34 Investigating the Use of Bayesian Network and *k*-NN Models to Develop Behaviours for Autonomous Robots

Isaac O. Osunmakinde, Chika O. Yinka-Banjo and Antoine Bagula

CONTENTS

34.1 INTRODUCTION

Bayesian network technology is very useful for encoding probabilistic knowledge as graphical structures. It is rapidly gaining popularity in modern artificial intelligence (AI) for solving real-life problems involving reasoning under uncertainty [1,2]. The most important benefit of using Bayesian networks in real-life applications is in carrying out probabilistic inference (or reasoning). Bayesian inference is a type of statistical inference in which probabilities are interpreted as degrees of belief and its fundamental computation is derived from Bayes' theorem [3]. The Network belief technology has been successfully used for reasoning in the areas of power transformer diagnosis [1], medical diagnoses [4], telecommunication networks [5] and so on. Knowledge is expensive to acquire and most of the time, there are no domain experts or knowledge engineers to interpret environments and model knowledge as Bayesian belief networks. Since data are cheap and contain useful information about the environments, Bayesian networks offer a great advantage that can capture and encode this hidden information as knowledge. *k*-Nearest neighbour (*k*-NN) is a nonparametric instance-based learning as it allows a hypothesis of model complexity to grow with data sizes. *k*-NN is based on minimum distance from a query instance to all training samples to determine the *k*-NN, which spans the entire input texture space. Prediction of the query instance is taken as majority votes of the *k*-NN. The *k*-NN model has been successfully used for prediction or reasoning in the areas of face recognition [2], traffic accident prediction [6] and fault detection [7].

The recent literature addresses robot slope-walking problem [8], using the machine learning technique of *k*-NN and robot localization problems [9,10] with wide applicability of the Bayesian theorem. There is not enough focus on the autonomous robot behaviour problem using the predictive power of *k*-NN and Bayesian network models as learning and reasoning techniques to manage autonomous robot navigation with respect to collision avoidance. The reasons for not using the models in this area could be as a result of their challenges, such as determination of appropriate *k*th value in *k*-NN and the computational intensity of Bayesian learning, in autonomous applications. This is a motivation for raising the research questions: (i) To what extent can a robot autonomously manage its behaviours when navigating without collision in a static environment where it was trained using teleoperation? and (ii) To what extent can a robot autonomously manage its behaviours when navigating without collision in a dynamic or different environment from where it was trained using teleoperation? An environment is said to be static when obstacles do not move from their positions while it is dynamic when obstacles are moving. To address these challenges and the questions, we propose an approach of training a robot to avoid obstacles through teleoperation and thereafter use the knowledge acquired to develop behaviours for autonomously navigating in various environmental sensing conditions using the learning and predictive capabilities of *k*-NN and Bayesian network models. The chosen behaviour or navigational direction of the robot determines the control command values of translational and rotational velocities the robot uses for navigation. This work integrates a *k*th set measure to *k*-NN for determining an appropriate *k*th value where robot behaviour to a new sensors reading is predicted based on majority voting. Using real-life publicly available ultrasound sensors minimum readings to obstacles on a number of comparative evaluations in static and dynamic environments, our experimental results show that both predictive learning paradigms developed as collision avoidance models (CAM) are capable of dealing with uncertainties during autonomous robot navigation. This excellent performance suggests a wider application of the behavioural models which learn tasks and command robot successfully without collisions in an unknown environment in industry.

The rest of this chapter is arranged as follows: Section 34.2 presents useful theories, which includes Bayesian network and *k*-NN models; Section 34.3 presents the proposed modelling for behavioural and collision avoidance for robots which includes perception of sensor data, learning and reasoning processes of the approaches; Section 34.4 critically presents experimental evaluations of the approaches on number of comparative evaluations in static and dynamic environments using publicly available minimum ultrasound sensors readings to obstacles. The average performance evaluation of the models is also compared based on a configuration of four sensors and related work is presented in Section 34.5. Section 34.6 concludes the chapter.

34.2 SOME USEFUL CONCEPTS AND MODELS

In this section some useful concepts and theories are discussed: (a) Bayesian networks modelling concepts and (b) nearest-neighbour model.

34.2.1 Bayesian Network Models

A Bayesian belief network is formally defined as a directed acyclic graph (DAG) represented as $G = \{X(G), A(G)\}$, where $X(G) = \{X_1, \ldots, X_n\}$, vertices (variables) of the graph G and $A(G) \subseteq X(G) \times X(G)$, set of arcs of G. The network requires discrete random values such that if there exists random variables $X_1, \ldots, X_n$ with each having a set of some values $x_1, \ldots, x_n$ then, their joint probability density distribution is defined as

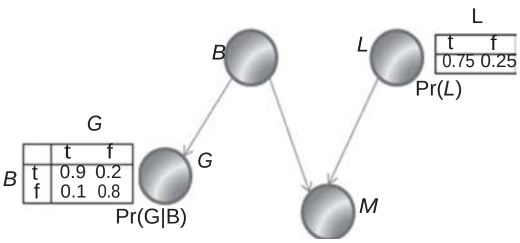

FIGURE 34.1 A simple Bayesian network model of a block lifting machine.

$$\mathrm{pr}(X_1,...,X_n) = \prod_{i=0}^{n} \mathrm{pr}(X_i \mid \pi(X_i)) \tag{34.1}$$

where $\pi(X_i)$ represents a set of probabilistic parent(s) of child X_i [3]. A parent variable otherwise refers to as *cause* has a dependency with a child variable known as *effect*. Every variable X with a combination of parent(s) values on the graph G captures probabilistic knowledge as conditional probability table (CPT). A variable without a parent encodes a marginal probability. For the purposes of illustrating BN, Figure 34.1 shows the DAG and the CPTs of a BN model as the core reasoning component of an intelligent system. In this case, it describes the operation of a block-lifting machine. The operation is monitored with the following attributes: battery (B), movement (M), liftable (L) and gauge (G) [3]. Each of the attributes contains states true (t) and false (f), with their associated probabilities captured as CPTs, such as L having $t = 0.75$ and $f = 0.25$. Figure 34.1 depicts conditional dependencies of the attributes which best describe the complexity of variables of the block lifting machine. For instance, in Figure 34.1, G is conditionally dependent on B and it is computed as $\mathrm{Pr}(G|B)$. Also, M is conditionally independent of G which implies that it is computed as $\mathrm{Pr}(M|B, L)$. The estimation of the probabilities, using the maximum likelihood estimate (MLE) algorithms, captured as CPTs results from the environment, for example, obstacle distances perceived by the robot sensors. A Bayesian network can be modelled by eliciting the probabilistic knowledge from domain experts, if the environment is small. For a more complex domain like robot environment, the most suitable BN is learned from the environment captured as samples using learning algorithms described in Refs. [11,12]. Having a BN model in place, a probabilistic inference is required for reasoning about any situation and the beliefs (or probabilities) of possible outcomes are propagated in a model based on the evidence of the situation. Understanding various obstacle distances is a possible situation that can be acted upon by the inference. The Bayesian inference accounts for the uncertainty capability of BNs through the Bayes' theorem shown in the following equation [13]

$$\mathrm{Pr}(X_i \mid X_j) = \frac{\mathrm{Pr}(X_j \mid X_i) \times \mathrm{Pr}(X_i)}{\mathrm{Pr}(X_j)} \tag{34.2}$$

The constituents of Equation 34.2 are: (i) $\mathrm{Pr}(X_i \mid X_j)$ is the posterior probability called the original degree of belief when the likelihood and prior are combined, (ii) $\mathrm{Pr}(X_j \mid X_i)$ is the likelihood function which is referred to as the conditional probability of what we know (evidence) based on what we do not know (query) and (iii) $\mathrm{Pr}\ (X_i)$ is the prior probability of X_i before making any observations; here, the marginal probability $\mathrm{Pr}\ (X_j)$ is a measure of the impact that observations have on the degree of beliefs.

34.2.2 *k*-nn Models

k-NN [14] is a non-parametric instance-based learning as it allows a hypothesis of model complexity to grow with data sizes. *k*-NN is based on minimum distance from a query instance to all training samples to determine the k-NN, which span the entire input space. The Euclidean distance of lower-dimensional space is commonly applied for computing the minimum distance in this step. The Euclidean distance for two-dimensional space, say points $x = (x_1, x_2)$ and $y = (y_1, y_2)$, is given as

$$\mathrm{d}(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2} \tag{34.3}$$

Prediction of the query instance is taken as majority votes of the k-NNs. The idea is that any point x is likely to be similar to those points in the neighbourhood of x. The choice of parameter value k is critical but k-NN is advantageously robust to uncertainty or noisy training samples. This is the simplest k-NN, but more sophisticated versions can be proposed.

34.3 BEHAVIOURAL AND CAM FOR ROBOTS

Figure 34.2 illustrates an unstructured indoor environment with scattered chairs and tables as obstacles where behaviour could be developed to support the navigation of a pioneer robot. We illustrate an experimental set-up with wall-following navigation task [15] of a robot, which uses 24 ultrasound sensors arranged circularly around its waist. The numbering of the ultrasound sensors starts at the front of the robot and increases in clockwise direction. Sensor readings are sampled at a rate of 9 samples/s and the data samples were collected at the same time step, as the robot navigates through a room following the wall in a clockwise direction, for 4 rounds. Ultrasound sensors send out an ultrasonic pulse and then wait for a response [16]. When the pulse leaves the device, it travels through the air until it collides with an object or obstacles, at which point an echo is reflected back. This echo is then sensed by the ultrasonic sensor. The sent pulse is anywhere from 40–200 kHz, but is typically in the 40–50 kHz range.

34.3.1 Perception of Ultrasound Sensor Data in Real Life

From the wall-following navigation task with the mobile robot [15], different data samples were captured from the environment based on three sensor configurations. The first configuration captures

FIGURE 34.2 Experimental in-door environment developing behaviours for a pioneer robot for avoiding obstacles of chairs, tables and wall. (Authors' Lab and Work environment.)

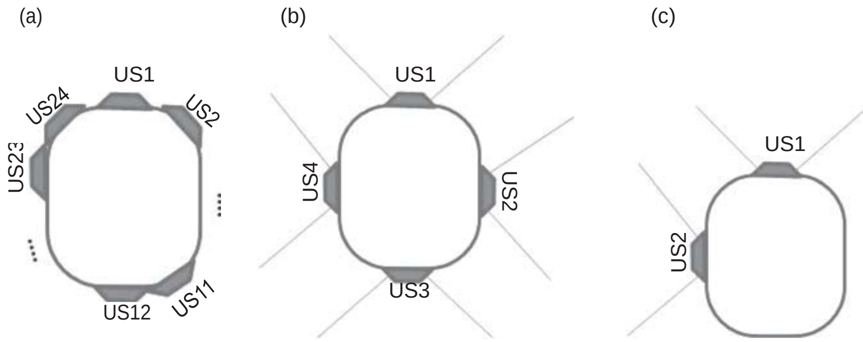

FIGURE 34.3 Three configurations of ultrasound sensors arranged circularly around a robot. (a) 24 Sensors. (b) 4 Sensors. (c) 2 Sensors.

the raw values of the measurements of all 24 ultrasound sensors, which are precisely tagged as US_1, US_2,. . ., US_{24} as shown in Figure 34.3a. This configuration consists of the minimum sensor readings among those within 15° arcs located around the robot. The second configuration captures four sensor readings named simplified distances. The simplified distances are referred to as the *front distance, left distance, right distance* and *back distance*. These distances consist, respectively, of the minimum sensor readings among those within 60° arcs located at the front, left, right and back parts of the robot as shown in Figure 33.3b. The third configuration captures only the front and left simplified distances and consists of the minimum sensor readings among those within 60° arcs located at the front and left of the robot as shown in Figure 34.3c. The robot is teleoperated for learning in the environment and behaviours were captured as it perceives obstacles. The ultrasound minimum sensor readings from the obstacles determine the behaviour or the navigational direction of the robot. The four directions defined for Figures 34.3a and b are Move-Forward, Slight-Right-Turn, Sharp-Right-Turn and Slight-Left-Turn, while two directions Move-Forward and Slight-Right-Turn are defined for the configuration in Figure 34.3c. The chosen direction of the robot determines the control command values, which are translational and rotational velocities the robot uses for navigation. Sensor readings and their associated robot actions captured from any of the three configurations are used for training the BN and *k*-NN models as they learn the environment.

34.3.2 Bayesian Learning and Reasoning Process to Robot Behaviour

Figure 34.4 illustrates the stages required to learn a BN model from an environment captured as sensor readings. As illustrated in Figure 34.4, learning such models from the environment can be decomposed as follows into sub-problems of: (a) data discretization as a pre-processing step, (b) learning a suitable network structure, (c) learning the associated conditional probability tables (CPTs) and (d) model visualization. Data discretization classifies numerical data into their corresponding interval values relative to the patterns in the data attributes. William and co-workers and Osunmakinde and Potgieter [11,12] have presented many algorithms including genetic and hill climbing algorithms to learn Bayesian networks from datasets. Its characteristics of capturing knowledge in dependency variables make it suitable for handling uncertainty problems, such as noisy sensor readings. Having a Bayesian network model in place after teleoperation of a robot, a probabilistic inference is required for reasoning about any positions of obstacles and the beliefs (or probabilities) of possible outcomes are propagated in a model based on the observations of the obstacles. In this research, we want to predict the probability Pr (or most likely) action, which is not known, based on the current knowledge of the ultrasound sensor readings *d* to obstacles that the robot understands as shown in Equation 34.4. For instance, for an obvious situation, a robot may navigate towards a freest direction, but preferential treatment is given to navigational targets.

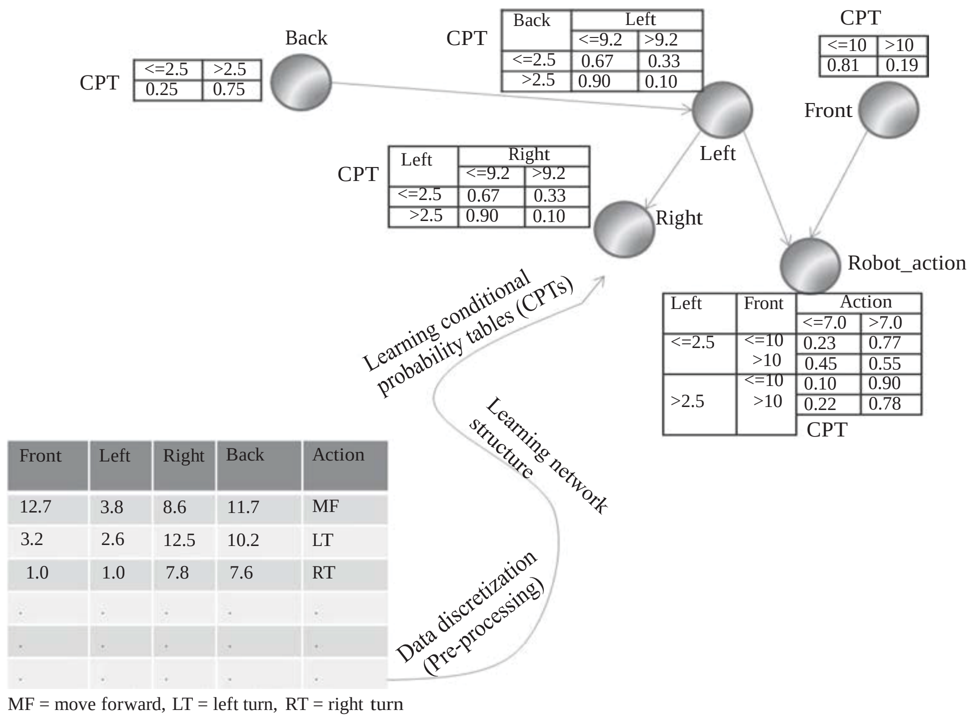

FIGURE 34.4 Learning stages of a BN where front node = US_1, right = US_2, back = US_3 and left = US_4.

$$\text{pr}\left(\textbf{Action}?\,|\,\text{US}_1 = d_1, \text{US}_2 = d_2,\ \text{US}_3 = d_3, ..., \text{US}_n = d_n\right) \tag{34.4}$$

Using Bayes' theorem in Equation 34.2, Equation 34.3 implies

$$\Rightarrow \frac{\Pr(\text{US}_1 = d_1, \text{US}_2 = d_2, \text{US}_3 = d_3, \ldots, \text{US}n = d_n \mid \text{Action}) \times \Pr(\text{Action})}{\Pr(\text{US}_1 = d_1, \text{US}_2 = d_2, \text{US}_3 = d_3, \ldots, \text{US}n = d_n)}$$

This is a Bayesian inference problem with more information in Ref. [13]. If a robot is expected to keep moving towards forward directions, then the back sensors would not participate in the reasoning process even if the sensors read the freest. The robot then negotiates among the due *forward*, *slight-right-turn* and *slight-left-turn* as the next most likely behaviour from the BN model.

34.3.3 The *k*-NN Modelling and Reasoning Process to Robot Behaviour

The adaptation of the *k*-NN model to address the robot behaviour in collision avoidance relies heavily on (i) the minimum distance from the new query instance to all training obstacle instances captured by the robot's knowledge, (ii) the choice of *k*th value and (iii) biasness check in the training instances. With regard to the minimum distance, this research proposes *n*-dimensional Euclidean measure based on the number *n* of sensors' readings that would participate in the choice of action as shown

$$\text{d}(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2 + \quad + (x_n - y_n)^2} \tag{34.5}$$

where query instance $x = (x_1, ..., x_n)$ and training instance $y = (y_1, \ldots, y_n)$.

The minimum distance eventually indicates a specific direction at which, for many purposes, the robot's action behaves as if it were concentrated on that chosen sensor. Given these numerous distances computed from all training samples, it is reasonable to choose the kth value as a number of sets s of distances and dynamically increases s with a step whenever there is a tie during the voting scheme. With regard to the biasness check, this chapter introduces an idea of making the training instances having equal number of robot actions since its prediction of new obstacle distances is based on majority votes. Now the robot autonomously navigates and perceives a new instance of sensor readings from obstacles. Without another expensive teleoperation or training, can the robot accurately predict its behaviour towards this new instance? The adaptation of the k-NN model requires the implementation of the following algorithm (k-NN approach):

```
INPUT:  Training & query sets of obstacle distances
OUTPUT: Predicted Robot Behaviours

Step 1: Specify training set
Step 2: Determine the neighbourhood size as kth set
Step 3: Compute the distance between a query instance and all training
        instances using Equation 34.5
Step 4: Determine nearest neighbours using the kth set minimum distance
Step 5: Assess the training actions/behaviours of the nearest Neighbours
Step 6: Predict robot behaviour for the query instance
Step 7: Repeat steps (3)–(6) for other query instances as perceived by the
        robot sensors
```

34.3.4 Evaluation Scheme

In this section, the performances of the approaches investigated are studied through an evaluation scheme commonly used in practice based on n-fold cross validation technique [13] as well as measuring the execution time. Cross validation sometimes called rotation estimation, is the most generally applicable strategy in model selection in machine learning since it does not rely on any probabilistic assumptions. Here the dataset is partitioned into a number n mutually disjoint folds and leave-one-out cross validation (LOO) for testing model performance while the remainder is used for training. This process is repeated n times to find the overall performance of the approaches. In this research, since learning from the environment and training of robot is carried out at the teleoperation phase, the execution speed of the model reasoning when a robot is autonomously reacting to obstacles is an important issue based on various sensor configurations.

34.4 EXPERIMENTAL EVALUATIONS OF THE CAM

One of the objectives of the investigation of the behavioural modelling approaches is to bring theory to practice with an emphasis on application to collision avoidance work. This section describes the experiments we conducted for evaluating the performances of the approaches for developing behaviours for robots using two machine learning models on three sets of real-life datasets based on sensor configurations. The models considered are BN and k-NN as described above. As described in the experimental setup of Section 34.3.2, the three datasets are captured from: (i) 24 ultrasound sensors, (ii) 4 ultrasound sensors and (iii) 2 ultrasound sensors. 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. We 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. These experiments were carried out specifically on a machine processor by implementing the *k*-NN in MATLAB and the BN in GeNile software [17]. The training samples extracted for robot to navigate towards the freest direction contains 200 samples, but going backwards is not an option since the robot is required to follow wall forward in a clockwise direction. 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.

34.4.1 Performance Accuracy of CAM in a Static Environment Using BN and *k*-NN

This section addresses our first research question presented at the introduction. Imagine a robot being allowed to autonomously avoid obstacles in an environment where it was trained-static. We therefore conducted experiments for finding the impact of the BN and the *k*-NN models on collision avoidance with the expectation of selecting a better model in terms of predicting accurate behaviour for robot when it perceives obstacles in similar positions already seen during training. It examines the consistency of the model in predicting robot behaviours. The BN model in Figure 34.5 is learned from the training sensor samples using the GeNle software and the results depicted by Table 34.1 are a summary of the average performance of the two models in terms of testing with five random samples repeated three times. The cross validation is modified differently here since the environment is static; the test samples form part of the training instances for evaluating the consistencies of the models. For each set of samples, the *Ei* indicates an instance of evidence of obstacle distances to the robot in the four sensor directions of front, left, right and back. Every instance is used to predict the robot behaviour or action using the reasoning processes of the BN in Equation 34.4 and the majority voting scheme of the *k*-NN described in Section 34.3.4. In all the sets, the expected robot behaviour (ERB) and the results of the predicted robot behaviour (PRB) revealed that autonomous collision avoidance using the BN and the *k*-NN are accurate in a static environment. One could observe that the readings of the back sensors formulates part of the learning proces, but do not participate in predicting the behaviour as the robot gives preference to its goal—follow a wall in a clockwise direction. Observe in Table 34.1 that the BN model tremendously indicates better beliefs or confidence for the robot's behaviour than the *k*-NN. For illustration, Figure 34.6 presents a pictorial representation of the belief results from the first partition. However, these results suggest that using the two models to develop behaviours for robot assist the controller in determining the translational and rotational velocities values for navigation.

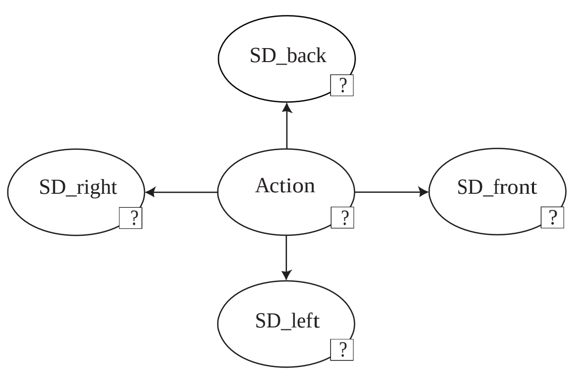

FIGURE 34.5 BN model for collision avoidance using real-life robot sensor readings.

TABLE 34.1
Robot Behaviour in a Static Environment with Real-Life Samples Using Three-Fold Cross Validation

	Percepts of Obstacle Distances				ERB = Expected Robot Behaviour; PRB = Predicted Robot Behaviour		
E_i	SD_Front	SD_Left	SD_Right	SD_Back	ERB	BN PRB (%)	*k*-NN PRB (%)
E_1	1.687	0.449	2.332	0.429	Slight-right	Slight-right (80.2)	Slight-right (78.69)
E_2	1.327	1.762	1.37	2.402	Slight-left	Slight-left (99.7)	Slight-left (90.0)
E_3	1.318	1.774	1.359	3.241	Slight-left	Slight-left (99.7)	Slight-left (90.0)
E_4	1.525	0.739	1.379	0.689	Move forward	Move forward (88)	Move forward (38.0)
E_5	0.786	0.661	2.748	0.689	Shift-right	Shift-right (89.2)	Shift-right (59.62)
Test Samples 1: Accuracy = 5/5 = 100%							*5/5 = 100%*
E_1	1.19	2.29	1.414	2.369	Slight-left	Slight-left (99.7)	Slight-left (90.0)
E_2	1.586	0.758	1.357	0.634	Front	Front (88.1)	Move forward (38.1)
E_3	0.762	0.482	1.697	0.473	Shift-right	Shift-right (85.4)	Shift-right (40.65)
E_4	1.637	0.474	1.715	0.465	Slight-right	Slight-right (94.2)	Slight-right (50.0)
E_5	0.79	0.779	1.345	0.67	Shift-right	Shift-right (96.2)	Shift-right (59.62)
Test Samples 2: Accuracy = 5/5 = 100%							*5/5 = 100%*
E_1	1.311	2.636	1.456	2.204	Slight-left	Slight-left (99.7)	Slight-left (90.0)
E_2	1.419	0.7	1.415	0.795	Move forward	Move forward (88.1)	Move forward (36.36)
E_3	0.799	0.664	2.478	1.239	Shift-right	Shift-right (99.7)	Shift-right (45.45)
E_4	1.636	0.475	1.707	0.47	Slight-right	Slight-right (94.2)	Slight-right (49.37)
E_5	2.644	0.69	2.084	0.955	Move forward	Move forward (98.7)	Move forward (63.29)
Test Samples 3: Accuracy = 5/5 = 100%							*5/5 = 100%*
Average Accuracy = 100%							

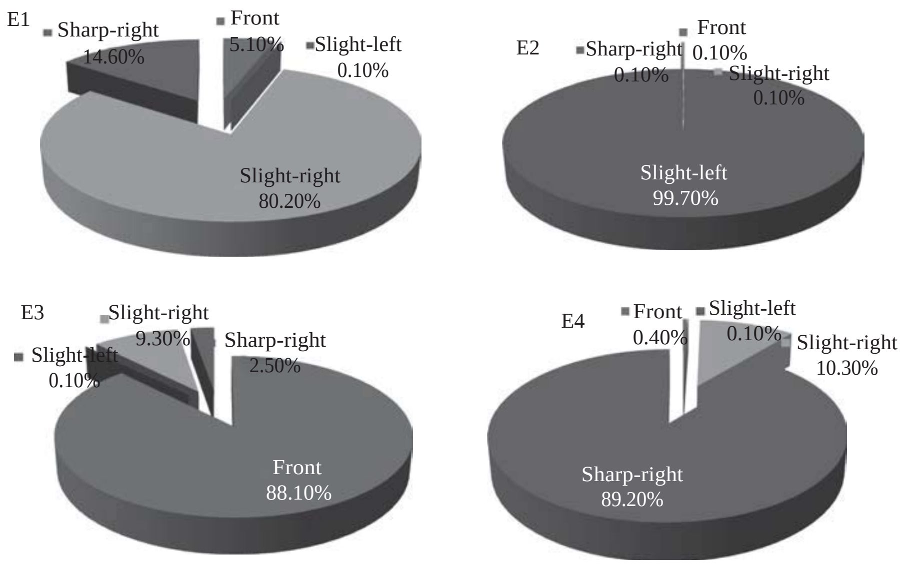

FIGURE 34.6 Presentation of the predicted behaviour using BN in a static environment from the first cross validation of Table 34.1.

34.4.2 Performance Accuracy of CAM in a Dynamic Environment Using BN and *k*-NN

This section addresses our second research question presented at the introduction. Imagine a robot being allowed to autonomously avoid obstacles in a new environment different from where it was trained-dynamic. We also conducted experiments similar to the static above to evaluate the collision avoidance capability of the two models when navigating in a new environment. The results depicted in Table 34.2 are a summary of the average performance of the models in terms of three-fold cross validation, where the test set are separated from the training instances and appear 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 and the *k*-NN are also promising in a dynamic environment. Observe in Table 34.2 that the BN model tremendously indicates average accuracy of 93.3% better beliefs for the robot's behaviour than the 73.3% of *k*-NN, probably due to the use of prior beliefs of the BN. The choice of *k*th value still needs more improvement. Figure 34.7 also presents a pictorial representation of the belief results from the first partition. However, the results suggest that using the two models to develop behaviours for robot assists the controller in determining the translational and rotational velocities values for navigation.

34.4.3 Comparing Average Performance Evaluations of the Models

From the results of evaluation in Table 34.3, we specifically access the average performance accuracies of the BN and *k*-NN collision avoidance approaches with respect to static and dynamic environments ranging from one to six cross validations. In Figure 34.8, one can see that the trend of the error on BN is lowered in the dynamic case compared to the error trend on *k*-NN. This obviously implies that a higher trend of accuracy is better for predicting behaviours for robots.

34.5 RELATED WORK

The recent literature [18–20] addresses the behaviour-based systems some of which were originally inspired by biological systems, but more work on developing behaviours for robots was recommended in Ref. [21] to assist in the control architecture. In Ref. [18], modelling a biological behaviour is studied

TABLE 34.2
Robot Behaviour in a Dynamic Environment with Real-Life Samples Using Three-Fold Cross Validation

	Percepts of Obstacle Distances						
E_i	SD_Front	SD_Left	SD_Right	SD_Back	ERB	BN PRB (%)	*k*-NN PRB (%)
E_1	2.651	0.625	1.599	0.795	Move forward	Move forward (94.1)	Move forward (66.67)
E_2	2.885	0.623	1.606	0.814	Move forward	Move forward (99.2)	Move forward (72.58)
E_3	0.894	0.649	1.071	1.085	Shift-right	Shift-right (93)	Shift-right (43.59)
E_4	1.501	0.492	1.816	1.28	Slight-right	Slight-right (79.5)	Move forward (38.1)
E_5	1.523	0.485	1.8	1.069	Slight-right	Slight-right (51.6)	Slight-right (46.99)
Test Samples 1: Accuracy = 5/5 = 100%							*4/5 = 80%*
E_1	2.581	0.613	1.619	0.852	Move forward	Move forward (99.3)	Move forward (65.78)
E_2	2.828	0.607	1.626	0.871	Move forward	Move forward (99.2)	Move forward (74.24)
E_3	0.854	0.628	1.016	1.168	Shift-right	Move forward (49.6)	Move forward (36.51)
E_4	1.511	0.49	1.82	1.27	Slight-right	Slight-right (79.5)	Shift-right (37.59)
E_5	0.784	0.487	1.797	1.156	Shift-right	Shift-right (96.5)	Shift-right (40.65)
Test Samples 2: Accuracy = 4/5 = 80%							*3/5 = 60%*
E_1	2.549	0.599	1.633	0.889	Move forward	Move forward (99.3)	Move forward (64.1)
E_2	2.544	0.597	1.639	0.908	Move forward	Move forward (99.3)	Move forward (64.1)
E_3	0.873	0.642	1.053	1.105	Shift-right	Shift-right (93)	Move forward (35.38)
E_4	1.617	0.475	1.854	1.169	Slight-right	Slight-right (51.6)	Slight-right (54.55)
E_5	0.789	0.49	1.864	1.076	Shift-right	Shift-right (79.1)	Shift-right (70.42)
Test Samples 3: Accuracy = 5/5 = 100%							*4/5 = 80%*
Average Accuracy = 93.3%							*73.3%*

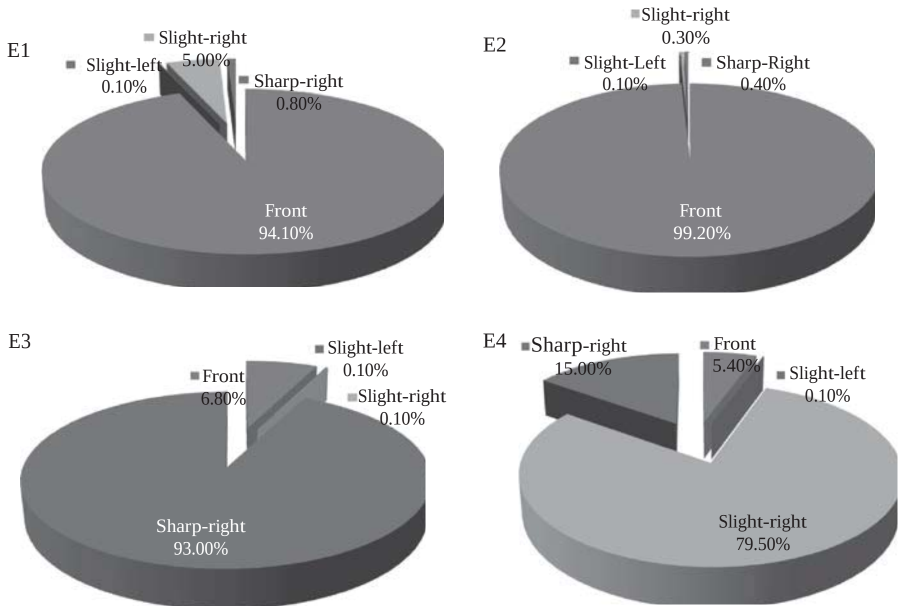

FIGURE 34.7 Presentation showing the predicted behaviour using BN in a dynamic environment from first cross validation of Table 34.2.

TABLE 34.3
Average Performance Evaluation Results of BN and *k*-NN Models

		Bayesian Networks (BN)		*k*-Nearest Neighbour (*k*-NN)	
Environment	**CVn**	**Accuracy (%)**	**Error (%)**	**Accuracy (%)**	**Error (%)**
Static	1	100	0	100	0
	2	100	0	100	0
	3	100	0	100	0
Dynamic	4	100	0	80	20
	5	80	20	60	40
	6	100	0	80	20

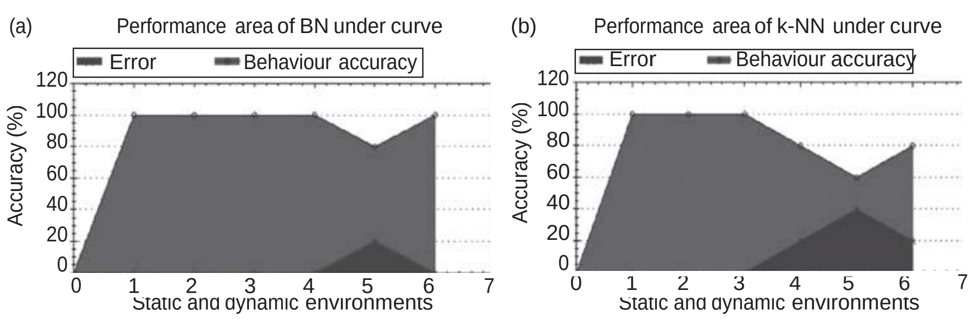

FIGURE 34.8 The BN accuracy under the green area in (a), at CV levels 4–6, is higher than that of the accuracy when compared to the *k*-NN model in (b).

by investigating problems in neuroethology by building physical robot models of biological sensorimotor systems. For instance, Robots are believed to mimic the behaviour of biological systems, but do they model complex behaviours very well, such as emotional expressions? It is argued that in building robot models biological relevance is more effective than loose biological inspiration. This reflects the view that biological behaviour needs to be studied and modelled in the context of real problems faced by real robots in real environments. A proposal of vision-based mobile robot, which can find the location of doors and can traverse doors in complex environments, is presented in Ref. [19]. A Principal Component Analysis (PCA) algorithm using a vision sensor and a fuzzy controller are used for obstacle avoidance and door traversal behaviours. Hongjun et al. [20], proposed a novel method for sensor planning using mobile robot localization based on Bayesian network inference. In their work, they proved that an autonomous robot cannot always determine its unique situation by local sensing information only. The reason is that, the sensor is prone to errors and a slight change of robotic behaviour deteriorates the sensing result. In [21], an attempt was made to obtain an adequate BN model for achieving a door-crossing behaviour using sonar sensors available on their robot platform. Concerning the performance of their obtained behaviour, they proposed that their results should be compared with other approaches, indicating that more work is required on developing behaviours for robots. As they pointed out that they are missing a common frame of reference with other approaches, this chapter experimented with a real-life publicly available navigation data. In our previous work [22], we mentioned that Bayesian network can be used for developing behaviours for autonomous robots for avoiding collisions in the environments. In the approach, an unstructured environment was simulated and information of the obstacles generated was used to build the BN model for avoiding collisions. A proposal to use a real-life robot data and test with more approaches were presented.

34.6 CONCLUDING REMARKS

In this chapter, BN and *k*-NN models have been successfully investigated and applied to developing behaviours for robots for avoiding collisions in static and dynamic environments. The performance of the learning mechanism of both models in the experiments using a real-life robot observation show that the predictive power of BN and *k*-NN models are valuable for their consistency in handling robot collision in a static environment. However, based on the scope and the data used, BN proved better with 93.3% compared to 73.3% of *k*-NN accuracy in the dynamic environment due to its probabilistic calculus in handling uncertainties. The choice of the *k*th minimum value in *k*-NN might need more improvement for an improved prediction to avoid collision. It is worth noting that the chosen direction of the robot assists in determining the control command values, which are translational and rotational velocities the robot uses for navigation. Hence, this investigation contributes to an attempt of using machine learning models for developing behaviours within the robotics domains. The results of this chapter extend our previous work in developing behaviours for a robot as reported in [22], which implemented only the BN model using simulated obstacle data samples. This research work can further be explored in different forms in future work: (i) developing behaviours for conditional collision avoidance by moving from a starting position to a goal position; (ii) develop behaviours based on other sensor configurations as the 24 sensors may be computationally intensive while the two sensors may be faster; and (iii) develop cooperative behaviours for multi-robot systems to avoid conflicts among robot team members.

REFERENCES

1. J.G. Rolim, P.C. Maiola, H.R. Baggenstoss, A.R.G. da Paulo, Bayesian networks application to power transformer diagnosis, *IEEE Lausanne Power Tech*, pp. 999–1004, ISBN: 978-1-4244-2189-3, 2008.
2. A. Thamizharasi, Performance analysis of face recognition by combining multiscale techniques and homomorphic filter using fuzzy K nearest neighbor classifier, *IEEE International Conference on Communication Control and Computing Technologies (ICCCCT)*, pp. 394–401, ISBN: 978-1-4244-7769-2, 2010.

3. N. Nilsson, *Artificial Intelligence, a New Synthesis*, 1st edition. San Fransisco, USA: Morgan Kaufmann Publishers, 1998.
4. Y. Sun, S. Lv and Y. Tang, Construction and application of Bayesian network in early diagnosis of Alzheimer disease's system, *International Conference on Complex Medical Engineering, IEEE/ICME*, pp. 924–929, ISBN: 978-1-4244-1078-1, 2007.
5. I.O. Osunmakinde and A. Potgieter, Immediate detection of anomalies in call data—An adaptive intelligence approach. *Proceedings of the 10th Southern African Telecommunications Networks and Applications International Conference (SATNAC)*, Mauritius. ISBN: 978-0-620-39351-5, 2007.
6. Y. Lv, S. Tang and H. Zhao, Real-time highway traffic accident prediction based on the *k*-nearest neighbor method, *International Conference on Measuring Technology and Mechatronics Automation, ICMTMA '09*, pp. 547–550, ISBN: 978-0-7695-3583-8, 2009.
7. G. Verdier and A. Ferreira, Fault detection with an adaptive distance for the *k*-nearest neighbors rule, *International Conference on Computers & Industrial Engineering, CIE*, pp. 1273–1278, ISBN: 978-1-4244-4135-8, 2009.
8. J. Nagasue, Y. Konishi, N. Araki, T. Sato and H. Ishigaki, Slope-Walking of a biped robot with *k* nearest neighbor method, *Fourth International Conference on Innovative Computing, Information and Control (ICICIC)*, pp. 173–176, ISBN: 978-1-4244-5543-0, 2010.
9. H. Zhou and S. Sakane, Sensor planning for mobile robot localization—A hierarchical approach using a Bayesian network and a particle filter, *IEEE Transactions on Robotics*, 24, 481–487, ISSN: 1552-3098, 2008.
10. S.I. Roumeliotis and G.A. Bekey, Distributed multirobot localization, robotics and automation, *IEEE Transactions on Robotics and Automation*, 18(5), 781–795, ISSN: 1042-296X, 2002.
11. H. William and G. Haipeng, P. Benjamin, & S. Julie, A Permutation genetic algorithm for variable ordering in learning Bayesian networks from data. *Proceedings of Genetic and Evolutionary Computation Conference*, Morgan Kaufmann Publishers Inc, pp. 383–390, 2002.
12. I.O. Osunmakinde and A. Potgieter, Emergence of optimal Bayesian networks from datasets without backtracking using an evolutionary algorithm. *Proceedings of the Third IASTED International Conference on Computational Intelligence*, Banff, Alberta, Canada, ACTA Press, ISBN: 978-0-88986-672-0, pp. 46–51, 2007.
13. S. Russell and P. Norvig, *Artificial Intelligence, A Modern Approach*, 2nd edn., Prentice-Hall Series Inc. NJ, 2003.
14. J. Arroyo and C. Mate, Forecasting histogram time series with *k*-nearest neighbors methods, *International Journal of Forecasting*, 25, 192–207, 2009.
15. D. Newman, S. Hettich, C. Blake and C. Merz, UCI *Repository of Machine Learning Databases* (University of California, Department of Information and Computer Science, Irvine, CA). DOI = http://www.ics.uci.edu/$\sim$mlearn/MLRepository.html; (last accessed 2011).
16. N. Harper and P. McKerrow, Recognizing plants with ultrasonic sensing for mobile robot navigation, *Journal of Robotics and Autonomous Systems*, 34, 71–82, 2001.
17. GeNle 2.0, *Decision Systems Laboratory*, University of Pittsburgh, URL =http://genie.sis.pitt.edu, 2009.
18. B. Webb, Can robots make good models of biological behavior? *Journal of Behavioral and Brain Sciences*, 24, 2001.
19. M.-W. Seo, Y.-J. Kim and M.-T. Lim, Door traversing for a vision based mobile robot using PCA, in: *Lecture Notes on Artificial Intelligence*, Springer-Verlag, pp. 525–531, 2005.
20. Z. Hongjun and S. Shigeyuki, Mobile robot localization using active sensing based on Bayesian network inference, *Journal of Robotics and Autonomous Systems*, 55, 292–305, 2007.
21. E. Lazkano, B. Sierra, A. Astigarraga and J.M. Martinez-Otzeta, On the use of Bayesian networks to develop behaviors for mobile robots, *Journal of Robotics and Autonomous Systems*, 55, 253–265, 2007.
22. C. Yinka-Banjo, I.O. Osunmakinde and A. Bagula, Collision avoidance in unstructured environments for autonomous robots: A behavioral modeling approach, In *Proceedings of the International Conference on Control, Robotics and Cybernetics (ICCRC)*, New Delhi, India, IEEE, ISBN: 978-1-4244-9709-6, pp. 297–303, 2011.