International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Machine Learning-Based Movement Scheduling and Management for Autonomous Mobile Robot Navigation

[1]Dasarathy A. K., [2]Shakuli Saxena, [3]Dr. Manish Soni, [4] Minnu Sasi

Submitted: 20/08/2023 **Revised**: 11/10/2023 **Accepted**: 22/10/2023

Abstract: Introduce autonomous mobile robot navigation in a few sentences, along with its significance in numerous industries. A discussion of the difficulties in attaining effective and adaptable movement scheduling and management for autonomous robots is required. Emphasize the advantages of using machine learning approaches to solve these problems. In this study, we recommend movement scheduling and management based on an Augmented Gradient Support Vector Machine (AG-SVM) for autonomous mobile robot navigation. Assemble a comprehensive dataset with historical information on the movements of mobile autonomous robots in different contexts. Gather data on the positions and speeds of the robots, the environment, the order of the tasks, and any pertinent sensor data. By removing outliers, dealing with missing values, and normalizing the data, we can clean and preprocess the acquired dataset. To extract pertinent features for the movement scheduling and management activity, if necessary, perform feature engineering. The dataset's most beneficial components that help with movement planning and management are taken out using the Histogram of Oriented Gradients (HOG). This technique helps to reduce dimensionality and improve the efficiency of learning algorithms. AG-SVM is used to manage and schedule movements. To improve the deployment of autonomous robots in various industries, it is important to emphasize the importance of adaptive and effective movement scheduling and management.

Keywords: *autonomous mobile robot navigation, Augmented Gradient Support Vector Machine (AG-SVM), movement scheduling, management, Histogram of Oriented Gradients (HOG)*

1. Introduction

Machine learning-based movement is the application of machine learning algorithms and methods to allow autonomous or semi-autonomous movement in a variety of systems, including robots, autonomous vehicles, drones, and virtual characters. It entails teaching these systems to continuously increase their mobility capabilities over time without explicit programming. In machine learning-based mobility, the system often makes use of sensor inputs, such as camera pictures, lidar data, or other environmental information, to detect and comprehend its surroundings [1]. Once this data has been processed and analyzed, it employs machine learning algorithms to identify and extract pertinent patterns and characteristics. The system can create suitable movement instructions based on the learned patterns and make intelligent judgments. A large collection of samples, either labeled or unlabeled, representing various movement circumstances is often gathered throughout the training phase. The objective is to allow the system to generalize from training data and modify its movement tactics for novel, unforeseen scenarios. The term "machine learning-based movement" refers to a variety of activities, including navigation, route planning, obstacle avoidance, object tracking, gesture recognition, and locomotion control [2]. These systems may be made more effective, adaptable, and responsive by using machine learning, which will also increase their reactivity to changing settings. The capacity of a robot to travel through its surroundings autonomously and without human assistance is referred to as autonomous mobile robot navigation. To help the robot detect its surroundings, plot a course, and carry out the essential operations to get where it's going, it makes use of a variety of algorithms, sensors, and control systems. The robot senses its surroundings using sensors like cameras, lidar, sonar, or infrared sensors. Obstacles, landmarks, and other important elements are all provided by these sensors [3]. Based on the sensor data, the robot creates a map or model of its surroundings. To comprehend the layout, the placements of obstacles, and other navigational signals, utilize this map. The robot decides where it is in the mapped surroundings on its own. Simultaneous localization and mapping (SLAM) methods, which integrate sensor data and movement information to properly predict the robot's location, may help with this. The robot plots a collision-free route from its present location to the intended site after

[1]*Professor, Department of Civil Engineering, Jain (Deemed-to-be University), Bangalore-27, India, Email Id: ak.dasarathy@jainuniversity.ac.in*
[2]*Assistant Professor, College of Agriculture Sciences., Teerthanker Mahaveer University, Moradabad, Uttar Pradesh, India, Email id: shakuli2803@gmail.com*
[3]*Assistant Professor, School of Engineering & Technology, Jaipur National University, Jaipur, india, Email Id: manishsoni@jnujaipur.ac.in*
[4]*Assistant Professor, School of Agriculture, Dev Bhoomi Uttarakhand University, Uttarakhand, India, Email Id: agri.minnu@dbuu.ac.in*

it has a map of the surrounding area and is aware of its location. To create a viable or ideal route, robot dynamics, & other restrictions [4]. A robot continually recognizes and avoids obstacles in its path while it is moving. To safely navigate around obstacles, it makes adjustments to its trajectory, speed, or direction based on real-time sensor data. The intended course is converted into real robot movements by the control system. To guarantee the correct execution of the planned trajectory, this comprises motor control, motion planning, and feedback systems. Machine learning methods may be used by autonomous mobile robots to enhance their navigational skills [5]. They may adjust their behavior in response to changing environmental circumstances, improve their route planning algorithms, or learn from prior experiences. An autonomous mobile robot can move on its own in a range of contexts, such as confined areas, rough terrain, or intricate industrial settings, by combining these components. The objective is to make it possible for the robot to go quickly and securely to its destination while avoiding hazards and making deft judgments based on its observation of the surroundings [6]. Utilizing machine learning methods to optimize the scheduling and administration of robot movements in a dynamic environment is known as machine learning-based movement scheduling and management for autonomous mobile robot navigation. The motions of many robots working in the same space may be planned using machine learning. Machine learning models can discover the most effective way to distribute movement duties while taking into account elements like robot capabilities, energy consumption, workload balance, and job prioritization [7]. The trajectory planning process for autonomous mobile robots may be improved with the use of machine learning techniques. These algorithms gain knowledge from past or simulated data to forecast ideal courses and provide smooth trajectories that save energy, lessen the danger of collisions, and take into consideration robot dynamics and environmental limits. Management of traffic Machine learning approaches may assist in traffic management and collision avoidance in areas where several autonomous mobility robots coexist. Robots can make smart choices to avoid traffic, negotiate lanes, and coordinate their movements to maximize overall efficiency by assessing sensor data and learning from prior encounters [8]. Autonomous mobile robots can make adaptive judgments while navigating thanks to machine learning methods. Robots may dynamically change their movement methods, choose other routes, or vary their behaviors to improve performance and adapt to changing situations by continually learning from sensor inputs, environmental changes, and human preferences. To find irregularities in robot motions or system problems, machine learning methods may be applied. Machine learning models can detect anomalous behavior, raise alarms, and start remedial steps to guarantee safe and dependable navigation by continuously monitoring sensor data, control signals, and robot responses [9]. Mobile autonomous robots may pick up information from user interactions and feedback. Machine learning models may modify their movement tactics to match user expectations by taking into account user choices, comments, or demonstrations. This boosts user happiness and the caliber of the navigation experience. Autonomous mobile robots can navigate effectively, adapt to changing situations, and optimize their motions for better performance thanks to machine learning-based movement scheduling and management algorithms. These methods employ previous and current data as well as human interactions to identify trends, anticipate outcomes, and improve the ability of autonomous robot navigation [10].

2. Related Works

This study provides a standardized structure for incorporating task scheduling and routing control on a shop floor powered by mobile autonomous robots, a more and more popular IM pattern. We explicitly suggest a multi-agent architecture that includes human beings, machines, and mobile robots. Like any other cyber-physical system, the design of the fundamental software platform and the selection of the underlying algorithm affect how well IM systems work [11]. The study suggests a reinforcement learning method in which an agent constructs pathways on a predetermined layout while being rewarded based on several parameters that reflect the intended properties of the system. The findings demonstrate that, for an elevated amount of AMRs running in the system, the suggested technique outperforms the conventional shortest-path-based strategy in terms of throughput and reliability. while the demand for high throughput necessitates the operation of a relatively high number of AMRs in comparison to the size of the area in which the robots work, the adoption of the suggested technique is advised [12]. The question under study was whether geographic data mining, digital twins based on simulation, and real-time monitoring technologies might enhance remote sensing robots. Shiny software was used to produce the flow diagram of evidence-based data that was gathered and handled using the Preferred Reporting Items for Systematic Reviews and Meta-analysis (PRISMA) criteria. For the first bibliometric mapping (data visualization) Dimensions was used, while VOSviewer was used for the layout algorithms [13]. The complexity of robotic jobs and settings tends to increase along with the complexity of control interfaces. Traditional input methods including touch, voice, and gesture may not be appropriate for all users. Participants with glide limitations may not be able to manage such systems, even though they are the ones who require robotic aid the most. Although some users can put out the effort to get comfortable with a robotic system [14]. This study, to design a special architecture that allows these users to converse with a robotic service assistant only by thinking in a closed-loop setting. One of the system's interconnected components is the brain-computer interface (BCI), which uses non-invasive neuronal signal recording and co-adaptive deep learning. Other components include high-level task planning based on referring expressions, navigation and manipulation planning, as well as environmental perception [15]. This work tackles the problem of autonomously mapping unknown small celestial bodies while passing by them close by. Here, a Deep Reinforcement Learning (DRL)-based forecast approach is proposed to increase surface mapping effectiveness by the intelligent autonomous selection of the image capture epochs. Learned policies are compared to standard policies in a series of conceivable scenarios, and the Neural Fitted Q (NFQ) and Deep Q Network (DQN) techniques are examined [16].

3. Methodology

An advanced method for enabling efficient and intelligent navigation of robots in a variety of contexts, machine learning-based movement scheduling and management is a key component of autonomous mobile robot navigation. Fig.1 shows the flow of this study.

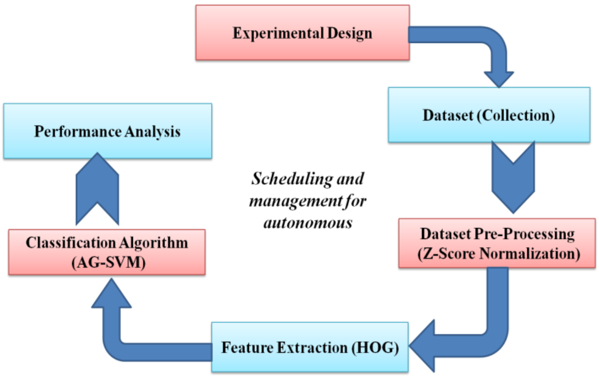

Fig.1. Flow of this study

3.1 Data set

The suggested adaptive algorithm's inputs consist of the distances between the left and right wheels and their estimated probabilities from the AMR at each moment. The output velocity in this context is referred to as V_{out}, and it is the same as the difference between the calculated probabilities of velocities of both wheels. The calculated value is contrasted with the intended velocity or V_D. The difference between V_D and V_{OUT} is used to calculate the error signal, E(x). The block coordinate descent method makes use of this signal to update the weights W1, W2, and W3 in each iteration, which is represented by a combined weight vector, i.e., $P_k = [D\ P(V_{RW})\ P(V_{LW})]$. The dataset has 2920 total data points, or (2920 x 6) = 17520, for each of the six sets. The remaining 5256 data points are used for validation, leaving 12264 data points for training.

3.2 Preprocessing using Z-Score Normalization

Z-score normalization, also known as zero-mean normalization, is the process of normalizing any output descriptors by calculating the normalized mean and standard deviation for each parameter over several test datasets. Each attribute is given its mean and standard deviation. The generalized equation 1 specifies the replacement as follows:

$$v' = \frac{v-\mu_A}{\sigma_A} \quad (1)$$

where A differences in values" stands for the attribute's standard deviation and A means" stands for the attribute's σ_(A)= As a result, every attribute in the dataset has zero fluctuation and zero significance. Before constructing a trainee collection and starting the training method, each training sample in the data set is first put through the Z-Score normalization process. A training data collection's average, variance, and statistical significance for each statistic must be calculated, noted, and utilized as weights in the final system design. It is a preprocessing phase in the architecture of neural networks. Given that the neural network was trained on a different kind of dataset, its outputs may vary significantly from the normalized data. By reducing their volume, this statistical normalization has the benefit of lessening the effects of data abnormalities.

3.3 Feature extraction using Histogram of Oriented Gradients (HOD)

HOG is a well-liked feature extraction method in computer vision that is used for object detection and identification. It records the local gradient data existing in a picture and transforms it into a feature vector. HOG is often used to characterize the quality of a particular local gradient distribution. When utilized for identifying targets, it achieves excellent results. Because of this, the HOG feature may be effectively used to represent the nature of the local gradient allocation.

Determine the gradient's magnitude and orientation:

Using these formulas, we can calculate the A and B orientation intensity gradients:

$$H_w = n(w+1,z) - J(w-1,z) \quad (2)$$

$$h_z = w(w,z+1) - J(w,z-1) \quad (3)$$

where H_w and h_z are the horizontal and vertical gradients, respectively; the gradient amplitude m(x, z) indicates the variance in the size of the grey level.

Equations (4) and (5) may also be used to compute gradient amplitude and direction:

$$n(w,z) = \sqrt{H_w^2 + H_w^2} \quad (4)$$

$$\theta(w,z)=arctan\left(\frac{H_z}{H_w}\right) \quad (5)$$

3.3.1 The Process of a Cell Histogram

Histograms are built by tallying the frequency with which each orientation of a gradient appears in each of the bins designated for that orientation. The direction of each gradient is used to determine which bin it is added to. The histogram represents regional edge directions via its quantification of the cell's internal gradient orientation distribution. To make the histogram more forgiving of little changes in brightness and contrast, normalization is an available option. Each bin value is divided by the total bin value in L1 normalization, whereas in L2 normalization, each bin value is divided by the Euclidean norm of the histogram vector. The local edge information in an image is captured by the HOG method as it builds histograms of cells based on gradient orientations. To further describe the picture in tasks like object identification and recognition, these histograms are employed as features.

3.3.2 Block specification

A single feature vector is constructed by concatenating the normalized histograms of each of the image's blocks. In addition to encapsulating information about local gradients in individual cells, this feature vector also includes spatial correlations as measured by block normalization. In HOG, the normalizing of cell contributions within blocks is a crucial step for dealing with differences in light and contrast. When the HOG feature descriptor is applied to a group of cells inside a block and then normalized, it becomes more resistant to variations in illumination and contrast.

3.3.3 The block's gradient is normalized

The HOG algorithm is often used to extract features from images. By segmenting a picture into tiny cells, generating histograms of gradient orientations inside each cell, and stringing together the feature vector, it can capture local gradient information. Each histogram's density is determined.

$$U^* = \sqrt{\frac{u}{||U||_L + 1.1\varepsilon'}} \quad (6)$$

One definition of density is the fraction of total occurrences or values that fall inside a certain interval. The density quantifies the extent to which gradients are concentrated across histograms.

3.4 Augmented Gradient- Support Vector Machine (AG-SVM)

AG-SVM is a kind of supervised learning that uses a predefined function to predict the label of an output based on the input values. It may improve the model's capacity to generalize by minimizing the errors of the sample points while lowering the structural hazards. Assume that there are l sample points and n indices in the datasets that need to be categorised.

AG-SVM, a well-known supervised ML technique, is used for both classification and regression. It can process difficult information and work in high-dimensional feature spaces. The primary objective of AG-SVM is to locate a hyperplane that optimally separates data points into their respective classes while reducing the distance between the hyperplane and the data points that are closest to the hyperplane. AG-SVM aims to maximize the margin and minimize the classification error. To get started, amass some training data that has already been labeled so that each data point already knows what category it belongs to. AG-SVM is a binary classifier; hence the data must be separated into those two categories. Problems involving many types may need multiple methods to solve. Filter the information to extract useful properties for the classification task. It is possible to encode categorical features, although AG-SVM performs better with numerical features. If you want to be sure that no one feature is overpowering the learning process, you should normalize or standardize the feature values. Min-max scaling and z-score normalization are two common types of scaling employed. The AG-SVM model is trained at the SVM Model Training step by feeding it the labeled data. The optimal hyperplane is the one that produces the largest gap between the two sets of data. Using a technique of mathematical optimization, AG-SVM can resolve this problem. Data may be moved into a higher dimensional space where linear separation is possible by the use of kernel functions in AG-SVM. Some typical kernel functions are the linear kernel, polynomial kernel, Gaussian (RBF) kernel, and sigmoid kernel. Which kernel should be utilized is determined by the complexity of the problem and the kind of data being processed. To find the optimal hyperplane, AG-SVM optimization requires the resolution of a quadratic programming problem. The optimization process seeks to maximize profit while lowering a cost function that punishes poorly classified data points. To determine the support vectors, the Lagrange multipliers are computed based on the data points that are on the boundary or extremely near to it. This might be accomplished with the help of strategies like grid search or randomized search. AG-SVM has found use in many different areas, including text classification, image recognition, bioinformatics, and even the financial sector. They are widely used because of their robust theoretical foundation, ability to deal with a wide variety of data distributions and resistance to overfitting. The foregoing discussion allows us to formulate the following constrained optimization problem as a definition of the optimal separation surface.

$$\phi(u) = \frac{1}{2}\|u\|^2 = \frac{1}{2}(u.u) \quad (7)$$

Equation (5-7) allows us to define the Lagrange function as follows:

$$K(u, a, \alpha) = \frac{1}{2}(u.u) - \sum_{j=1}^{m} \quad \alpha_j\{[(u.v_j) + a] - 1\} \quad (8)$$

$$\sum_{j=1}^{m} \quad z_j \alpha_j = 0 \quad (9)$$

$$\alpha_j \geq 0, j = 1,2, \dots, m \quad (10)$$

Where $\alpha_j > 0$ is the coefficient of Language. The problem is finding the Lagrange function's minimum of .. u and α. u and α seek partial differential and make them equal to zero, The original issue may be reduced to the dual issue below, which is straightforward: the restrictions that Equation represents (11-14):

$$R(\alpha) = \sum_{j=1}^{m} \quad \alpha_j - \frac{1}{2}\sum_{j,i=1}^{m} \quad \alpha_j \alpha_i z_j z_i (v_j . v_i) \quad (11)$$

$$u^* = \sum_{j=1}^{m} \quad \alpha_j^* z_j v_j \quad (12)$$

$$\alpha_j(z_j(u.v_j + a) - 1) = 0, j = 1,2, \dots, m \quad (13)$$

$$e(v) = sgn\{(u^*.v) + a^*\} = sgm\{\sum_{j=1}^{m} \quad \alpha_j^* z_j (v_j.v) + a^*\} \quad (14)$$

4. Results and Discussion

This section discusses in detail the findings of the proposed methodology (AG-SVM) with the existing methods utilized in this research are Conventional Neural Networks (CNN), Deep Neural Networks (DNN), and long short-term memory (LSTM). To analyse the efficiency of the proposed method, parameters such as accuracy, precision, recall, and f1-score are utilized in this research. Here TP denotes true positive, TN denotes true negative, FP denotes false positive, and FN denotes false negative.

Table.1. Numerical outcomes of proposed and existing methods

Methods	Accuracy %	Precision %	F1-score %	Recall %
CNN [17]	75	70	80	73
DNN [18]	78	77	70	87
LSTM [19]	83	85	83	89
AG-SVM [Proposed]	96	89	92	95

A. Accuracy

A difference between the result and the true number is caused by inadequate precision. The percentage of actual outcomes reveals how balanced the data is overall. Accuracy is assessed using an equation (15).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (15)$$

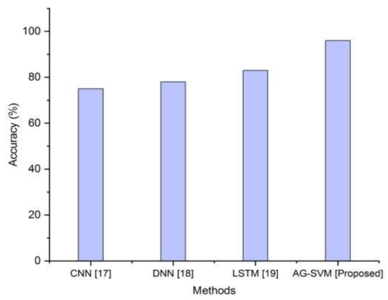

Fig.2. Comparison of Accuracy

Fig.2 shows the comparable values for the accuracy measures. When compared to existing methods like CNN, which has an accuracy rate of 75%, DNN, which has an accuracy rate of 78%, and LSTM, which has an accuracy rate of 86.64%, the recommended method's AG-SVM value is 96%. The proposed AG-SVM has greater accuracy than existing approaches and works well in categorizing autonomous mobile robot navigation.

B. Precision

The most crucial standard for accuracy is precision, it is clearly defined as the percentage of properly categorised cases to all instances of predictively positive data. Equation (16) is used to compute the precision.

$$Precision = \frac{TP}{TP+FP} \quad (16)$$

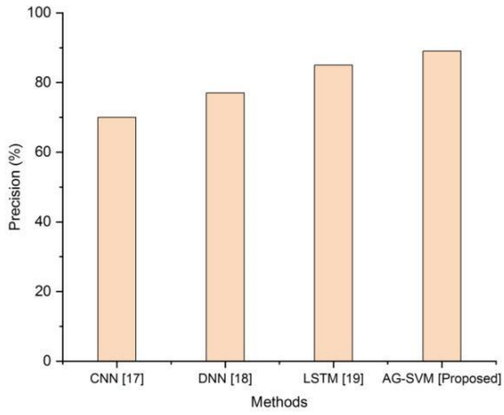

Fig.3. Comparison of Precision

Comparable values for the precision measures are shown in Fig.3. This proves the suggested strategy may provide performance results that are superior to those obtained by the current study methods. The precision of the proposed approach AG-SVM is 89%, which performs better than existing outcomes. Include DNN, CNN, and LSTM precision rates are 77%, 70%, and 85%.

C. Recall

The potential of a model to identify each important sample within a data collection is known as recall. The recall is calculated using equation (17).

$$Recall = \frac{TP}{TP+FN} \quad (17)$$

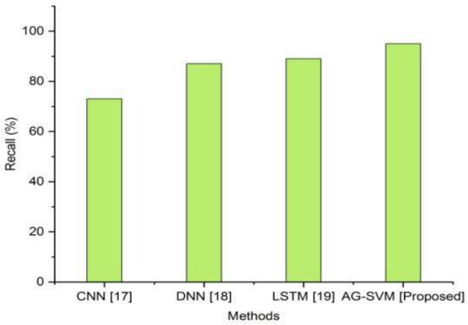

Fig.4. Comparison of Recall

Fig.4 shows the comparative data for the recall metrics. Recall rates for CNN were 73%, DNN 87%, LSTM 89%, and AG-SVM 95%.The proposed method performed better than the current results with a recall of 95%.

D. F1-score

The harmonic mean of the proposed model is computed to merge "recall and precision" into a single component called the f1-score. Equation (18) is used to determine the f1-score.

$$F1 - score = \frac{2\times precision \times recall}{precision+recall} \quad (18)$$

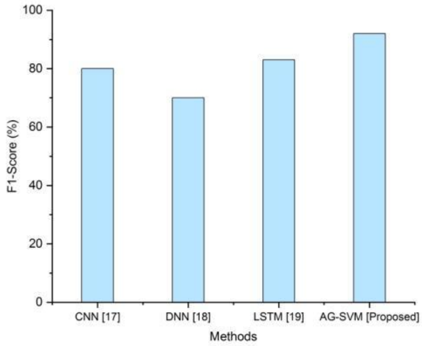

Fig.5. Comparison of F1-score

Fig.5 shows the comparative data for the recall metrics. Recall rates for CNN 80%, DNN 70%, LSTM 83%, and AG-SVM 92%.The suggested method outperforms current results with an F1-score of 95%.

5. Conclusion

The AG-SVM algorithm, created especially for this research, makes use of augmented gradient optimization methods to enhance decision-making skills, resulting in more precise and effective movement scheduling and control. In general, the recommended AG-SVM algorithm and HOG feature extraction contribute to the system's the amazing performance in terms of accuracy 96%, precision 89%, recall 95%, and F1-score 92% in the context of movement scheduling and management for autonomous mobile robot navigation [20]. These metrics pertain to the system's ability to navigate autonomously within its environment. Autonomous mobile robot navigation is much improved by the machine learning-based movement scheduling and management system that includes HOG feature extraction and the suggested AG-SVM method. The remarkable performance metrics show its promise in several fields, such as manufacturing, supply chain management, and security. In further research, it may be possible to develop intuitive and natural interfaces that enable people to offer high-level commands or preferences to the robot. This will make it possible for humans and robots to effectively cooperate in shared workplaces or on joint projects.

References

[1] Jokić A, Petrović M, Kulesza Z, Miljković Z. Visual Deep Learning-Based Mobile Robot Control: A Novel Weighted Fitness Function-Based Image Registration Model. InNew Technologies, Development and Application IV 2021 May 12 (pp. 744-752). Cham: Springer International Publishing.

[2] Hu H, Jia X, He Q, Fu S, Liu K. Deep reinforcement learning based AGVs real-time scheduling with the mixed rule for the flexible shop floor in industry 4.0.

Computers & Industrial Engineering. 2020 Nov 1;149:106749.

[3] Zeng J, Ju R, Qin L, Hu Y, Yin Q, Hu C. Navigation in unknown dynamic environments based on deep reinforcement learning. Sensors. 2019 Sep 5;19(18):3837.

[4] Eskandari M, Savkin AV. Deep Reinforcement Learning Based Joint 3D Navigation and Phase Shift Control for Mobile Internet of Vehicles Assisted by RIS-equipped UAVs. IEEE Internet of Things Journal. 2023 May 18.

[5] Fan T, Long P, Liu W, Pan J. Distributed multi-robot collision avoidance via deep reinforcement learning for navigation in complex scenarios. The International Journal of Robotics Research. 2020 Jun;39(7):856-92.

[6] Wesselhöft M, Hinckeldeyn J, Kreutzfeldt J. Controlling fleets of autonomous mobile robots with reinforcement learning: a brief survey. Robotics. 2022 Aug 30;11(5):85.

[7] Le AV, Kyaw PT, Veerajagadheswar P, Muthugala MV, Elara MR, Kumar M, Nhan NH. Reinforcement learning-based optimal complete water-blasting for autonomous ship hull corrosion cleaning system. Ocean Engineering. 2021 Jan 15;220:108477.

[8] Pei M, An H, Liu B, Wang C. An improved dyna-q algorithm for mobile robot path planning in an unknown dynamic environment. IEEE Transactions on Systems, Man, and Cybernetics: Systems. 2021 Jul 26;52(7):4415-25.

[9] Ho TM, Nguyen KK, Cheriet M. Federated deep reinforcement learning for task scheduling in a heterogeneous autonomous robotic system. IEEE Transactions on Automation Science and Engineering. 2022 Nov 14.

[10] Chehelgami S, Ashtari E, Basiri MA, Masouleh MT, Kalhor A. Safe deep learning-based global path planning using a fast collision-free path generator. Robotics and Autonomous Systems. 2023 May 1;163:104384.

[11] Agrawal A, Won SJ, Sharma T, Deshpande M, McComb C. A multi-agent reinforcement learning framework for intelligent manufacturing with autonomous mobile robots. Proceedings of the Design Society. 2021 Aug;1:161-70.

[12] Kozjek D, Malus A, Vrabič R. Reinforcement-learning-based route generation for heavy-traffic autonomous mobile robot systems. Sensors. 2021 Jul 14;21(14):4809.

[13] Andronie M, Lăzăroiu G, Iatagan M, Hurloiu I, Ștefănescu R, Dijmărescu A, Dijmărescu I. Big Data Management Algorithms, Deep Learning-Based Object Detection Technologies, and Geospatial Simulation and Sensor Fusion Tools in the Internet of Robotic Things. ISPRS International Journal of Geo-Information. 2023 Jan 21;12(2):35.

[14] Thakur A, Das S, Kumari R, Mishra SK. Machine Learning based Intelligent Model for Path Planning Obstacle Avoidance in Dense Environments for Autonomous Mobile Robot.

[15] Piccinin M, Lunghi P, Lavagna M. Deep Reinforcement Learning-based policy for autonomous imaging planning of small celestial bodies mapping. Aerospace Science and Technology. 2022 Jan 1;120:107224.

[16] Ho DK, Ben Chehida K, Miramond B, Auguin M. Learning-Based Adaptive Management of QoS and Energy for Mobile Robotic Missions. International Journal of Semantic Computing. 2019 Dec;13(04):513-39.

[17] Jokić A, Petrović M, Kulesza Z, Miljković Z. Visual Deep Learning-Based Mobile Robot Control: A Novel Weighted Fitness Function-Based Image Registration Model. InNew Technologies, Development and Application IV 2021 May 12 (pp. 744-752). Cham: Springer International Publishing.

[18] Lee S, Kim Y, Kahng H, Lee SK, Chung S, Cheong T, Shin K, Park J, Kim SB. Intelligent traffic control for autonomous vehicle systems based on machine learning. Expert Systems with Applications. 2020 Apr 15;144:113074.

[19] Niranjan DR, VinayKarthik BC. Deep learning-based object detection model for autonomous driving research using Carla simulator. In2021 2nd international conference on smart electronics and communication (ICOSEC) 2021 Oct 7 (pp. 1251-1258). IEEE.

[20] Sundar Ganesh, C. S. & Daisy Mae, R. B. (2022). A Study on the Usage of Robot Navigation to Determine Robot's Own Position in its Frame of Reference and then to Plan a Path towards Some Goal Location. Technoarete Transactions on Industrial Robotics and Automation Systems (TTIRAS). 2(4), 1–6.

[21] Kapoor, E. ., Kumar, A. ., & Singh , D. . (2023). Energy-Efficient Flexible Flow Shop Scheduling With Due Date and Total Flow Time. International Journal on Recent and Innovation Trends in Computing and Communication, 11(2s), 259–267. https://doi.org/10.17762/ijritcc.v11i2s.6145

[22] Paul Garcia, Ian Martin, Laura López, Sigurðsson

Ólafur, Matti Virtanen. Enhancing Student Engagement through Machine Learning: A Review. Kuwait Journal of Machine Learning, 2(1). Retrieved from http://kuwaitjournals.com/index.php/kjml/article/view/163

[23] Kumar, S.A.S., Naveen, R., Dhabliya, D., Shankar, B. M., Rajesh, B. N. Electronic currency note sterilizer machine (2020) Materials Today: Proceedings, 37 (Part 2), pp. 1442-1444.