LEARNING OF ROBOT NAVIGATION TASKS BY PROBABILISTIC NEURAL NETWORK

Mücella ÖZBAY KARAKUŞ[1] and Orhan ER[2]

[1]Department of Computer Engineering, Bozok University, Yozgat, Turkey
mucella-ozbay@hotmail.com
[2]Department of Electrical and Electronics Engineering,
Bozok University, Yozgat, Turkey
orhaner2009@gmail.com

ABSTRACT

This paper reports results of artificial neural network for robot navigation tasks. Machine learning methods have proven usability in many complex problems concerning mobile robots control. In particular we deal with the well-known strategy of navigating by "wall-following". In this study, probabilistic neural network (PNN) structure was used for robot navigation tasks. The PNN result was compared with the results of the Logistic Perceptron, Multilayer Perceptron, Mixture of Experts and Elman neural networks and the results of the previous studies reported focusing on robot navigation tasks and using same dataset. It was observed the PNN is the best classification accuracy with 99,635% accuracy using same dataset.

KEYWORDS

Wall-following Robot Navigation, Artificial Neural Network, Robot Control Systems

1. INTRODUCTION

The days of robots existing exclusively in the assembly line and in fiction are over; they are moving into our homes and workplaces and starting to become a part of everyday life. However, in order for them to be productive members of society, robots still need researchers and engineers to give them many of the skills we take for granted, such as object recognition and the ability to navigate closed spaces [1].

Recent advances robotics technologies have already made enormous contributions in many industrial areas. There are numerous robotic applications found in a variety of areas in our society such as surveillance systems, quality control systems, AGVs (autonomous guided vehicles), and cleaning machines [2, 3]. Robots are now expected to become the next generation of rehabilitation assistants for elderly and disabled people such as intelligent wheelchairs. By integrating intelligence into a powered wheelchair, a robotic wheelchair has the ability to safely transport a user to their desired destination. In order to achieve successful navigation in a narrow hallway, a robot must exhibit fundamental abilities such as recognizing a corridor and detecting and avoiding collisions. The corridor navigation agent, for example, processes a captured image and identifies a corridor using machine vision techniques, and the collision detection agent avoids walls and obstacles by using neural network to interpret the sensor inputs. Because incremental learning is vital in robots that need to adapt rapidly and respond appropriately to new or unexpected events that occur in its vicinity.

Sundarapandian et al. (Eds) : ICAITA, SAI, SEAS, CDKP, CMCA-2013
pp. 23–34, 2013. © CS & IT-CSCP 2013 DOI : 10.5121/csit.2013.3803

Where robots are required to act as assistants in domestic environments or in security applications when dealing with the general public, an important part of an intelligent engagement is the ability to learn during the interaction. Moreover, without this ability, the robot can use only previously-acquired knowledge and will make decisions based only on a subset of the information potentially available. To take full part in communication involving humans or other robots, a robot needs the twin capabilities of learning in real time (at a sufficient rate that the knowledge can be subsumed during the interaction) and incrementally (be used to augment the existing stored knowledge) [4]. Due to recent developments within the robotics industry, building a robot has become much more effortless with the aid of commercially available robot kits. Using these kits allows us to possibly reuse the robot and the robot control program since the kit is usually provided with useful development tools [5].

Here, an important topic is the optimization of an initial neural network by learning. Experience with neural networks shows that a fast development of controllers is possible. Therefore different neural network models have been developed for the optimization of controllers in recent years (e.g. [6.]). Systems based on soft computing methods are needed for the realization of high-MIQ (Machine Intelligence Quotient) products, i.e. machines that can mimic the ability of human mind to operate on uncertain and imprecise data. These techniques will also continue to be the key methods for the realization of interactive intelligent systems, because intelligent interaction with the environment and with human requires a flexible, adaptive machine behavior and the ability to handle with imprecise and uncertain measurements or further system inputs [7].

Fundamentally, the behaviour of a robot is a result of the interaction of three factors: the robot's hardware, the robot's controller, and the environment the robot is operating in. The robot acquires information from the environment through its sensors, which provides the input signals to the controller. The controller computes the desired motor commands and the robot performs these commands in the environment to achieve the desired task [8]. Given that sensing and the actions of a robot are coupled dynamically, given the sensitivity of robot sensor's to slight changes in the environment, the robot environment interaction exhibits complex, non linear, often chaotic and usually unpredictable characteristics [9,10]. Because of this, the task of robot programming, designing a control program to achieve a desired behaviour is difficult. Unlike other engineering disciplines, there is no formal, theory-based design methodology which the robot programmer can follow to program a robot to achieve a desired task. Nevertheless, we have previously shown that the robot programming process can be automated: sensor-motor competences in mobile robotics applications can be modelled automatically and algorithmically, using robot training and system identification methods [11].

Many contemporary problems concerning mobile robots can be solved using different learning algorithms. The most common learning technique in robot systems is reinforcement learning. It allows generating robot’s policy (mapping between a world state and actions) using feedback from the environment. Real systems are often too complex for traditional learning techniques. Complexity problem may be solved by learning distribution. Also using supervised learning instead of reinforcement learning can give better results. This leads us to the need of machine learning framework for mobile robot control. The integration of a learning algorithm with a mobile robot controller is not a straightforward task. Typically robot on-board computer has limited resources and computational power, which reduces learning capabilities [12].

The PNN structures provide a general solution to pattern classification problems by following an approach developed in statistics, called Bayesian classifiers. The PNN uses a supervised training set to develop distribution functions within a pattern layer. Training of the PNN is much simpler than other ANNs structures. However, the pattern layer can be quite huge if the distinction between categories is varied and at the same time quite similar in special areas [13]. Because of the PNN provides a general solution to pattern classification problems, it is suitable for the disease diagnosis systems [14-17].

Some advantages of the PNN are

1. Very fast learning and recalling process;
2. No iteration for weight regulations in learning process;
3. No predecision for the number of hidden layers and the number of hidden nodes in each layer. With the predetermined;
4. Training samples, the number of hidden nodes could be effectively determined.
5. Limit number of samples for training;
6. Adaptability for architectural changes.

This study offers an alternative approach that robot navigation tasks may be useful for a prospect of early finding target. In this study, probabilistic neural network (PNN) structure was used for robot navigation tasks. The PNN result was compared with the results of the Logistic Perceptron, Multilayer Perceptron, Mixture of Experts and Elman neural networks and the results of the previous studies reported focusing on robot navigation tasks and using same dataset.

2. MATERIAL AND METHOD

2.1. Data Source

The UCI machine learning repository contains many real-world data sets that have been used by a variety of investigators. We have performed several experiments in order to test the Machine Learning Library's capabilities. We used datasets from a popular datasets repository - UCI Machine Learning Repository [18]. Below we present learning results for Wall-Following Robot Navigation Data Set. The task associated with the dataset is a classification. The goal is to learn a robot's move that should be performed in order to avoid collisions with walls. The data set collected as the SCITOS G5 robot's measurements of 24 ultrasound sensors arranged circularly around its 'waist' that navigates through the room following the wall in a clockwise direction.
This robot is built for indoors use, moving by means of a differential motor system, carrying its 60kg platform at speeds up to 1.4m/s, being able to rotate 360 degrees and to push loads up to 50kg. The robot has 570mm × 735mm × 610mm in height, length and width, respectively. With a battery autonomy of 12h. Its computer hardware consists of a motherboard, Mini ITX, Intel Core Duo Processor 1.6 or 2.0GHz, 2GB of RAM, 120GB of HD, with PS/2 input for keyboard and mouse, 5 USB inputs, 3 Firewire inputs, TV-out, RS232, VGA, SPDIF output, LVDS, and input to RJ45 Ethernet 10/100 and wireless network on-board, IEEE 802.11a/b/g standard. It uses Linux (Fedora 9) equipped with a collection of libraries C++ API Robot, developed by MetraLabs to enable creation of software for control and navigation of the SCITOS G5 platform. The robot has encoders to calculate the position with 460 ticks by rotation of the wheel, a collision sensor, a belt of 24 ultrasound sensors, with a range within 20cm − 300cm, a laser sensor SICK S300 and a robotic head with an omni-directional camera [19].

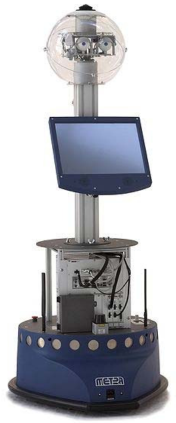

Figure 1. Robot SCITOS G5 [19].

The Wall-Following Robot Navigation has a very small number of classes, which correspond to four different directions for the robot (Move-Forward, Slight-Right-Turn, Sharp-Right-Turn, Slight-Left-Turn) data set that has 5456 instances with 24 continuous predictor variables and 1 class variable. The predictor variables describe US1: ultrasound sensor at the front of the robot (reference angle: 180°), US2: ultrasound reading (reference angle: 165°), US3: ultrasound reading (reference angle: -150°), US4: ultrasound reading (reference angle: -135°), US5: ultrasound reading (reference angle: -120°), US6: ultrasound reading (reference angle: -105°), US7: ultrasound reading (reference angle: -90°), US8: ultrasound reading (reference angle: -75°), US9: ultrasound reading (reference angle: -60°), US10: ultrasound reading (reference angle: -45°), US11: ultrasound reading (reference angle: -30°), US12: ultrasound reading (reference angle: -15°), US13: reading of ultrasound sensor situated at the back of the robot (reference angle: 0°), US14: ultrasound reading (reference angle: 15°), US15: ultrasound reading (reference angle: 30°), US16: ultrasound reading (reference angle: 45°), US17: ultrasound reading (reference angle: 60°), US18: ultrasound reading (reference angle: 75°), US19: ultrasound reading (reference angle: 90°), US20: ultrasound reading (reference angle: 105°), US21: ultrasound reading (reference angle: 120°), US22: ultrasound reading (reference angle: 135°), US23: ultrasound reading (reference angle: 150°), US24: ultrasound reading (reference angle: 165°) [20].The whole dataset consists of about 5456 labelled examples. The actual available moves and class distributions are presented in Table 1.

Table 1. Characteristics of Wall-Following Robot Navigation Data Set

Move	Samples in dataset	Class for NN
Move-Forward	2205 samples (40.41 %)	1
Slight-Right-Turn	826 samples (15.13 %)	2
Sharp-Right-Turn	2097 samples (38.43 %)	3
Slight-Left-Turn	328 samples (6.01 %)	4

The provided files comprise three different data sets. The first one contains the raw values of the measurements of all 24 ultrasound sensors and the corresponding class label Sensor readings are sampled at a rate of 9 samples per second. The second one contains four sensor readings named 'simplified distances' and the corresponding class label. These simplified distances are referred to as the 'front distance', 'left distance', 'right distance' and 'back distance'. They consist, respectively, of the minimum sensor readings among those within 60 degree arcs located at the front, left, right and back parts of the robot. The third one contains only the front and left simplified distances and the corresponding class label. Files with different number of sensor readings were built in order to evaluate the performance of the classifiers with respect to the number of inputs [21]. Besides the navigation decision-making, the thread is also responsible for sending, through the wireless network, the following information to the database located at the support computer: 24 ultrasound readings, the (x, y) coordinates and the angular position, the translational and rotational speed of the robot at that time, the label number of the action that was performed according to the settings of the environment and the four simplified distances. The arrangement of objects located in the test room can be seen in Figure 2 [19].

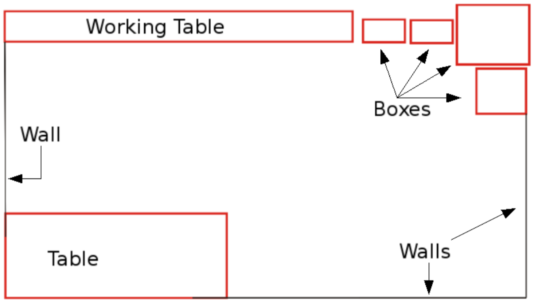

Figure 2. Sketch of the robot's navigation environment [19].

2.2. Navigation

Navigation is one of the most important problems in designing and developing intelligent mobile robots. Staying operational, i.e. avoiding dangerous situations such as collisions and staying within safe operating conditions (temperature, radiation, exposure to weather, etc.) comes first. But if any tasks are to be performed that relate to specific places in the robot environment, navigation is a must. Robot navigation is defined by the ability of a mobile robot to determine its own position in its frame of reference and then to plan a path towards some goal location. If the environment is unknown to the robot, then the path planning stage has no sense. In this case, the navigation strategy is purely reactive. The inputs to the mobile robot navigator are the target position and the sensor system data. If there are no obstacles between the robot and its target, the navigation path is just a straight line between them. If an obstacle is detected, some avoidance strategy is required. Potential function based methods, neural networks, and fuzzy logic based controllers, trained with a heuristic database of rules, are among the possibilities following. Figure 6 shows the position of motion control (for obstacle avoidance) and exploration (navigation) compared to other mobile robot research areas [22].

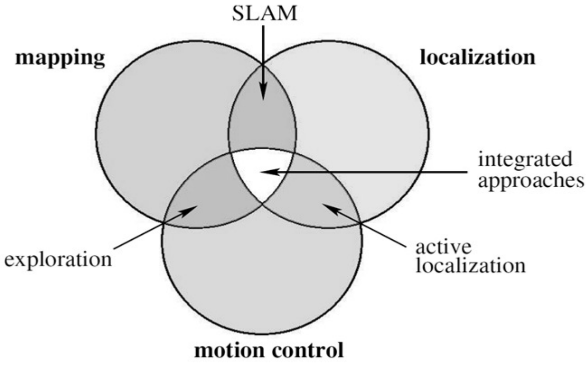

Figure 3: Relationships between mobile robot research areas [23].

2.3. Wall-Following

As mentioned before, in order to move around large objects or navigate from one room to another, a wall following method is required. In some cases, such as when the robot encounters U-shaped objects or should navigate to another room to reach its target, the mobile robot needs to drive in the opposite direction of the target. This is possible if the direction of the goal is not considered in the wall following method. Therefore, in this module the target's direction does not affect the decision made by the neural networks. Upon reaching a wall or a wide obstacle a left or right direction is selected for navigating around the wall. Until the object avoidance action is not triggered again, the direction of wall-following will be maintained. For example, if the robot is following the wall and keeping right (the walls will always be on the right side of the robot), then it will never change the wall-following direction unless the object-avoidance action is required. In other words, the direction to follow the wall is decided only at the time of changing from object-avoidance to wall-following [22].

2.4. Sensors

The problem of mobile robot navigation, includes three fundamental matters; map building, localization and path planning. This problem refers to planning a path to a specified target, executing this plan based on sensor readings, and is the key to the robot performing some particular tasks. Artificial Neural networks are increasingly being used in various fields of machine learning, including pattern recognition, speech production and recognition, signal processing, medicine, and business. In the recent years, artificial neural networks, including feedforward neural network, self organizing neural network, principal component analysis (PCA), dynamic neural network, support vector machines (SVM), neuro-fuzzy approach, etc., have been extensively employed in the field of mobile robot navigation because of their properties such as nonlinear mapping, ability to learn from examples, good generalization performance, massively parallel processing, and ability to approximate an arbitrary function given sufficient number of neurons. For a mobile robot to identify where it is or how it got there, or to be able to reason about where it has gone, sensors are necessary. For measuring the distance that wheels have traveled along the ground and for measuring inertial changes and external structure in the environment, the sensors can be flexible and mobile. The may be roughly divided into two classes: internal state sensors, such as accelerometers, gyroscopes, and external state sensors, such as laser sensors, infrared sensors, sonar, and visual sensors. The internal state sensors such as accelerometers and gyroscopes provide the internal information about the robot's movements. The external state sensors, such as laser, infrared sensors, sonar, and visual sensors, provide the external information about the environment [22].

2.5. Artificial Neural Network

The probabilistic neural network (PNN) developed by Donald Specht [13] is a network formulation of 'probability density estimation (PDF)'. It is a model based on competitive learning with a 'winner takes all attitude' and the core concept based on multivariate probability. The PNN provides a general solution to pattern classification problems by following an approach developed in statistics, called Bayesian classifiers. The network paradigm also uses Parzen Estimators which were developed to construct the PDF required by Bayes theory. The architecture of probabilistic neural network is shown as Figure 4.

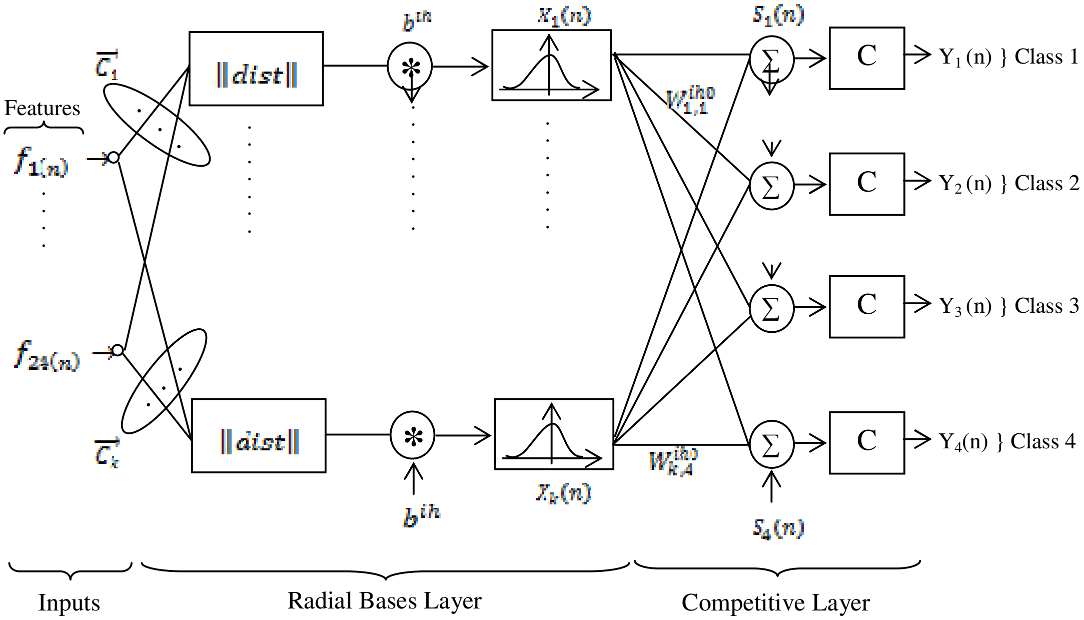

Figure 4. Architecture of PNN

The PNN structure used in this study consists of the Input Layer, Pattern Layer, Summation Layer and the Output Layer [15-17]. The input layer does not perform any computation and simply distributes the input values to all of the pattern units. All pattern layer neurons simultaneously receive the 24-dimensional real valued input vector. The pattern layer consists of a set of radial basis functions. All of the radial basis functions are the same type (Gaussian).

According to the pattern *f* from input layer, the neuron f_{ij} of the pattern layer computes its output

$$\phi_{ij}(f) = \frac{1}{(2\pi)^{d/2}\sigma^d}\exp\left[-\frac{(f-f_{ij})^T(f-f_{ij})}{2\sigma^2}\right] \qquad (1)$$

where d denotes the dimension of the pattern vector f, f_{ij} is the neuron vector and σ is the smoothing parameter controlling the sensitivity of the Gaussian function. The summation layer neurons sum the inputs from the pattern layer

$$p_{ij}(f) = \frac{1}{(2\pi)^{d/2}\sigma^d}\frac{1}{m}\sum_{j=1}^{m}\exp\left[-\frac{(f-f_{ij})^T(f-f_{ij})}{2\sigma^2}\right] \quad (2)$$

where m denotes the total number of samples. The formulation of the output units of the PNN is a model based on a competitive algorithm with a ‘winner takes all’ stand [24]. The output layer has two-input neurons and produces binary outputs as shown in Figure 5.

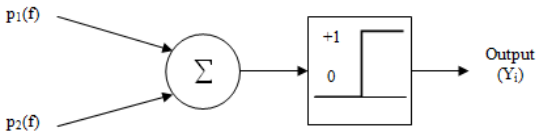

Figure 5. Architecture of output layer

Probabilistic neural network is a kind of radial basis network suitable for classification problems. PNN uses the supervised learning where the data is divided into two parts, the training and the testing part. The performance of PNN is related to two procedures: training procedure and recall procedure [24].

3. PREVIOUS STUDY

These datasets used are publicly available sensor readings from the University of California Irvine (UCI) machine learning repository, which most researchers use to validate their techniques. In practice, the major contributing factors that affect accuracy of robot behaviour to obstacle avoidance are the model learning process, number of sensors considered and speed of reasoning based on the sensor configurations. At first data set donated to the UCI repository was created by the researchers at the Federal University of Ceará in Portugal did just that when they investigated using several learning systems to help their robot navigate a room without colliding with walls or objects [19]. Among the tools they tried were logistic perceptrons and multilayer perceptron networks. They found that, unless they used short-term memory mechanisms, the logistic perceptron was unable to command the robot around the room without short-term memory functions. They concluded from these results that this classification task is not linearly separable.
Ref [1] was compared performance of tools not just in terms of classification rate but also in terms of ease of use and speed of computation. She first studied the multilayer perceptron and she produced with this tool an average classification rate of 80,2% over all runs and sets. So next she tried k-nearest neighbour classifier. At the end of each run she was rewarded with a classification rate that hovered around 32%. Then she experimented with the maximum likelihood classifier. Its average classification rate was found to be 89,3% which tops the MLP's rate by almost 10%.
Ref [12] used the dataset as 60% of samples from each class' category were included in a training set, and the other 40% formed a test set. C4.5 algorithm (Naïve Bayes and a Data Mining algorithm: Apriori) was used in a supervised learning process. The classifier performed, assigning the correct category to 97.5% of samples from the test set. It should be noted that the overall execution time was high, because in a tested version all the computations were performed in Erlang.

In Ref [21], the performance of the different classifier methods like Bagging, Dagging, Decorate, MultiClassClassifier, and MultiboostAB were compared. In this Bagging was best algorithm to finding the accuracy. In these Experiment the same robot navigation datasets were used. Classification Accuracy and Time was Calculated by 10-fold validation method.

Ref [25] studied with the same data set for the MaxDistance strategy which performs best in the first few iterations. They think that they introduced a new framework for the classification with labels.

In Ref [20], they selected five applicable normalization methods and then they normalized selected data sets afterward they calculated the accuracy of classification algorithm before and after normalization. In study the SVM algorithm was used in classification. Eventually Data Envelopment Analysis (DEA) was used for ranking normalization methods. They have used four data sets in order to rank the normalization methods due to increase the accuracy then using DEA and AP-model outrank these methods. They had maximum 77,52% success rate with vector normalization method for wall following robot navigation data set.

Ref [26] investigated the performance of the proposed method using several UCI data sets [18] and compare the obtained results with other well-known feature extraction methods: PCA (principal component analysis - extracts features by maximizing the variance of data), LDA (Linear discrimination analysis), SDA (subclass discriminant analysis) and the (mutual information) MI-based feature extraction method proposed in [27]. The MI-based method is also known as MRMI-SIG. In study, a novel DA method for feature extraction based on MI was proposed. This method was called MIDA. The goal of MIDA was to create new features by transforming the original features such that the transformation simultaneously maximizes the MI between the transformed features and the class labels and minimizes redundancy. In contrast to other DA (Discriminant Analysis) algorithms that are based on second order statistics, the proposed method was based on information theory that was able to compare the nonlinear relationships between random variables (i.e., between a vector of features and the class label). The proposed method was evaluated using six data sets from UCI databases. The third data set they used was the wall-following robot navigation data set that consists of 24 features. Its objective was to test the hypothesis that a seemingly simple navigation task is actually a nonlinearly separable classification task. The experiments have shown that the MIDA method performs comparably to existing methods. MIDA's performance was frequently better than other methods.

Ref [28] conducted three main experiments to compare the performance achieved by the different models on the four ultrasound sensors configuration in terms of; (a) collision avoidance efficiency in static environments, (b) collision avoidance efficiency in dynamic environments and (c) comparing average performance evaluations of the models. Using three-fold cross validation, the full dataset was divided into three partitions as randomly. Some samples from one of the partitions were selected for testing and the others were used for training. Generally, while the training dataset is approximately 95% of all data, the rest of the 5% data are used as testing samples. Since the cross-validation technique was used, this process was repeated three times. They applied three-fold cross validation, where the test set were separated from the training instances and appeared as new obstacle readings to the models. In all the folds, the ERB and the results of the PRB revealed that autonomous collision avoidance using the BN (Bayesian Network) and the k-NN were also promised in a dynamic environment. BN model tremendously indicated average accuracy of 93.3% better believed for the robot's behaviour than the 73.3% of k-NN.

4. CONCLUSIONS

These studies have applied a different method to the wall-following mobile robot navigation problem using same dataset. According to result, it is seen that the most suitable neural network structure is PNN structure for classifying wall-following mobile robot data.

In this study, the best result for the average classification accuracy was obtained using PNN structure by %99,635 as seen in the Table 2.

Table 2. Comparison of the results for studies.

Name	**Method**	**Rate (%)**
SAME DATA and NEURAL NETWORK STRUCTURES		
Ref [1]	Maximum Likelihood Classifier	89.30
SAME DATA and DIFFERENT STRUCTURES		
Ref [20]	Data Envelopment Analysis (DEA)	77.52
Ref [26]	Mutual Information Discriminant Analysis (MIDA)	92.1
Ref [28]	Bayesian Network	93.30
Ref [12]	Naïve Bayes and a Data Mining algorithm: Apriori	97.50
Ref [21]	Bagging	99.28
SAME DATA and NEURAL NETWORK (PNN) STRUCTURES		
Our Study	Probabilistic Neural Network	99.635

It is found that the neural network structures generally show good performances for wall-following mobile robot navigation problem. This classification accuracy is highly reliable for such a problem because only a few samples were misclassified by the system. Finally, ANN structures can be helpful as learning based decision support system to contribute to the engineers in their projects.

In future we may conduct the same experiments with different datasets instead of multiple dataset, Multiclass and combine few ensemblers with the different base classifier to study how the ensemblers combined with the base classifiers boost the performance accuracy could.

REFERENCES

[1] DiVall M. (2010) “Neural Network Learning of Robot Navigation Tasks”,*ECE/ME/CS 53,* Final Report.

[2] Trahanias, P.E. and et al. 1997. “Navigational support for robotic wheelchair platforms: an approach that combines vision and range sensors.” *Proceedings of the 1997 IEEE International Conference on Robotics and Automation, Albuquerque, NM,* 12, pp. 65-70.

[3] Wijesoma, W. S., Khaw, P. P., and Teoh, E. K. 2001. “Sensor modeling and fusion for fuzzy navigation of an AGV.” *International Journal of Robotics and Automation* 16(1): pp. 14-25.

[4] Swere E. , Mulvaney D. and Sillitoe I. (2006). “A Fast Memory-Efficient Incremental Decision Tree Algorithm in its Application to Mobile Robot Navigation” , *International Conference on Intelligent Robots and Systems*, *pp. 645-650*, Beijing, China.

[5] Y. Ono, H. Uchiyama, W. Potter (2004). “A Mobile Robot For Corridor Navigation: A Multi-Agent Approach”, *ACM-SE 42 Proceedings of the 42nd annual Southeast regional conference*, pp. 379-384.

[6] H.R. Berenji and P. Khedkar, (1992) “Learning and tuning fuzzy logic controllers through reinforcements”, *IEEE Trans. Neural Networks,* 3 724-740.

[7] Nadine Tschichold-Gürrman, “The neural network model RuleNet and its application to mobile robot navigation.” *Institute of Robotics, ETH Zürich, CH 8092 Zürich, Switzerland Fuzzy Sets and Systems* 85 (1997) 287-303

[8] U. Nehmzow, (2003), “Mobile Robotics: A Practical Introduction”, *2nd ed., Springer Verlag*, 2003.

[9] U. Nehmzow, (2003) “Quantitative analysis of robot-environment interaction-Towards ``scientific mobile robotics''”, *International Journal of Robotics and Autonomous Systems* 44: pp. 55-68.

[10] U. Nehmzow, K. Walker, (2003) “The behaviour of mobile robot is chaotic”, *AISB Journal* 1(4) pp.373-388.

[11] U. Nehmzowa, O. Akanyeti, S.A. Billings, (2010) “Towards modelling complex robot training tasks through system identification”, *Robotics and Autonomous Systems* 58: pp. 265-275

[12] L. Beben, B. Sniezynski, W. Turek and K. Cetnarowicz , (2012) “Architecture of an Erlang-Based Learning System for Mobile Robot Control” , *Proceedings of the 5th International Workshop on Evolutionary and Reinforcement Learning for Autonomous Robot Systems”*, pp.45-48.

[13] Specht D.F. (1990), “Probabilistic neural networks”, *Neural Networks* 3, 109–118.

[14] Er O, Yumusak N, Temurtas F. (2010) “Chest diseases diagnosis using artificial neural networks”, *Expert Systems with Applications*, 37 (12) 7648-7655.

[15] Er O, Sertkaya C, Temurtas F and Tanrikulu AC. (2009), “A Comparative Study on Chronic Obstructive Pulmonary and Pneumonia Diseases Diagnosis using Neural Networks and Artificial Immune System”, *Journal of Medical Systems*, 33 (6), 485-492.

[16] Er O, Temurtas F and Tanrikulu AC. (2010), “Tuberculosis Disease Diagnosis Using Artificial Neural Networks”, *Journal of Medical Systems*, 34 (3), 299-302.

[17] Temurtas F., (2009) “A comparative study on thyroid disease diagnosis using neural networks”, *Expert Systems with Applications*, 36, 944-949.

[18] UCI machine learning repository (http://archive.ics.uci.edu/ml/), 2013.

[19] Freire, A.L; Barreto, G.A. ; Veloso, M. ; Varela, A.T. (2009). “Short-term memory mechanisms in neural network learning of robot navigation tasks: A case study”, *6th Latin American Robotics Symposium (LARS), pp. 1-6.*

[20] M. Eftekhary, P. Gholami, S. Safari and M. Shojaee, (2012), “Ranking Normalization Methods for Improving the Accuracy of SVM Algorithm by DEA Method”, *Modern Applied Science Published by Canadian Center of Science and Education*; Vol. 6, No. 10, pp. 26-36.

[21] P.Kalaiselvi, C.Nalini, (2013), “A Comparative Study of Meta Classifier Algorithms on Multiple Datasets”, *International Journal of Advanced Research in Computer Science and Software Engineering*, Volume 3, Issue 3,pp. 654-659.

[22] Dezfoulian, Seyyed Hamid, (2012) "A Generalized Neural Network Approach to Mobile Robot Navigation and Obstacle Avoidance" pp.102.

[23] C. Stachniss, D. Hahnel, and W. Burgard, (2004). “Exploration with active loop-closing for FastSLAM,” *Intelligent Robots and Systems,(IROS 2004). Proceedings IEEE/RSJ International Conference*, pp. 1505-1510.

[24] Delen, D., Walker, G., Kadam, A., (2005) Predicting breast cancer survivability: A comparison of three data mining methods, *Artificial Intelligence in Medicine*, 34(2), 113–127.

[25] Nicolas Cebron · Fabian Richter · Rainer Lienhart, ““I can tell you what it’s not”: active learning from counterexamples”, *Prog Artif Intell* (2012), vol.1, pp:291–301.

[26] Shadvar A., (2012), “Dimension Reduction By Mutual Information Discriminant Analysis”, *International Journal of Artificial Intelligence & Applications (IJAIA*), Vol.3, No.3, pp. 23-35.

[27] K. E. Hild II, D. Erdogmus, K. Torkkola, J. C. Principe, (2006) “Feature extraction using information theoretic learning,” *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. 28, no. 9, pp. 1385-1392.

[28] Isaac O. Osunmakinde, ChikaO. Yinka-Banjo and Antoine Bagula, (2012), “Investigating the Use of Bayesian Network and k-NN Models to Develop Behaviours for Autonomous Robots”, *Mobile Intelligent Autonomous Systems*, No:34, pp. 751-764.

Authors

Mücella Özbay Karakuş was born in Elazig (1984). She is a Lecturer in the Department of Computer Engineering at the University of Bozok and she has been a faculty member since 2008. She completed her MSc and her undergraduate studies at the Fırat University. She is making her PhD thesis at the Bozok University. Her research interests lie in the area of Artificial Neural Network, Image processing, Intelligent robot technology and Mechatronics Engineering . She has collaborated actively with researchers in other disciplines of computer science and robotics.

Orhan ER was born in Mardin (1978). He received the B.Sc., M.Sc., and Ph.D. degrees from the Sakarya University of Turkey, in 2001, 2004, and 2009, respectively, B.Sc., M.Sc. degrees in computer engineering and Ph.D. degree in electronics engineering. Currently, he is an assistant Professor at the Faculty of Engineering and Architecture. His research interests are performance modeling and optimization of artificial neural networks and artificial immune systems.