal paths [1].

B. Safe Artificial Potential Field

On top of the idea of traditional APF, SAPF introduces the safety constraints in order to enhance the obstruction handling and help the point avoid local minima or other path tracking problems such as oscillation in a constrained space that are caused by traditional APF method. The concept of SAPF enhances the standard APF method by taking into account safe distance margins in conjunction with dynamic repulsive forces related to proximity to obstacles in planning paths within cluttered spaces.

In SAPF, both the attractive as well as repulsive forces are changed as it keeps a safe distance from obstacles so that collision can be avoided and instead, its buffer zone is ensured. Although the position of the initial position still has the same charge as the obstacles, thus producing a repulsive force, the goal has an opposite charge, thus making an attractive force for the goal. However, SAPF changes these forces dynamically according to safety margins around obstacles. Model of the potential field in SAPF The distance between the robot and the goal is represented by d. Let us denote the potential field as U. The attractive potential field remains quite similar to APF but can be scaled for a smoother transition.

The repulsive potential field in SAPF is designed to keep the robot at a safe distance from obstacles, introducing a more gradual force as the robot approaches them. This dynamic adjustment is based on the distance d(q) to the nearest obstacle, and Q^*_{safe} is the safe distance threshold [2]. The repulsive potential is given by (2)

$$U_{rep}(q) = \begin{cases} \frac{1}{2}\eta\left(\frac{1}{d(q)} - \frac{1}{Q^*_{safe}}\right), & d(q) \le Q^*_{safe} \\ 0, & d(q) > Q^*_{safe} \end{cases}$$

(6)

In contrast to the APF, where the repulsive force is abrupt near obstacles, SAPF introduces a smoother transition that adjusts the repulsive force based on the robot's proximity to an obstacle, ensuring a gradual deceleration when approaching an obstacle. The total potential field U is the sum of the attractive and repulsive fields using (3)

The resultant force acting on the robot, which drives its movement, is derived from the gradient of the potential field using (4) and (5)

$$\square = \nabla\square = \square_{\square\square\square} + \square\square_{\square\square\square}$$

SAPF provides several advantages over standard APF. It introduces a safety margin around obstacles and smoother transitions in repulsive forces, which reduces the likelihood of local minima prevalent in the oscillations observed in very packed environments. The approach would be fruitful in navigating narrow passages or dynamic environments with multiple obstacles that a robot has to navigate through.

However, the SAPF is not without its weaknesses. The dynamic adjustment of the repulsive force can make the navigation speed slower in highly densely packed regions because it avoids rushing through dangerous environments by keeping as much distance as possible from obstacles. In addition, even though risk of being caught in a local minimum is reduced, it is still present to some extent, and SAPF will also fail in significantly complex obstacle configurations as well as extreme cases of obstacle density.

C. Vortex Artificial Potential Field

The Vortex Artificial Potential Field (VAPF) is the modified version of traditional APF. It has been designed to overcome some of the limitations of the original approach in the context of complicated configurations. The new method of VAPF introduces a rotation component in addition to regular potential fields around the potential field space. This is the standard approach utilized in APF so that the robot actually avoids obstacles by producing a vortex-like behaviour around them. This algorithm avoids a robot getting stuck in local minima or experiencing oscillations within tight spaces, which are common drawbacks in the standard APF.

In the VAPF method, the robot is still under an attractive force towards the goal and a repulsive force away from obstacles, while a vortex or swirl force near obstacles is introduced, causing the robot to "rotate" around obstacles while maintaining advance toward the goal. This helps guide the robot around complex-shaped obstacles without getting stuck in local minima.

Representation of VAPF in mathematical terms would include attractive and repulsive potential fields along with the extra vortex potential. Let $d(q,q_{goal})$ be the distance between the robot's position q and the goal q_{goal}, and d(q) be the distance between the robot and the nearest obstacle. The attractive and repulsive potentials are defined similarly to APF from equation (1) and (2)

The vortex potential is introduced near obstacles to create a rotational force that helps the robot navigate around them. This component is added to the repulsive potential field to create a tangential force that encourages circular motion near obstacles. The vortex potential can be modelled as:

$$U_{vortex}(q) = \frac{\lambda}{d(q)} \tag{7}$$

Where:

λ is a scaling factor controlling the strength of the vortex force. d(q) is the distance between the robot and the obstacle.

The resulting vortex force, F_{vortex}, is the negative gradient of the vortex potential:

$$\square_{\square\square\square\square\square\square} = -\nabla\square_{\square\square\square\square\square\square} \tag{8}$$

This vortex force adds a tangential component to the robot's movement, creating a swirling effect around obstacles. The total potential field is a combination of the attractive, repulsive, and vortex components:

$$U = U_{att} + \Sigma U_{rep} + U_{vortex} \tag{9}$$

The total force acting on the robot is the gradient of the combined potential field:

$$F = \nabla U = F_{att} + \Sigma F_{rep} + F_{vortex} \tag{10}$$

This will give Vortex APF the giant advantage in many path-planning tasks where traditional APF would fail. By incorporating the rotational force around the obstacle, VAPF avoids getting caught at local minima or oscillation in tight spaces. This vortex action will preserve the forward movement of the robot while moving around the obstacles to make the path-planning process efficient and safe [3].

However, the method has some disadvantages as well. The vortex component creates unwanted rotations when the obstacles are sparsely distributed or spread widely enough that it might increase the path length. In addition, much effort is needed for the fine-tuning of parameters like vortex force strength with the complexity of the environment. A relatively high vortex force strength causes too much spiraling of the robot around the obstacles, meaning a lot of delays in reaching the goal.

D. Dynamic Window Approach

The Dynamic Window Approach (DWA) is a widely-used path planning technique, often employed as the default algorithm in Robot Operating System (ROS). Unlike Artificial Potential Field (APF) algorithms, which typically generate a single trajectory, DWA continuously evaluates multiple candidate trajectories in real-time. These trajectories are generated based on the robot’s dynamic constraints, and the most optimal path is selected to guide the robot toward its goal while avoiding obstacles.

DWA is classified as a local path planning algorithm, which enables the robot to react to dynamic obstacles in real-time. The algorithm not only considers the environment but also integrates the robot's kinematics into the decision-making process, ensuring that the chosen trajectory is feasible given the robot's velocity and acceleration limits [4]. In 2D robot kinematics, the robot is typically simplified as a circular shape for computational ease. The position of the robot is represented by coordinates x and y, while θ denotes the robot's orientation with respect to the horizontal axis.

The robot's movement can be described by the following equations based on its velocities:

$$\square = \square + \square \cdot \square\square\square\square + \square \cdot \square\square\square\left(\frac{\square}{2} + \square\right) \tag{11}$$

$$\square = \square + \square \cdot \square\square\square\square + \square \cdot \square\square\square\left(\frac{\square}{2} + \square\right) \tag{12}$$

$$\theta = \omega dt \tag{13}$$

Where:

- u represents the velocity along the x-axis.
- v represents the velocity along the y-axis.
- ω is the rotational velocity around the z-axis.
- dt is the time increment.

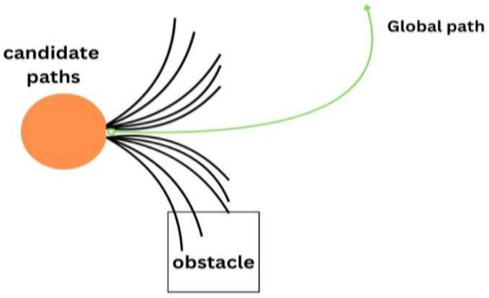

Fig 2: Dynamic Window Approach of Mobile robot

In the DWA method, candidate trajectories are generated based on various velocity pairs and are discretized into time instances through sampling. The algorithm then evaluates each trajectory to ensure that it is collision-free and within the robot's velocity limits. Trajectories that do not satisfy these constraints or intersect with obstacles are discarded.

The best trajectory is selected by optimizing a cost function as shown in Fig. 2. The cost function evaluates each candidate trajectory based on three primary criteria:

Heading: Measures how effectively the trajectory directs the robot toward the goal.

Clearance: it computes the clearance from the closest obstacle to the robot.

Velocity: Prioritizes trajectories with higher forward velocities to maximize motion efficiency.

An optimal trajectory is that one minimizes the cost function and which blends goal-directed movement with basic avoidance of obstacles and efficient velocity usage [4,5].

Once the best trajectory is chosen, the robot follows this trajectory for a brief time. After that time interval, the robot recalculates new candidate trajectories based on updated sensor information, and repeats the process until it eventually manages to achieve its goal. This iterative approach allows DWA to adapt to real-time dynamic changes in the environment. DWA is great for dynamic, high-speed environments or areas with a dense obstacle field. It considers the kinematic constraints of the robot to ensure that the chosen trajectories are kinematically feasible and collision free. Furthermore, it is really flexible when using real-time evaluation of several possible trajectories in cluttered environments.

However, the local trajectory optimization inherent in DWA can lead to suboptimal paths in certain situations, particularly when the robot encounters narrow passages or complex obstacle configurations. This limitation arises because DWA focuses only on short-term trajectory optimization, which may prevent the robot from identifying globally optimal paths.

III. Neural Network Implementation

Combining two or more algorithms will increase the solution's effectiveness and quality [16]. So, the mentioned local path planning algorithms are used under an artificial neural network to produce minimal time to reach goal. An artificial neural network (ANN) can represent the mapping relationship between the perceptual and behavioural domains, which is what path planning in this study.

Artificial neural networks, or ANNs, are artificial adaptive systems that draw inspiration from how the human brain functions. The neural network has neurons known as units which are ordered as layers to make a whole ANN. It has an input layer, output layer, and hidden layers. From the outside world, the input layer takes the data and it sends to the hidden layers for the transformation of valuable output data. This data is further given to the output layer from which the output is taken. The main advantage of the neural network is that the entire network won't stop working if any one node stops working. This can be very useful for a system with insufficient data. After creating the network, it has to be trained. There are two ways to train: one is supervised learning and the other is unsupervised learning. In supervised learning the network and the desired output with the inputs. Whereas in unsupervised learning the network has to learn on its own [7].

Neural networks take data and train themselves to predict the pattern. The basic structural element of neural networks is a neuron which computes the basic mathematical calculation. Each neuron is connected to the neuron in the next layer through channels called weights. The inputs are multiplied with the weights and passed as input to the next hidden layer which has neurons with mathematical values assigned to it as bias, which is added to the input. The input is then passed to a threshold function called activation function, this function determines whether the neuron will get activated or not. The activated neuron transmits data to the next layer, this is called forward propagation. In the output layer the neuron with the highest value determines the output. The predicted output is compared with the actual output to evaluate the result of the model. From the deviation in the prediction the values of the neurons are changed in magnitude throughout the structure, this is called backward propagation. This cycle of forward and backward propagation is continued till the weights of the neurons attain a value with which this model predicts the most accurately [8].

Feed-forward back-propagation network that is an ANN architecture consisting of an input layer, output layer, and the several hidden layers is adopted here. The path selection algorithm by ANN is shown in Fig. 3.

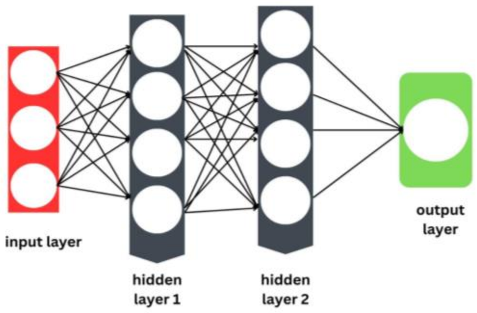

Fig 3: Neural Network Architecture

IV. Methodology

This work develops an ANN-based model for autonomously selecting the most appropriate path planning algorithm that suits an application with a static obstacle environment for a mobile robot. The method looks to optimize the navigation of a robot by dynamically choosing the best algorithm that minimizes traversal time and keeps the effectivity and safety of obstacle avoidance intact.

Test environment for evaluating the performance of different path-planning algorithms was developed. The objective here would be to move a robot from a predefined start point to a goal point using each of the developed path-planning algorithms: Artificial Potential Field (APF), Safe Artificial Potential Field (SAPF), Dynamic Window Approach (DWA), and Vortex Artificial Potential Field (VAPF). The Neural network is trained with number of random with the time data taken from the Local Path Planning algorithms under different arrangement of obstacles keeping the initial and goal points same in this case. Thus the neural network model formulates the relation between the obstacle arrangement and the time taken for the local path planning algorithms. The model is trained to identify the algorithm which takes the minimal time to reach goal. From this greater manipulation of the mobile robot is achieved.

The Program was developed for 10 static obstacles. The obstacles were presumed to be circles with a uniform radius of 0.25 units scattered randomly in a finite 2D workspace. Some random center coordinates of the obstacles were generated to build different geometries. In each experiment, the target was to reach the target within 60 seconds. Each algorithm was tested to determine the traversal time.

For each configuration, by logging the time taken to reach the goal by each algorithm. It is declared a run of an algorithm unsuccessful in this case if it cannot reach the goal within this window of 60 seconds. This paper adopted the time of traversal for a successful run as performance metric. It is the best algorithm configuration with the smallest number of traversal time. The process was repeated for different trials, but the arrangement of obstacles was randomly changed every time for a large and varied dataset. The dataset included the coordinates of obstacles as well as the time taken by each algorithm to traverse through them.

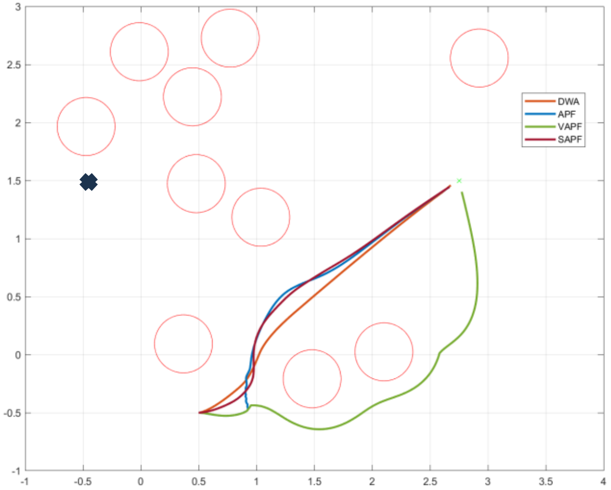

Fig 4: MATLAB Output with different Paths

Multiple trials are conducted with different randomly generated obstacle setups to gather sufficient data. This data is then fed into an Artificial Neural Network (ANN) developed using TensorFlow. The ANN is trained to predict the best-performing path planning algorithm based on the obstacle centre coordinates, aiming to select the most time-efficient algorithm for future navigation scenarios. This model serves to automate the process of selecting the optimal path planning algorithm in real-time, enhancing efficiency and decision-making in robotic navigation.

V. Result And Discussion

The result of the model is detailed below using classical performance metrics like F1 score and overall efficiency of the ANN prediction model is approximately 85% which can be further tuned to increase the accuracy. This opens a whole new approach to path planning with potential in higher accuracy and increased efficiency in navigation of the mobile robots.

An F1 score of 0.80 signifies the balance of Precision and Recall for a classifier machine learning model. Indicating the quality of this model's performance and reflecting on its precision and recall accuracy [9].

Table 1: Performance of the Classifier Model

	Precision	Recall	F1 Score
APF	0.81	0.96	0.88
SAPF	0.92	0.41	0.56
VAPF	1	0.35	0.51
DWA	0.9	0.81	0.85
Macro avg	0.91	0.63	0.7
Weighted avg	0.86	0.85	0.84

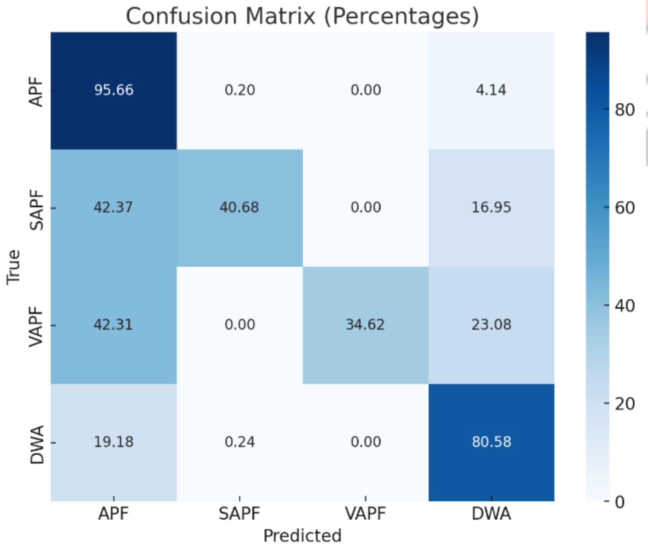

Fig 5: Confusion Matrix for the Classifier Model

The confusion matrix for a summary table is a representation classification model's performance. It supports both binary and multi-class classification.

The confusion matrix represents the correct number of true positives and the correct number of true negatives [9]. Additionally, it depicts the weaknesses of the model: false negatives (FN) comprise the cases that are missed, whereas the false positives (FP) are termed "false alarms".

A confusion matrix enables you to see how often each of them happens together in one place. For this reason, you can count the number of correct predictions and errors of each type at the same time. The confusion matrix proves to have important insights into model performance regarding the prediction of the correct path planning algorithm. While it indeed shows that the model correctly predicts 95.66% of true APF cases while, at times confusing it with DWA (4.14%). The SAPF algorithm failed as a model as it correctly classified only

40.68% of SAPF cases, with misclassifying 42.37% cases as APF, and 16.95% cases as DWA. The VAPF algorithm has a weak classification rule with only 34.62% of true VAPF cases classified. It usually classified VAPF as APF (42.31%), DWA (23.08%), thus indicating tremendous difficulty in the differentiation of the VAPF algorithm from other algorithms. On the other hand, for the DWA model, it has done a pretty good job in terms of accuracy, reporting an accuracy of 80.58% although it misclassifies 19.18% of DWA cases as APF. It is doing well for APF and DWA but gets confused between SAPF and APF, and it really struggles heavily to distinguish VAPF from other algorithms. These results suggest that the model could be hugely improved in terms of discriminating these very similar algorithms, especially SAPF and VAPF.

VI. Conclusion

This research demonstrated how an ANN model can be designed and employed so as to be able to exactly predict the best path planning algorithm for mobile robots working within static environments. Using obstacle centre coordinates as input and calculating time taken by different algorithms, the model thus achieved 85% accuracy although this can be improved upon if tuned further. This method raises the potential of using dynamic path planning using machine learning models to yield maximum efficiency. The use of ANN streamlines decision-making in real time, allowing for automatic selection of the optimal algorithm determined by the terrain, and enables enhancing the efficiency of systems for self-navigation. Such approach opens new avenues toward further increasing the adaptability and performance of mobile robots in complex environments, and it presents a highly promising tool for real-world robotic applications.

References

[1] Zhengtian Wu, Jinyu Dai, Baoping Jiang, Hamid Reza Karimi, “Robot path planning based on artificial potential field with deterministic annealing”, ISA Transactions, vol. 138, 2023.

[2] Souza, Rafael Monteiro Jorge Alves, Gabriela Vieira Lima, Aniel Silva Morais, Luís Cláudio Oliveira-Lopes, Daniel Costa Ramos, and Fernando Lessa Tofoli, “Modified artificial potential field for the path planning of aircraft swarms in three-dimensional environments” Sensors 22, no. 4 , p.1558, 2022.

[3] R. Szczepanski, “Safe Artificial Potential Field - Novel Local Path Planning Algorithm Maintaining Safe Distance From Obstacles”, in IEEE Robotics and Automation Letters, vol. 8, no. 8, pp. 4823-4830, Aug. 2023.

[4] Yanjie Cao, Norzalilah Mohamad Nor, “An improved dynamic window approach algorithm for dynamic obstacle avoidance in mobile robot formation”, Decision Analytics Journal, vol. 11, 2024.

[5] D. Fox, W. Burgard and S. Thrun, “The dynamic window approach to collision avoidance”, in IEEE Robotics & Automation Magazine, vol. 4, no. 1, pp. 23-33, March 1997.

[6] Rafal Szczepanski, “Comparison of Local Path Planning algorithms”, MATLAB Central File Exchange. Retrieved September 21, 2024.

[7] McCulloch WS, Pitts WH, “A logical calculus of the ideas immanent in nervous activity”, Bull Math Biophys,1943.

[8] Maind, Sonali B., and Priyanka Wankar, “Research paper on basic of artificial neural network”, International Journal on Recent and Innovation Trends in Computing and Communication 2, no. 1, pp.96-100, 2014.

[9] Krstinić, Damir, Maja Braović, Ljiljana Šerić, and Dunja Božić-Štulić. “Multi-label classifier performance evaluation with confusion matrix”, Computer Science & Information Technology 1, pp.1-14, 2014.

[10] Zhou, C., Huang, B., Fränti, P, “A review of motion planning algorithms for intelligent robots”, J Intell Manuf, vol.33, pp.387–424, 2022.

[11] Karimi, Zohreh. "Confusion Matrix." Encycl. Mach. Learn. Data Min., no. October 260, 2021.

[12] Hicks SA, Strümke I, Thambawita V, Hammou M, Riegler MA, Halvorsen P, Parasa S, “On evaluation metrics for medical applications of artificial intelligence”, Sci Rep, 2022 Apr 8.

[13] Grossi, Enzo, and Massimo Buscema, “Introduction to artificial neural networks”, European journal of gastroenterology & hepatology 19, no. 12, pp. 1046-1054, 2007.

[14] Bekey, George A., and Kenneth Y. Goldberg, eds, “Neural networks in robotics”, Springer Science & Business Media, vol. 202, 2012.

[15] Rao, D. H., “Neural networks in robotics and control: some perspectives”, Proceedings of IEEE/IAS International Conference on Industrial Automation and Control. IEEE, 1995.

[16] Zhang, Han-ye, Wei-ming Lin, and Ai-xia Chen, “Path planning for the mobile robot: A review”, Symmetry 10, vol. 10, pp.450, 2018.

[17] Sabry, Ahmad & Nordin, Farah & Ameer, Hadeer & Kadir, Zainal, “Fault Detection and Diagnosis of Industrial Robot Based on Power Consumption Modeling”, IEEE Transactions on Industrial Electronics, 2019.

[18] Prabhu, S.M., Garg, D.P, “Artificial neural network based robot control: An overview”, J Intell Robot Syst 15, pp.333–365, 1996.