A Robot Obstacle Avoidance Method based on Random Forest HTM Cortical Learning Algorithm

Ahmed Mahdi Abdulkadium
College of Agriculture, Information System, Computer Science, Al-Qasim Green University, Iraq.
E-mail: ahmed_mahdi@uoqasim.edu.iq

Received August 02, 2020; Accepted October 03, 2020
ISSN: 1735-188X
DOI: 10.14704/WEB/V17I2/WEB17067

Abstract

Robotics mainly concern with the movement of robot with improvement obstacle avoidance, this issue is handed. It contains of a Microcontroller to process the data, and Ultrasonic sensors to detect the obstacles on its path. Artificial intelligence is used to predict the presence of obstacle in the path. In this research random forest algorithm is used and it is improved by RFHTMC algorithm. Deep learning mainly compromises of reducing the mean absolute error of forecasting. Problem with random forest is time complexity, as it involves formation of many classification trees. The proposed algorithm reduces the set of rules which is used for classification model, to improve time complexity. Performance analysis shows an significant improvement in results as compare to other deep learning algorithm as well as random forest. Forecasting accuracy shows 8% improvement as compare to random forest with 26% reduced operation time.

Keywords

Decision Tree, Neural Network, Mobile Robots, HTM Cortical Algorithm.

Introduction

Now a days, an application of robotics in present in every industry and home appliances. It helps to automate many complicated task. With the advancement in static or steady task, concept of mobile robot also has a widespread. Mobile robot is able to relocate itself using own algorithm. Which can handle task in offices, hospitals and home. This is called as Anthropomorphic robots. Anthropomorphic robots are able to perform real world task which needs movement, it uses noise detection and obstacle avoidance algorithm. Artificial intelligence has a capability to predict the occurrence of any obstacle using inputs of sensor node. Sensor inputs like voice and camera images helps to predict the occurrence of any obstacle present in its way. Using Voice data of obstacle detection involves many challenges, in some real case scenario, a source of voice cannot be an obstacle, like falling of water or sound of wind.

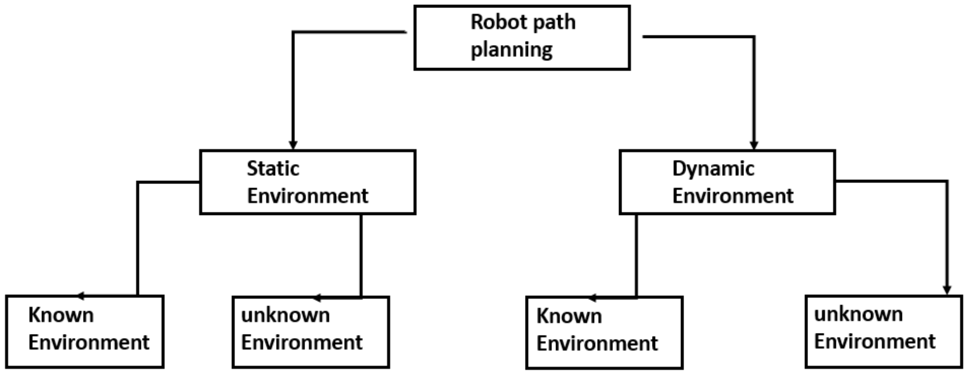

Figure 1 Robot path planning

Path planning is an important primitive for autonomous mobile robots that lets robots find the shortest or optimal path between two points. Robot path planning is classified as path planning in static environment and dynamic environment [11]. For static environment the path is fixed and robot has to travel the same path. Hence output robot path planning algorithm is required. In case of Dynamic Environment the two end points could be variable or unknown at the initial state. In case of known environment, the positions of obstacles are already known, hence there is no need of detecting obstacles, but in real world environment things are continuously moving like in factory, home. This positions may change, hence there is a requirement to avoid this dynamically moving obstacles. Obstacle detection is a complex task for many robotics systems. For a static base environment, robots should avoid the self-collisions and objects present in the environment. For dynamic environment the obstacle avoidance is complicated as robot need to handle the detection for static as well as dynamic object.

Artificial intelligence has proven the acceptable accuracy of prediction in many field where data is present for analysis. Applications where there is a lot of data for study purpose can be applied to machine learning algorithm which can predict the future data with acceptable accuracy rate.

Application of Artificial Intelligence

Pharmaceutical and medical research: Artificial Intelligence is used in many chemical research which helps in advancement in pharma and drugs [12]. It can also integrate with microscope in medical treatment which takes cell data as input and provide the required information about nature of cells.

Defence and Aerospace: Artificial intelligence in defence provides an opportunity to identify enemy based on data provided to machine learning model. It can also be integrated with drones to track the specific group or person [13]. Same is used in aerospace technologies and satellites.

Electronics and telecommunication: Electronic market is currently focusing on the advancement in the electronic product with AI. This provides automation and reduces the human requirement in many applications [14]. Many home appliances are now using AI for their enhancement, it generally uses voice and images for data processing purpose which can automate task like operating AC temperature, switch management.

Robotics: with the initial research in robotics, AI helps to perform many common applications like calculations, monitoring. With advancement now it is possible to perform action which requires movement. This redefines the usage of Artificial Intelligence in the robotic field. In this research advanced random forest decision tree algorithm is used to detect and avoid the obstacle present in the way. Artificial Intelligence is used in robotic navigations. Images captured form the camera is used to predict the obstacles in travelling path. There exists many simulating technologies which helps to integrate the AI technologies. Many data mining tools supports bots API this includes Python, R and MATLAB. Being an open source R and python has many libraries which connect with API and determines the hidden pattern present in the robotic datasets [10]. This libraries reduces the coding efforts and helps researcher for analysis purpose.

This paper mainly focuses on the object detection using artificial intelligence and decision tree. Camera present on the robot is used to capture the images, this images are input to machine learning algorithm, using which an object can be classified as obstacle, movable object or no object. This helps the movement of robot. Object detection using images also involves many major challenges like there involves many features in one image, makes it more complex to understand, along with that similarity between two different objects is also a major challenge, object detection of sofa and bed is a best example of this case, where there is a lot of similar feature, but they belong to different class. Machine learning model can learn the weightage of each obstacle by which it can predict the impact of obstacle with respect to robotic motion.

This paper presents the needs of AI in Robotics and proposed algorithm to enhance the performance of existing robotic devices. In the first Introduction section, existing approach, need of AI in robotics and problems occurs while knowledge extraction is studied. Section II describes previous researches with respect to data mining in robotics is described. Section III explores the advancement in decision trees to enhance the performance of navigation technologies. Section IV and V presents experimental analysis, conclusion and future work.

Related Work

Obstacle avoidance is a crucial step in robotic system, many researches are going on into finding the optimised machine learning algorithm for obstacle detection. Rapidly-Exploring Random Tree (RRT) Planner and Randomized A* (RA*) algorithm has proven a good accuracy rate in detecting the obstacles present in the way [7]. Even though this algorithm perform the obstacle avoidance in static environment but in dynamic and unknown environment accuracy of this algorithm falls drastically this require an alternative system which can support dynamically moving obstacles.

In research Analysis of fuzzy rules for robot path planning, Various fuzzy logic methods are conveniently employed for path planning of mobile robots as they are most helpful in handling uncertain data. It uses short paths to determine the obstacles and shortest path simultaneously. Various membership function including trapezoidal and Triangular, Bell and Trapezoidal is used to determine path [2]. In Dual Expanded Guide Circle (Dual-EGC) algorithm for obstacle avoidance of remotely operated mobile robot, Expanded Guide Circle algorithm is used to detect the presence of any obstacle in the path. Smoothening technique is applied EGC algorithm to avoid the zig zag nature of predicted path [1]. In Control a mobile robot in Social environments by considering human as a moving obstacle, accuracy of algorithm is increased considering human actions is used. Fuzzy logic is used to determine the human gestures and action and path detection algorithm were used to determine path [3]. BCM Obstacle Avoidance Algorithm which is based on ultrasonic sensor is used to detect hanging objects. This research captures the images in 3 dimensions and uses previous stored data to predict the possibility of path [4].

Application of this features is being used in hospitals, In research obstacle avoidance of hospital ward inspection robots in complex environment, vision-laser sensors is used to capture the images. Proposed algorithm collects the dimensions of the object and the distance of object from robot. Using this dataset**,** virtual obstacle object is created and stored in queue for processing. Using this virtual object path is defined without obstacle collision [5]. In this research Stereo Vision and Bayesian Approach is proposed to identify the obstacles present in the path. This research uses stereo camera to pick the multiple objects at the same time**,** which reduces the time complexity also detects obstacles with accuracy rate better than existing algorithms [6].

1) Architecture of Robot

This section explains the Architecture of Robot. The robot which is used in this research is equipped with two wheels. As illustrated in figure 2, architecture of robot is consist of camera, three distance sensors, raspberry Pi kit, capacitors, wheels and motors. Camera and three distance sensors are input devices to the system. Camera capture the still images from the external environment while sensors detects the distance from various objects. 5 A driver DC motor and capacitor is used to control the motion and voltage. Wheels are operated based on the output of the proposed algorithm.

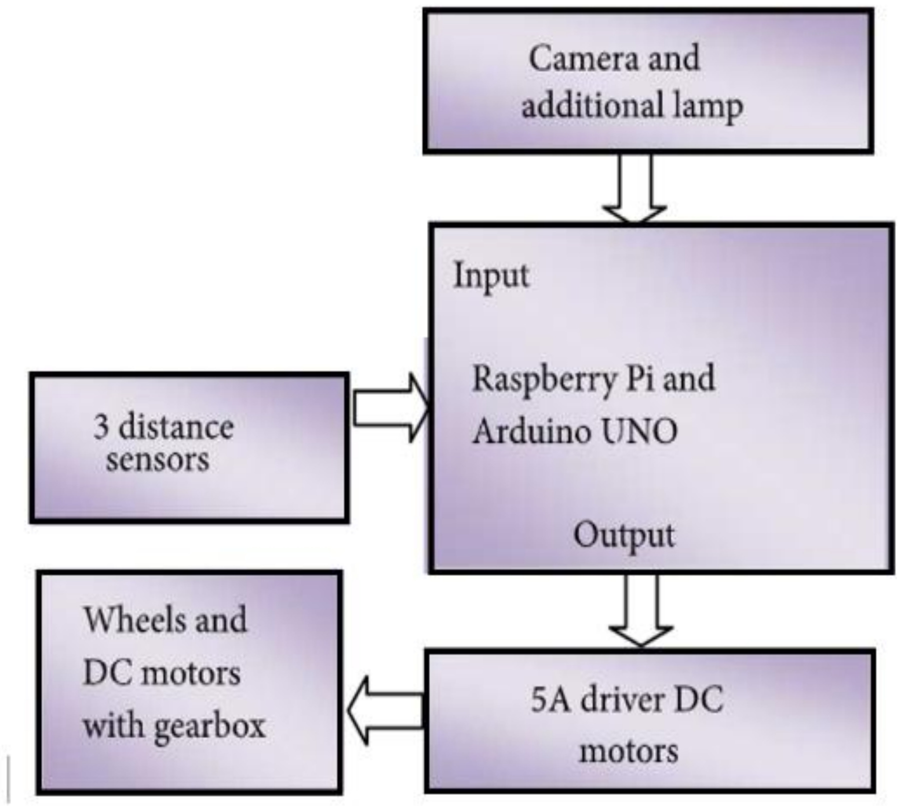

Figure 2 Architecture of mobile robot

Three ultrasonic sensors measure the distance from the obstacle. This sensors uses 40 KHz of frequency with maximum angle of deviation of 30 degree. The ultrasonic sensor has high frequency, high sensitivity and high penetrating power therefore it can easily detect the external or deep objects. Although is difficult to detect the small objects, camera captures this images and AI controls the motor.

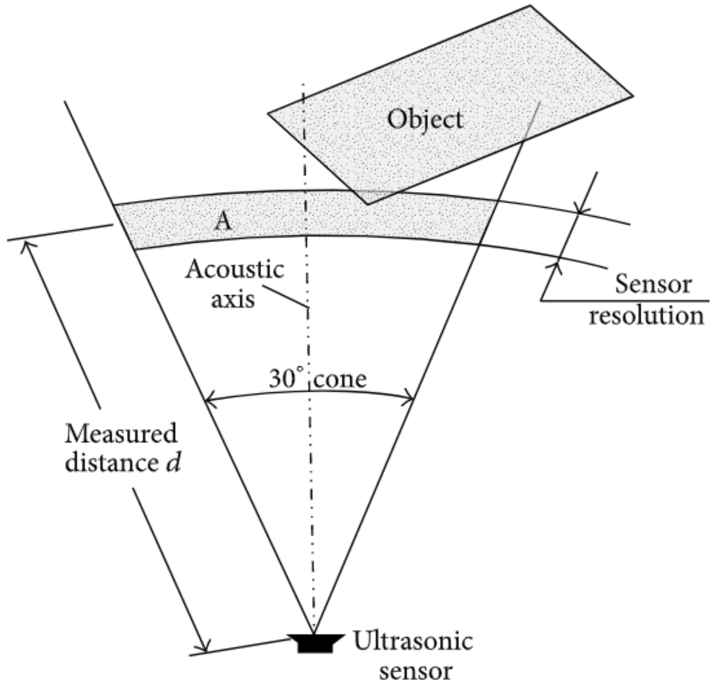

Figure 3 Two-dimensional projection ultrasonic sensor

Ultrasonic sensors is used to detects the objects by short ultrasonic burst and its echo to measure the distance from the object. The measure of this burst is 40 kHz. It hits the object and bounce back to sensor where ping sensor capture the echo and terminate with. Width of this pulse is used to measure the distance of the object from sensors.

Random Forest Algorithm

Data collected from the sensors and camera is used as input to the artificial intelligence algorithm. Decision trees are capable of handling the attributed data. It is a tree like model, which includes probability of output and cost. It is a type of conditional algorithm which is being used in many recent research to find the near optimised solution. It is mostly used in decision analysis task, in this research robot has to make a decision with respect to the obstacle present in the path.

There exist different many decision tree algorithm, in this research random forest decision tree algorithm is used for the analysis purpose, the output of random forest is compared with other decision trees and neural network algorithms it is observed that the prediction accuracy of random forest is more as compare to other algorithm. This section describes the principle, pseudo code working of random forest algorithm.

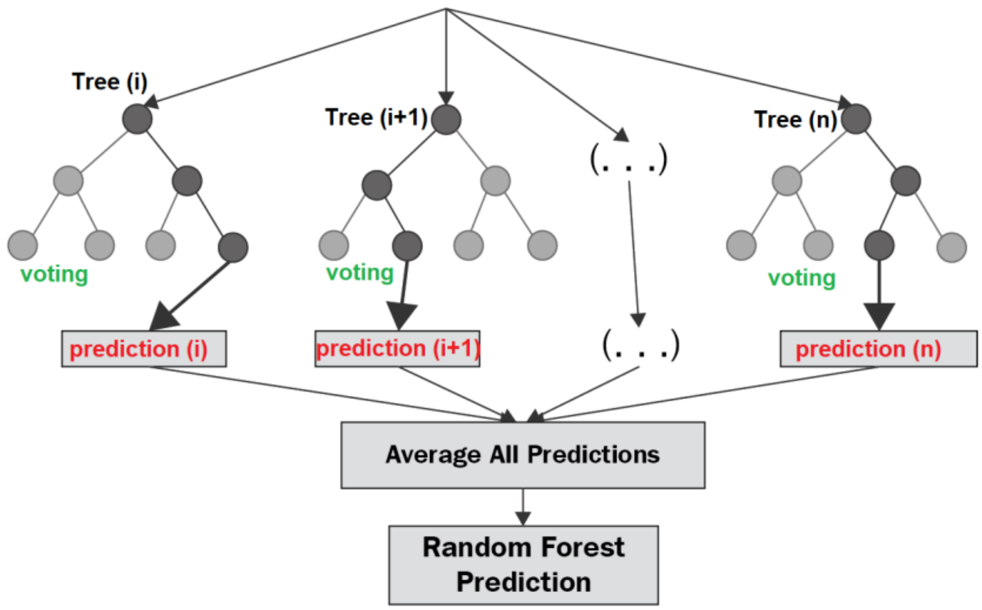

Figure 4 Random Forest tree

Random forest is an ensemble decision tree method of classification algorithm. It creates multiple tree in training phase. N numbers of trees are constructed, where n is an input. Each tree is constructed based on random subsets taken from the original training dataset [8]. Using this approach n numbers of trees are constructed and voting technique is used to decide output class prediction. It uses bootstrap aggregation or bagging to reduces the variance, and improves data stability. Bagging creates bags which were filled with randomly selected data from the whole training set, size of each of this bag is less the training dataset, each of this bag is used as a training set to for training which reduce the variance. It also avoids the data overfitting issue and improves the accuracy of algorithm. Each tree is created using the Information Gain and Entropy of subset. Voting is used to predict output class. Votes are provided to each class while selection and class with highest Votes is selected as prediction output.

Entropy is calculated by

$$H(S) = \sum_{x \in X} p(x) \log_2 (1/p(x)) \ (1)$$

Information Gain is calculated by

$$IG(A, S) = H(S) - \sum_{t \in T} p(t) H(t) \ \ (2)$$

Where, H(s) – Entropy, S –dataset, X – set of classes, S P(x)- total number of elements in x class to total number of elements in set S, H(s) – Entropy of set s., H(t)-Entropy of subset t.

Algorithm of Random Forest

1) Sample set is used to create n number of subsets
2) Using information gain and entropy, decision tree is constructed for each subset.
3) Votes are assigned to each class
4) Class with highest vote is selected as output predication.

In this research R is used to collect the data and perform data modelling. R is an open source programming language with many AI libraries. In this research *randomForest* and *RWeka* libraries were used.

```
r = randomForest(DEC~.,x)
```
This function is used to get the random forest tree with output class variable.
```
pred.prob <- predict(r,t,type="prob")
```
predict function provides the probability distribution for each output class.

Limitation of Random Forest Algorithm

1) Data Overfitting issue which may contain a noisy subset which will drop the accuracy rate of the algorithm extensively.
2) Random forest allow random seeds expect that any modification in machine learning model is difficult.

HTM Cortical Algorithm

It is a machine learning algorithm which aims to provides predictions based on algorithmic and structural properties of neocortex. It models the single layers from the cortex. Continues streams of input patterns is the input to the algorithm. It represents the sequences of the pattern in the repeated structure [9]. It grabs the hidden patterns present in the algorithm, to predict future pattern. Like other neural networks, HTM uses training on stored data, and hidden patterns were recognised and used to predict the future value. In HTM predictions are based on the patterns present in the earlier layer. HTM also has a variable length memory using which if a different previous layers can be accessed.

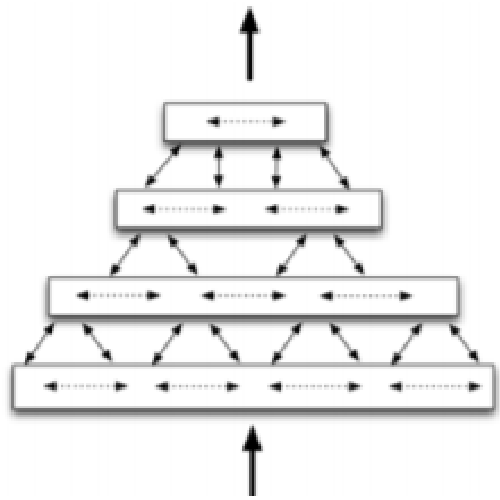

Figure: HTM regions with four levels

Algorithm Step for HTM Cortical Algorithm

1. Create a sparse distributed environment of input.

   Sparse represents that only few neurons are active while distributed means many neurons needs to be activated. Small activate neurons are sufficient to perform prediction, but highly interconnected environment is suggested, which is called as sparse distributed environment.

2. Create a representation of input with respect to previous inputs

   HTM algorithm needs to understand the structure of the input data present in the whole data set as well as in previous level. In this phase algorithm perform two steps 1) learning 2) Inference. In learning phase, learning initialized with the data present in the dataset with on-line learning, and then algorithm learns from the each level present in the model. In Inference, knowledge extracted from the each of previous level is used to define the new level. In this way more optimised solution based on the input is predicted for next level using inference.

3. Create a prediction based on current input with respect to previous inputs.

   This is the final step in the algorithm. Output of second step, that is interconnection between levels is used to predict the next level. In this step, neurons from each column of region is activated in sparse distributed environment. The output of this step is not only used in next level, but it is used in many levels which created in future.

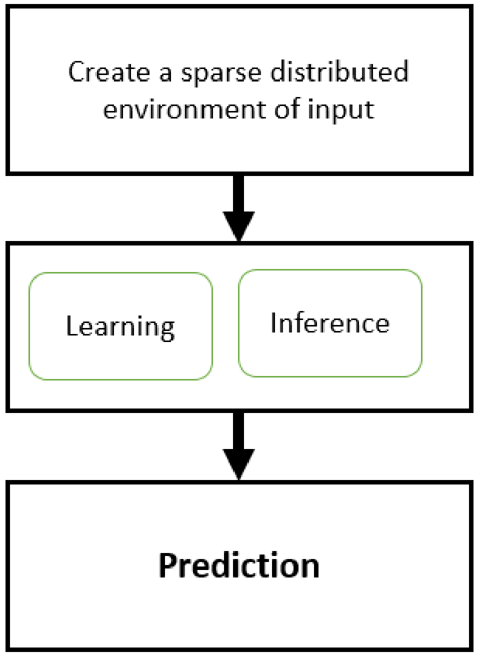

Figure: HTM cortical Algorithm

Performance of Random forest can be improved by reducing the mean absolute percentage error in training phase. It is also known as test error or generalization error. This error is defined as

$$Prederror\ (\mathrm{L}) = \mathrm{X}, \mathrm{Y}\,\{\mathrm{L}(\mathrm{Y}, \mathrm{L}(\mathrm{X}))\}(3)$$

Where, L is the learning set for ϕ L, X is a random vector and Y is categorical variable.

For improving the accuracy of the algorithm in HTM cortical algorithm is integrated with Random forest. The proposed Random Forest HTM Cortical Learning Algorithm consist of specification of both. It reduces the issue of data overfitting which is observed in random forest and it also reduces the iteration required for HTM cortical algorithm. Important specification for the proposed algorithm is prediction error. This algorithm reduces the prediction error for the sensor data. As described in figure 4 random forest, its prediction error for each level is analysed by:

$$Prederror = \sum_{i=1}^{n} (Prederror\ (\mathrm{L}) = \mathrm{X}, \mathrm{Y}\,\{\mathrm{L}(\mathrm{Y}, \mathrm{L}(\mathrm{X}))\})\ (4)$$

To reduce the prediction error random forest is merged with HTM and equation for calculating prediction error is computed. In tree, two vertices are connected with each other using single path. In a rooted tree, only one node acts a root of the whole tree. For this research we assume that the tree is a directed graph, where all the edges has a direction. In this structure if the edge is connecting node t 1 directed to node t2, then node t2 is considered as child node of t1.In rooted tree, node is considered as an internal node if it has one or more child node, and considered as a terminal if there exist no child node. It is also called as leave node. HTM algorithm divides the whole space X(t) into subspaces, which is like a sparse distributed in HTM algorithm described earlier. Using this total prediction error in each subspace is calculated, summation of this error subspace error provides the prediction error for algorithm. This is described in equation 5.

$$terr = \sum_{i=1}^{n1}(Prederror(\mathrm{Li})) + \sum_{d=1}^{n2}(Prederror(\mathrm{Ld}))..\sum_{s=1}^{nn}(Prederror(\mathrm{Ls})) \quad (5)$$

From this observation, it is clear that HTM reduces the transition between the sparse distributed environment. In some model, transition is present in the liner series, but in most of the datasets it reduces the iteration with high accuracy. HTM provides a prediction based on the inputs of many levels. For this transitions were stored in the spatial pattern, to reduce the disk space.

Pseudo code for Proposed Algorithm is as below:

1) Create a Random Forest Tree and Take User input
   Create_RF_tree
   Accept User input, X as predictor and Y as target
   Input T_test is predictor of t_dataset

2) Testing of test data is carried out initially. Each column and then each cell from the active columns and cells of column is selected as described in HTM Algorithm. Prediction state of each level is checked, depending on the prediction state and segment sequence, random forest algorithm is used to classify. Output for each level is predicted. For each prediction, mean absolute percentage error is calculated.

```
Foreach coulum in active_cols(t)
{
Var predict = false;
For(i=0 to i<Cell_count())
{
 If(predictive_State(cell, i, t-1) == true)
 {
 s = get_Active_Segment (cell, i, t-1, active_State)
 if (s.sequence_Segment == true)
 {
 model= Random_Forest_Classifier()
```

```
model.fit(X, y)
active_State(cell, i, t) = 1
}
}
If( bu_Predicted == false)
{
Foreach (col in column)
{
active_State(cell, i, t) = 1
}
}
Foreach (cell in cells)
{
Foreach (segment in segments)
{
If(segmnt_Activ(c, i, s, t))
{
predicted= model.predict(T_test)
}
}
}
```

3) Check the value of prediction, in this step model fitting is performed and this step is iterated for all the nodes until the child node is detected using HTM algorithm by dividing each active path as sub active segments.

```
Call Predict(x)
Set t = t0
While( t == child node)
{
 t = child node t0 of t
}
Return t
```

Experimental Results

In this section performance of the algorithm is analysed using cross validation, real time testing on dataset, this result is compared among the other Artificial intelligence algorithms. Initially the cross validation test is performed on the dataset to find the model which is best suitable for sensor data which is collected from robot camera. Robot camera used to capture the images and sensor is used to capture the distance, based on the result outcome of robotic movement the training data set is prepared. This is used in preliminary testing. Result of Random forest on present dataset is compared with Support vector machine, J48 decision tree, Naive Bayes classifier. These all are mostly used algorithms in the AI predictions. N-fold cross validation is used to measure the performance of the model built using this algorithms. Weka tool is used to perform this testing. Value of N is set to 10. 10-fold cross validation result is as below:

Table 1 Comparison of AI algorithms with respect to Robotic dataset

Algorithm	Correctly classified instance	Mean absolute error
J48	82.345 %	0.2561
Support vector machine	85.92 %	0.2114
Naïve Bayes classifier	83.12%	0.329
Random Forest	86.226%	0.1943

As described in table 1, correctly classified instances and mean absolute error is considered in analysing the performance of the algorithm. It was observed that correctly classified instances for Random forest algorithm is highest which is 86.22%, while the mean absolute error is 0.1943. This observation clearly indicates better performance of random forest with respect to other machine learning or Artificial intelligence algorithms.

Although with the provided dataset, Random forest has highest accuracy rate however as we discussed in section 3, Random forest has a tendency of data overfitting, which try to fit the existing data, while if new data with less relevance with model observed then accuracy of the model drops.

To overcome the drawback, and enhance the performance of the system using knowledge present at different level, HTM cortical algorithm is integrated with random forest decision tree, as random forest had highest accuracy rate. For analysis purpose mean absolute error of the random forest and proposed algorithm is plotted.

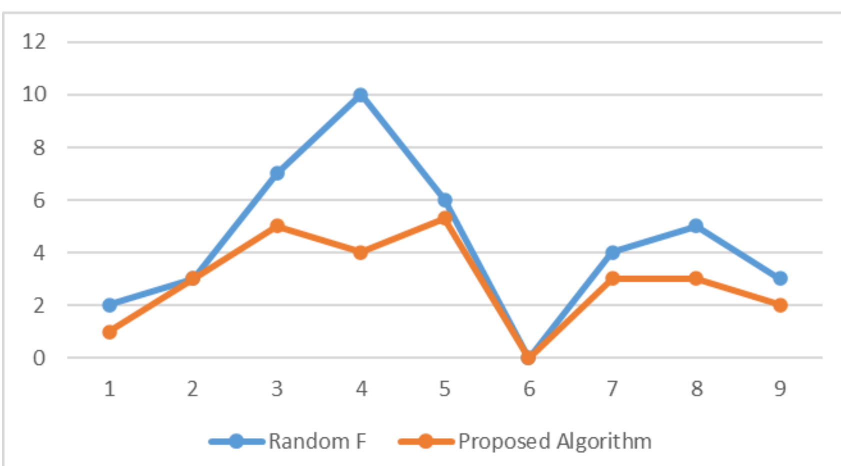

Chart 1 Comparison of mean absolute error

In chart1, X axis shows the mean absolute error of random forest and proposed algorithm, while Y axis represents the number of iterations. It is clearly observed that, mean absolute error in case of Random forest is high initially, it reduces as algorithm iterates more. However in case of proposed algorithm the mean absolute error in initial iteration is control, and it reduces further as iteration goes on increasing, to achieve the threshold random forest took 9 iteration while proposed algorithm took 7 iterations. This shows that with proposed algorithm the number of iteration required for the prediction is less. This will eventually reduce the time required for prediction.

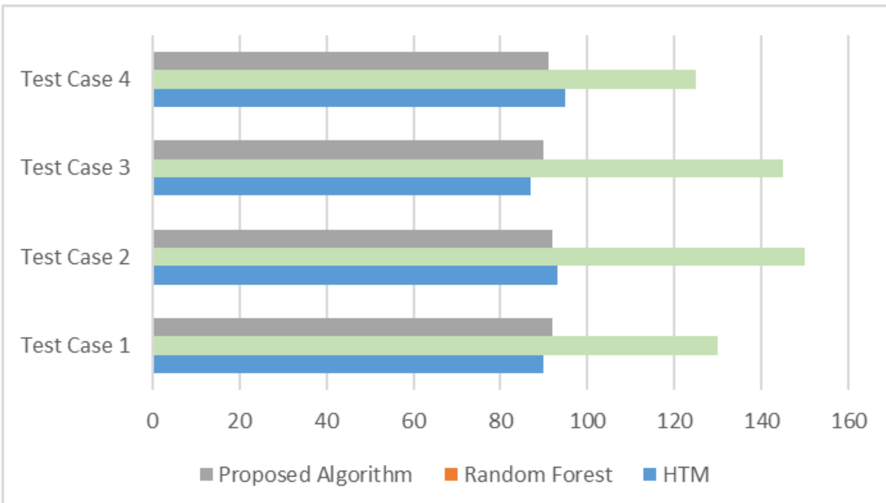

Chart 2 Comparison of time complexity

For analysing the time complexity of proposed method, four test cases with approximately similar sized and different dataset is prepared. Machine learning model is prepared for this 3 algorithm, and time required for prediction is tested. It is observed that the time required for random forest machine learning algorithm is highest. While HTM cortical and proposed algorithm has similar time complexity. This is because even though proposed algorithm reduces the iteration, it requires some extra time to build random forest algorithm. However considering the accuracy and minute enhancement in time complexity, proposed algorithm proven to be the best for real time robotic application, where there is plenty dataset and location and size of the objects were dynamic.

For the real time testing, robot is configured with the proposed algorithm, with sensor and camera as described in architecture. Path is configured with various static and dynamic objects along with this four different environments were considered. Obstacle avoidance accuracy of the algorithm is measured. Chart 3 illustrates the performance of proposed algorithm with respect to random forest and HTM cortical algorithm.

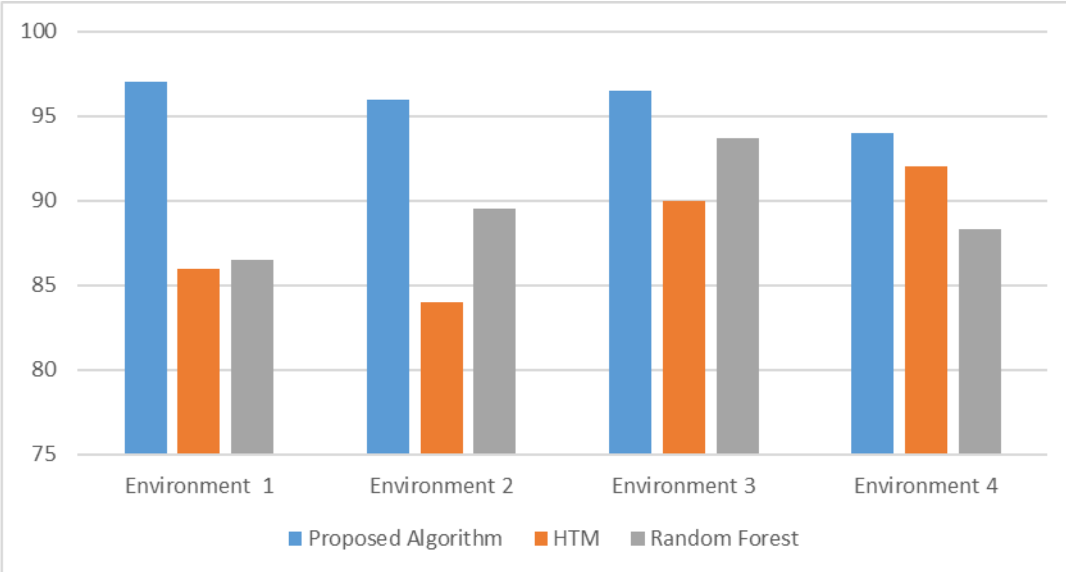

Chart 3 Real time performance Comparison

Chart 3 shows that proposed algorithm has a real time accuracy rate of 93 to 96% in variable changing environment. While HTM and Random forest's real time accuracy rate is approximately 87% and 89% respectively.

Conclusion

In this research proposed algorithm is configured with the robotic system, and the dataset as well as real time accuracy is analysed. It is observed that the without configuring any algorithms robot cannot perform task which involves a moment, as obstacles are always present. Artificial intelligence is capable of predicting the presence of such obstacles using machine learning algorithm and object detection techniques. Similar to Machine learning, HTM algorithm is also used in many research related to prediction. After comparing accuracy of the random forest on training dataset with other machine learning algorithm, it is selected as integrated with HTM cortical algorithm.

Accuracy of Random forest, HTM and proposed algorithm has compared. It is observed that the accuracy of the proposed algorithm is enhanced approximately 8 to 10%, even though this algorithm uses random forest tree, it reduces the required iterations for HTM algorithm, hence time complexity of the proposed algorithm is also in acceptable range.

References

Kim, D.H., & Kim, G.W. (2017). Dual Expanded Guide Circle (Dual-EGC) algorithm for obstacle avoidance of remotely operated mobile robot. *In IEEE 14th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, 226-227.

Sandeep, B.S., Supriya, P. (2016). Analysis of fuzzy rules for robot path planning. *In IEEE International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 309-314.

Tanha, S.D.N., Dehkordi, S.F., & Korayem, A.H. (2018). Control a mobile robot in Social environments by considering human as a moving obstacle. *In IEEE 6th RSI International Conference on Robotics and Mechatronics (IcRoM)*, 256-260.

Qijun, L., Xiuyan, Z. (2014). An Improved BCM Obstacle Avoidance Algorithm for Outdoor Patrol Robot. *In IEEE Sixth International Conference on Measuring Technology and Mechatronics Automation*, 87-90.

Yu, J., Wu, H., Wang, X. (2016). The research on obstacle avoidance of hospital ward inspection robots in complex environment. *In IEEE Chinese Control and Decision Conference (CCDC)*, 5242-5247.

Budiharto, W. (2012). Multiple Moving Obstacles Avoidance for Humanoid Service Robot Using Stereo Vision and Bayesian Approach. *In IEEE Sixth Asia Modelling Symposium*, 1-6.

Eleni, K., & Signe, M. (2017). Path Following, Obstacle Detection and Obstacle Avoidance for Thrusted Underwater Snake Robots. *IEEE International Conference on Robotics and Biomimetics (ROBIO)*.

Abbas, M.A. (2018). Improving deep learning performance using random forest HTM cortical learning algorithm. *In IEEE First International Workshop on Deep and Representation Learning (IWDRL)*, 13-18.

Kamble, R.A. (2017). Short and long term stock trend prediction using decision tree. *In IEEE International Conference on Intelligent Computing and Control Systems (ICICCS)*, 1371-1375.

Wang, X.H., & Chen, C.A. (2015). Research on competence evaluation system of data mining tools. *In IEEE International Conference on Mechatronics and Automation (ICMA)*, 1-5.

Dai, S., Huang, H., Wu, F., Xiao, S., Zhang, T. (2009). Path planning for mobile robot based on rough set genetic algorithm. *In IEEE Second International Conference on Intelligent Networks and Intelligent Systems*, 278-281.

Kaddi, S.S., & Patil, M.M. (2017). Pharma data analytics: An emerging trend. *In IEEE International Conference on Smart Technologies for Smart Nation (SmartTechCon)*, 1307-1311.

Sumari, A.D.W., Ahmad, A.S. (2017). The application of cognitive artificial intelligence within C4ISR framework for national resilience. *In IEEE Fourth Asian Conference on Defence Technology-Japan (ACDT)*, 1-8.

Li, W., Ying, J. (2008). Design and analysis artificial intelligence (AI) research for power supply—power electronics expert system (PEES). *In IEEE Twenty-Third Annual IEEE Applied Power Electronics Conference and Exposition*, 2009-2015.

Savulescu, C., Polkowski, Z. (2017). An approach for systems identification using artificial inteligence. *In IEEE 9th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, 1-4.